Diplomarbeit

Schnelle Varianten des Generative Topographic Mapping

angefertigt am
Institut für Numerische Simulation

vorgelegt der
Mathematisch-Naturwissenschaftlichen Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn

Dezember 2009

von
Alexander Hullmann
aus
Engelskirchen

Inhaltsverzeichnis

1 Einleitung

Die Menge an hochdimensionalen Daten nimmt stetig zu. Moderne Informationstechnologien machen es möglich, immer größere Datenmengen zu erheben, zu speichern und zu versenden. Der entscheidende Arbeitsschritt ist jedoch die Verarbeitung und Analyse, denn unser Interesse gilt meist nicht den Daten selbst, sondern den in ihnen enthaltenen Informationen.

Hochdimensionale Daten sind beispielsweise Texte, Bilder, Finanzzeitreihen, Messwerte und Umfrageergebnisse. Das digitalisierte Foto eines Autos würde auch dann noch Marke und Modell erkennen lassen, wenn einige Pixel beliebig verändert würden. Offenbar ist die hochdimensionale Darstellung eines Bildes als Vektor von Farbwerten robust in Hinblick auf kleine Veränderungen. Diese Robustheit ist gleichbedeutend mit Redundanz, denn die Störung einzelner Pixel verändert nicht die Information, welches Auto dargestellt wird.

Diese Redundanz ist Ausdruck des Prinzips, dass hochdimensionale Daten intrinsisch niederdimensional sein können. Würden wir Kantenlängen und Volumen von Würfeln messen und die Messwerte gegeneinander auftragen, so lägen sie alle auf einer Kurve. Der funktionale Zusammenhang zwischen beiden Größen führt dazu, dass die zweidimensionalen Datenpunkte auf einer eindimensionalen Struktur liegen. Offenbar schränken lineare und nichtlineare Zusammenhänge den Teilbereich des Raums ein, in dem sich Datenpunkte befinden können.

Da im Hochdimensionalen unsere Intuition und Vorstellungskraft versagen, benötigen wir numerische Verfahren, um Zusammenhänge in den Daten zu erkennen und zu beschreiben. Eine niederdimensionale Projektion ermöglicht neben der Visualisierung auch die effiziente Weiterverarbeitung hochdimensionaler Daten. Der prominenteste Vertreter der Verfahren zur Dimensionsreduktion ist die Hauptkomponentenanalyse oder Principal Component Analysis (PCA). Da die PCA nur lineare Zusammenhänge erkennen kann, haben sich viele weitere Verfahren etabliert. Bekannte Beispiele sind Multidimensional Scaling (MDS), Local Tangent Space Alignment (LTSA), Sammons Nonlinear Mapping (SNM), Curvilinear Component Analysis (CCA), Curvilinear Distance Analysis (CDA), Principal Manifolds, Kohonens SOM, Generative Topographic Mapping (GTM), Locally Linear Embedding (LLE), Isomap, Laplacian Eigenmaps und Kernel PCA. Einen Überblick über die genannten Verfahren geben [HNS08, CP97, LV07].

Im Mittelpunkt dieser Arbeit steht eine modifizierte Formulierung des Generative Topographic Mapping, die eine problemangepasste Verwendung von Diskretisierungen und Quadraturregeln erlaubt und somit höhere Einbettungsdimensionen ermöglicht. Die Grundannahme des GTM ist, dass Datenpunkte im D-dimensionalen Raum von L latenten Variablen $\mathbf{x} = (x_1, \ldots, x_L)$ und einer Abbildung $\boldsymbol{y} : [0,1]^L \to \mathbb{R}^D$ erzeugt worden sind. Der Hyperwürfel $[0,1]^L$ wird hierbei auch als Latent-Space bezeichnet, da jedes $\mathbf{x} \in [0,1]^L$ die latenten Variablen $x_1, \ldots, x_L$ in einem Vektor zusammenfasst. Mit dem Expectation-Maximization-Algorithmus werden das Mapping $\boldsymbol{y}$ und die inverse Varianz β optimiert, um die Wahrscheinlichkeitsdichte der Daten möglichst gut zu rekonstruieren.

Wir fassen an dieser Stelle die Beiträge dieser Arbeit zusammen:

- Wir formulieren das GTM mit einem gleichverteilten Latent-Space-Prior, so dass statt endlicher Summen Integrale über $[0,1]^L$ entstehen. Durch angepasste Quadraturregeln oder das Ausnutzen einer Produktstruktur des Integranden sind Effizienzsteigerungen möglich. Das klassische GTM geht mit der Wahl einer bestimmten Quadraturregel aus dieser Formulierung hervor.

- Wir verallgemeinern das GTM-Funktional auf allgemeine Dichten im Datenraum und beschreiben ein mit dem EM-Algorithmus vergleichbares Minimierungsverfahren – jedoch ohne die Verwendung von stochastischen Begriffen.

- Wir beschreiben und implementieren ein Sparse GTM, das Dünngitterinterpolation zur Diskretisierung der Komponentenfunktionen von $\boldsymbol{y}$ und Dünngitterquadratur zur Integration über den Latent-Space verwendet. Die Anzahl von Quadraturpunkten und Freiheitsgraden reduziert sich von $\mathcal{O}(h^{-L})$ auf $\mathcal{O}(h^{-1} \cdot (\log h^{-1})^{L-1})$, wobei h die Maschenweite in eine Koordinatenrichtung bezeichnet. So wird bei fixer Latent-Space-Dimension der sogenannte „Fluch der Dimension“, das heisst die exponentielle Abhängigkeit der Laufzeit von L, bis auf den $\log h^{-1}$-Term aufgehoben. Die Approximationsgüte verschlechtert sich hierbei lediglich um einen logarithmischen Faktor $(\log h^{-1})^{L-1}$, siehe [BG04, GG98].

- Wir entwickeln die theoretischen Grundlagen für ein Low-Rank GTM, bei dem die Komponentenfunktionen von $\boldsymbol{y}$ mit

$$y_d(\mathbf{x}) = \sum_{r=1}^{P} s_r^d \prod_{i=1}^{L} g_{r,i}^d(x_i)$$

  diskretisiert werden. Hierbei sind die Funktionen $g_{r,i}^d$ selbst wieder Unbekannte, siehe [BGM09].

- Wir entwickeln und implementieren ein Low-ANOVA GTM mit

$$\boldsymbol{y}(\mathbf{x}) = \sum_{d=1}^{D} \mathbf{v}_d g_d(x_{p(d)})$$

  und orthonormalen $\{\mathbf{v}_d\}_{d=1}^D$. Mit dieser Diskretisierung zerfallen alle L-dimensionalen Integrale in Produkte von eindimensionalen Integralen, und die exponentielle Abhängigkeit der Laufzeit von der Latent-Space-Dimension wird vollständig aufgehoben. Das Mapping $\boldsymbol{y}$ hat zwar signifikant weniger Freiheitsgrade als bei anderen Diskretisierungen, es können jedoch noch Nichtlinearitäten in den Daten erfasst werden. Somit liegt die Mächtigkeit des Low-ANOVA GTM zwischen der PCA und dem klassischen GTM.

- Wir beschreiben und implementieren ein GTM, das statt klassischer Gauß-Kerne

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{\sigma^2}\right)$$

die p-Gauß-Kerne

$$k_p(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{d\,(\mathbf{x}, \mathbf{y})^p}{\sigma^p}\right)$$

mit $d(x, y) = \|\cdot\|_p$ verwendet. In [FWV05] wird nachgewiesen, dass sich mit p-Gauß-Kernen und $d(x, y) = \|\cdot\|_2$ in hohen Dimensionen Lokalität herstellen lässt. Wir weisen dieses Resultat auch für unsere Metrik nach.

- Wir konstruieren mit dem GTM einen Klassifikator. Mit diesem zeigen wir experimentell Wechselwirkungen zwischen hochdimensionalem Rauschen und der intrinsischen Dimension von Daten.

Im Folgenden geben wir einen kurzen Überblick über den Aufbau der Arbeit: In Kapitel 2 diskutieren wir die Erwartungen an ein Verfahren zur Dimensionsreduktion und stellen die Principal Component Analysis und das GTM vor. Unterabschnitte über *density modelling* und den Expectation-Maximization-Algorithmus erläutern die Hintergründe des Generative Topographic Mapping. In Kapitel 3 behandeln wir den Zusammenhang zwischen der Maximum-Likelihood-Bestimmung und der Minimierung der Kullback-Leibler-Divergenz von Modell und empirischer Verteilung. Dies ist die Grundlage für die Definition unseres GTM-Funktionals, dessen Minimierung wir erläutern. In Kapitel 4 stellen wir die Quadratur und die Interpolation auf dünnen Gittern in allgemeiner Form vor. Wir klären, inwieweit die Voraussetzungen für deren Anwendung auf das GTM gegeben sind. Wir beschreiben die einzelnen Minimierungsschritte in Pseudocode. Nachdem wir auf numerische Aspekte der GTM-Implementierung eingegangen sind, stellen wir den Bezug zwischen dem Regularisierungsoperator und Hilberträumen mit reproduzierendem Kern her. Die Kapitel 5 und 6 behandeln die Low-Rank und Low-ANOVA Diskretisierungen für das Mapping $\boldsymbol{y}$. Die Verwendung der euklidischen Norm $\|\cdot\|_2$ oder ihrem Quadrat ist in hohen Dimensionen problematisch, da der sogenannte „Concentration-of-Measure“-Effekt auftritt. In Kapitel 7 stellen wir das Prinzip der Lokalität vor, und weisen nach, dass dies bei unseren p-Gauß-Kernen gegeben ist. Wir beschreiben die Implementierung der p-Gauß-Kern GTM. Schließlich folgt das Kapitel 8 mit numerischen Experimenten. Wir testen unsere GTM-Varianten mit verschiedenen synthetischen Datensätzen und Beispielen aus der Literatur. Wir konstruieren einen Klassifikator und zeigen, dass mit der Literatur vergleichbare Erkennungsraten erreicht werden.

Es ist mir eine besondere Freude, an dieser Stelle den Personen, die mich bei der Erstellung dieser Arbeit unterstützt haben, meinen Dank auszusprechen.

Zuallererst möchte ich mich bei meinem Betreuer Prof. Dr. Michael Griebel bedanken. Sein Engagement, Vorlesungen über spezielle Themen für einen kleinen Zuhörerkreis zu halten, und die intensive Betreuung meiner Arbeit waren sehr motivierend. Ohne die vielen Anregungen, Diskussionen und Literaturempfehlungen wären viele Erkenntnisse nicht entstanden. Des Weiteren möchte ich mich bei Priv.-Doz. Dr. Marc Alexander Schweitzer für die Übernahme des Zweitgutachtens bedanken.

Die Atmosphäre in der Arbeitsgruppe war sehr gut, und bei Fragen und für Diskussionen standen mir die Türen aller Mitarbeiter offen. Mein besonderer Dank gilt Christian Feuersänger, der mich sehr unterstützt und mit mir seine Erfahrungen im Bereich der Datenanalyse geteilt

hat. Nahezu alle Abbildungen in dieser Arbeit habe ich mit seinem PGFPLOTS-Paket erstellt.

Bei Bastian Bohn und Jens Oettershagen möchte ich mich für das Korrekturlesen der Arbeit und viele unterhaltsame und fachliche Gespräche bedanken. Die ausgezeichneten Arbeitsbedingugen am Institut für Numerische Simulation und der interessante Austausch auf den Blockseminaren 2007 und 2009 haben ebenfalls zum Erfolg dieser Arbeit beigetragen.

Schließlich möchte ich meinen Eltern meine große Dankbarkeit für ihre vielfältige Unterstützung ausdrücken.

2 Dimensionsreduktion

In diesem Kapitel werden die wichtigsten Aspekte der Dimensionsreduktion dargestellt. In Inhalt und Aufbau ist es an die ersten Kapitel von [LV07] angelehnt.

2.1 Aufgaben der Dimensionsreduktion

Wie bereits in der Einleitung beschrieben wurde, ist das Ziel der Dimensionsreduktion die Reduzierung von Redundanzen. Eine optimale Methode sollte über folgende Fähigkeiten verfügen:

1. Bestimmung der Anzahl der latenten Variablen,
2. Einbettung der Daten zur Dimensionsreduktion und
3. Einbettung der Daten zur Bestimmung der latenten Variablen.

In den folgenden Unterabschnitten werden wir die einzelnen Punkte genauer beschreiben. Auch wenn wir das Generative Topographic Mapping erst in Abschnitt 2.3 ausführlich vorstellen werden, erwähnen wir bereits an dieser Stelle, ob es die entsprechende Funktionalität bietet.

2.1.1 Bestimmung der Anzahl der latenten Variablen

Zu Beginn der Analyse hochdimensionaler Daten muss die intrinsische Dimensionalität bestimmt werden. Sie steht stellvertretend für die Anzahl der latenten Variablen, die die Daten generiert haben, kann jedoch auch gebrochene Werte annehmen. In [LV07] werden unterschiedliche Dimensionsbegriffe motiviert, nämlich die Konzepte der topologischen Dimension, der fraktalen Dimension und der q-Dimension. Letztere hat als Spezialfälle die Kapazitätsdimension, die Informationsdimension und die Korrelationsdimension. Diese Dimensionsbegriffe sind nicht Gegenstand dieser Arbeit. Für den Begriff der Korrelationsdimension verweisen wir auf die Arbeit von Grassberger und Procaccia [GP83].

Das Generative Topographic Mapping stellt keinen Mechanismus zur Dimensionsschätzung zur Verfügung, sondern erwartet eine Vorgabe für die Anzahl L von latenten Variablen. L ist somit ein Hyperparameter und sollte mit Sorgfalt gewählt werden, denn eine Unterschätzung der intrinsischen Dimension führt in der Regel zu schlechten Ergebnissen.

2.1.2 Einbettung der Daten zur Dimensionsreduktion

Eine niedrige intrinsische Dimension P der Daten legt nahe, dass eine topologische Struktur vorliegt und der Datenraum nicht komplett ausgefüllt wird. In diesem Fall möchten wir die Daten in einen niederdimensionalen Raum einbetten, der besser ausgefüllt wird. Ziele hierbei sind

- eine bessere Visualisierung der Daten,
- eine kompaktere Repräsentation und
- eine effizientere Weiterverarbeitung.

Um Daten mit einer niedrigen intrinsischen Dimension P einzubetten, müssen mehrere Annahmen gemacht werden. Zum einen müssen die Daten auf einer P-dimensionalen Struktur liegen, zum anderen muss es möglich sein, sie in einen Raum einzubetten, dessen Dimension näher an P als an D ist. Beides ist nicht zwingend der Fall, da P global bestimmt wird, und lokal möglicherweise nicht gilt.

Eine niederdimensionale Einbettung muss die Struktur der Mannigfaltigkeit, auf der die Daten liegen, respektieren. Lee und Verleysen klassifizeren in [LV07] Verfahren danach, ob sie paarweise Abstände oder die Topologie der Daten erhalten. Der Erhalt der Topogie beschränkt sich auf qualitative Aussagen der Art „Punkt A ist näher an Punkt B als an Punkt C“, die meist den relevanten Anteil der Abstandsfunktion darstellen. Das GTM lässt sich eindeutig den topologieerhaltenden Verfahren zuordnen, und die Einbettung der Daten zur Dimensionsreduktion ist seine zentrale Anwendung.

2.1.3 Einbettung zur Bestimmung der latenten Variablen

Dimensionsreduktion führt in erster Linie zu einer Reduzierung der Variablen, die die Daten beschreiben. Die Bestimmung oder Wiederherstellung der latenten Variablen geht hingegen etwas weiter. So kann beispielsweise die statistische Unabhängigkeit der latenten Variablen gefordert werden, und das Verfahren muss dies bei der Bestimmung der niederdimensionalen Repräsentation der Daten berücksichtigen.

Die latenten Variablen des GTM sind unabhängig identisch und uniform auf $[0, 1]$ verteilt. Insofern ermittelt dieses Verfahren in der Regel nicht die *richtigen* latenten Variablen. Allerdings bietet das Low-ANOVA GTM aus Kapitel 6 eine ähnliche Nebenbedingung: Das Modell nimmt an, dass sich der Datenraum in orthogonale Unterräume zerlegen lässt, zwischen denen keine statistischen Abhängigkeiten bestehen. Diese Annahme hat jedoch eine deutliche Einschränkung des Modells zur Folge.

2.2 Principal Component Analysis

Die Principal Component Analysis ist eines der ältesten und bekanntesten Verfahren für multivariate Datenanalyse und Data Mining. Die PCA wurde von Pearson im Rahmen einer biologischen Anwendung Anfang des letzten Jahrhunderts eingeführt, siehe [Pea01]. Davon unabhängig wurde die PCA für stochastische Prozesse von Karhunen entwickelt und von Loève generalisiert, siehe [Kar46, Loe48].

2.2.1 Modell und Berechnung

Das Modell der PCA nimmt an, dass die D-dimensionalen Datenpunkte oder Samples $\mathcal{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$ aus einer linearen Transformation von L latenten Variablen hervorgegangen sind,

die wir in dem Vektor $\mathbf{x} = (x_1, \ldots, x_L)^T$ zusammenfassen. Mit der entsprechenden Matrix $\boldsymbol{W} \in \mathbb{R}^{D \times L}$ gilt dann

$$\mathbf{t} = \boldsymbol{W}\mathbf{x}.$$

Wir nehmen für die latenten Variablen $(x_1, \ldots, x_L)^T$ eine Gauß-Verteilung an, aber auch eine Gleichverteilung ist möglich. Entscheidend ist die Spaltenorthogonalität von $\boldsymbol{W}$, das heisst sie muss $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_L$ erfüllen. Durch eine einfache Translation erreichen wir, dass die Beobachtungen um den Nullpunkt zentriert sind, also dass $\mathbb{E}[\mathbf{t}] = \mathbf{0}$ gilt, und nehmen dies auch für die latenten Variablen an.

Die Frage der Skalierung der Datenraumdimensionen ist von besonderer Wichtigkeit. Sollte eine Dimension für eine Länge stehen, können wir diese in Metern oder Millimetern angeben, erwarten von der PCA jedoch das gleiche Ergebnis. Dies ist jedoch nur dann der Fall, wenn wir vor der eigentlichen PCA die Datenraumdimensionen normieren, das heisst durch ihre jeweilige Standardabweichung teilen. Hier müssen jedoch zwei Aspekte beachtet werden:

- Eine Variable mit Standardabweichung 0 kann so nicht behandelt werden.
- Wenn eine Variable mit geringer Standardabweichung verrauscht wird, hat das Rauschen einen hohen Anteil an ihrer Standardabweichung. Das Rauschen kann durch die Normierung verstärkt werden, was besonders nachteilig ist, da die PCA diese Dimension aufgrund ihrer Unabhängigkeit von den anderen Raumdimensionen als besonders signifikant einschätzen würde.

Man kann die PCA über verschiedene Minimierungskriterien motivieren. Wir entscheiden uns für die statistische Perspektive von Hotelling, siehe [Hot33]. Durch die Unkorreliertheit der latenten Variablen ist die Kovarianzmatrix

$$\boldsymbol{C}_{\mathbf{x}} = \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$$

diagonal. In der Regel gilt diese Unkorreliertheit für die Beobachtungen oder Datenpunkte nach der Achsentransformation mit der Matrix $\boldsymbol{W}$ nicht mehr. Das Ziel der PCA ist nun, die unkorrelierten latenten Variablen zu rekonstruieren. Es gilt

$$\boldsymbol{C}_{\mathbf{t}} = \mathbb{E}\left[\mathbf{t}\mathbf{t}^T\right] = \mathbb{E}\left[\boldsymbol{W}\mathbf{x}\mathbf{x}^T\boldsymbol{W}^T\right] = \boldsymbol{W}\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]\boldsymbol{W}^T = \boldsymbol{W}\boldsymbol{C}_{\mathbf{x}}\boldsymbol{W}^T.$$

Mit $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_L$ können wir

$$\boldsymbol{C}_{\mathbf{x}} = \boldsymbol{W}^T\boldsymbol{C}_{\mathbf{t}}\boldsymbol{W} \tag{2.1}$$

folgern. Durch eine Eigenwertzerlegung der symmetrischen Matrix $\boldsymbol{C}_{\mathbf{t}}$ erhalten wir

$$\boldsymbol{C}_{\mathbf{t}} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T,$$

wobei $\boldsymbol{V}$ eine Matrix mit normierten Eigenvektoren $\{\mathbf{v}_d\}_{d=1}^D$ und $\boldsymbol{\Lambda}$ eine Diagonalmatrix mit den entsprechenden Eigenwerten λ_d in absteigender Reihenfolge ist. Gleichung (2.1) wird durch Einsetzen zu

$$\boldsymbol{C}_{\mathbf{x}} = \boldsymbol{W}^T\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T\boldsymbol{W}. \tag{2.2}$$

Unter der Annahme, dass das PCA-Modell vollständig erfüllt wird, sind nur die ersten L Eigenwerte $\{\lambda\}_{d=1}^L$ größer als 0. Des Weiteren müssen die Spalten von $\boldsymbol{W}$ im Spann der ersten L

Spalten von $\boldsymbol{V}$ liegen. Wir setzen

$$\boldsymbol{W} := \boldsymbol{V}\boldsymbol{I}_{D\times L},$$

was zu

$$\boldsymbol{C}_{\mathbf{x}} = \boldsymbol{I}_{L\times D}\boldsymbol{\Lambda}\boldsymbol{I}_{D\times L},$$

bzw. zu

$$\boldsymbol{C}_{\mathbf{x}} = \mathrm{diag}(\lambda_1, \ldots, \lambda_L, 0, \ldots, 0)$$

führt. Offenbar sind die Eigenwerte in $\boldsymbol{\Lambda}$ die Varianzen der latenten Variablen. Nun haben wir die Modellparameter der PCA vollständig bestimmt.

In den allermeisten Fällen werden die Daten nicht vollständig dem PCA-Modell entsprechen. Hierfür bestehen mehrere Gründe:

- Die einzelnen Dimensionen der $\mathbf{t}$-Vektoren können verrauscht sein.
- Die Verteilung der $\mathbf{t}$-Vektoren folgt dem PCA-Modell, aber der Übergang zur empirischen Verteilung der Samples $\{\mathbf{t}_n\}_{n=1}^N$ verursacht einen Fehler.
- Die niederdimensionalen Strukturen, auf denen sich die Daten befinden, sind nicht linear.

Die genannten Punkte führen dazu, dass es kaum 0-Eigenwerte gibt. Der beschriebene Lösungsweg bleibt jedoch gültig, wenn wir uns auf die größten Eigenwerte beschränken. Wenn wir die Varianz von $\mathbf{t}$ als

$$\sigma_{\mathbf{t}}^2 := \mathrm{tr}(\boldsymbol{C}_{\mathbf{t}}) = \sum_{d=1}^{D} \lambda_d$$

definieren, so erhalten wir eine Lösung, die die mit L latenten Variablen erfassbare Varianz maximiert. In [LV07] wird gezeigt, dass die PCA ebenfalls den quadratischen Rekonstruktionsfehler minimiert.

2.2.2 Fähigkeiten

Der Erfolg der PCA ist nicht nur der Einfachheit des Verfahrens, sondern auch der breiten Anwendbarkeit geschuldet. Tatsächlich werden alle drei der in Abschnitt 2.1 beschriebenen Fähigkeiten umgesetzt.

Bestimmung der Anzahl der latenten Variablen

Wenn das PCA-Modell vollständig respektiert wird, sind $D - L$ der D Eigenwerte der Kovarianzmatrix $\boldsymbol{C}_{\mathbf{t}}$ gleich 0. Wie wir bereits wissen, ist dies häufig nicht der Fall. Um Rauschen von den latenten Variablen zu unterscheiden, muss die Varianz des Rauschens kleiner sein als die Varianz der Daten. In diesem Fall können wir das Spektrum heranziehen: Wir plotten die Eigenwerte $\{\lambda_d\}_{d=1}^D$ in absteigender Reihenfolge und suchen nach einer Sprungstelle, wo sich die „großen“ von den „kleinen“ Eigenwerten trennen lassen. Die großen Eigenwerte entsprechen hierbei den latenten Variablen, die kleinen dem Rauschen.

Sollte es sehr viele latente Variablen geben, die nur teilweise eine höhere Varianz als das Rauschen haben, ist es immer noch möglich, die Dimension so zu reduzieren, dass ein möglichst

großer Anteil der Varianz erhalten bleibt. Um beispielsweise 95% der Varianz zu erhalten, wählen wir L minimal, so dass

$$0.95 \leq \frac{\sum_{d=1}^{L} \lambda_d}{\sum_{d=1}^{D} \lambda_d}$$

gilt.

Letzten Endes sind die beschriebenen Methoden heuristisch und haben nur dann Aussagekraft, wenn die Daten aus einem PCA-Modell stammen. Dennoch muss die Anzahl der latenten Variablen nicht vorgegeben werden, sondern wird vom Verfahren geschätzt.

Einbettung der Daten zur Dimensionsreduktion

Nachdem die Anzahl der latenten Variablen bestimmt worden ist, kann ein Datenpunkt $\mathbf{t}$ durch

$$\mathbf{x} := \boldsymbol{I}_{L \times D} \boldsymbol{V}^T \mathbf{t} \tag{2.3}$$

in den Raum der latenten Variablen projiziert werden. Diese Darstellung entspricht einem Basiswechsel und dem Entfernen der $D - L$ am wenigsten wichtigen Raumrichtungen.

Einbettung zur Bestimmung der latenten Variablen

Neben der Dimensionsreduktion kann die PCA auch die latenten Variablen separieren. Dies geschieht in Formel (2.3) durch die Multiplikation mit $\boldsymbol{V}^T$.

Es wurde vorausgesetzt, dass die Beobachtungen aus einer Rotation von Gauß-verteilten latenten Variablen hervorgehen. Diese strengen Voraussetzungen können etwas abgeschwächt werden: Es ist ausreichend, für die Spalten von $\boldsymbol{W}$ Orthogonalität statt Orthonormalität zu fordern. In diesem Fall werden die latenten Variablen bis auf Permutation und Skalierungsfaktoren korrekt bestimmt. Wenn $\boldsymbol{W}$ eine beliebige Matrix ist, kann die PCA immer noch latente Variablen entlang orthogonaler Richtungen bestimmen.

Da die Kumulanten höherer Ordnung wie beispielsweise Schiefheit (3. Ordnung) oder Wölbung (4. Ordnung) für gaußsche Variablen gleich 0 sind, erzielen wir auf diese Weise vollständig unabhängige latente Variablen. Bei anderen Verteilungen müssen diese Kumulanten jedoch berücksichtigt werden, siehe [CA02, HKO01].

2.2.3 Beispiele

Im ersten Beispiel ziehen wir 2000 Zufallsvektoren $\mathbf{x} = (x_1, x_2)^T$, wobei x_1 und x_2 unabhängig Gauß-verteilt mit Erwartungswert 2 und Varianzen $\sigma_1^2 = 4$ und $\sigma_2^2 = 1$ sind. Sie werden mit der Matrix

$$\boldsymbol{W} = \begin{pmatrix} -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{3}} & 0 \end{pmatrix}$$

in den $\mathbb{R}^3$ abgebildet. Mit Hilfe der Hauptkomponentenanalyse können wir sowohl die $\boldsymbol{W}$-Matrix als auch die Varianzen σ_1^2 und σ_2^2 schätzen. Die Ergebnisse sind

$$\tilde{\boldsymbol{W}} = \begin{pmatrix} 0.5811 & 0.7040 \\ 0.5736 & -0.7102 \\ 0.5773 & -0.0031 \end{pmatrix}$$

mit den Varianzen $\tilde{\sigma}_1^2 = 3.9553$ und $\tilde{\sigma}_2^2 = 1.0023$. Abgesehen vom Schätzfehler stimmen die ermittelten Varianzen mit denen der latenten Variablen überein. Auch die orthonormalen Richtungen der $\tilde{\boldsymbol{W}}$-Matrix entsprechen bis auf das Vorzeichen den tatsächlichen Modellparametern. Die Datenpunkte und der rekonstruierte Latent-Space sind in Abbildung 2.1 zu sehen.

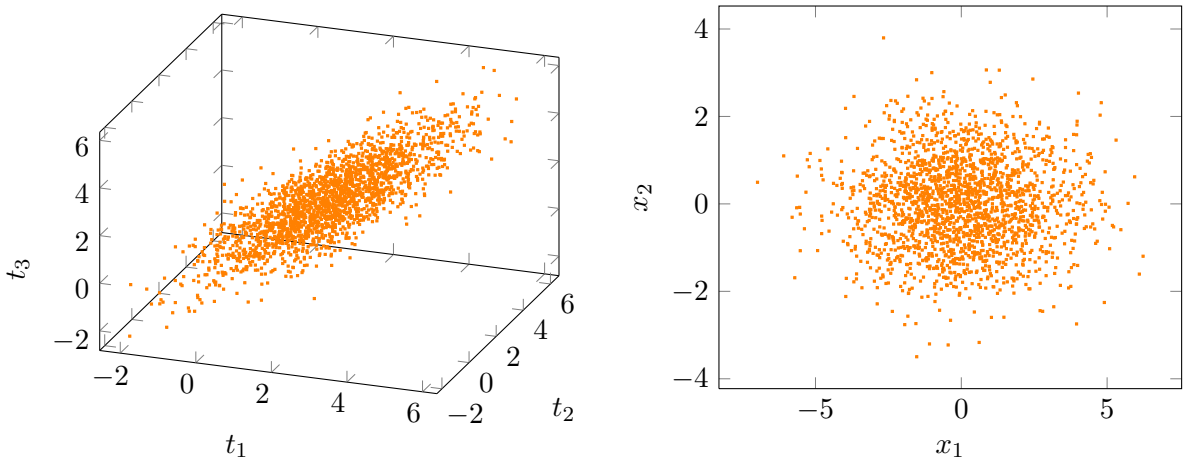

Abb. 2.1: Gauß-verteilte Zufallsvariablen erzeugen mit einem PCA-Modell die links dargestellten Datenpunkte $\mathbf{t} \in \mathbb{R}^3$. Mit der Hauptkomponentenanalyse können die latenten Variablen $\mathbf{x} \in \mathbb{R}^2$ rekonstruiert werden (rechts).

In einem zweiten Beispiel gehen wir nicht mehr von Gauß-verteilten Zufallsvariablen aus. Stattdessen nehmen wir an, dass x_1 auf $[0, 2]$ und x_2 auf $[0, 1]$ uniform verteilt sind. Wir verwenden die Abbildungsmatrix

$$\boldsymbol{W} = \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \\ 0 & 1 \end{pmatrix},$$

um die Zufallsvektoren in den $\mathbb{R}^3$ abzubilden. Wir lassen hier die Orthonormalitätsbedingung für die Spalten von $\boldsymbol{W}$ fallen. Offenbar entspricht dieser Datensatz nicht dem PCA-Modell, eine Hauptkomponentenanalyse ist dennoch anwendbar. Die Schätzung auf der Basis von 2000 Samples ergibt

$$\tilde{\boldsymbol{W}} = \begin{pmatrix} -0.8970 & 0.3853 \\ -0.4128 & -0.5552 \\ -0.1579 & -0.7371 \end{pmatrix}$$

mit den Varianzen $\tilde{\sigma}_1^2 = 0.4194$ und $\tilde{\sigma}_2^2 = 0.1315$. Die geschätzten Varianzen passen nicht zu

den latenten Variablen, was in Anbetracht der Tatsache, dass sie nicht normalverteilt und die Spalten von $\boldsymbol{W}$ nicht orthonormal sind, nicht anders zu erwarten ist. Die Richtungen von $\tilde{\boldsymbol{W}}$ liegen jedoch innerhalb der zweidimensionalen Ebene, die von $\boldsymbol{W}$ aufgespannt wird. Anhand der niederdimensionalen Projektion der Datenpunkte in Abbildung 2.2 ist erkennbar, dass die rekonstruierten latenten Variablen nicht unabhängig sind wie im ersten Beispiel. Wir stellen fest, dass sich die Hauptkomponentenanalyse mit kleinen Abstrichen auf allgemeine lineare Strukturen anwenden lässt.

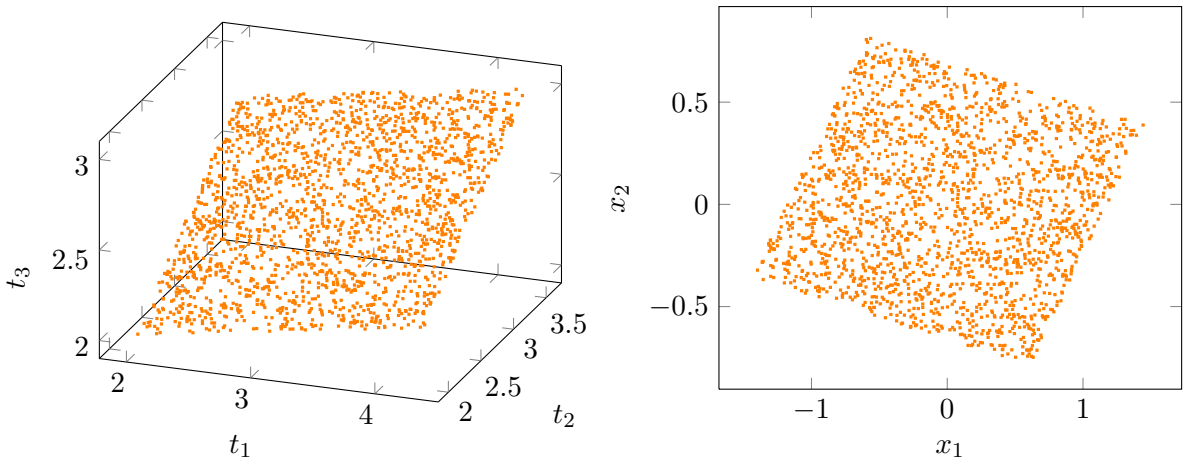

Abb. 2.2: Bei uniform verteilten latenten Variablen erkennt die PCA die Hauptachsen in den Datenpunkten $\mathbf{t} \in \mathbb{R}^3$ (links) und kann sie in den $\mathbb{R}^2$ einbetten (rechts).

Nun stellen wir noch eine intrinsisch zweidimensionale Struktur vor, die in [LV07] als Benchmark verwendet wird, und unter dem Namen „Swiss Roll“ bekannt ist. In Abbildung 2.3 sehen wir, dass die Hauptachsenanalyse die Swiss Roll nicht erkennt. Die Struktur wird nicht „abgerollt“, sondern die t_2-Richtung geht einfach verloren. In der Folge liegen Punkte nahe beieinander, deren Abstände auf der Struktur gemessen sehr groß sind. Diese Schwäche der PCA motiviert Verfahren, die mit Nichtlinearitäten umgehen können, wie beispielsweise das Generative Topographic Mapping.

2.3 Generative Topographic Mapping

Das Generative Topographic Mapping (GTM) wurde 1996 von Bishop, Svensén und Williams veröffentlicht, siehe [BSW98b]. Es verwendet wie die PCA ein Modell, das davon ausgeht, dass sich die Daten mit einer kleinen Anzahl latenter Variablen beschreiben lassen. Im Gegensatz zur PCA sind jedoch auch nichtlineare Transformationen der latenten Variablen möglich.

Ausgangspunkt für das GTM war der Self Organizing Map (SOM)-Algorithmus, siehe Kohonen [Koh82]. Dieser Algorithmus hat seinen Ursprung in der Neurobiologie und fasst eine Menge von Datenvektoren $\{\mathbf{t}_n\}_{n=1}^N$ im D-dimensionalen Raum zu einer Menge von Referenzvektoren zusammen, die in der Regel auf einer zweidimensionalen Struktur liegen. Das GTM überwindet einige Nachteile der SOM:

- Das Fehlen einer Kostenfunktion,

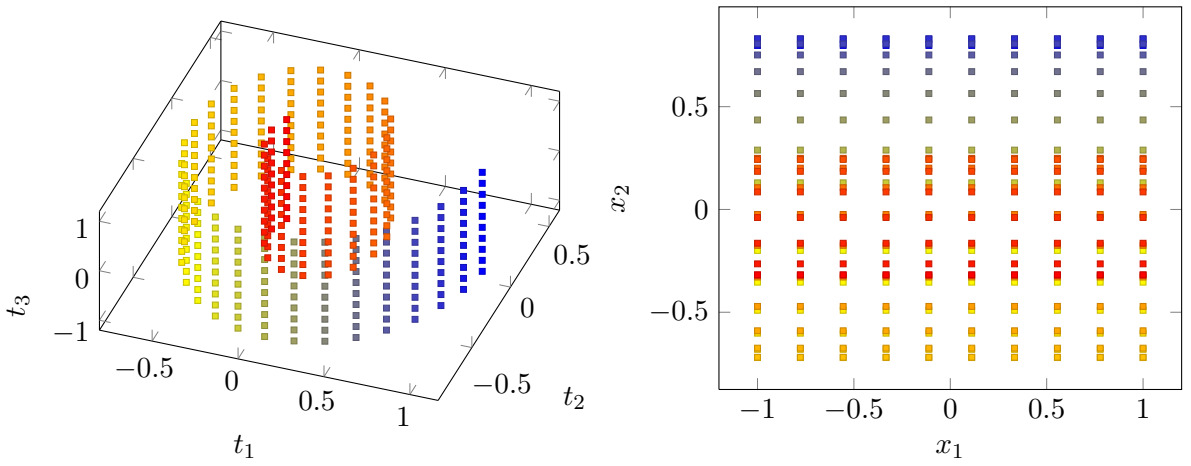

Abb. 2.3: Die Struktur der Swiss Roll (links) und ihre zweidimensionale Einbettung mit der PCA (rechts).

- das Fehlen einer theoretischen Basis für die korrekte Wahl der Nachbarschafts- und Lernparameter,
- das Fehlen eines allgemeinen Konvergenzbeweises und
- die Tatsache, dass keine Wahrscheinlichkeitsverteilung im Datenraum definiert wird.

2.3.1 Density Modelling

Das *density modelling* wurde 1994 von MacKay ausfürlich beschrieben, siehe [Mac95]. Es konstruiert mit Hilfe der latenten Variablen eine Dichte im Datenraum. Durch die Anwendung des Satz von Bayes erhalten wir unter der Annahme eines Priors eine echte Wahrscheinlichkeitsverteilung auf dem Raum der Modellparameter.

Seien wie bisher $\mathcal{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$ eine endliche Menge von unabhängig identisch verteilten Samples im $\mathbb{R}^D$. Wir möchten nun einen Parametersatz θ bestimmen, der mit unserem Modell eine Dichte im Datenraum erzeugt, die zu den Samples passt. Es liegt nahe, die Likelihood

$$L(\theta) = p(\mathcal{T} \mid \theta)$$

zu maximieren. Im Fall von Samples, die unabhängig identisch verteilt sind, gilt für die Log-Likelihood

$$\mathcal{L}(\theta) := \log L(\theta) = \log p(\mathcal{T} \mid \theta) = \log \prod_{n=1}^{N} p(\mathbf{t}_n \mid \theta) = \sum_{n=1}^{N} \log p(\mathbf{t}_n \mid \theta). \tag{2.4}$$

An dieser Stelle verwenden wir ein klein geschriebenes $p(\cdot)$, da wir für Samples aus einem kontinuierlichen Modell keine Wahrscheinlichkeit, sondern nur eine Wahrscheinlichkeits*dichte* angeben können. Ein anderer Aspekt von einer endlichen Menge von Samples ist, dass sogenanntes „Overfitting“ auftreten kann: In diesem Fall passt sich das Modell den Beobachtungen

so weit an, dass die Ergebnisse auf anderen Samples der gleichen Verteilung schlechter werden. Dies kann durch einen Prior $p(\theta)$ verhindert werden, der jedem Parametersatz – unabhängig von den Daten – eine Wahrscheinlichkeitsdichte zuordnet.

Durch Anwendung des Satz von Bayes

$$p(\theta \mid \mathcal{T}) = \frac{p(\mathcal{T} \mid \theta)p(\theta)}{p(\mathcal{T})} \propto L(\theta)p(\theta)$$

erhalten wir zu gegebenen Daten eine Wahrscheinlichkeitsverteilung auf dem Raum der Modellparameter, was bei der Penalized-Likelihood-Maximierung nicht der Fall ist. In den meisten Fällen sind wir am Modus dieser Verteilung interessiert. Für die sogenannte Maximum-a-Posteriori (MAP)-Bestimmung auf dem Raum der Modellparameter gilt

$$\begin{aligned} \theta_{\text{opt}} &= \arg\max_{\theta} p(\theta \mid \mathcal{T}) \\ &= \arg\max_{\theta} \log L(\theta)p(\theta) \\ &= \arg\max_{\theta} (\mathcal{L}(\theta) + \log p(\theta)). \end{aligned}$$

Hier sehen wir die Verbindung zur Penalized-Likelihood-Maximierung, denn $S(\theta) := \log p(\theta)$ können wir als den Regularisierungsterm auffassen, der unerwünschte Parametersätze bestraft und somit Overfitting vermeidet.

In [Mac95] werden zwei Gründe angeführt, warum die Verwendung des Satz von Bayes trotz dieser Äquivalenz vorteilhaft ist:

- Durch die Wahrscheinlichkeitsverteilung auf dem Raum der Modellparameter ist es möglich, die Unsicherheit des Ergebnisses zu bestimmen.

- Üblicherweise wird der Regularisierungsterm $S(\theta)$ mit einem Vorfaktor λ gewichtet, zu dessen Bestimmung Kreuzvalidierung nötig ist. Es ist prinzipiell möglich, diese Prozedur durch eine weitere Anwendung des Satz von Bayes zu ersetzen, um den optimalen Vorfaktor zu bestimmen.

In [GH08] wird darauf hingewiesen, dass das Lebesgue-Maß auf unendlichdimensionalen Parameterräumen nicht definiert ist, und in diesem Fall bei der MAP-Bestimmung keine gültige Wahrscheinlichkeitsdichte für den Prior angegeben werden kann. In der Praxis sind die Parameterräume in der Regel endlichdimensional, in der Theorie ist dieser Einwand jedoch relevant, beispielsweise für Konvergenzresultate.

Im Folgenden zählen wir die Komponenten auf, mit denen wir unser *density modelling* eindeutig beschreiben können.

- Die Anzahl der latenten Variablen L:
  L ist damit auch die Dimension des Latent-Space, also des Raums der latenten Variablen in vektorieller Schreibweise.

- Der Prior auf dem Latent-Space $p(\mathbf{x})$:
  Hierbei ist $\mathbf{x} \in \mathbb{R}^L$ ein Zufallsvektor.

- Die Abbildung zwischen Latent-Space und Datenraum $\boldsymbol{y} : \mathbb{R}^L \to \mathbb{R}^D$:
  Sie ist stetig und sollte möglichst glatt sein. Die Dimensionsreduktion wird erst durch diese Verbindung zwischen den L- und D-dimensionalen Räumen ermöglicht. Dieses Mapping wird vollständig durch den Parametersatz θ bestimmt, so dass $\boldsymbol{y}(\mathbf{x})$ eine Abkürzung für $\boldsymbol{y}(\mathbf{x}; \theta)$ darstellt.

- Die Fehlerfunktion $G_n(\mathbf{x}, \theta) = \log p(\mathbf{t}_n \mid \mathbf{x}, \theta)$:
  Sie führt eine Rauschkomponente in das Modell ein. Mit ihr kann auch der Posterior im Latent-Space
  $$p(\mathbf{x} \mid \mathbf{t}_n, \theta) = \frac{p(\mathbf{t}_n \mid \mathbf{x}, \theta) p(\mathbf{x})}{p(\mathbf{t}_n \mid \theta)} = \frac{\exp(G_n(\mathbf{x}, \theta)) p(\mathbf{x})}{p(\mathbf{t}_n \mid \theta)}$$
  bestimmt werden, wobei
  $$p(\mathbf{t}_n \mid \theta) = \int p(\mathbf{t}_n \mid \mathbf{x}, \theta) p(\mathbf{x}) d\mathbf{x} \tag{2.5}$$
  die Normalisierungskonstante ist.

- Der Optimierungsalgorithmus:
  Dieser bestimmt $\arg\max_\theta p(\theta \mid \mathcal{T})$. In der Praxis wird dies mit der Maximierung der Log-Likelihood, zum Beispiel durch Gradientenabstieg auf dem Ausdruck (2.5) erreicht.

Im Folgenden stellen wir vor, wie die Elemente des *density modelling* im Fall des GTM aussehen.

- Die Fehlerfunktionen $G_n(\mathbf{x}, \theta)$ sind so gewählt, dass $p(\mathbf{t} \mid \mathbf{x}, \theta)$ eine in $\boldsymbol{y}(\mathbf{x})$ zentrierte Gauß-Verteilung mit inverser Varianz β ist. Der Parameter β hängt wie das Mapping $\boldsymbol{y}$ von θ ab, so dass β eine abkürzende Schreibweise für $\beta(\theta)$ ist. Es gilt also
  $$p(\mathbf{t} \mid \mathbf{x}, \theta) \sim \mathcal{N}\left(\boldsymbol{y}(\mathbf{x}), \beta^{-1} \boldsymbol{I}\right)$$
  oder
  $$p(\mathbf{t}_n \mid \mathbf{x}, \theta) = \left(\frac{\beta}{2\pi}\right)^{\frac{D}{2}} \exp\left(-\frac{\beta}{2} \|\mathbf{t}_n - \boldsymbol{y}(\mathbf{x})\|^2\right).$$
  Eine andere Betrachtungsweise ist, dass die Fehlerfunktionen G_n bis auf Konstanten dem quadrierten euklidischen Abstand zwischen jeweils einem Datenpunkt und dem Bild eines Punktes aus dem Latent-Space unter $\boldsymbol{y}$ entsprechen. Dies ist nicht unproblematisch, da der sogenannte Concentration-of-Measure-Effekt zu unintuitivem Verhalten der euklidischen Norm in hochdimensionalen Räumen führt, siehe [Ver02]. Eine alternative Norm wird in Kapitel 7 dieser Arbeit vorgestellt.

- Der Prior im Latent-Space hat die Gestalt
  $$p(\mathbf{x}) = \frac{1}{K} \sum_{i=1}^{K} \delta_{\mathbf{x}_i}(\mathbf{x}) = \begin{cases} 0, & \text{wenn } \mathbf{x} \neq \mathbf{x}_i \\ \frac{1}{K}, & \text{wenn } \mathbf{x} = \mathbf{x}_i. \end{cases}$$
  Hierbei dürfen die Latent-Space-Samples $\{\mathbf{x}_i\}_{i=1}^K$ nicht mit den latenten Variablen $(x_1, \ldots, x_L)^T$

verwechselt werden, ein Sample $\mathbf{x}_i \in \mathbb{R}^L$ beschreibt den Zustand von *allen* latenten Variablen. Die Menge $\{\mathbf{x}_i\}_{i=1}^K$ ist auf einem Gitter auf $[0,1]^L$ angeordnet, weshalb wir zukünftig mit Latent-Space $[0,1]^L$ und nicht $\mathbb{R}^L$ meinen.

Den Prior $p(\mathbf{x})$ als Summe von Dirac-Deltas zu wählen, hat den Vorteil, dass das Integral (2.5) in eine Summe zerfällt, also

$$p(\mathbf{t}_n \mid \theta) = \frac{1}{K} \sum_{i=1}^{K} p(\mathbf{t}_n \mid \mathbf{x}_i, \theta)$$

gilt. Die Log-Likelihood (2.4) hat dann die Gestalt

$$\mathcal{L}(\theta) = \sum_{n=1}^{N} \log \left(\frac{1}{K} \sum_{i=1}^{K} p(\mathbf{t}_n \mid \mathbf{x}_i, \theta) \right).$$

Hier wird deutlich, dass das GTM ein *constrained mixture model* ist: Die Datendichte wird durch K Gauß-Kerne approxmiert, die in ihrer Bewegungsfreiheit durch das Mapping $\boldsymbol{y}$ eingeschränkt sind. Genau diese Einschränkung führt dazu, dass die Topologie der Daten in der L-dimensionalen Projektion weitestgehend erhalten bleibt.

- Das Mapping $\boldsymbol{y} : [0,1]^L \to \mathbb{R}^D$ diskretisieren wir durch ein lineares Modell

$$\boldsymbol{y}(\mathbf{x}) = \boldsymbol{W}\Phi(\mathbf{x}),$$

wobei $\boldsymbol{W} \in \mathbb{R}^{D \times M}$ und $\Phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x}))^T$ gilt. Die $\phi_j(\mathbf{x}) : [0,1]^L \to \mathbb{R}$ sind in der Regel nichtlineare Basisfunktionen. In [BSW98b, Sve98] werden Gauß-Kerne als Basisfunktionen vorgeschlagen, die um eine Teilmenge der $\{\mathbf{x}_i\}_{i=1}^K$ zentriert sind. Die Abstimmung von der Anzahl der Basisfunktionen, deren Varianz σ^2 und der Anzahl K an Gitterpunkten ist entscheidend für eine erfolgreiche Dimensionsreduktion.

- Als Optimierungsalgorithmus wird der Expectation Maximization (EM)-Algorithmus eingesetzt. Diese Wahl ist typisch für ein *mixture model* von Gauß-Kernen. In dem folgenden Unterabschnitt 2.3.2 wird der EM-Algorithmus einschließlich eines Konvergenzbeweises vorgestellt.

2.3.2 Expectation Maximization-Algorithmus

Der EM-Algorithmus wird eingesetzt, um bei gegebenen Werten der Zufallsvariablen Z eine Likelihood-Maximierung der Modellparameter θ durchzuführen. Hierbei seien Y und Z diskrete Zufallsvariablen, die in den Raum S abbilden. Beabsichtigt ist eine kompakte oder instruktive Darstellung von Z durch die Variable Y. Die folgende sehr allgemeine Darstellung ist angelehnt an [NH98] und wird zu einem späteren Zeitpunkt auf das GTM angewandt.

Die gemeinsame Verteilung von Y und Z sei mit θ parametrisiert. Für Z hat die Randverteilung die Gestalt

$$P(z \mid \theta) = \sum_{y \in S} P(y, z \mid \theta),$$

wobei $z \in S$ gilt. Zu gegebenem $z \in S$ suchen wir das θ, welches $\mathcal{L}(\theta) = \log P(z \mid \theta)$ maximiert. Der EM-Algorithmus wird mit einem $\theta^{(0)}$ initialisiert und wiederholt für $s = 1, 2, \ldots$ folgende Schritte:

- **E-Schritt:**
  Bestimme die Verteilung $\mu^{(s)}(y), y \in S$ für Y mit

$$\mu^{(s)}(y) = P\left(y \mid z, \theta^{(s-1)}\right).$$

- **M-Schritt:**
  Bestimme $\theta^{(s)}$ als das θ, welches

$$\mathbb{E}_{\mu^{(s)}}\left[\log P\left(y, z \mid \theta\right)\right]$$

  maximiert.

Der E-Schritt entspricht hierbei der Ermittlung der wahrscheinlichsten Verteilung für Y, während der M-Schritt eine Maximum-Likelihood-Bestimmung des Parameters θ durchführt. Das Verfahren setzt voraus, dass die Bestimmung des optimalen θ unter der Annahme einer vorgegebenen Y-Verteilung einfach ist. Im Folgenden werden wir beweisen, dass in jeder Iteration die Log-Likelihood $\mathcal{L}(\theta)$ erhöht wird.

Definition 2.1 (EM-Funktional)**.** Sei μ die Verteilung von Y und θ ein Parametersatz. Wir definieren das EM-Funktional

$$\mathcal{F}(\mu, \theta) := E_\mu\left[\log P(y, z \mid \theta)\right] + H(\mu), \tag{2.6}$$

wobei $H(\mu) = -\mathbb{E}_\mu\left[\log \mu(y)\right]$ die Entropie der Zufallsvariablen Y mit Verteilung μ ist, siehe zum Entropiebegriff auch Abschnitt 3.1.

Bis auf das Vorzeichen entspricht die Funktion $\mathcal{F}$ der Helmholtzschen freien Energie, die eine wichtige Rolle in der statistischen Physik spielt. Wenn wir bei dieser Interpretation bleiben, entsprechen die physikalischen Zustände den Werten von Y, während die Energie eines Zustands $-\log P(y, z \mid \theta)$ ist. Die folgenden beiden Lemmata geben Eigenschaften von $\mathcal{F}$ wieder, die bekannten Fakten aus der statistischen Physik entsprechen:

- Die Boltzmann-Verteilung über die Zustände minimiert die freie Energie und
- die freie Energie hängt mit dem Logarithmus der kanonischen Zustandssumme zusammen.

Lemma 2.2. *Für festes θ existiert eine eindeutige Verteilung von Y, die $\mathcal{F}(\mu, \theta)$ maximiert. Sie ist durch $\mu_\theta(y) = P(y \mid z, \theta)$ gegeben und hängt stetig von θ ab.*

Beweis. Die Lösung μ_θ des Maximierungsproblems befindet sich an einem kritischen Punkt von $\mathcal{F}(\mu, \theta)$ mit der Nebenbedingung $\sum_{y \in S} \mu_\theta(y) = 1$ und kann mit Hilfe eines Lagrange-Multiplikators gefunden werden. Hierbei steht der Gradient von $\mathcal{F}$ normal auf der durch die Nebenbedingung definierten Oberfläche, so dass ein festes λ existiert mit

$$\lambda = \frac{\partial \mathcal{F}}{\partial \mu_\theta(y)}(\mu_\theta, \theta) = \log P(y, z \mid \theta) - \log \mu_\theta(y) - 1$$

für alle $y \in S$. Wir folgern, dass die $\mu_\theta(y)$, welche $\mathcal{F}(\cdot, \theta)$ maximieren, proportional zu $P(y, z \mid \theta)$ sind. Die Normalisierung $\sum_{y \in S} \mu_\theta(y) = 1$ ergibt, dass $\mu_\theta(y) = P(y \mid z, \theta)$ als einzige Lösung in Frage kommt. Wenn $P(y, z \mid \theta)$ stetig von θ abhängt, gilt dies auch für $\mu_\theta(y)$. □

Lemma 2.3. *Sei $\mu(y) = P(y \mid z, \theta)$. Dann gilt $\mathcal{F}(\mu, \theta) = \log P(z \mid \theta)$.*

Beweis.

$$\begin{aligned}
\mathcal{F}(\mu, \theta) &= \mathbb{E}_\mu \left[\log P(y, z \mid \theta)\right] + H(\mu) \\
&= \mathbb{E}_\mu \left[\log P(y, z \mid \theta)\right] - \mathbb{E}_\mu \left[\log P(y \mid z, \theta)\right] \\
&= \mathbb{E}_\mu \left[\log P(y, z \mid \theta) - \log P(y \mid z, \theta)\right] \\
&= \mathbb{E}_\mu \left[\log P(z \mid \theta)\right] \\
&= \log P(z \mid \theta)
\end{aligned}$$

□

Nun können wir für den EM-Algorithmus eine alternative Formulierung wählen:

- **E-Schritt:**
  Setze $\mu^{(s)}$ auf das μ, welches
  $$\mathcal{F}(\mu, \theta^{(s-1)})$$
  maximiert.
- **M-Schritt:**
  Setze $\theta^{(s)}$ auf das θ, welches
  $$\mathcal{F}(\mu^{(s)}, \theta)$$
  maximiert.

Satz 2.4. *Die Iterationen sind äquivalent zu dem E- und M-Schritt der klassischen Formulierung.*

Beweis. Dass die E-Schritte äquivalent sind, ist eine direkte Folge von Lemma 2.2. Dass die M-Schritte gleich sind, folgt daraus, dass der Entropieterm $H(\mu)$ in Gleichung (2.6) nicht von θ abhängt, und somit $\mathbb{E}_\mu \left[\log P(y, z \mid \theta)\right]$ wie im M-Schritt der ersten Formulierung maximiert wird. □

In dieser Formulierung ist es offensichtlich, dass der Algorithmus mit Ausnahme von Sattelpunkten zu den μ^* und θ^* konvergiert, die $\mathcal{F}$ lokal maximieren. Der folgende Satz stellt den Bezug zu $\mathcal{L}(\theta)$ her.

Satz 2.5. *Wenn $\mathcal{F}(\mu, \theta)$ ein lokales Maximum bei μ^* und θ^* hat, hat auch $\mathcal{L}(\theta)$ ein lokales Maximum bei θ^*. Für ein globales Maximum μ^* und θ^* hat auch $\mathcal{L}(\theta)$ ein globales Maximum bei θ^*.*

Beweis. Durch die Kombination der Lemmata 2.2 und 2.3 erkennen wir, dass $\mathcal{L}(\theta) = \log P(z \mid \theta) = \mathcal{F}(\mu_\theta, \theta)$ für beliebige θ gilt. Insbesondere gilt dies für $\mathcal{L}(\theta^*) = \mathcal{F}(\mu_{\theta^*}, \theta^*) = \mathcal{F}(\mu^*, \theta^*)$. Um zu zeigen, dass θ^* ein lokales Maximum von $\mathcal{L}$ ist, müssen wir zeigen, dass in der Nähe

kein $\theta^\dagger$ existiert mit $\mathcal{L}(\theta^\dagger) > \mathcal{L}(\theta^*)$. Wäre dem so, würde $\mathcal{F}(\mu^\dagger, \theta^\dagger) > \mathcal{F}(\mu^*, \theta^*)$ gelten, wobei $\mu^\dagger = \mu_{\theta^\dagger}$ ist. Da μ_θ stetig von θ abhängt, müsste $\mu^\dagger$ in der Umgebung von μ^* liegen, was ein Widerspruch zur lokalen Maximalität von $\mathcal{F}(\mu^*, \theta^*)$ ist. Der Beweis für das globale Maximum erfolgt analog und kommt ohne Stetigkeitsforderung aus. □

2.3.3 Umsetzung des EM-Algorithmus

Nachdem in Unterabschnitt 2.3.2 der EM-Algorithmus in allgemeiner Form dargestellt wurde, wird er nun auf das GTM-Modell aus Unterabschnitt 2.3.1 angewendet. Anzumerken ist, dass der EM-Algorithmus zunächst keinen Prior über die Modellparameter θ berücksichtigt, obwohl dieser im GTM-Modell für die Anwendung des Satz von Bayes notwendig ist. Wir stellen den zunächst unregularisierten GTM-Algorithmus aus [BSW98b] vor und verweisen darauf, dass sich der Regularisierungsterm problemlos in den M-Schritt integrieren lässt.

Zur Übersichtlichkeit werden in der folgenden Tabelle 2.1 die verwendeten Bezeichner zusammengefasst.

Bezeichner	Bedeutung
D	Dimension des Datenraums
L	Dimension des Latent-Space
$\mathbf{x}$	L-dimensionaler Punkt im Latent-Space
$\mathbf{t}$	D-dimensionaler Datenpunkt im Datenraum
N	Anzahl der Datenpunkte
M	Anzahl der Basisfunktionen
K	Anzahl der Gitterpunkte im Latent-Space
$\phi_j(x)$	Basisfunktion $\phi_j : [0,1]^L \to \mathbb{R}$
$\Phi(x)$	Basisfunktionsvektor $(\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x}))^T$
$\boldsymbol{y}(\mathbf{x})$	Mapping von $\mathbf{x}$ in den Datenraum
$\boldsymbol{W}$	Parametermatrix für Mapping $\boldsymbol{y}(\mathbf{x}) = \boldsymbol{W}\Phi(\mathbf{x})$ mit $\boldsymbol{W} \in \mathbb{R}^{D \times M}$
β	Inverse Varianz des Rauschens vom Bild von $\boldsymbol{y}$
θ	Modellparameter, die die Matrix $\boldsymbol{W}$ und die inverse Varianz β festlegen

Tabelle 2.1: Bezeichner und ihre Bedeutung im GTM-Kontext

Folgende Schritte setzen das GTM in der klassischen Form um:

- **Initialisierung** $s = 0$**:**
  Zu Beginn muss ein $\theta^{(0)}$ bestimmt werden. Hierzu führen wir eine PCA auf den Daten durch und initialisieren $\boldsymbol{y}^{(0)}$ so, dass das Bild $\boldsymbol{y}^{(0)}\left([0,1]^L\right)$ eine Hyperebene im Datenraum ist, die sich entlang der L Hauptachsen mit der größten Varianz orientiert. Die inverse Varianz $\beta^{(0)}$ sollte sich in der gleichen Größenordnung wie die Varianzen des PCA-Modells bewegen.

- **E-Schritt** $s \to s + 1$**:**
  Zu gegebenen Modellparametern $\theta^{(s)}$ werden die Posterior-Verteilungen der latenten Variablen bestimmt. Sie werden beim GTM *Responsibilities* genannt. Für den Datenpunkt $\mathbf{t}_n$

und Vektor $\mathbf{x}_i$ im Latent-Space gilt

$$R^{(s)}(\mathbf{t}_n, \mathbf{x}_i) = \frac{p(\mathbf{t}_n \mid \mathbf{x}_i, \theta^{(s)})}{\sum_{i'=1}^{K} p(\mathbf{t}_n \mid \mathbf{x}_{i'}, \theta^{(s)})}. \tag{2.7}$$

- **M-Schritt** $s+1$**:**
  Auf Grundlage der Posterior-Verteilungen $R^{(s)}(\mathbf{t}_n, \cdot)$ können nun die Parameter $\theta^{(s+1)}$, also $\boldsymbol{y}^{(s+1)}$ und $\beta^{(s+1)}$, eindeutig bestimmt werden. Die Log-Likelihood mit eingesetzten *Responsibilities* hat die Gestalt

$$\mathcal{L}_R(\theta^{(s+1)}) = \sum_{n=1}^{N} \sum_{i=1}^{K} R^{(s)}(\mathbf{t}_n, \mathbf{x}_i) \log p(\mathbf{t}_n \mid \mathbf{x}_i, \theta^{(s+1)}). \tag{2.8}$$

  Die Abbildung $\boldsymbol{y}^{(s+1)}$ lässt sich in diskretisierter Form durch die Parametermatrix $\boldsymbol{W}^{(s+1)}$ beschreiben. Einsetzen in Formel (2.8) und Ableiten nach jedem Matrixeintrag führt zur Gleichung

$$\sum_{n=1}^{N} \sum_{i=1}^{K} R^{(s)}(\mathbf{t}_n, \mathbf{x}_i) \left(\boldsymbol{W}^{(s+1)} \Phi(\mathbf{x}_i) - \mathbf{t}_n \right) \Phi^T(\mathbf{x}_i) = 0,$$

  welche sich auch als Matrixgleichung

$$\boldsymbol{\Phi}^T \boldsymbol{G} \boldsymbol{\Phi} \boldsymbol{W}^T = \boldsymbol{\Phi}^T \boldsymbol{R} \boldsymbol{T} \tag{2.9}$$

  schreiben lässt. Hierbei ist $\boldsymbol{\Phi}$ eine $K \times M$-Matrix mit $(\boldsymbol{\Phi})_{ij} = \phi_j(x_i)$, $\boldsymbol{T}$ eine $N \times D$-Matrix mit $(\boldsymbol{T})_{nk} = (\mathbf{t}_n)_k$, $\boldsymbol{R}$ eine $K \times N$-Matrix mit $(\boldsymbol{R})_{in} = R^{(s)}(\mathbf{t}_n, \mathbf{x}_i)$ und $\boldsymbol{G}$ eine $K \times K$-Diagonalmatrix mit

$$(\boldsymbol{G})_{ii} = \sum_{n=1}^{N} R^{(s)}(\mathbf{t}_n, \mathbf{x}_i).$$

  $\beta^{(s+1)}$ lässt sich anschließend mit

$$\frac{1}{\beta^{(s+1)}} = \frac{1}{ND} \sum_{n=1}^{N} \sum_{i=1}^{K} R^{(s)}(\mathbf{t}_n, \mathbf{x}_i) \left\| \boldsymbol{W}^{(s+1)} \Phi(\mathbf{x}_i) - \mathbf{t}_n \right\|^2$$

  exakt bestimmen.

2.3.4 Niederdimensionale Projektion

Es gibt zwei Möglichkeiten, einen Punkt $\mathbf{t}_n \in \mathbb{R}^D$ in den Latent-Space einzubetten:

- **Mean-Projektion**
  Wir projizieren $\mathbf{t}_n$ auf den Erwartungswert seiner Posterior-Verteilung im Latent-Space, also

$$\mathbb{E}[\mathbf{x} \mid \mathbf{t}_n, \theta] = \sum_{i=1}^{K} p(\mathbf{x}_i \mid \mathbf{t}_n, \theta) \mathbf{x}_i = \sum_{i=1}^{K} R(\mathbf{t}_n, \mathbf{x}_i) \mathbf{x}_i.$$

- **Mode-Projektion**
  Wir projizieren $\mathbf{t}_n$ auf einen Punkt $\mathbf{x}_i \in [0,1]^L$ im Latent-Space, wobei wir i mit

$$\arg\max_i R(\mathbf{t}_n, \mathbf{x}_i)$$

  bestimmen.

2.4 Einordnung von Verfahren

In Abschnitt 2.1 sind mögliche Funktionalitäten von Verfahren zur Dimensionsreduktion vorgestellt worden. In diesem Abschnitt behandeln wir weitere charakteristische Eigenschaften, die Verleysen in [LV07] beschreibt, und ordnen die PCA und das GTM entsprechend ein.

2.4.1 Harte oder weiche Dimensionsreduktion

Probleme, in denen die Daten hunderte oder sogar tausende Dimensionen haben, erfordern eine „harte Dimensionsreduktion“ (*hard dimensionality reduction*). Zu dieser Klasse gehören Klassifizierungs- und Mustererkennungsprobleme im Zusammenhang mit Bildern oder Sprache. Die PCA ist wegen ihres simplen Modells und den wenigen Parametern hierfür sehr gut geeignet.

„Weiche Dimensionsreduktion“ (*soft dimensionality reduction*) wird bei Daten verwendet, die nicht zu hochdimensional sind. Bei den einzelnen Koordinaten der Daten handelt es sich in der Regel um verschiedene Variablen mit eindeutiger Interpretation. Das GTM ist der „weichen Dimensionsreduktion“ zuzurechnen.

2.4.2 Traditionelles oder generatives Modell

Das Modell eines Verfahrens beschreibt die Verbindung zwischen latenten und beobachteten Variablen. Es ist bemerkenswert, dass die Verbindung in zwei Richtungen gehen kann: von den latenten zu den beobachteten Variablen oder von den beobachteten zu den latenten. Letztere Verbindung ist häufiger und intuitiver: Wir möchten aus den Beobachtungen die latenten Variablen rekonstruieren.

Generative Modelle (*generative models*) hingegen modellieren die Beobachtungen als eine Funktion der latenten Variablen. Dieser etwas komplexere Ansatz ist näher am Prozess, wie Daten tatsächlich generiert werden. Es ist jedoch nötig, abwechselnd die latenten Variablen und die Beobachtungen zu betrachten, um die Modellparameter zu bestimmen. Wie man bereits am Namen erkennt, gehört das Generative Topographic Mapping dieser Verfahrensklasse an. Dies gilt auch für die PCA, wobei hier die Bestimmung der Modellparameter einfach ist.

2.4.3 Lineares oder nichtlineares Modell

Die Verfahren zur Dimensionsreduktion lassen sich unterteilen in solche, deren Modelle linear sind, und solche, deren Modelle auch Nichtlinearitäten in den Daten erfassen können.

Die PCA ist ein Beispiel für ein lineares Verfahren. Als solches hat es den Vorteil, dass die Anzahl der Modellparameter klein ist. Die Projektion von D-dimensionalen Daten auf eine

L-dimensionale Ebene hat $\mathcal{O}(L \cdot D)$ Parameter und wird bereits durch $L + 1$ Datenpunkte eindeutig bestimmt.

Das GTM ist als nichtlineares Verfahren wesentlich mächtiger, was jedoch auch einige Nachteile mit sich bringt. Das GTM hat eine wesentlich höhere Anzahl von Parametern, die in der Regel exponentiell von L abhängt. Dies bedeutet zum einen einen hohen Rechenaufwand und zum anderen, dass wesentlich mehr Datenpunkte zur Parameterbestimmung erforderlich sind. Zudem führt die Arbeit mit Nichtlinearitäten in aller Regel dazu, dass das Verfahren nur lokale Optima findet.

2.4.4 Kontinuierliches oder diskretes Modell

Während einige Modelle ein kontinuierliches Mapping zwischen latenten Variablen und Beobachtungen erzeugen, existieren andere, die eine Abbildung nur auf einer endlichen Menge von Punkten bestimmen. Die PCA und das GTM erzeugen beide ein kontinuierliches Mapping. Die SOM [Koh82] hingegen hat ein diskretes Modell.

Ein kontinuierliches Modell ist insofern wünschenswert, als dass man die Ergebnisse auf Datenpunkte übertragen kann, die nicht bei der Bestimmung der Modellparameter beteiligt waren. Während bei der PCA und dem GTM jeder beliebige Punkt mit dem konstruierten Mapping eingebettet werden kann, müssen bei diskreten Verfahren neue Punkte mit Hilfe bereits bekannter Punkte interpoliert werden. Eine Besonderheit in Bezug auf das GTM ist, dass zwar ein kontinuierliches Mapping existiert, der Latent-Space dennoch durch die $\{\mathbf{x}_i\}_{i=1}^{K}$ diskret ist.

2.4.5 Implizites oder explizites Mapping

Dieses Kriterium hängt eng mit dem vorherigen Punkt zusammen. Ein explizites Mapping gibt für jeden Datenpunkt seine niederdimensionale Repräsentation an. Das PCA und das GTM hingegen konstruieren ein implizites Mapping, da die Modellparameter eine Abbildung definieren und nicht explizit für jeden Datenpunkt seine niederdimensionale Darstellung angeben.

Verfahren zur Vektorquantisierung sind prinzipiell explizit, bilden jedoch wieder eine eigene Klasse, da sie mehreren Datenpunkten eine gemeinsame niederdimensionale Projektion zuordnen.

2.4.6 Integrierte oder externe Bestimmung der Dimensionalität

In Abschnitt 2.1.1 wurde bereits auf die Fähigkeit, die Anzahl der latenten Variablen L zu bestimmen, eingegangen. Die allermeisten Verfahren wie auch das GTM sind nicht in der Lage, eine Dimensionsschätzung vorzunehmen, und erwarten L als einen externen Hyperparameter. Die PCA bildet hier eine Ausnahme, da sie anhand der Eigenwerte der Kovarianzmatrix die Anzahl der latenten Variablen bestimmen kann. Dies wurde bereits in Unterabschnitt 2.2.2 beschrieben.

Bisher wurden die Begriffe „Anzahl latenter Variablen“ und „intrinsische Dimensionalität“ als Synonyme gebraucht. Die Anzahl latenter Variablen des Verfahrens entspricht der Einbettungsdimension, diese muss jedoch nicht gleich der intrinsischen Dimension der Daten sein.

- Wenn die Einbettungsdimensions kleiner als die intrinsische Dimensionalität der Daten ist, erzeugt dies in der Regel „schlechtere“ Ergebnisse. In einigen Fällen ist dies dennoch

sinnvoll, zum Beispiel wenn eine zwei- oder dreidimensionale Darstellung zur Visualisierung erwünscht ist.

- Die Einbettungsdimension sollte größer als die intrinsische Dimensionalität der Daten sein, wenn in den Daten Nichtlinearitäten enthalten sind, die das Modell nicht anders auflösen kann. Die PCA könnte beispielsweise einen Kreis im $\mathbb{R}^2$ nicht als eindimensionale Struktur erkennen, sondern würde zwei latente Variablen benötigen, um ihn verlustfrei rekonstruieren zu können.

2.4.7 Geschichtete oder eigenständige Einbettung

Bei einer „geschichteten Einbettung“ (*layered embedding*) sind zwei Einbettungen mit den Dimensionen L und L' in den ersten $\min(L, L')$ Dimensionen identisch. Dies bedeutet, dass jede Einbettung von D-dimensionalen Daten mit L Dimensionen durch eine D-dimensionale Einbettung und dem Entfernen der letzten $D - L$ Dimensionen erzielt werden kann. Dies ist charackteristisch für spektrale Methoden, und auch die PCA ist dieser Klasse zuzuordnen: Das Hinzunehmen oder Entfernen von Dimensionen betrifft die restlichen Dimensionen nicht, weil die Hauptachsen orthogonal sind.

Andere Verfahren, das GTM eingeschlossen, müssen in der Regel für verschiedene Einbettungsdimensionen neu gestartet werden.

2.4.8 Ein oder mehrere Koordinatensysteme

Dimensionsreduktion bedeutet nicht zwingend, dass höherdimensionale Daten in genau ein Koordinatensystem eingebettet werden. Möglicherweise gibt es verschiedene Bereiche in den Daten mit unterschiedlich hoher intrinsischer Dimensionalität. Es ist auch denkbar, dass zwei Modelle je eine Hälfte der Daten perfekt erklären können, jedoch keines deren Gesamtheit. Hier hilft das Aufteilen der Daten und die Einbettung in verschiedene niederdimensionale Koordinatensysteme.

Gerade bei der PCA ist dies empfehlenswert, da das Modell sehr restriktiv ist, aber auf linearen Teilbereichen der Daten sehr gute Ergebnisse erzielt. Das Verfahren einer lokalen PCA wird in [KL97] dargestellt. Auch für das GTM existieren Erweiterungen auf mehrere Koordinatensysteme, siehe beispielsweise [TN01].

2.4.9 Optionale oder erforderliche Vektor-Quantisierung

Bei einer sehr großen Datenmenge kann es ratsam sein, die Dimensionsreduktion mit einer kleineren Menge von repräsentativen Beobachtungen durchzuführen. Auf diese Weise soll bei wesentlich geringerem Rechenaufwand die Verteilung der Punkte im Datenraum weitestgehend erhalten bleiben.

Der wünschenswerte Fall, dass mehr Daten als benötigt vorliegen, tritt in der Realität eher selten auf. Deshalb wird die Vektor-Quantisierung meist nicht durchgeführt. Bei der SOM hingegen ist sie Teil des Verfahrens.

2.4.10 Batch- oder Online-Algorithmus

Es wird zwischen Batch- und Online-Algorithmen unterschieden. Für den Lernprozess verlangt ein Batch-Algorithmus, dass alle Datenpunkte vorliegen, während Online-Algorithmen eine bestehende Lösung um neue, bisher unbekannte Datenpunkte ergänzen kann. Das GTM ist den Verfahren mit Batch-Algorithmen zuzurechnen, wobei in [BSW98a] eine inkrementelle Variante beschrieben wird. Die PCA verwendet ebenfalls einen Batch-Algorithmus, obwohl auch Online-Varianten existieren.

2.4.11 Exakte oder approximative Optimierung

Ob exakt oder approximativ optimiert wird, hängt eng mit dem Batch- oder Online-Charakter eines Verfahrens zusammen. Batch-Algorithmen verwenden meist eine algebraische Prozedur, mit der sich die Lösung in geschlossener Form angeben lassen kann. Hierbei ist die gefundene Lösung exakt, das heisst optimal. Die PCA ist hierfür ein Beispiel.

Bei Online-Algorithmen wird meist eine stochastische Gradientenmethode verwendet, wie sie auch in [Mac95] beschrieben wird. Diese Verfahren haben mitunter keine garantierte Konvergenz, und sind somit der approximativen Optimierung zuzurechnen. Allerdings können die Online-Varianten zu Batch-Varianten mit garantierter Konvergenz modifiziert werden, siehe [RM51]. Das Problem, dass nur ein lokales Optimum gefunden wird, bleibt jedoch bestehen.

Approximative Verfahren sind den exakten in Hinblick auf die Komplexität der Zielfunktion überlegen. Während sich die Modellparameter der PCA exakt und schnell bestimmen lassen, hat das GTM wesentlich mehr Freiheitsgrade und kann Nichtlinearitäten erkennen – zu dem Preis, dass der Rechenaufwand höher ist und die Lösung meist nur lokal optimal ist.

2.4.12 Das zu optimierende Kriterium

Die wohl wichtigste Charakteristik eines Verfahrens ist das bei der Dimensionsreduktion optimierte Kriterium. Die Dimensionsreduktion folgt meistens geometrischen Erwägungen: Man nimmt an, dass die verrauschten Datenpunkte auf einer niederdimensionalen Struktur oder Mannigfaltigkeit liegen. Um diese Mannigfaltigkeit unabhängig vom verwendeten Koordinatensystem zu charakterisieren, werden relative Beziehungen zwischen den Datenpunkten betrachtet. Es gibt zwei große Klassen von Verfahren, je nachdem welche Art von Beziehung berücksichtigt wird.

- Ein Kriterum ist, dass die paarweisen Abstände der Punkte bei der niederdimensionalen Projektion weitestgehend erhalten bleiben. Hierdurch werden wesentliche Eigenschaften der Mannigfaltigkeit auch im Niederdimensionalen beibehalten. Da die PCA neben der Dekorrelation auch den quadratischen Rekonstruktionsfehler minimiert, gehört sie in diese Kategorie.

- Das GTM hingegen ist den topologieerhaltenden Verfahren zuzuordnen. Wie wir bereits in Abschnitt 2.1 beschrieben haben, beschränken sich diese auf qualitative Aussagen der Art „Punkt A ist näher an Punkt B als an Punkt C“. Der entscheidende Vorteil des GTM gegenüber vielen abstandserhaltenden Verfahren ist, dass nicht der direkte Abstand zweier Punkte entscheidend ist, sondern die relativen Abstände der Punkte auf der Mannigfaltigkeit.

3 Modifizierte Formulierung des GTM

So wie wir das GTM in Abschnitt 2.3 vorgestellt haben, ist es zur zwei- oder dreidimensionalen Einbettung höherdimensionaler Daten gut geeignet. Es existieren jedoch auch spezifische Einschränkungen, die je nach Anwendung relevant sein können.

- Wie in Unterabschnitt 2.3.1 bereits erwähnt wurde, hat der Prior über die Modellparameter nur für endlich diskretisierte Mappings $\boldsymbol{y}$ eine Dichte. Dies ist in der Praxis unproblematisch, lässt jedoch keine Formulierung mit einem unendlichdimensionalen Funktionenraum für $\boldsymbol{y}$ zu.

- Die derzeitige Formulierung enthält nicht viel Theorie zur richtigen Diskretisierung von $\boldsymbol{y}$ und zum Latent-Space Prior $p(\mathbf{x})$. In [BSW98b] wird darauf hingewiesen, dass die Latent-Space-Samples $\{\mathbf{x}_i\}_{i=1}^K$ als Monte-Carlo-Approximation aufgefasst werden können. Zudem sollen „$\mathcal{O}(100)$“ Punkte im 2σ-Abstand vom Zentrum der Gauß-Basisfunktionen liegen. Mit diesen Erfahrungswerten werden gute Ergebnisse erzielt, sie verlieren jedoch beim Übergang zu anderen Diskretisierungen und Quadraturformeln ihre Gültigkeit.

- Die Maximum-Likelihood-Formulierung erwartet eine endliche Menge von unabhängig identisch verteilten Samples. Es wäre interessant, das GTM auch für kontinuierliche Wahrscheinlichkeitsdichten im Datenraum zu formulieren.

- Die Anzahl der Gitterpunkte im Latent-Space K skaliert exponentiell mit der Latent-Space-Dimension L. Dies spielt bei der zweidimensionalen Visualisierung noch keine Rolle. Sollte die Dimensionsreduktion ein Vorverarbeitungsschritt für ein anderes Verfahren sein und Daten drei- oder höherdimensional einbetten, wird dies im Hinblick auf die Laufzeitkomplexität schwierig.

Bevor wir eine modifizierte Formulierung des GTM vorschlagen, fassen wir die relevanten Ergebnisse zusammen. In Unterabschnitt 2.3.1 wurde das GTM stochastisch über die MAP-Bestimmung oder die Penalized-Likelihood-Maximierung motiviert. In Unterabschnitt 2.3.2 wurde gezeigt, dass der Expectation-Maximization-Algorithmus im Grunde ein Funktional $\mathcal{F}(\mu, \theta)$ maximiert. Lemma 2.3 stellt den Bezug zwischen diesem Funktional und der Log-Likelihood her.

In Abschnitt 3.1 werden wir zeigen, dass eine Maximierung der Log-Likelihood im Wesentlichen der Minimierung der Kullback-Leibler-Divergenz von Beobachtungen und GTM-Modell entspricht. Dies motiviert die Definition des GTM-Funktionals als Summe von Kreuzentropie und Regularisierungsterm. In Abschnitt 3.2 werden wir ein Minimierungsverfahren für dieses Funktional vorstellen, welches eine deterministische Variante des EM-Algorithmus darstellt. Dies ermöglicht uns

- die Verwendung des Maßes μ im Datenraum statt einer endlichen Menge von Samples,

- die Verwendung einer uniformen Verteilung im Latent-Space statt eines regulären Gitters und
- eine überwiegend deterministische statt einer rein stochastischen Formulierung, wodurch theoretische Resultate aus der Funktionsrekonstruktion und numerischen Integration anwendbar werden.

Das bisherige GTM geht bei der entsprechenden Wahl der Quadraturregel und der Basisfunktionen aus dieser Formulierung hervor, es werden jedoch effizienzsteigernde Modifikationen möglich.

3.1 Kullback-Leibler-Divergenz und Maximum-Likelihood

Ein sehr grundlegendes Konzept der Informationstheorie ist die Entropie einer Zufallsvariablen. Mit der Entropie wird die mittlere Überraschung über den Ausgang eines Zufallsexperiments gemessen. Im Folgenden erstellen wir eine zusammenfassende Übersicht, für eine umfangreiche Darstellung sei auf [Sha06] verwiesen.

Definition 3.1 (Entropie im Diskreten)**.** Es sei $(\Omega, \mathcal{A}, P)$ ein diskreter Wahrscheinlichkeitsraum und $X\colon \Omega \to S$ eine Zufallsvariable. Die Entropie von X ist

$$H(X) := -\sum_{s\in S} P(X=s)\log P(X=s) = -\mathbb{E}[\log P(X=s)].$$

Definition 3.2 (Entropie im Kontinuierlichen)**.** Es sei $(\Omega, \mathcal{A}, P)$ ein Wahrscheinlichkeitsraum und $X\colon \Omega \to \mathbb{R}$ eine Zufallsvariable mit Verteilung μ. Hierbei sei μ absolut stetig bezüglich des Lebesgue-Maßes. Dann ist die Entropie von X

$$H(X) := -\mathbb{E}_\mu\left[\log d\mu\right].$$

Wie bereits angedeutet wurde, wird mit der Entropie die Unsicherheit über den Ausgang eines Zufallsexperiments gemessen. Eine hohe Entropie bedeutet in diesem Kontext maximale Ungewissheit oder ein hohes Maß an Überraschung, eine niedrige Entropie einen fast sicheren Ausgang des Experiments. Trotz der augenscheinlichen Ähnlichkeit unterscheiden sich die diskrete und die kontinuierliche Entropiedefinition deutlich im Wertebereich. Für endliche S liegt die Entropie in $[0, \log|S|]$, für kontinuierliche Räume in $(-\infty, 0]$.

Das Konzept der Entropie lässt sich verallgemeinern zur Kullback-Leibler-Divergenz oder relativen Entropie, die die Überraschung relativ zu einer bekannten Verteilung misst. Falls das Zufallsexperiment exakt dieser Verteilung folgt, ist die relative Entropie gleich 0. Sie wächst mit steigender Diskrepanz zwischen Annahme und Beobachtungen. Man spricht man von einer „Divergenz“, weil sie nicht symmetrisch ist.

Definition 3.3 (Kullback-Leibler-Divergenz)**.** Seien μ, ν zwei Wahrscheinlichkeitsverteilungen, wobei ν absolut stetig bezüglich μ ist ($\nu \ll \mu$). Dann bezeichnet

$$D(\mu\|\nu) := -\mathbb{E}_\mu\left[\log\frac{d\nu}{d\mu}\right]$$

die Kullback-Leibler-Divergenz von ν und μ. Falls $\nu \ll \mu$ nicht gilt, ist $D(\mu\|\nu) := \infty$.

Lemma 3.4. *$D(\mu\|\nu) \geq 0$. $D(\mu\|\nu) = 0$ gilt genau dann, wenn $\nu = \mu$ fast überall.*

Beweis. Wegen der Jensenschen Ungleichung gilt

$$D(\mu\|\nu) = -\mathbb{E}_\mu \left[\log \frac{d\nu}{d\mu}\right] \geq -\log \mathbb{E}_\mu \left[\frac{d\nu}{d\mu}\right] = -\log 1 = 0.$$

Bei Gleichheit ist $\frac{d\nu}{d\mu}$ fast sicher konstant, was den zweiten Teil des Lemmas beweist. □

Sind μ und ν Verteilungen auf dem $\mathbb{R}^D$ mit Dichtefunktionen $p(\mathbf{t})$ und $q(\mathbf{t})$, so gilt

$$\begin{aligned} D(\mu\|\nu) &= \int_{\mathbb{R}^D} p(\mathbf{t}) \log \frac{p(\mathbf{t})}{q(\mathbf{t})} d\mathbf{t} && (3.1)\\ &= \int_{\mathbb{R}^D} p(\mathbf{t}) \log p(\mathbf{t}) d\mathbf{t} - \int_{\mathbb{R}^D} p(\mathbf{t}) \log q(\mathbf{t}) d\mathbf{t} && (3.2)\\ &= -H(\mu) - \int_{\mathbb{R}^D} p(\mathbf{t}) \log q(\mathbf{t}) d\mathbf{t} && (3.3) \end{aligned}$$

Diese Form des Abstands zwischen zwei Wahrscheinlichkeitsdichten ist ein Speziallfall der Csiszár-Divergenz, wie sie in [AS06] beschrieben wird.

Definition 3.5 (Csiszár-Divergenz)**.** Sei $h : \mathbb{R} \to \mathbb{R}$ eine konvexe linksseitig stetige Funktion, und p und q zwei Wahrscheinlichkeitsdichten auf $\mathbb{R}^D$. Dann ist die Csiszár-Divergenz definiert als

$$C_h(p, q) = \int_{\mathbb{R}^D} p(\mathbf{t}) h\left(\frac{q(\mathbf{t})}{p(\mathbf{t})}\right) d\mathbf{t}.$$

Offenbar ergibt sich die Kullback-Leibler-Divergenz aus der Csiszár-Divergenz mit $h(\xi) = -\log \xi$. Die Kullback-Leibler-Divergenz mit vertauschten Parametern ergibt sich aus $h(\xi) = \xi \log \xi$. Mit dieser Verallgemeinerung des Divergenz-Begriffs ist die Möglichkeit verbunden, bestehende Verfahren mit anderen Divergenzen auf neue Aufgabengebiete anzuwenden. Der Kullback-Leibler-Divergenz kommt jedoch eine besondere Bedeutung zu, weil sie eng mit der Log-Likelihood verbunden ist.

Die MAP-Methode aus Unterabschnitt 2.3.1 führt zu dem zu maximierenden Funktional

$$\begin{aligned} \mathcal{F}(\theta) &= \log p(\mathcal{T} \mid \theta) && (3.4)\\ &= \underbrace{\sum_{n=1}^{N} \log p(\mathbf{t}_n \mid \theta)}_{\text{Log-Likelihood}} + \underbrace{\log p(\theta)}_{\text{Regularisierung}} - \underbrace{\log p(\mathcal{T})}_{\text{Konstante}}. && (3.5) \end{aligned}$$

Sei ν_θ die Wahrscheinlichkeitsverteilung im Datenraum mit Dichte $g_\theta(\mathbf{t})$, die das GTM mit Parametersatz θ erzeugt. Zusätzlich sei μ die Wahrscheinlichkeitsverteilung im Datenraum, die wir durch die empirische Verteilung der Samples

$$\frac{1}{N} \sum_{n=1}^{N} \delta_{\mathbf{t}_n}(\mathbf{t})$$

approximieren wollen. Die empirische Verteilung ist nicht absolut-stetig bezüglich des Lebesgue-Maßes, wir können jedoch eine Wahrscheinlichkeitsdichte $f(\mathbf{t})$ angeben, die diese Summe von Punktmaßen beliebig genau annähert, so dass

$$-\int_{\mathbb{R}^D} f(\mathbf{t}) \log g_\theta(\mathbf{t}) d\mathbf{t} \approx -\frac{1}{N} \sum_{n=1}^{N} \log g_\theta(\mathbf{t}_n) \tag{3.6}$$

gilt. Die durch f erzeugte Verteilung nennen wir μ_f. Die Gleichung (3.3) ergibt mit Einsetzen von (3.5) und (3.6)

$$\begin{aligned} D(\mu_f \| \nu_\theta) + \underbrace{H(\mu_f)}_{Konstante} &\approx -\frac{1}{N} \sum_{n=1}^{N} \log g_\theta(\mathbf{t}_n) && (3.7) \\ &= -\frac{1}{N} \sum_{n=1}^{N} \log p(\mathbf{t}_n \mid \theta) && (3.8) \\ &= -\frac{1}{N} \mathcal{F}(\theta) + \underbrace{\frac{1}{N} \log p(\theta)}_{\text{- Regularisierung}} - \underbrace{\frac{1}{N} \log p(\mathcal{T})}_{\text{Konstante}}. && (3.9) \end{aligned}$$

An der Zeile (3.9) erkennen wir, dass eine Maximierung der Bayesschen Formulierung des GTM eine Minimierung der regularisierten Kullback-Leibler-Divergenz von GTM-Dichte und Datenraumdichte ist. Diese Dichterekonstruktion ist neben der Penalized-Likelihood-Maximierung und der Bayesschen Formulierung eine dritte Perspektive auf das GTM.

3.2 GTM-Funktional

Der vorherige Abschnitt motiviert die Definition des folgenden zu minimierenden GTM-Funktionals.

Definition 3.6 (GTM-Funktional)**.** Sei μ die Wahrscheinlichkeitsverteilung der Datenpunkte. Das GTM mit Parametersatz θ erzeuge die Verteilung ν_θ mit Dichte $g_\theta(\mathbf{t})$ im Datenraum. Wir definieren das GTM-Funktional

$$\mathcal{G}(\theta) := \underbrace{-\int_{\mathbb{R}^D} \log g_\theta(\mathbf{t}) d\mu(\mathbf{t})}_{\text{Kreuzentropie}} + \underbrace{\lambda \cdot S(\theta)}_{\text{Regularisierung}} . \tag{3.10}$$

Die Kreuzentropie entspricht hierbei $D(\mu \| \nu_\theta) + H(\mu)$.

Der bisherige Prior $p(\mathbf{x}) = \frac{1}{K} \sum_{i=1}^{K} \delta_{\mathbf{x}_i}(\mathbf{x})$ ist gitterartig angelegt. Ein Nachteil dieser Konstruktion ist die exponentielle Abhängigkeit von der Dimension des Latent-Space, denn die Anzahl der $\mathbf{x}_i$ skaliert exponentiell mit der Raumdimension L. Wir verwenden nun eine uniforme Verteilung, die auch der Grenzwert der Gitterverteilungen unter schwacher Konvergenz ist:

$$p(\mathbf{x}) := \begin{cases} 1, & \text{wenn } \mathbf{x} \in [0,1]^L \\ 0 & \text{sonst.} \end{cases}$$

Definition 3.7 (GTM-Dichte)**.** Der Parametersatz θ definiere ein Mapping $\boldsymbol{y} : [0,1]^L \to \mathbb{R}^D$ und eine inverse Varianz β. Die GTM-Dichte im Datenraum zum Parameter θ ist

$$g_\theta(\mathbf{t}) := \left(\frac{\beta}{2\pi}\right)^{\frac{D}{2}} \int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\mathbf{t}_n - \boldsymbol{y}(\mathbf{x})\|^2\right) d\mathbf{x}. \tag{3.11}$$

Wir setzen nun die Dichte des GTM aus Definition 3.7 in das GTM-Funktional ein. Ausserdem verwenden wir nicht mehr die Notation θ für die Modellparameter, sondern schreiben zukünftig $\boldsymbol{y}$ und β aus. Es folgt

$$\begin{aligned} \mathcal{G}(\boldsymbol{y},\beta) &= -\int_{\mathbb{R}^D} \log \int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\mathbf{t} - \boldsymbol{y}(\mathbf{x})\|^2\right) d\mathbf{x} d\mu(\mathbf{t}) && (3.12)\\ &\quad -\frac{D}{2}\log\frac{\beta}{2\pi} + \lambda \cdot S(\boldsymbol{y}). && (3.13) \end{aligned}$$

In dieser Form lässt sich $\mathcal{G}$ nicht gut minimieren. In Lemma 2.3 des EM-Algorithmus wurde der Zusammenhang zwischen dem Logarithmus der kanonischen Zustandssumme und der Helmholtzschen freien Energie hergestellt. Eine ähnliche Umformung rechnen wir nun im Kontinuierlichen nach. Zunächst benötigen wir die Definition 3.8.

Definition 3.8 (Responsibilities)**.** Wir definieren eine Funktion $R : \mathbb{R}^D \times [0,1]^L \to (0,\infty)$, die jedem Datenpunkt $\mathbf{t} \in \mathbb{R}^D$ seine Posterior-Verteilung im Latent-Space zuordnet.

$$R(\mathbf{t},\mathbf{x}) := \frac{\exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2\right)}{\int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}') - \mathbf{t}\|^2\right) d\mathbf{x}'} \tag{3.14}$$

Satz 3.9. *Das GTM-Funktional lässt sich umformen zu*

$$\begin{aligned} \mathcal{G}(\boldsymbol{y},\beta) &= \int_{\mathbb{R}^D}\int_{[0,1]^L} \frac{e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-\mathbf{t}\|^2} d\mathbf{x}'} \log \frac{e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-t\|^2} d\mathbf{x}'} d\mathbf{x} d\mu(\mathbf{t})\\ &\quad +\frac{\beta}{2}\int_{\mathbb{R}^D}\int_{[0,1]^L} \frac{e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-\mathbf{t}\|^2} d\mathbf{x}'} \|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2 d\mathbf{x} d\mu(\mathbf{t})\\ &\quad -\frac{D}{2}\log\frac{\beta}{2\pi} + \lambda \cdot S(\boldsymbol{y}). \end{aligned}$$

Beweis. Wir integrieren über die Entropien der *Responsibilities*.

$$\begin{aligned} &\int_{\mathbb{R}^D} H\left(R(\mathbf{t},\cdot)\right) d\mu(\mathbf{t})\\ =\ &\int_{\mathbb{R}^D} -\int_{[0,1]^L} \log\left(\frac{e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-\mathbf{t}\|^2} d\mathbf{x}'}\right) \frac{e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-\mathbf{t}\|^2} d\mathbf{x}'} d\mathbf{x} d\mu(\mathbf{t})\\ =\ &\int_{\mathbb{R}^D} -\int_{[0,1]^L} \frac{e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-\mathbf{t}\|^2} d\mathbf{x}'} \left(-\frac{\beta}{2}\right) \|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2 d\mathbf{x} d\mu(\mathbf{t}) \end{aligned}$$

$$
\begin{aligned}
&+ \int_{\mathbb{R}^D} - \underbrace{\int_{[0,1]^L} \frac{e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-\mathbf{t}\|^2} d\mathbf{x}'} d\mathbf{x}}_{=1} \left(- \log \int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-\mathbf{t}\|^2} d\mathbf{x}' \right) d\mu(\mathbf{t}) \\
= \; & \frac{\beta}{2} \int_{\mathbb{R}^D} \int_{[0,1]^L} \frac{e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-\mathbf{t}\|^2} d\mathbf{x}'} \|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2 d\mathbf{x} d\mu(\mathbf{t}) \\
& + \int_{\mathbb{R}^D} \log \int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2} d\mathbf{x} d\mu(\mathbf{t})
\end{aligned}
$$

Das Integral aus der letzten Zeile kennen wir aus Gleichung (3.12). Die Behauptung ergibt sich durch Einsetzen und Vereinfachung. □

Bemerkenswert ist, dass die zweite Zeile des GTM-Funktionals aus Satz 3.9 dem M-Schritt des GTM in Gleichung (2.8) entspricht. Den Entropie-Term aus der ersten Zeile kennen wir in der Form von der klassischen GTM-Formulierung nicht. Er entsteht jedoch implizit durch die Verwendung des Expectation Maximization-Algorithmus und entspricht dem Entropie-Term auf der rechten Seite von Gleichung (2.6).

3.3 Regularisierungsterm

Der Regularisierungsterm $S(f)$ vermeidet Overfitting, indem der Funktionenraum von $\boldsymbol{y}$ eingeschränkt wird. Zunächst betrachten wir den eindimensionalen Fall mit

$$S(f) = \|Gf\|_{L^2}^2,$$

wobei G ein spezieller Differentialoperator ist. Dieser setzt bei der Lösung des Minimierungsproblems

$$\arg\min_f C(f) + \lambda \cdot S(f)$$

die von G geforderte Glattheit durch. Unter bestimmten Voraussetzungen kann λ als Lagrange-Multiplikator für die Minimierung von $C(f)$ unter der Nebenbedingung $\|f\| = c$ aufgefasst werden, siehe [FG09, Aro50, Wah90].

Bei der GTM-Funktionalminimierung minimieren wir eine vektorwertige Funktion. Um dies effizient durchzuführen, muss der Regularisierungsterm einige Voraussetzungen erfüllen:

- Er muss additiv über die Datenraumdimensionen sein, also die Darstellung

  $$S(\boldsymbol{y}) = S_1(y_1) + \cdots + S_D(y_D)$$

  zulassen, wobei

  $$\boldsymbol{y}(\mathbf{x}) = \begin{pmatrix} y_1(\mathbf{x}) \\ \vdots \\ y_D(\mathbf{x}) \end{pmatrix}$$

  gilt. In Unterabschnitt 3.4.2 werden wir zeigen, dass die Minimierung der einzelnen Komponenten von $\boldsymbol{y}$ unabhängig voneinander erfolgen kann, so dass die Laufzeit linear von

der Datenraumdimension D abhängt. Dies ist jedoch nur dann möglich, wenn der Regularisierungsterm keine Abhängigkeiten zwischen den D Datenraumdimensionen erzeugt.

- Jedes einzelne $S_d(y_d)$ muss in der Form

$$S_d(y_d) = \sum_{q=1}^{Q} \|G_q y_d\|_{L^2}^2$$

mit speziellen Differenzialoperatoren G_q darstellbar sein. Weil wir zunächst davon ausgehen, dass alle Komponenten gleich regularisiert werden und damit $S_1 \equiv \cdots \equiv S_D$ gilt, ist es nicht nötig Q und G_q mit d zu parametrisieren.

Nun zeigen wir, warum diese Voraussetzungen notwendig sind. Hierfür diskretisieren wir $\boldsymbol{y}$ und wählen die bereits bekannte Darstellung

$$\boldsymbol{y}(\mathbf{x}) = \boldsymbol{W}\Phi(\mathbf{x}),$$

wobei $\boldsymbol{W} \in \mathbb{R}^{D\times M}$ gilt und $\Phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x}))^T$ ein Basisfunktionsvektor ist. Welche Basisfunktionen wir verwenden und wie M mit L skaliert, ist noch offen. Der Regularisierungsterm in Dimension $d = 1, \ldots, D$ lässt sich nun darstellen als

$$S_d(y_d) = \sum_{q=1}^{Q} \left\| G_q \sum_{j=1}^{M} w_{dj}\phi_j \right\|_{L^2}^2 = \sum_{q=1}^{Q} \sum_{j,j'=1}^{M} w_{dj} w_{dj'} \left\langle G_q\phi_j, G_q\phi_{j'} \right\rangle_{L^2},$$

womit eine Minimierung in Richtung der $\boldsymbol{W}$-Matrixeinträge durch ein lineares Gleichungssystem möglich ist.

Für $L = 1$ führt die Seminorm $\|\nabla \boldsymbol{y}\|_{L^2}^2$ dazu, dass die Länge der eindimensionalen Struktur im Datenraum bestraft wird, siehe [FG09]. Eine solche geometrische Interpretation ist wünschenswert. Entsprechende Strafterme für Flächen und Volumina sind jedoch nicht mehr additiv über die Datenraumdimensionen. In [FG09] wird als Alternative eine generalisierte Variation von Hardy und Krause zur Regularisierung von Mannigfaltigkeiten vorgeschlagen, die auch die niederdimensionalen Ränder einer Mannigfaltigkeit bestraft. Dies ist bei den Principal Manifolds sinnvoll, denn hier wirkt die „Anziehungskraft“ der Datenpunkte häufig auf die Ränder. Im Fall des GTM ist diese Regularisierung nicht zwingend geboten, da die Datenpunkte gemäß der *Responsibilities* stets auf die gesamte Mannigfaltigkeit wirken.

Nun stellen wir einige Normen vor, die in dieser Form auch in dem Sparse GTM implementiert sind.

- **L^2-Norm:**
  Die L^2-Norm setzen wir durch $Q = 1, G_1 = Id$ um. Mit dieser Norm ist das Modell nicht mehr translationsinvariant. Dies macht bei der Dimensionsreduktion erst Sinn, wenn die Datenpunkte standardmäßig um die 0 zentriert werden.

- **H^1-Norm:**
  Die Norm
  $$\|y_d\|_{H^1}^2 = \sum_{|\alpha|_1 \leq 1} \|D^\alpha y_d\|_{L^2}^2$$
  realisieren wir mit $Q = D + 1, G_1 = \frac{\partial}{\partial x_1}, \ldots, G_D = \frac{\partial}{\partial x_D}, G_{D+1} = Id$. Die Einzelableitungen sorgen für Glattheit von y_d, die Norm enthält jedoch noch den nicht-translationsinvarianten L^2-Anteil.

- **H^1-Seminorm:**
  Die Seminorm
  $$|y_d|_{H^1}^2 = \sum_{|\alpha|_1 = 1} \|D^\alpha y_d\|_{L^2}^2$$
  setzen wir mit $Q = D, G_1 = \frac{\partial}{\partial x_1}, \ldots, G_D = \frac{\partial}{\partial x_D}$ um. Diese translationsinvariante Norm ist sehr gut geeignet, die Regularität von y_d sicherzustellen. Wie in [Gar04] jedoch angemerkt wird, ist die Existenz von stetigen Repräsentanten für Funktionen aus H^1 für $D > 1$ nicht mehr garantiert.

- **$H^{1,\mathbf{mix}}$-Norm:**
  Die Norm
  $$|y_d|_{H^1}^2 = \sum_{|\alpha|_\infty \leq 1} \|D^\alpha y_d\|_{L^2}^2$$
  setzen wir mit $Q = 2^D$ und allen gemischten Ableitungen um. Diese Norm hat noch einen translationsinvarianten L^2-Anteil, sorgt jedoch für stetige Einbettbarkeit unabhängig von der Dimension D, siehe ebenfalls [Gar04].

- **$H^{1,\mathbf{mix}}$-Seminorm:**
  Die Norm
  $$|y_d|_{H^{1,\mathrm{mix}}}^2 = \sum_{|\alpha|_\infty = 1} \|D^\alpha y_d\|_{L^2}^2$$
  entspricht der $H^{1,\mathrm{mix}}$-Norm nur ohne L^2-Anteil.

- **Höchster $H^{1,\mathbf{mix}}$-Term:**
  Mit $Q = 1, G_1 = \frac{\partial^D}{\partial x_1 \cdots \partial x_D}$ realisieren wir den höchsten Term der $H^{1,\mathrm{mix}}$-Norm. Dieser garantiert jedoch noch keine stetige Einbettbarkeit.

3.4 Funktionalminimierung

Satz 3.10. *Das in Abschnitt 3.2 bestimmte GTM-Funktional $\mathcal{G}$ ist nicht konvex und hat mehrere lokale Minima.*

Beweis. Wir nehmen an, $\mathcal{G}$ sei konvex, und $(\boldsymbol{y}, \beta)$ ein globales Optimum. Sei $\varphi : \mathbb{R}^L \to \mathbb{R}^L$ eine einfache Variablenpermutation. Da $\boldsymbol{y} \circ \varphi$ und β die gleiche Dichte im Datenraum erzeugen, gilt

$$\mathcal{G}(\boldsymbol{y}, \beta) = \mathcal{G}(\varphi \circ \boldsymbol{y}, \beta).$$

Wegen der Konvexität von $\mathcal{G}$ gilt für alle $\lambda \in (0,1)$

$$\mathcal{G}(\lambda \boldsymbol{y} + (1-\lambda)\boldsymbol{y} \circ \varphi, \beta) \leq \lambda \mathcal{G}(\boldsymbol{y}, \beta) + (1-\lambda)\mathcal{G}(\boldsymbol{y} \circ \varphi, \beta) = \mathcal{G}(\boldsymbol{y}, \beta).$$

Aufgrund der Optimalität von $(\boldsymbol{y}, \beta)$ gilt ausserdem

$$\mathcal{G}(\lambda \boldsymbol{y} + (1-\lambda)\varphi \circ \boldsymbol{y}, \beta) \geq \mathcal{G}(\boldsymbol{y}, \beta).$$

Also ist $\mathcal{G}$ für alle $\lambda \in (0,1)$ konstant. Da dies in aller Regel nicht der Fall ist, was bereits an einfachen Beispielen nachvollziehbar ist, kann $\mathcal{G}$ nicht konvex sein. □

Wir konstruieren nun ein Minimierungsverfahren, das an den Alternating-Least-Squares-Algorithmus angelehnt ist, siehe [KB89].

Definition 3.11 ($\mathcal{K}$-Funktional)**.** Wir definieren das $\mathcal{K}$-Funktional

$$\begin{aligned} \mathcal{K}(\psi, \boldsymbol{y}, \beta) &:= \int_{\mathbb{R}^D} \int_{[0,1]^L} \psi(\mathbf{t}, \mathbf{x}) \log \psi(\mathbf{t}, \mathbf{x}) d\mathbf{x} d\mu(\mathbf{t}) \\ &+ \frac{\beta}{2} \int_{\mathbb{R}^D} \int_{[0,1]^L} \psi(\mathbf{t}, \mathbf{x}) \|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2 d\mathbf{x} d\mu(\mathbf{t}) \\ &- \frac{D}{2} \log \frac{\beta}{2\pi} + \lambda \cdot S(\boldsymbol{y}), \end{aligned}$$

wobei $\psi : \mathbb{R}^D \times [0,1]^L \to (0, \infty)$ eine Funktion mit

$$\int_{[0,1]^L} \psi(\mathbf{t}, \mathbf{x}) d\mathbf{x} = 1$$

für alle $\mathbf{t} \in \mathbb{R}^D$ ist.

Lemma 3.12 beschreibt eine wesentliche Eigentschaft des $\mathcal{K}$-Funktionals.

Lemma 3.12. *Für die zu $\boldsymbol{y}$ und β passenden Responsibilities R, siehe Definition 3.8, erhalten wir das ursprüngliche $\mathcal{G}$-Funktional*

$$\mathcal{K}(R, \boldsymbol{y}, \beta) = \mathcal{G}(\boldsymbol{y}, \beta).$$

Beweis. Gleichheit ergibt sich durch einfaches Einsetzen der Definition von R. □

Nun minimieren wir $\mathcal{K}$ durch ein Iterationsverfahren. Zunächst wählen wir ein initiales $\boldsymbol{y}^{(0)}$ und $\beta^{(0)}$. Dann setzen wir fort

$$\begin{aligned} \psi^{(0)} &:= \arg\min_{\psi} \mathcal{K}(\psi, \boldsymbol{y}^{(0)}, \beta^{(0)}) \\ \boldsymbol{y}^{(1)} &:= \arg\min_{\boldsymbol{y}} \mathcal{K}(\psi^{(0)}, \boldsymbol{y}, \beta^{(0)}) \\ \beta^{(1)} &:= \arg\min_{\beta} \mathcal{K}(\psi^{(0)}, \boldsymbol{y}^{(1)}, \beta) \end{aligned}$$

$$\begin{aligned}
\psi^{(1)} &:= \arg\min_{\psi} \mathcal{K}(\psi, \boldsymbol{y}^{(1)}, \beta^{(1)}) \\
\boldsymbol{y}^{(2)} &:= \arg\min_{\boldsymbol{y}} \mathcal{K}(\psi^{(1)}, \boldsymbol{y}, \beta^{(1)}) \\
\beta^{(2)} &:= \arg\min_{\beta} \mathcal{K}(\psi^{(1)}, \boldsymbol{y}^{(2)}, \beta) \\
&\vdots
\end{aligned}$$

Bevor wir zeigen, warum dieses Verfahren ein lokales Minimum von $\mathcal{G}$ findet, betrachten wir die einzelnen Minimierungsschritte.

3.4.1 Minimierung im ersten Parameter

Der folgende Satz ist die Entsprechung von Lemma 2.2 und sagt aus, dass die zu $\boldsymbol{y}$ und β gehörenden *Responsibilities* das $\mathcal{K}$-Funktional im ersten Parameter minimieren.

Satz 3.13. *Sei $\mathcal{K}$ wie oben definiert. Dann gilt*

$$R = \arg\min_{\psi} \mathcal{K}(\psi, \boldsymbol{y}, \beta).$$

Beweis. Sei ψ^* eine Funktion, für die $\mathcal{K}(\cdot, \boldsymbol{y}, \beta)$ minimal ist. Da wir nicht gefordert haben, dass $\psi^*(\mathbf{t}, \mathbf{x})$ stetig von $\mathbf{t}$ abhängt, können wir die $\psi^*(\mathbf{t}, \cdot)$ unabhängig voneinander betrachten. Sie müssen für alle $\mathbf{t} \in \mathbb{R}^D$ mit positiver Dichte $d\mu(\mathbf{t})$

$$\int_{[0,1]^L} \psi^*(\mathbf{t}, \mathbf{x}) \log \psi^*(\mathbf{t}, \mathbf{x}) d\mathbf{x} + \frac{\beta}{2} \int_{[0,1]^L} \psi^*(\mathbf{t}, \mathbf{x}) \|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2 d\mathbf{x}$$

unter der Nebenbedingung

$$\int_{[0,1]^L} \psi^*(\mathbf{t}, \mathbf{x}) d\mathbf{x} = 1$$

minimieren. Wir wählen ein beliebiges $z \in L^1([0,1])$ mit

$$\int_{[0,1]^L} z(\mathbf{x}) d\mathbf{x} = 0.$$

Aufgrund der Optimalität von $\psi^*(\mathbf{t}, \cdot)$ muss für $s \in \mathbb{R}$

$$\begin{aligned}
\frac{\partial}{\partial s} \Bigg[& \int_{[0,1]^L} (\psi^*(\mathbf{t}, \mathbf{x}) + s z(\mathbf{x})) \log (\psi^*(\mathbf{t}, \mathbf{x}) + s z(\mathbf{x})) \, d\mathbf{x} \\
& + \frac{\beta}{2} \int_{[0,1]^L} (\psi^*(\mathbf{t}, \mathbf{x}) + s z(\mathbf{x})) \, \|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2 d\mathbf{x} \Bigg]_{s=0} \quad = \quad 0
\end{aligned}$$

gelten. Wir folgern

$$\begin{aligned}
& \int_{[0,1]^L} z(\mathbf{x}) \log \psi^*(\mathbf{t},\mathbf{x}) d\mathbf{x} + \underbrace{\int_{[0,1]^L} z(\mathbf{x}) d\mathbf{x}}_{=0} + \frac{\beta}{2} \int_{[0,1]^L} z(\mathbf{x}) \|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2 d\mathbf{x} = 0 \\
\Leftrightarrow \quad & \int_{[0,1]^L} z(\mathbf{x}) \log \psi^*(\mathbf{t},\mathbf{x}) d\mathbf{x} - \int_{[0,1]^L} z(\mathbf{x}) \log \exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2\right) d\mathbf{x} = 0 \\
\Leftrightarrow \quad & \int_{[0,1]^L} z(\mathbf{x}) \log \frac{\psi^*(\mathbf{t},\mathbf{x})}{\exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2\right)} d\mathbf{x} = 0.
\end{aligned}$$

Da die letzte Zeile für alle Funktionen z mit $\int z(\mathbf{x}) d\mathbf{x} = 0$ gilt, können wir folgern, dass

$$\log \frac{\psi^*(t,x)}{\exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2\right)}$$

bis auf eine Nullmenge konstant ist. Damit ist $\psi^*(\mathbf{t},\mathbf{x})$ proportional zu $\exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2\right)$. Aus

$$\int_{[0,1]^L} \psi^*(\mathbf{t},\mathbf{x}) d\mathbf{x} = 1$$

folgt

$$\psi^*(\mathbf{t},\mathbf{x}) = \frac{\exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2\right)}{\int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2\right) d\mathbf{x}} = R.$$

□

Der Satz 3.13 zeigt, dass die *Responsibilities* R das Funktional $\mathcal{K}$ im ψ-Parameter minimieren. Dies ist entscheidend, um den Bezug zu $\mathcal{G}$ herzustellen, wie in Unterabschnitt 3.4.4 gezeigt wird.

3.4.2 Minimierung im zweiten Parameter

In Abschnitt 3.3 wurden bereits die Voraussetzungen für die Optimierung in $\boldsymbol{y}$ erläutert.

Definition 3.14 (M-Schritt-Minimierung)**.** Wir bezeichnen die Minimierung in $\boldsymbol{y}$-Richtung in Anlehnung an den EM-Algorithmus „M-Schritt-Minimierung“

$$\begin{aligned}
& \arg\min_{\boldsymbol{y}} \mathcal{K}(\psi^{(s)}, \boldsymbol{y}, \beta^{(s)}) \\
= \quad & \arg\min_{\boldsymbol{y}} \frac{\beta}{2} \int_{\mathbb{R}^D} \int_{[0,1]^L} \psi^{(s)}(\mathbf{t},\mathbf{x}) \|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2 d\mathbf{x} d\mu(\mathbf{t}) + \lambda \cdot S(\boldsymbol{y}).
\end{aligned}$$

Wesentlich an dieser Stelle ist, dass sich die rechte Seite der M-Schritt-Minimierung als eine Summe über die D Dimensionen schreiben lässt, so dass die einzelnen y_d-Komponenten unabhängig voneinander minimiert werden können. $\|\cdot\|^2$ erfüllt diese Voraussetzung, und der Regularisierungsterm $S(\boldsymbol{y})$ muss entsprechend gewählt werden.

Für die d-te Dimension gilt mit dem Einsetzen der diskretisierten Darstellung von y_d

$$\begin{aligned}
&\arg\min_{y_d} \quad \frac{\beta}{2}\int_{\mathbb{R}^D}\int_{[0,1]^L}\psi^{(s)}(\mathbf{t},\mathbf{x})\left(y_d(\mathbf{x})-\mathbf{t}_d\right)^2 d\mathbf{x}d\mu(\mathbf{t}) + \lambda\cdot S_d(y_d)\\
=&\arg\min_{(w_{d1},\ldots,w_{dM})} \quad \frac{\beta}{2}\int_{\mathbb{R}^D}\int_{[0,1]^L}\psi^{(s)}(\mathbf{t},\mathbf{x})\left(\sum_{j=1}^{M} w_{dj}\phi_j(\mathbf{x})-\mathbf{t}_d\right)^2 d\mathbf{x}d\mu(\mathbf{t})\\
&\qquad +\lambda\sum_{q=1}^{Q}\sum_{j,j'=1}^{M} w_{dj}w_{dj'}\left\langle G_q\phi_j, G_q\phi_{j'}\right\rangle_{L^2}.
\end{aligned}$$

Ableiten nach w_{dc} ergibt

$$\begin{aligned}
&\frac{\partial}{\partial w_{dc}}\mathcal{K}(\psi^{(s)},\boldsymbol{y},\beta^{(s)}) = 0\\
\Leftrightarrow\quad &\sum_{j=1}^{M}\left(\int_{\mathbb{R}^D}\int_{[0,1]^L}\psi^{(s)}(\mathbf{t},\mathbf{x})\phi_j(\mathbf{x})\phi_c(\mathbf{x})d\mathbf{x}d\mu(\mathbf{t}) + \frac{2\lambda}{\beta}\sum_{q=1}^{Q}\left\langle G_q\phi_j, G_q\phi_c\right\rangle_{L^2}\right) w_{dj}\\
=\quad &\int_{\mathbb{R}^D}\int_{[0,1]^L}\psi^{(s)}(\mathbf{t},\mathbf{x})\mathbf{t}_d\phi_c(\mathbf{x})d\mathbf{x}d\mu(\mathbf{t}).
\end{aligned}$$

Dies führt mit $d=1,\ldots,D$ zu D linearen Gleichungssystemen über alle w_{d*}-Einträge

$$\left(\boldsymbol{A}+\frac{2\lambda}{\beta}\boldsymbol{C}_d\right)\mathbf{w}_d = \mathbf{b}_d, \tag{3.15}$$

wobei $\mathbf{w}_d = (w_{d1},\ldots,w_{dM})^T$, $\boldsymbol{A}\in\mathbb{R}^{M\times M}$ mit

$$(\boldsymbol{A})_{cj} = \int_{\mathbb{R}^D}\int_{[0,1]^L}\psi^{(s)}(\mathbf{t},\mathbf{x})\phi_c(\mathbf{x})\phi_j(\mathbf{x})d\mathbf{x}d\mu(\mathbf{t}), \tag{3.16}$$

$\boldsymbol{C}_d\in\mathbb{R}^{M\times M}$ mit

$$(\boldsymbol{C}_d)_{cj} = \sum_{q=1}^{Q}\left\langle G_q\phi_c, G_q\phi_j\right\rangle_{L^2} \tag{3.17}$$

und $\mathbf{b}_d\in\mathbb{R}^M$ mit

$$(\mathbf{b}_d)_c = \int_{\mathbb{R}^D}\int_{[0,1]^L}\psi^{(s)}(\mathbf{t},\mathbf{x})\mathbf{t}_d\phi_c(\mathbf{x})d\mathbf{x}d\mu(\mathbf{t})$$

gilt. Bemerkenswert ist, dass die $\boldsymbol{A}$-Matrix weder von d noch den Daten abhängt. Wenn über alle Dimensionen gleich regularisiert wird, gilt dies auch für die Regularisierungsmatrix $\boldsymbol{C}_d$. Eine Ersetzung von $d\mu(\mathbf{t})$ durch diskrete Samples und die Verwendung einer Trapezregel-Quadratur für das $d\mathbf{x}$-Integral beschreibt den M-Schritt des klassischen GTM. Dieser Zusammenhang wird in Abschnitt 3.5 dargestellt.

Wir können das lineare Gleichungssystem auch als ein Galerkin-gewichtetes Residuum auf-

fassen. Die Rücksubstitution von $\sum_{j=1}^{M} w_{dj}\phi_j(\mathbf{x}) = y_d(\mathbf{x})$ ergibt

$$\begin{aligned}
&\sum_{j=1}^{M}\left(\int_{\mathbb{R}^D}\int_{[0,1]^L}\psi^{(s)}(\mathbf{t},\mathbf{x})\phi_j(\mathbf{x})\phi_c(\mathbf{x})d\mathbf{x}d\mu(\mathbf{t}) + \frac{2\lambda}{\beta}\sum_{q=1}^{Q}\langle G_q\phi_j, G_q\phi_c\rangle_{L^2}\right) w_{dj} \\
= &\int_{\mathbb{R}^D}\int_{[0,1]^L}\psi^{(s)}(\mathbf{t},\mathbf{x})t_d\phi_c(\mathbf{x})d\mathbf{x}d\mu(\mathbf{t}) \\
\Leftrightarrow \quad &\int_{\mathbb{R}^D}\int_{[0,1]^L}\psi^{(s)}(\mathbf{t},\mathbf{x})y_d(\mathbf{x})\phi_c(\mathbf{x})d\mathbf{x}d\mu(\mathbf{t}) + \frac{2\lambda}{\beta}\sum_{q=1}^{Q}\langle G_q y_d, G_q\phi_c\rangle_{L^2} \\
= &\int_{\mathbb{R}^D}\int_{[0,1]^L}\psi^{(s)}(\mathbf{t},\mathbf{x})t_d\phi_c(\mathbf{x})d\mathbf{x}d\mu(\mathbf{t}).
\end{aligned}$$

Diese Gleichung muss für alle $c = 1, \ldots, M$ erfüllt sein, womit $\phi_c(\mathbf{x})$ zu einer Testfunktion wird.

3.4.3 Minimierung im dritten Parameter

Die Optimierung in β ist unproblematisch. Einfaches Differenzieren ergibt

$$\begin{aligned}
&\frac{\partial}{\partial\beta}\mathcal{K}(\psi^{(s)}, \boldsymbol{y}^{(s+1)}, \beta) = 0 \\
\Leftrightarrow \quad &\frac{1}{\beta} = \frac{1}{D}\int_{\mathbb{R}^D}\int_{[0,1]^L}\psi^{(s)}(\mathbf{t},\mathbf{x})\|\boldsymbol{y}^{(s+1)}(\mathbf{x}) - \mathbf{t}\|^2 d\mathbf{x}d\mu(\mathbf{t}).
\end{aligned}$$

3.4.4 Konvergenz

In jedem der beschriebenen Schritte wird der Funktionalwert $\mathcal{K}$ weiter minimiert. Da die Kullback-Leibler-Divergenz von unten durch die 0 beschränkt ist, ist der Funktionalwert ebenfalls von unten beschränkt, woraus die Konvergenz des Verfahrens folgt. Während $\mathcal{G}$ nicht konvex in $\boldsymbol{y}$ ist, lässt sich $\mathcal{K}(\psi, \boldsymbol{y}, \beta)$ in jedem Parameter exakt minimieren. Allerdings ist der zulässige Suchraum $(\psi, \boldsymbol{y}, \beta)$ von $\mathcal{K}$ größer als der von $\mathcal{G}$. Der Satz 3.13 zeigt, dass eine Minimierung der ersten Komponente von $\mathcal{K}$ die zu $\boldsymbol{y}$ und β gehörigen *Responsibilities* ergibt, womit wir nach Lemma 3.12 zurück in den Suchraum von $\mathcal{G}$ gelangen. Mit

$$\mathcal{K}(\psi^{(s)}, \boldsymbol{y}^{(s)}, \beta^{(s)}) = \mathcal{K}(R^{(s)}, \boldsymbol{y}^{(s)}, \beta^{(s)}) = \mathcal{G}(\boldsymbol{y}^{(s)}, \beta^{(s)})$$

ist der Bezug zwischen $\mathcal{K}$ und $\mathcal{G}$ hergestellt.

Einen Sonderfall stellen Sattelpunkte dar, an denen eine Funktion in jeder Einzelrichtung optimal ist, obwohl es sich nicht um ein Extremum handelt. Als Beispiel für einen Sattelpunkt betrachten wir die Stelle $(0, 0)$ der Funktion

$$f(x, y) = x^2 + y^2 - 4xy.$$

f hat einen kritischen Punkt in $(0, 0)$, aber die Hesse-Matrix ist indefinit, das heisst es handelt sich weder um ein Maximum noch um ein Minimum, siehe Abbildung 3.1. In Unterabschnitt 6.6.2 beschreiben wir einen Fall, bei dem die Sattelpunktproblematik zu beobachten ist.

Hiervon abgesehen hat sie keine besondere Bedeutung im GTM-Kontext.

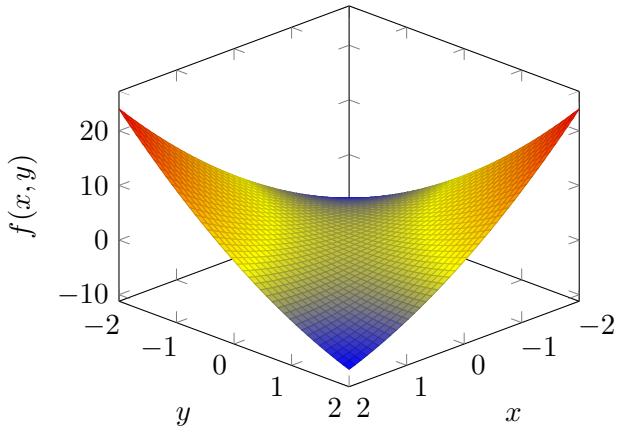

Abb. 3.1: Funktion f mit Sattelpunktpunkt an der Stelle $(0,0)$.

3.5 Bezug zum klassischen GTM

In diesem Abschnitt werden wir nachrechnen, dass die Matrixgleichung (2.9), wie sie beim klassischen GTM entsteht, ein Spezialfall unserer Formulierung ist. Hierzu müssen wir Integrale berechnen, wie sie in Gleichung (3.16) vorkommen. In diesem Abschnitt diskretisieren wir $d\mathbf{x}$-Integrale über $[0,1]^L$ mit Quadraturpunkten einer L-dimensionalen tensorierten Mittelpunktsregel. Sie liegen gitterartig auf $[0,1]^L$ und entsprechen den $\{\mathbf{x}_i\}_{i=1}^K$ des klassischen GTM, also

$$Q_K f := \frac{1}{K} \sum_{i=1}^{K} f(\mathbf{x}_i).$$

Integrale über $\mathbb{R}^D$ mit dem Maß $d\mu(\mathbf{t})$ zerfallen durch das Einsetzen der empirischen Verteilung

$$\mu(\mathbf{t}) = \frac{1}{N} \sum_{n=1}^{N} \delta_{t_n}(\mathbf{t})$$

zu einer Summe über die N Datenpunkte.

3.5.1 E-Schritt

Es wurde bereits gezeigt, dass die *Responsibilities* aus Definition 3.8 das $\mathcal{K}$-Fuktional im ersten Parameter minimieren. Dies bedeutet, dass zu gegebenen $\boldsymbol{y}^{(s)}$ und $\beta^{(s)}$

$$\psi^{(s)}(\mathbf{t},\mathbf{x}) = R^{(s)}(\mathbf{t},\mathbf{x}) = \frac{\exp\left(-\frac{\beta^{(s)}}{2}\|\boldsymbol{y}^{(s)}(\mathbf{x}) - \mathbf{t}\|^2\right)}{\int_{[0,1]^L} \exp\left(-\frac{\beta^{(s)}}{2}\|\boldsymbol{y}^{(s)}(\mathbf{x}') - \mathbf{t}\|^2\right) d\mathbf{x}'}$$

gilt. Wir setzen die Quadraturregel ein und erhalten für $\mathbf{t}_n$ und $\mathbf{x}_i$

$$\psi^{(s)}(\mathbf{t}_n, \mathbf{x}_i) \approx \frac{K \exp\left(-\frac{\beta^{(s)}}{2}\|\boldsymbol{y}^{(s)}(\mathbf{x}_i) - \mathbf{t}_n\|^2\right)}{\sum_{i'=1}^{K} \exp\left(-\frac{\beta^{(s)}}{2}\|\boldsymbol{y}^{(s)}(\mathbf{x}_{i'}) - \mathbf{t}\|^2\right)} = K\tilde{R}^{(s)}(\mathbf{t}_n, \mathbf{x}_i).$$

Hierbei bezeichnen wir mit $\tilde{R}^{(s)}$ die diskreten *Responsibilities* des klassischen GTM aus Formel (2.7). Offenbar entsprechen sie bis auf den Faktor K den $\psi^{(s)}$ mit eingesetzter Quadraturregel ausgewertet in $(\mathbf{t}_n, \mathbf{x}_i)$.

3.5.2 M-Schritt

Wir betrachten den Eintrag cj der $\boldsymbol{A}$-Matrix nach Einsetzen der Quadraturregeln und formen um

$$\begin{aligned}
(\boldsymbol{A})_{cj} &= \int_{\mathbb{R}^D}\int_{[0,1]^L} \psi^{(s)}(\mathbf{t}, \mathbf{x})\phi_c(\mathbf{x})\phi_j(\mathbf{x}) d\mathbf{x} d\mu(\mathbf{t}) \\
&\approx \frac{1}{NK}\sum_{n=1}^{N}\sum_{i=1}^{K} \psi^{(s)}(\mathbf{t}_n, \mathbf{x}_i)\phi_c(\mathbf{x}_i)\phi_j(\mathbf{x}_i) \\
&= \frac{1}{NK}\sum_{i=1}^{K} \phi_c(\mathbf{x}_i)\left(\sum_{n=1}^{N} K\tilde{R}^{(s)}(\mathbf{t}_n, \mathbf{x}_i)\right)\phi_j(\mathbf{x}_i) \\
&= \frac{1}{N}\left(\boldsymbol{\Phi}^T\boldsymbol{G}\boldsymbol{\Phi}\right)_{cj}.
\end{aligned}$$

Wir erkennen, dass es sich bis auf den Faktor N um das Matrixprodukt auf der linken Seite der Formel (2.9) handelt. Für den c-ten Eintrag des b_d-Vektors gilt analog

$$\begin{aligned}
(b_d)_c &= \int_{\mathbb{R}^D}\int_{[0,1]^L} \psi^{(s)}(\mathbf{t}, \mathbf{x})\,(\mathbf{t})_d\,\phi_c(\mathbf{x}) d\mathbf{x} d\mu(\mathbf{t}) \\
&\approx \frac{1}{NK}\sum_{n=1}^{N}\sum_{i=1}^{K} \psi^{(s)}(\mathbf{t}_n, \mathbf{x}_i)\,(\mathbf{t}_n)_d\,\phi_c(\mathbf{x}_i) \\
&= \frac{1}{NK}\sum_{i=1}^{K} \phi_c(\mathbf{x}_i)\left(\sum_{n=1}^{N} K\tilde{R}^{(s)}(\mathbf{t}_n, \mathbf{x}_i)\,(\mathbf{t}_n)_d\right) \\
&= \frac{1}{N}\sum_{i=1}^{K}\left(\boldsymbol{\Phi}^T\right)_{ci}(\boldsymbol{R}\boldsymbol{T})_{id} = \frac{1}{N}\left(\boldsymbol{\Phi}^T\boldsymbol{R}\boldsymbol{T}\right)_{cd}.
\end{aligned}$$

Nun ist erkennbar, dass die Matrix-Gleichung aus Formel (2.9) entsteht, wenn man die linearen Gleichungssysteme für $d = 1, \ldots, D$ spaltenweise hintereinander reiht. Der Regularisierungsterm ist hiervon komplett unabhängig und wurde aus Gründen der Übersichtlichkeit weggelassen.

Eine Interpretation ist folgende: Die Latent-Space-Verteilung $p(\mathbf{x}) = \frac{1}{K}\sum_{i=1}^{K}\delta_{x_i}(\mathbf{x})$ des klassischen GTM realisiert effektiv die Quadraturregel eines GTM mit kontinuierlicher Latent-

Space-Verteilung $p(\mathbf{x}) = 1$. Da das regelmäßige Gitter die Glattheit der Basisfunktionen und der exp-Funktion nicht ausnutzt, sind hier Verbesserungen möglich. Diese Betrachtungsweise bildet die Grundlage für die Verwendung von dünnen Gittern sowohl für die Quadratur als auch für die Basisfunktionen.

4 Sparse GTM

Die auf einem Gitter angeordneten Latent-Space-Samples $\{\mathbf{x}_i\}_{i=1}^K$ unterliegen dem Fluch der Dimension (*curse of dimensionality*). Dieser Begriff wurde 1961 von Bellmann geprägt, siehe [Bel61], und beschreibt die bei vielen Problemen auftretende exponentielle Abhängigkeit des Rechenaufwands von der Dimension. Wenn h die Maschenweite bezeichnet, so gilt $K = \mathcal{O}(h^{-L})$, was zur Folge hat, dass eine Halbierung der Maschenweite den Rechenaufwand um den Faktor 2^L steigert. Die Grenzen heutiger Computer werden so bereits bei einer mittleren Dimensionsanzahl schnell erreicht.

Der Fluch der Dimension lässt sich nicht allgemein brechen, da er häufig eine theoretische untere Schranke markiert. Erst durch das Einbeziehen von Vorwissen, das bedeutet das Einschränken der Problemklasse, sind Verfahren mit einer günstigeren Laufzeitkomplexität möglich. Um dies beim GTM zu erreichen, verwenden wir eine Dünngitterdiskretisierung für das Mapping $\boldsymbol{y}$ und eine Dünngitterquadratur anstelle der Latent-Space-Samples.

In den Abschnitten 4.1 und 4.2 werden wir die Interpolation und Quadratur mit dünnen Gittern in allgemeiner Form vorstellen. In den Unterabschnitten 4.1.1 und 4.2.3 gehen wir darauf ein, inwieweit die Voraussetzungen für die Anwendung auf das GTM erfüllt sind.

In Abschnitt 4.3 beschreiben wir die Umsetzung des Sparse GTM und analysieren die Laufzeit. Abschnitt 4.4 behandelt numerische Aspekte des M-Schritt-Gleichungssystems und der *Responsibilities*-Berechnung. In Abschnitt 4.5 stellen wir eine Formulierung des Sparse GTM mit Hilberträumen mit reproduzierendem Kern vor. Schließlich werden in Abschnitt 4.6 Laufzeitaspekte der verschiedenen GTM-Varianten behandelt.

4.1 Dünngitter-Diskretisierung

Die folgende Darstellung der Konstruktion einer Dünngitterbasis orientiert sich an [FG09]. Für eine ausführlichere Beschreibung mit Fokus auf partielle Differentialgleichungen siehe [BG04].

Dünne Gitter wurden ursprünglich für die effiziente Diskretisierung d-dimensionaler elliptischer Probleme zweiter Ordnung entwickelt, und basieren auf Tensorprodukten von eindimensionalen Multiskalenfunktionen. Die Koeffizienten einer hinreichend glatten Lösung fallen mit einer bestimmten Rate ab. Bei Funktionen mit beschränkten r-ten Ableitungen führt das Abschneiden der Reihe zu einer Dünngitter-Basis, die nur $\mathcal{O}(h^{-1}\log(h^{-1})^{d-1})$ Freiheitsgrade benötigt, im Gegensatz zu $\mathcal{O}(h^{-d})$ bei einer vollen Basis. Bei der Verwendung von stückweisen Polynomen vom Grad $r-1$ als Basisfunktionen ist der Dünngitter-Diskretisierungsfehler mit $\mathcal{O}(h^r\left(\log h^{-1}\right)^{d-1})$ in der L^2-Norm nur geringfügig schlechter als der Fehler $\mathcal{O}(h^r)$ der Vollgitter-Basis.

Diese Eigenschaften machen die dünnen Gitter zu einer guten Wahl für das GTM. Wir verwenden für die Komponentenfunktionen von $\boldsymbol{y} : [0,1]^L \to \mathbb{R}^D$ D voneinander unabhängige Diskretisierungen. Im Folgenden werden wir den skalaren Fall einer Funktion f behandeln, der

sich einfach auf $\boldsymbol{y}$ verallgemeinern lässt. Hierbei entspricht die Dimension d der Latent-Space-Dimension L und nicht der Datenraumdimension D.

Wir beschränken uns auf d-lineare Hütchenfunktionen als Dünngitterfunktionen. Im eindimensionalen Fall entspricht dies der Funktion $\phi(x)$,

$$\phi(x) := \begin{cases} 1 - |x|\,, & \text{wenn } x \in [-1, 1]\,, \\ 0 & \text{sonst.} \end{cases}$$

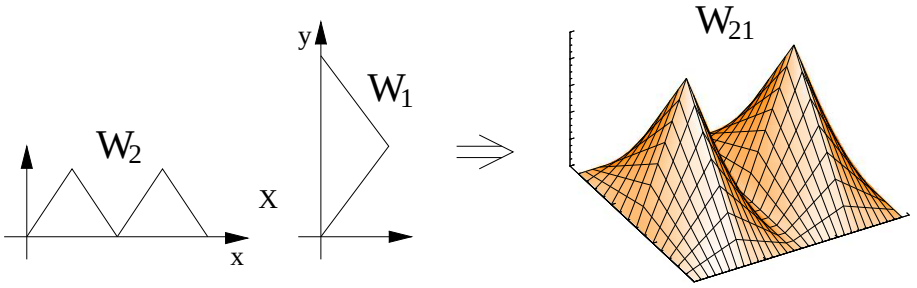

Abb. 4.1: Tensorproduktansatz für stückweise bilineare Basisfunktionen. Abbildung entnommen aus [FG09].

Durch Translation und Dilatation können aus dieser Funktion beliebige $\phi_{l_j,i_j}(x_j)$ mit zugehörigem Träger $\left[x_{l_j,i_j} - h_{l_j}, x_{l_j,i_j} + h_{l_j}\right] = \left[(i_j - 1)h_{l_j}, (i_j + 1)h_{l_j}\right]$ erzeugt werden, also

$$\phi_{l_j,i_j}(x_j) := \phi\left(\frac{x_j - i_j \cdot h_{l_j}}{h_{l_j}}\right).$$

Die resultierende eindimensionale Funktion wird in der Tensorproduktkonstruktion der d-linearen Basisfunktionen verwendet. Hierzu wird für jeden Punkt $\mathbf{x}_{\mathbf{l},\mathbf{i}} := \mathbf{i} \cdot \mathbf{h}_\mathbf{l}, \mathbf{0} \leq \mathbf{i} \leq 2^\mathbf{l}$ entsprechend der Abbildung 4.1 das Produkt

$$\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) := \prod_{j=1}^{d} \phi_{l_j,i_j}(x_j)$$

gebildet. Hierbei bezeichnet $\mathbf{l} = (l_1, \ldots, l_d) \in \mathbb{N}^d$ das multivariate Verfeinerungslevel, $\mathbf{i} = (i_1, \ldots, i_d) \in \mathbb{N}^d$ eine multivariate Position und $\mathbf{0} = (0, \ldots, 0)$ den Nullpunkt. Die Ungleichungen $\mathbf{0} \leq \mathbf{i} \leq 2^\mathbf{l}$ sind komponentenweise zu verstehen. Wir betrachten nun die Familie d-dimensionaler Gitter

$$\left\{T_\mathbf{l}, \mathbf{l} \in \mathbb{N}^d\right\}$$

auf $[0, 1]^d$ mit multivariaten Maschenweiten $\mathbf{h}_\mathbf{l} := (h_{l_1}, \ldots, h_{l_d}) := 2^{-\mathbf{l}}$. Das Gitter $T_\mathbf{l}$ ist äquidistant in jeder Koordinatenrichtung, in unterschiedlichen Richtungen sind jedoch verschiedene Maschenweiten zulässig. Die Gitterpunkte $\mathbf{x}_{\mathbf{l},\mathbf{i}}$ des Gitters $T_\mathbf{l}$ sind

$$\mathbf{x}_{\mathbf{l},\mathbf{i}} := (x_{l_1,i_1}, \ldots, x_{l_d,i_d}) := \mathbf{i} \cdot \mathbf{h}_\mathbf{l}, \quad \mathbf{0} \leq \mathbf{i} \leq 2^\mathbf{l}.$$

Die Funktionen $\phi_{\mathbf{l},\mathbf{i}}$ bilden eine Basis des Funktionenraums $V_\mathbf{l}$ von stückweise d-linearen Funktionen auf dem Gitter $T_\mathbf{l}$. Wir definieren

$$V_\mathbf{l} := \operatorname{span}\left\{\phi_{\mathbf{l},\mathbf{i}} \mid \mathbf{0} \leq \mathbf{i} \leq 2^\mathbf{l}\right\},$$

wobei Randfunktionen entsprechend abgeschnitten werden. Wir führen die hierarchischen Inkremente

$$W_\mathbf{l} := \operatorname{span}\left\{\phi_{\mathbf{l},\mathbf{i}} \;\middle|\; \begin{array}{ll} 1 \leq i_j \leq 2^{l_j} - 1, i_j \text{ ungerade}, & \text{falls } l_j > 0, \\ 0 \leq i_j \leq 1, & \text{falls } l_j = 0, \end{array} 1 \leq j \leq d\right\}$$

ein, für die die Gleichung

$$V_\mathbf{l} = \bigotimes_{\mathbf{t} \leq \mathbf{l}} W_\mathbf{t}$$

gilt. Die Träger der Basisfunktionen $\phi_{\mathbf{l},\mathbf{i}}$, die $W_\mathbf{l}$ aufspannen, sind für jedes $\mathbf{l} > 0$ paarweise disjunkt. Somit erhalten wir mit der Indexmenge

$$\mathbf{I}_\mathbf{l} := \operatorname{span}\left\{\mathbf{i} \in \mathbb{N}^d \;\middle|\; \begin{array}{ll} 1 \leq i_j \leq 2^{l_j} - 1, i_j \text{ ungerade}, & \text{falls } l_j > 0, \\ 0 \leq i_j \leq 1, & \text{falls } l_j = 0, \end{array} 1 \leq j \leq d\right\}$$

eine alternative Basis für $V_\mathbf{l}$, nämlich die hierarchische Basis

$$\{\phi_{\mathbf{k},\mathbf{i}} \mid \mathbf{i} \in \mathbf{I}_k, \mathbf{k} \leq \mathbf{l}\}.$$

Diese verallgemeinert durch Tensorproduktbildung die eindimensionale Basis aus Abbildung 4.2 auf den d-dimensionalen Fall. Mit den hierarchischen Inkrementen $W_\mathbf{l}$ definieren wir die Räume

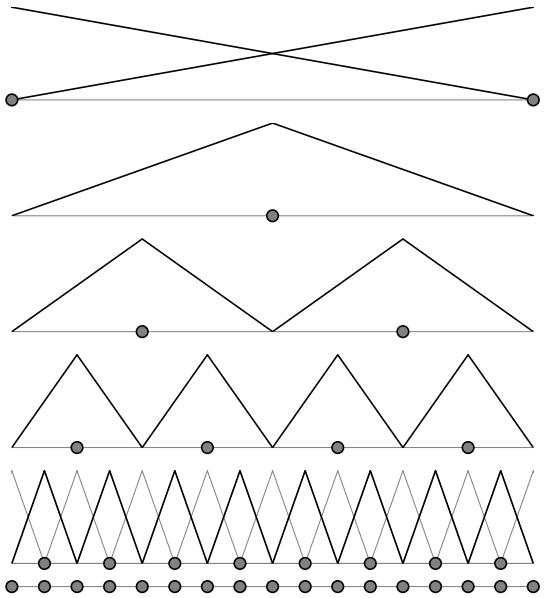

Abb. 4.2: Stückweise lineare hierarchische Basis. Abbildung entnommen aus [FG09].

$$V^{(d)} := \sum_{l_1=0}^{\infty} \cdots \sum_{l_d=0}^{\infty} W_{(l_1,\ldots,l_d)} = \bigotimes_{\mathbf{l}\in\mathbb{N}_0^d} W_{\mathbf{l}} \tag{4.1}$$

mit der hierarchischen Basis

$$\left\{\phi_{\mathbf{l},\mathbf{i}} \mid \mathbf{i} \in \mathbf{I}_{\mathbf{l}}, \mathbf{l} \in \mathbb{N}_0^d\right\}.$$

Es ist unschwer erkennbar, dass eine Funktion $f \in V^{(d)}$ eindeutig aufgespalten werden kann in

$$f(\mathbf{x}) = \sum_{\mathbf{l}} f_{\mathbf{l}}(\mathbf{x}), \quad f_{\mathbf{l}}(\mathbf{x}) = \sum_{\mathbf{i}\in\mathbf{I}_{\mathbf{l}}} v_{\mathbf{l},\mathbf{i}} \cdot \phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) \in W_{\mathbf{l}}, \tag{4.2}$$

wobei $v_{\mathbf{l},\mathbf{i}} \in \mathbb{R}$ die Koeffizienten der Darstellung von f in der hierarchischen Produktbasis sind. Nun folgt eine zentrale Beobachtung: Die Koeffizienten $v_{\mathbf{l},\mathbf{i}}$ zeigen ein spezifisches Abfallverhalten mit dem Level $\mathbf{l}$, wenn f beschränkte zweite Ableitungen hat, das heisst falls

$$f \in X^{q,r}(\overline{\Omega}) := \left\{u : \overline{\Omega} \to \mathbb{R} \mid D^{\boldsymbol{\alpha}} u \in L^q(\Omega), |\boldsymbol{\alpha}|_\infty \le r\right\}$$

mit $\Omega = (0,1)^d$ und $r = 2$ gilt. Hierbei bezeichnet

$$D^{\boldsymbol{\alpha}} f := \frac{\partial^{|\boldsymbol{\alpha}|_1} f}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}$$

die schwache Ableitung zum Multiindex $\boldsymbol{\alpha} \in \mathbb{N}_0^d$ mit den Normen $|\boldsymbol{\alpha}|_1 := \sum_{j=1}^d \alpha_j$ und $|\boldsymbol{\alpha}|_\infty := \max_{1\le j\le d} \alpha_j$. Zweimalige partielle Integration und Ausnutzen der Produktstruktur ergibt nach [BG04] für $\mathbf{l} > 0$ die Integraldarstellung

$$v_{\mathbf{l},\mathbf{i}} = \int_{[0,1]^d} \psi_{\mathbf{l},\mathbf{i}} \cdot D^{\mathbf{2}} f(\mathbf{x}) d\mathbf{x}$$

für den Koeffizienten $v_{\mathbf{l},\mathbf{i}}$ der hierarchischen Darstellung (4.2). Hierbei ist $\psi_{l_j,i_j}(x_j) := -2^{-(l_j+1)} \cdot \phi_{l_j,i_j}(x_j)$, und weiterhin $\psi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) := \prod_{j=1}^d \psi_{l_j,i_j}(x_j)$. Für Koeffizienten der Randfunktionen mit einem $l_j = 0$ findet keine partielle Integration in der j-ten Richtung statt, sondern x_j wird in Abhängigkeit von i_j auf 0 oder 1 gesetzt. Die allgemeinere Formel lautet

$$v_{\mathbf{l},\mathbf{i}} = \left[\int_{[0,1]} \cdots \int_{[0,1]} \prod_{\substack{j=1 \\ l_j\neq 0}}^{d} \psi_{l_j,i_j}(x_j) \left(\prod_{\substack{j=1 \\ l_j\neq 0}}^{d} \frac{\partial^2}{\partial x_j^2}\right) f(x) d \left(\prod_{\substack{j=1 \\ l_j\neq 0}}^{d} x_j\right)\right]_{\mathbf{x}|_{\mathbf{l}=0} := \mathbf{x}_{\mathbf{l},\mathbf{i}}|_{\mathbf{l}=0}}.$$

Wir können die Koeffizienten abschätzen mit

$$|v_{\mathbf{l},\mathbf{i}}| \le 2^{-d} \cdot 2^{-2\cdot|\mathbf{l}|_1} \cdot |f|_{\mathbf{2},\infty} = \mathcal{O}(2^{-2\cdot|\mathbf{l}|_1}), \quad \mathbf{l} > 0,$$

wobei $|f|_{\boldsymbol{\alpha},\infty} := \|D^{\boldsymbol{\alpha}} f\|_\infty$ gilt. Somit fallen für Funktionen mit beschränkten zweiten Ableitungen aus dem Raum $X^{q,2}(\overline{\Omega})$ die hierarchischen Koeffizienten mit der Rate $2^{-2\cdot|\mathbf{l}|_1}$ ab. Ein detaillierter Beweis findet sich in [BG04].

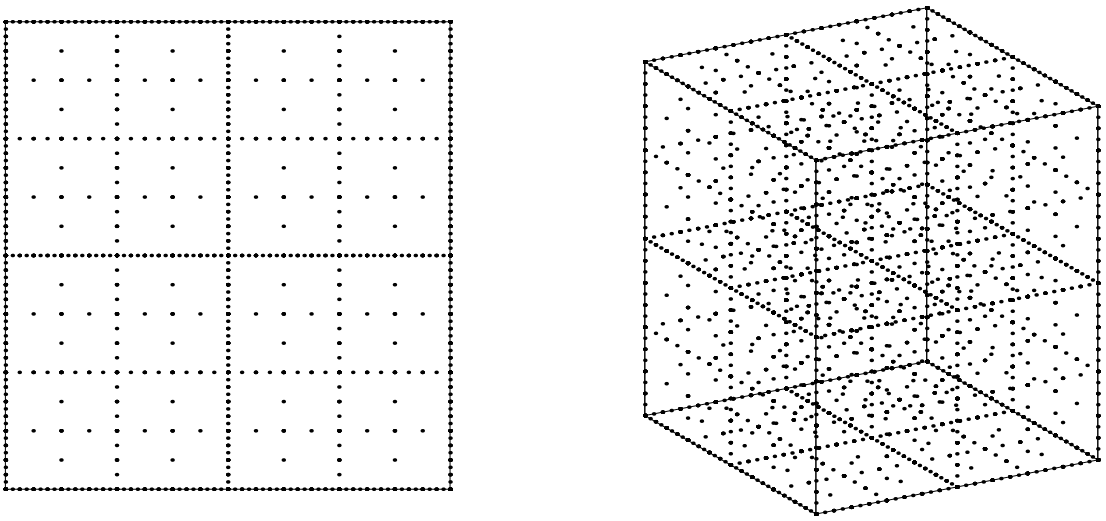

Abb. 4.3: Dünne Gitter in 2 und 3 Dimensionen. Abbildung entnommen aus [FG09].

Abhängig von der für uns relevanten Fehlernorm motiviert dieses Abfallverhalten verschiedene Trunkationen der Reihenentwicklung von f. Für ein gegebenes $k \in \mathbb{N}$ definieren wir den regulären Dünngitter-Raum

$$V_k^{(1)} := \bigotimes_{q(\mathbf{l}) \leq k} W_{\mathbf{l}} \tag{4.3}$$

mit

$$q(\mathbf{l}) := 1 + \sum_{\substack{m=1,\dots,d \\ l_m \neq 0}} (l_m - 1) \quad \text{und} \quad q(\mathbf{0}) = 0.$$

Die assoziierte trunkierte Reihe, also der Interpolant von f in $V_k^{(1)}$, ist somit

$$f_k^{(1)} := \sum_{q(\mathbf{l}) \leq k} \sum_{\mathbf{i}} v_{\mathbf{l},\mathbf{i}} \cdot \phi_{\mathbf{l},\mathbf{i}}.$$

Beispiele für zwei- und dreidimensionale reguläre dünne Gitter sind in Abbildung 4.3 dargestellt. Die Dimension des Raums $V_k^{(1)}$ lässt sich mit

$$|V_k^{(1)}| = \mathcal{O}\left(h_k^{-1} \cdot |\log_2 h_k|^{d-1}\right)$$

und $h_k = 2^{-k}$ abschätzen, wobei für den Interpolationsfehler einer Funktion f im regulären Dünngitterraum $V_k^{(1)}$

$$\left\| f - f_k^{(1)} \right\|_{L^p} = \mathcal{O}(h_k^2 \cdot k^{d-1})$$

gilt. Das konventionelle volle Gitter

$$V_k^{(\infty)} := \bigotimes_{|\mathbf{l}|_\infty \leq k} W_\mathbf{l}$$

realisiert mit $|V_k^{(\infty)}| = \mathcal{O}(h_k^{-d})$ Dimensionen einen L^p-Fehler der Ordnung $\mathcal{O}(h_k^2)$. Hier zeigt sich der Fluch der Dimension, denn die Maschenweite hängt exponentiell mit der Latent-Space-Dimension zusammen. Beim regulären Dünngitterraum zeigt sich der Fluch der Dimension nur in abgeschwächter Form im $|\log_2 h_k|$-Term. Dies sind die Fehlerraten zu einer fixen Dimension d, eine detaillierte Diskussion der Konstanten findet sich in [Gri06].

4.1.1 Voraussetzungen für die Dünngitterdiskretisierung

Wir haben beschrieben, wie eine skalare Funktion mit dünnen Gittern interpoliert werden kann. Die vektorwertige Funktion $\boldsymbol{y} : [0,1]^L \to \mathbb{R}^D$ diskretisieren wir, indem wir für jede der D Komponentenfunktionen eine Dünngitterbasis verwenden. Die Voraussetzung hierfür ist, dass die Komponentenfunktionen aus dem Raum

$$X^{q,r}(\overline{\Omega}) := \left\{ u : \overline{\Omega} \to \mathbb{R} \mid D^{\boldsymbol{\alpha}} u \in L^q(\Omega), |\boldsymbol{\alpha}|_\infty \leq r \right\}$$

mit $\Omega = (0,1)^d$ und $r = 2$ stammen. Dieser Raum ist echt enthalten in

$$\left\{ u : \overline{\Omega} \to \mathbb{R} \mid D^{\boldsymbol{\alpha}} u \in L^q(\Omega), |\boldsymbol{\alpha}|_1 \leq r \right\}.$$

Wegen der beschränkten zweiten gemischten Ableitungen machen wir trotz einer substanziellen Reduktion der Freiheitsgrade einen nur geringfügigen zusätzlichen Fehler. Somit gilt, dass wir auf einer kleineren Problemklasse

- mit der gleichen Anzahl von Freiheitsgraden einen kleineren Fehler machen oder
- für den gleichen Fehler weniger Freiheitsgrade benötigen.

Dies erinnert an das No-free-Lunch-Prinzip: Universell lässt sich die Komplexität eines Problems nicht brechen, Verbesserungen sind nur mit Vorwissen beziehungsweise zusätzlichen Voraussetzungen möglich. Die Frage, ob die mathematischen Voraussetzungen für die Dünngitterdiskretisierung beim GTM gegeben sind, lässt sich nicht theoretisch beantworten, sondern hängt mit der Anwendung zusammen.

An dieser Stelle sei erwähnt, dass das GTM in [BSW98b] Gauß-Basisfunktionen verwendet, die in der Regel hohe Glattheitsvoraussetzungen bei der Interpolation haben. Insofern ist die Forderung nach $H^{1,\text{mix}}$-Regularität nicht unrealistisch. Im Experimentierteil werden wir ein Beispiel für einen Datensatz geben, der sich mit einer Dünngitterdiskretisierung günstiger als mit einem vollen Gitter approximieren lässt, siehe Unterabschnitt 8.1.3.

Unabhängig davon gilt, dass wir nicht an einem L^2- oder L^∞-Fehler der einzelnen Komponentenfunktionen bei der Rekonstruktion einer Mannigfaltigkeit interessiert sind. Unser Zielfunktional misst die Divergenz zwischen zwei Wahrscheinlichkeitsdichten im Datenraum. Insofern sind Fehlerordnung und Komplexität von dünnen Gittern in erster Linie nur ein Hinweis darauf, dass die Anordnung der Freiheitsgrade günstiger ist als bei vollen Gittern.

4.1.2 Implementierung

Wie sich im noch folgenden Abschnitt 4.3 zeigen wird, ist es ausreichend, wenn unsere Implementierung der Dünngitterbasis folgende Schnittstelle bietet: Zu einem gegebenen Punkt $\mathbf{x} \in [0,1]^d$ muss sie alle Basisfunktionen mit $\mathbf{x} \in \operatorname{supp} \phi_n$ effizient bestimmen, und die Paare $(n, \phi_n(\mathbf{x}))$ zurückliefern. Hierbei gilt, dass

$$n := m(\mathbf{l}, \mathbf{i})$$

eine beliebige Nummerierung der tensorierten Hütchenfunktionen ist. Mit dieser einheitlichen Schnittstelle kann der GTM-Algorithmus mit verschiedenen Diskretisierungen kombiniert werden.

Eine wichtige Beobachtung ermöglicht die effiziente Funktionsauswertung: Die eindimensionalen Basisfunktionen in Abbildung 4.2 haben auf jedem Level disjunkte Träger, und die levelweise Vereinigung der Träger ist $[0,1]$. Wenn wir auf einem Level l eine Funktionsauswertung durchführen, existiert für jedes x genau ein i mit $x \in \operatorname{supp} \phi_{l,i}$. Dies gilt auch für die zwei Randfunktionen in der Abbildung 4.2 oben, wenn sie getrennt voneinander behandelt werden. Die Schnittpunkte der Träger zweier benachbarter Basisfunktionen sind vernachlässigbar, da dort beide Funktionen mit 0 ausgewertet werden. Die beschriebene Eigenschaft, dass zu jedem Punkt auf jedem Level maximal eine Basisfunktion nicht 0 ist, überträgt sich durch die Tensorproduktbildung

$$\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) := \prod_{j=1}^{d} \phi_{l_j,i_j}(x_j)$$

auf jedes der d-dimensionalen Level $\mathbf{l}$. Dies motiviert den Algorithmus 1, der für jedes Level die Basisfunktionen nummeriert und auswertet. Aus Gründen der Übersichtlichkeit werden die Randfunktionen nicht berücksichtigt.

Algorithmus 1 Punktauswertung in der Dünngitterbasis

Eingabe: Ein Punkt $\mathbf{x} \in [0,1]^L$
Ausgabe: Alle Tupel $(n, \phi_n(\mathbf{x}))$ mit $\phi_n(\mathbf{x}) \neq 0$

```
o ← 0        // Offset-Variable für die Funktionsnummer
for all 1 ≤ l ≤ (k, k, ..., k) and 1 ≤ q(l) ≤ k do
        v ← 0        // Funktionswert wird auf 0 gesetzt
        n ← o        // Funktionsnummer wird auf Offset gesetzt
        q ← 0        // Hilfsvariable zur Nummerierung
        i ← getBasisFunctionOnLevel(l)        // Einfache Operation in O(L)
        for j = 1 to d do
                n ← n + q · i_j
                v ← v · φ_{l_j,i_j}(x_j)        // Tensorierung
                q ← q · 2^{l_j − 1}
        end for
        print (n, v)        // Tupel zurückgeben
        o ← o + 2^{|l|_1 − d}
end for
```

4.2 Dünngitterquadratur

Theoretische Komplexitätsuntersuchungen in [TW88] bestätigen, dass die untere Schranke für den Rechenaufwand bei vielen Integrationsproblemen exponentiell mit der Dimension ansteigt. Dennoch bestehen Möglichkeiten, dem Fluch der Dimension zu begegnen:

- Eine gewisse Erleichterung verschafft der Übergang von einer garantierten Fehlerschranke zu einer stochastischen Formulierung mit Monte-Carlo-Integration und dem erwarteten Fehler, da hier die Konvergenzrate $N^{-\frac{1}{2}}$ unabhängig von der Dimension d gilt, siehe [Caf98]. Die Unabhängigkeit von der Dimension gilt jedoch nur für die Rate, da die Quadratwurzel der Varianz der zu integrierenden Funktion als Konstante eingeht. Mit Quasi-Monte-Carlo ist die Rate $(\log N)^d N^{-1}$ möglich.

- Eine alternative Herangehensweise ist, Verfahren für spezielle Funktionenklassen zu konstruieren. Es ist intuitiv klar, dass stärkere Voraussetzungen an die Funktionen eine effektivere Integration ermöglichen. Die Dünngitterquadratur erzielt auf diese Weise eine deutliche Effizienzsteigerung.

Die folgende Darstellung der Dünngitterquadratur in allgemeiner Form orientiert sich an [GG98]. In Unterabschnitt 4.2.3 werden wir nachweisen, dass die geforderten Glattheitseigenschaften bei unserem GTM erfüllt sind.

Wir betrachten die numerische Integration von Funktionen $f(\mathbf{x})$ aus der Funktionenklasse $\mathcal{F}$ auf dem d-dimensionalen Hyperwürfel $\Omega := [0,1]^d$. Es sei eine Folge von Quadraturregeln mit n_l^d Auswertungen gegeben, wobei $l \in N$ das Level bezeichnet. Eine Quadraturregel besteht aus Positionen $\mathbf{x}_{li}$ und Gewichten ω_{li}, und es gilt $n_l^d < n_{l+1}^d$. Die Anwendung der Regel lässt sich mit

$$Q_l^d f := \sum_{i=1}^{n_l^d} \omega_{li} \cdot f(\mathbf{x}_{li})$$

beschreiben. Die exakte Integration bzeichnen wir mit

$$I^d f := \int_\Omega f(\mathbf{x}) d\mathbf{x}.$$

Nun können wir den Quadraturfehler in Abhängigkeit von $l \in \mathbb{N}$ mit

$$E_l^d f := \left| I^d f - Q_l^d f \right|$$

ausdrücken.

4.2.1 Eindimensionale Trapezregel

Die Newton-Cotes-Formeln verwenden äquidistante Punkte und bestimmen die dazugehörigen Quadraturgewichte durch Integration der Lagrange-Polynome durch diese Punkte, siehe [DR84]. Eine Quadraturformel mit niedriger Glattheitsvoraussetzung ist die Trapezregel, die durch iterierte Anwendung auf größere Intervalle angewendet werden kann und dabei eine Ordnung verliert.

Wir betrachten zunächst eindimensionale Funktionen $f \in \mathcal{C}^r$ mit

$$\mathcal{C}^r := \left\{ \; f : [0,1] \to \mathbb{R} \;\middle|\; \left\| \tfrac{\partial^s}{\partial x^s} f \right\|_\infty < \infty, s \leq r \; \right\}.$$

Wir setzen für $l = 1$

$$n_l^1 := 1 \quad \text{mit} \quad Q_l^1 f := f\left(\frac{1}{2}\right),$$

und für $l > 1$

$$n_l^1 := 2^{l-1} + 1 \quad \text{mit} \quad Q_l^1 f := \sum_{i=1}^{n_l^1}{}'' 2^{1-l} \cdot f\left((i-1) \cdot 2^{1-l}\right).$$

Hierbei deutet $\sum''$ an, dass der erste und letzte Summand halbiert werden. Für Funktionen $f \in \mathcal{C}^2$ machen wir mit diesem Integrationsverfahren einen Fehler von

$$\left| E_l^1 f \right| = \mathcal{O}(2^{-2l}).$$

Im Fall unserer GTM liegt auch im schwachen Sinn keine zweifache Differenzierbarkeit vor. Dass die Rate der Trapezregel dennoch erreicht wird, weisen wir in Unterabschnitt 4.2.3 nach.

4.2.2 Smolyaks Konstruktion

Nun gehen wir zum mehrdimensionalen Fall mit $\Omega = [0,1]^d$ über. Smolyak hat ein Verfahren zur Konstruktion multivariater Quadraturregeln entwickelt, das auf der Tensorierung eindimensionaler Quadraturregeln basiert, siehe [Smo63]. Ähnlich wie im Fall der Dünngitterdiskretisierung werden beschränkte gemischte Ableitungen gefordert

$$\mathcal{C}_d^r := \left\{ \; f : \Omega \to \mathbb{R} \;\middle|\; \|D^{\boldsymbol{\alpha}} f\|_\infty < \infty \;\; , |\boldsymbol{\alpha}|_\infty \leq r \right\}.$$

Diese Räume entsprechen bei $[-1,1]^d$-Periodizität den Korobov-Räumen, siehe [Tem84],

$$\mathcal{E}_d^r := \left\{ f : \Omega \to \mathbb{R}, a(m_1, \ldots, m_d) = \mathcal{O}(|\overline{m}_1 \cdot \cdots \cdot \overline{m}_d|^{-r}) \right\}$$

mit $r > 1, \overline{m}_j := \max(1, m_j)$ und $a(m_1, \ldots, m_d)$ als den Fourier-Koeffizienten der Reihe

$$f(\mathbf{x}) = \sum_{m_1, \ldots, m_d = -\infty}^{\infty} a(m_1, \ldots, m_d) \cdot e^{-2\pi i (m_1 x_1 + \cdots + m_d x_d)}.$$

Nun betrachten wir eine Folge von eindimensionalen Quadraturregeln für eine univariate Funktion f

$$Q_l^1 = \sum_{i=1}^{n_l^1} \omega_{li} \cdot f(x_{li})$$

und definieren eine Differenz-Quadraturregel

$$\Delta_l^1 f := (Q_l^1 - Q_{l-1}^1) f \tag{4.4}$$

mit

$$Q_0^1 f := 0.$$

Smolyaks Konstruktion für d-dimensionale Funktionen f ist dann für $k \in \mathbb{N}$ und $\mathbf{l} \in \mathbb{N}^d$

$$Q_k^d f := \sum_{|\mathbf{l}|_1 \leq k+d-1} \left(\Delta_{l_1}^1 \otimes \cdots \otimes \Delta_{l_d}^1\right) f.$$

Das Tensorprodukt von d Quadraturregeln $\left(Q_{l_1}^1 \otimes \cdots \otimes Q_{l_d}^1\right)$ ist hierbei als die Summe über alle möglichen Kombination

$$\left(Q_{l_1}^1 \otimes \cdots \otimes Q_{l_d}^1\right) f := \sum_{i_1=1}^{n_{l_1}^1} \cdots \sum_{i_d=1}^{n_{l_d}^1} \omega_{l_1 i_1} \ldots \omega_{l_d i_d} \cdot f\left(x_{l_1 i_1}, \ldots, x_{l_d i_d}\right)$$

definiert. Eine einfache Produktformel lässt sich mit

$$\left(Q_k^1 \otimes \cdots \otimes Q_k^1\right) = \sum_{(1,\ldots,1) \leq \mathbf{l} \leq (k,\ldots,k)} \left(\Delta_{l_1}^1 \otimes \cdots \otimes \Delta_{l_d}^1\right) f$$

erzeugen und entspricht der Summation über den Würfel $|\mathbf{l}|_\infty \leq k$ anstelle des Simplex $|\mathbf{l}|_1 \leq k + d - 1$.

Diese Konstruktion ist analog zur Dünngitterinterpolation aus Abschnitt 4.1: In Formel (4.3) wird durch die Wahl des passenden q bestimmt, wie die Reihe abgeschnitten wird, und die Level der hierarchischen Basis können als Differenzen wie in in Formel (4.4) aufgefasst werden.

Smolyaks Konstruktion lässt sich mit der sogenannten Kombinationstechnik auch ohne die Differenzen $\Delta_{l_j}^1$ als

$$Q_k^d f = \sum_{l \leq |\mathbf{l}|_1 \leq k+d-1} (-1)^{k+d-|l|_1-1} \cdot \binom{d-1}{|\mathbf{l}|_1 - k} \cdot \left(Q_{l_1}^1 \otimes \cdots \otimes Q_{l_d}^1\right) f \tag{4.5}$$

darstellen, siehe [GSZ92]. Wir werden zur Erzeugung von Quadraturpunkten und Gewichten diese Darstellung verwenden. Für die folgende Fehlerabschätzung ist es erforderlich, dass die eindimensionale Quadraturregel $n_l^1 = O(2^l)$ und $\left|E_l^1 f\right| = \mathcal{O}\left((n_l^1)^{-r}\right)$ erfüllt. Dann benötigt die Dünngitterquadratur

$$n_l^d = \mathcal{O}\left(2^l \cdot k^{d-1}\right)$$

Funktionsauswertungen und erzeugt bei Funktionen $f \in \mathcal{C}^r$ den Quadraturfehler

$$\left|E_l^d f\right| = \mathcal{O}\left(2^{-lr} \cdot k^{(d-1)\cdot(r+1)}\right).$$

4.2.3 Voraussetzungen für die Dünngitterquadratur

Die eindimensionale Trapezregel hat zweite Ordnung für Funktionen $f \in \mathcal{C}^2$. In diesem Unterabschnitt behandeln wir die Frage, ob die geforderte Glatteit bei den Integralen des Sparse GTM gegeben ist. Wir können alle auftretenden Integrale über den Latent-Space auf diese drei

Typen zurückführen:

$$\int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\boldsymbol{W}\Phi(\mathbf{x}) - \mathbf{t}\|^2\right) d\mathbf{x},$$
$$\int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\boldsymbol{W}\Phi(\mathbf{x}) - \mathbf{t}\|^2\right) \phi_i(\mathbf{x}) d\mathbf{x} \quad \text{und}$$
$$\int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\boldsymbol{W}\Phi(\mathbf{x}) - \mathbf{t}\|^2\right) \phi_i(\mathbf{x})\phi_j(\mathbf{x}) d\mathbf{x}.$$

Die exp-Funktion ist unendlich oft differenzierbar, das Quadrat der euklidischen Norm ebenfalls. Der Integrand in $\mathbf{x}$ ist somit so glatt wie die Basis, die im Vektor Φ verwendet wird. Die lineare Hütchenbasis ist selbst im schwachen Sinne nicht zweifach differenzierbar. Sie ist stetig, einmal schwach differenzierbar und hat endlich viele „Knicke“ an den Stellen, wo Hütchen beginnen, enden, oder ihren höchsten Punkt haben. Dass endlich viele solcher Stellen die Fehlerordnung bei der Integration nicht beeinflussen, werden wir nun nachrechnen.

Satz 4.1. *Sei $f \in C([0,1])$ an endlich vielen Stellen $s_1, \ldots, s_W$ nicht differenzierbar und auf $[0,1] \setminus \bigcup_{z=1}^{W} \{s_z\}$ zweimal stetig differenzierbar. Dann hat die iterierte Trapezregel auf $[0,1]$ mit Maschenweite h die Fehlerordnung 2.*

Beweis. Wir wählen die Quadraturpunkte $x_i = i \cdot h$, und unterteilen unser Intervall auf diese Weise in $K := h^{-1}$ Abschnitte. Sei h so klein, dass in einem Abschnitt $[x_i, x_{i+1}]$ maximal ein Knick s_z liegt. Für die Abschnitte $[x_i, x_{i+1}]$, in denen f zweimal stetig differenzierbar ist, gilt nach der einfachen Trapezregel

$$\frac{x_{i+1} - x_i}{2}\left(f(x_i) + f(x_{i+1})\right) - \int_{x_i}^{x_{i+1}} f(x)dx = \frac{f^{(2)}(\xi_i)}{12}h^3$$

für ein $\xi_i \in [x_i, x_{i+1}]$. In den Abschnitten $[x_i, x_{i+1}]$, die einen Knick s_z beinhalten, ist die erste Ableitung durch $c := \|f^{(1)}\|_\infty$ beschränkt. Durch eine stückweise Anwendung des Hauptsatzes der Differential- und Integralrechnung gilt für $x \in [x_i, x_{i+1}]$

$$f(x_i) - (x - x_i)c \leq f(x) \leq f(x_i) + (x - x_i)c.$$

Für die rechte Intervallgrenze x_{i+1} gelten entsprechende Ungleichungen analog. Wir folgern

$$\frac{h}{2}f(x_i) - \frac{c}{8}h^2 \leq \int_{x_i}^{x_{i+h/2}} f(x)dx \leq \frac{h}{2}f(x_i) + \frac{c}{8}h^2.$$

Damit gilt

$$\left|\frac{x_{i+1} - x_i}{2}\left(f(x_i) + f(x_{i+1})\right) - \int_{x_i}^{x_{i+1}} f(x)dx\right| \leq \frac{c}{4}h^2.$$

Offenbar verlieren wir durch den Knick eine Ordnung. Dass sich dies nicht auf die iterierte Trapezregel übertragt, weisen wir nun nach. Es sei $c' := \max_i f^{(2)}(\xi_i)$ für alle Abschnitte i

ohne Knick. Für das Gesamtintervall gilt

$$\begin{aligned}\left|\frac{1}{K}\sum_{i=1}^{K}{}''f(x_i) - \int_0^1 f(x)dx\right| &\leq (K-W)\frac{c'}{12}h^3 + W\frac{c}{4}h^2 \\ &= \frac{c'}{12}h^2 - W\frac{c'}{12}h^3 + W\frac{c}{4}h^2 = \mathcal{O}(h^2).\end{aligned}$$

□

Der Satz 4.1 erklärt, wieso wir die Trapezregel auf eine Funktion mit endlich vielen Knicken anwenden können, ohne eine Ordnung zu verlieren.

Die Voraussetzungen des Satzes sind bei einer abzählbar unendlich großen Basis nicht mehr erfüllt: Für die Räume $V^{(d)}$ aus Gleichung (4.1) gilt, dass ihr Abschluss ohne Randfunktionen bezüglich der H^1-Norm der H_0^1-Raum ist, also $\overline{V}^{H^1} = H_0^1(\overline{\Omega})$ gilt, siehe [BG04]. Im Raum H^1 ist nicht garantiert, dass die Anzahl auftretender Knicke endlich ist. Ausserdem ist die Einbettung $H^m(\Omega) \hookrightarrow C(\Omega)$ nur für $m > \frac{d}{2}$ kompakt, womit ab zwei Dimensionen die Punktauswertung in H^1 nicht mehr stetig ist.

Wenn die Basis nicht endlich ist, benötigen wir einen Regularisierungsoperator, der die entsprechenden Glattheitseigenschaften sicherstellt. Wie in Abschnitt 3.3 dargestellt wurde, ist die Norm $\|\cdot\|_{H^{1,\text{mix}}}$ hierzu geeignet. Die Verwendung dieser Norm stellt sicher, dass die gemischten Ableitungen bis zur ersten Ordnung beschränkt sind, womit auch die Konvergenzvoraussetzungen der Dünngitterquadratur erfüllt sind.

4.2.4 Implementierung

Wir generieren die Quadraturpunkte und Gewichte über die Kombinationstechnik aus Formel (4.5). Da das Sparse GTM-Verfahren sehr häufig über den Latent-Space integriert, ist es sinnvoll, die Quadraturpunkte vorzuberechnen und dabei unnötige Funktionsauswertungen zu eliminieren. Wenn wir eine lexikographische Ordnung auf den d-dimensionalen Quadraturpunkten einführen und sie anschließend sortieren, so liegen identische Quadraturpunkte nebeneinander. Diese Vorgehensweise benötigt für K Quadraturpunkte den Aufwand $\mathcal{O}(K \cdot \log K)$.

Eine speicherplatzeffizientere Darstellung der Punkte ermöglicht zugleich das Zusammenfassen identischer Funktionsauswertungen in $\mathcal{O}(2^k)$, also der maximalen Punktanzahl in einer Raumrichtung. Die Datenstruktur orientiert sich am Präfixbaum: Wir verwenden eine Baumstruktur mit d Ebenen, die den einzelnen Raumdimensionen entsprechen. Die Knoten enthalten die entsprechenden Koordinaten, und die Blätter auf der $(d+1)$-Ebene die Quadraturgewichte. In Abbildung 4.4 wird die Datenstruktur exemplarisch dargestellt. Beim Einfügen von Quadraturpunkten werden entweder neue Knoten und ein Blatt angelegt, oder das Gewicht eines bereits bestehenden Blatts angepasst. Letzterer Fall entspricht dem Einsparen einer Funktionsauswertung.

Da die Nachfolger eines Knotens in einer verketteten Liste gespeichert sind, benötigen wir $\mathcal{O}(d \cdot 2^k)$ zum Einfügen von Quadraturpunkten. Dies ließe sich noch beschleunigen, ist jedoch nicht nötig, da der Baum nur einmal aufgebaut werden muss. Eine Traversierung ist in Linearzeit möglich.

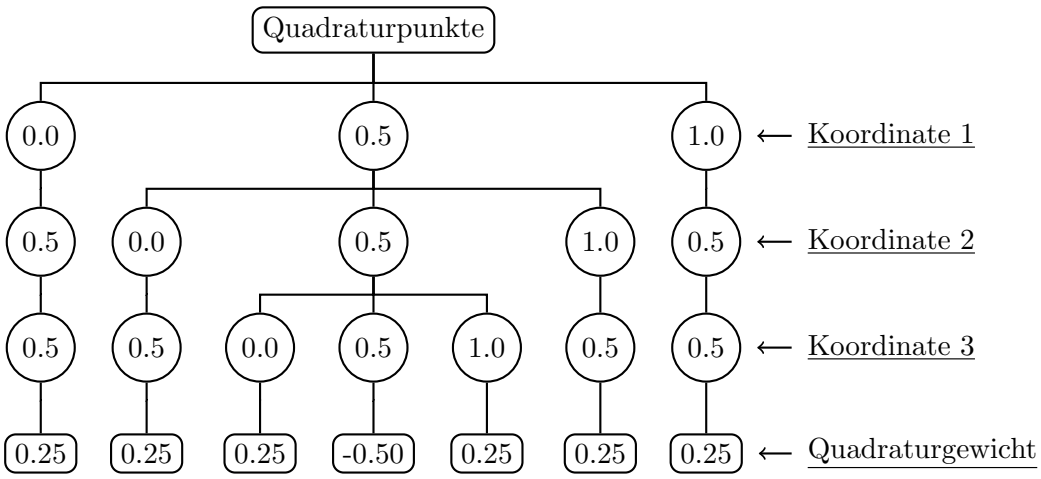

Abb. 4.4: Dünngitterquadratur-Datenstruktur für $d = 3$ und $l = 2$.

4.3 Umsetzung der Sparse GTM

In diesem Abschnitt beschreiben wir in Pseudocode die relevantesten Teile des Sparse GTM und schätzen schließlich die Laufzeit ab. Wir verwenden die empirische Verteilung einer endlichen Menge von Samples $\{\mathbf{t}_n\}_{n=1}^N$ anstelle einer kontinuierlichen Wahrscheinlichkeitsverteilung $d\mu(\mathbf{t})$ im Datenraum. Die $d\mathbf{x}$-Integrale diskretisieren wir mit einer Dünngitterquadraturregel bestehend aus den Punkten $\{\mathbf{x}_i\}_{i=1}^K$ und den Gewichten $\{\omega_i\}_{i=1}^K$. Die Parameter k_B und k_Q bezeichnen die Dünngitterlevel für Basis und Quadratur, und h_B und h_Q die dazugehörigen Maschenweiten. Die maximale Anzahl von Basisfunktionen oder Quadraturpunkten in eine Raumrichtung lässt sich mit $\mathcal{O}(h_B^{-1})$ beziehungsweise $\mathcal{O}(h_Q^{-1})$ abschätzen.

4.3.1 Punktauswertung des Mappings

Um die Dünngitterdiskretisierung für unser GTM zu nutzen, nummerieren wir die Basisfunktionen nach einem beliebigen Schema durch und fassen sie in dem Vektor

$$\Phi(\mathbf{x}) = \begin{pmatrix} \phi_1(\mathbf{x}) \\ \vdots \\ \phi_M(\mathbf{x}) \end{pmatrix}$$

zusammen. Das Mapping $\boldsymbol{y}$ wird mit

$$\boldsymbol{y}(\mathbf{x}) = \boldsymbol{W}\Phi(\mathbf{x})$$

diskretisiert, wobei $\boldsymbol{W}$ eine $D \times M$-Matrix ist. Dies entspricht einem Dünngitterbasisansatz für jede der D Komponentenfunktionen von $\boldsymbol{y} : [0,1]^L \to \mathbb{R}^D$. Algorithmus 2 wertet $\boldsymbol{y}(\mathbf{x})$ wesentlich schneller aus als eine naive Matrix-Vektor-Multiplikation.

Algorithmus 2 Punktauswertung von $\boldsymbol{y}(\mathbf{x})$

Eingabe: Ein Punkt $\mathbf{x} \in [0,1]^L$
Ausgabe: Der Vektor $\boldsymbol{y}(\mathbf{x}) \in \mathbb{R}^D$
$y \leftarrow (0,\ldots,0)^T$
for all $(j, \phi_j(\mathbf{x}))$ with $\phi_j(\mathbf{x}) \neq 0$ **do**
$y \leftarrow y + \phi_j(\mathbf{x}) \cdot \text{column}(\boldsymbol{W}, j)$
end for
return y

Bei der Abschneidefunktion der Dünngitterbasis

$$q(\mathbf{l}) := 1 + \sum_{\substack{m=1,\ldots,d \\ l_m \neq 0}} (l_m - 1)$$

mit $q(\mathbf{l}) \leq k_B$ liegt die Anzahl der zulässigen Levelkombinationen in $\mathcal{O}(k_B^L)$ oder $\mathcal{O}((\log h_B^{-1})^L)$. In Unterabschnitt 4.1.2 wurde beschrieben, dass zu jeder Levelkombination $\mathbf{l}$ maximal eine Basisfunktion ungleich 0 ist. Somit hat $\Phi(\mathbf{x})$ maximal $\mathcal{O}((\log h_B^{-1})^L)$ nicht-Null-Einträge, und der Aufwand für die Auswertung des Mappings $\boldsymbol{y}(\mathbf{x})$ liegt in $\mathcal{O}(D \cdot (\log h_B^{-1})^L)$.

4.3.2 Responsibilities-Berechnung

In diesem Unterabschnitt berechnen wir die *Responsibilities* aus Definition 3.8

$$R(\mathbf{t}, \mathbf{x}) = \frac{\exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2\right)}{\int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2\right) d\mathbf{x}}$$

vor. Das Einsetzen der Quadraturregeln und die Beschränkung auf $\mathbf{t} \in \{\mathbf{t}_n\}_{n=1}^N$ motiviert die folgende Definition.

Definition 4.2 (Responsibilities-Matrix)**.** Wir definieren zu gegebenem Mapping $\boldsymbol{y}$ und inverser Varianz β die *Responsibilities*-Matrix $\boldsymbol{R} \in \mathbb{R}^{K \times N}$ mit

$$(\boldsymbol{R})_{in} := \frac{\exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}_i) - \mathbf{t}_n\|^2\right)}{\sum_{i'=1}^K \omega_i \exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}_{i'}) - \mathbf{t}_n\|^2\right)}.$$

Offenbar gilt $R(\mathbf{t}_n, \mathbf{x}_i) \approx (\boldsymbol{R})_{in}$. Der Pseudocode zur Aufstellung der Matrix $\boldsymbol{R}$ ist in Algorithmus 3 dargestellt.

Die *Responsibilities*-Berechnung benötigt offenbar $\mathcal{O}(K \cdot N \cdot D \cdot (\log h_B^{-1})^L)$. Wie ein numerischer Unterlauf für große Distanzen zwischen einem $\mathbf{t}_n$ und den Quadraturpunkten $\mathbf{x}_i$ vermieden werden kann, wird in Unterabschnitt 4.4.2 erläutert.

Algorithmus 3 *Responsibilities*-Berechnung

Eingabe: Datenpunkte $\mathbf{t}_n$, Quadraturpunkte $\mathbf{x}_i$ und -gewichte ω_i, Mapping $\boldsymbol{y}$, inverse Varianz β

Ausgabe: Responsibilities-Matrix $\boldsymbol{R} \in \mathbb{R}^{K \times N}$

```
v_n ← 0        // N Normierungsvariablen mit 0 initialisieren
for i = 1 → K do
    for n = 1 → N do
        (R)_in ← exp(−(β/2) ‖y(x_i) − t_n‖²)
        v_n ← v_n + w_i · (R)_in        // Normierungsvariablen updaten
    end for
end for
for n = 1 → N do
    for i = 1 → K do
        (R)_in ← (R)_in ÷ v_n
    end for
end for
return R
```

4.3.3 M-Schritt-Gleichung

Wie wir aus Unterabschnitt 3.4.2 wissen, führt die M-Schritt-Minimierung zu D unabhängigen linearen Gleichungssystemen über alle w_{d*}-Einträge

$$\left(\boldsymbol{A} + \frac{2\lambda}{\beta}\boldsymbol{C}\right)\mathbf{w}_d = \mathbf{b}_d \tag{4.6}$$

mit $\mathbf{w}_d = (w_{d1}, \ldots, w_{dM})^T$, der $M \times M$-Matrix $\boldsymbol{A}$, der $M \times M$-Regularisierungsmatrix $\boldsymbol{C}$ und der rechten Seite $\mathbf{b}_d$, siehe (3.15). Die Regularisierungsmatrix kann bei Programmstart in $\mathcal{O}(M^2)$ aufgestellt werden, da sich ihre Einträge für alle Normen aus Abschnitt 3.3 analytisch bestimmen lassen. Bemerkenswert ist, dass die $\boldsymbol{A}$-Matrix weder von d noch von den Daten abhängt. Wir setzen die Quadraturregeln ein und approximieren die Matrixeinträge mit

$$\begin{aligned}(\boldsymbol{A})_{cj} &\approx \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K}\omega_i\,(\boldsymbol{R})_{in}\,\phi_c(\mathbf{x}_i)\phi_j(\mathbf{x}_i)\\ &= \sum_{i=1}^{K}\omega_i\left(\frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{R})_{in}\right)\phi_c(\mathbf{x}_i)\phi_j(\mathbf{x}_i).\end{aligned}$$

Für die rechte Seite gilt

$$\begin{aligned}(\mathbf{b}_d)_c &\approx \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K}\omega_i(\boldsymbol{R})_{in}\,(\mathbf{t}_n)_d\,\phi_c(\mathbf{x}_i)\\ &= \sum_{i=1}^{K}\frac{1}{N}\sum_{n=1}^{N}\omega_i(\boldsymbol{R})_{in}\,(\mathbf{t}_n)_d\,\phi_c(\mathbf{x}_i).\end{aligned}$$

Algorithmus 4 Aufstellen des M-Schritt-LGS

Eingabe: Datenpunkte $\mathbf{t}_n$, Quadraturpunkte $\mathbf{x}_i$ und -gewichte ω_i, *Responsibilities*-Matrix $\boldsymbol{R}$
Ausgabe: Matrix $\boldsymbol{A}$ und D Vektoren $\mathbf{b}_d$

```
for i = 1 to K do
    for all (j_1, φ_{j_1}(x_i)) with φ_{j_1}(x_i) ≠ 0 do
        for all (j_2, φ_{j_2}(x_i)) with φ_{j_2}(x_i) ≠ 0 do
            (A)_{j_1 j_2} ← (A)_{j_1 j_2} + ω_i · ( (1/N) Σ_{n=1}^{N} (R)_{in} ) · φ_{j_1}(x_i) · φ_{j_2}(x_i)
        end for
        for d = 1 to D do
            for n = 1 to N do
                (b_d)_{j_1} += (1/N) ω_i · (R)_{in} · (t_n)_d · φ_{j_1}(x_i)
            end for
        end for
    end for
end for
return (A, {b_d}_{d=1}^{D})
```

Wenn wir das Gleichungssystem schnell aufstellen wollen, gehen wir nicht Eintrag für Eintrag vor, sondern werten die Dünngitterbasis auf allen Quadraturpunkten aus und verteilen die Ergebnisse auf die $\boldsymbol{A}$-Matrix und die rechte Seite. Das Verfahren wird in Algorithmus 4 dargestellt. Die Summe $\frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{R})_{in}$ lässt sich in dem *Responsibilities*-Berechnungsschritt für alle $i = 1, \ldots, K$ ohne zusätzliche Laufzeitkomplexität vorberechnen. Deshalb liegt das Aufstellen des linearen Gleichungssystems in $\mathcal{O}\left(M^2 + K \cdot (\log(h_B^{-1})^L \left((\log(h_B^{-1})^L + D \cdot N\right)\right)$.

4.3.4 Lösungsverfahren

Bisher haben wir uns noch nicht auf ein Verfahren zur Lösung des linearen Gleichungssystems festgelegt. Mehrere Gründe sprechen für eine einfache LR-Zerlegung:

1. In dem noch folgenden Abschnitt 5.2 werden wir nachrechnen, dass die Integranden wegen der *Responsibilities* nicht separabel sind. Aus diesem Grund sind schnelle Matrix-Vektor-Produkt-Algorithmen, wie sie in [Feu05] in anderem Zusammenhang diskutiert werden, nicht anwendbar, so dass die Laufzeitkomplexität einer Matrix-Vektor-Multiplikation mit $\mathcal{O}(M^2)$ angegeben werden kann. Dies schränkt den Spielraum für ein Iterationsverfahren deutlich ein.

2. In dem nächsten Abschnitt 4.4 befassen wir uns mit der Kondition der Matrix $\boldsymbol{A}$. Es stellt sich heraus, dass sie beliebig schlecht ist, so dass die Anzahl der nötigen Schritte eines Iterationsverfahrens kaum abzuschätzen ist.

Da wir ein lineares Gleichungssystem mit D verschiedenen rechten Seiten haben, können wir eine Zerlegung der $\boldsymbol{A}$-Matrix D-mal nutzen. Unter der Annahme, dass das Berechnen der LR-Zerlegung $\mathcal{O}(M^3)$ und die Rücksubstitution $\mathcal{O}(M^2)$ benötigt, kommen wir auf eine Laufzeit für den M-Schritt von $\mathcal{O}(M^3 + D \cdot M^2)$. Wir verwenden die Implementation der LR-Zerlegung aus der GNU Scientific Library, siehe [GSL].

4.3.5 Beta-Berechnung

Die Beta-Berechnung erfordert das Bestimmen der rechten Seite von

$$\frac{1}{\beta} := \frac{1}{ND} \sum_{n=1}^{N} \sum_{i=1}^{K} \omega_i (\boldsymbol{R})_{in} \|\boldsymbol{y}(\mathbf{x}_i) - \mathbf{t}_n\|^2.$$

Die Summation erfolgt ohne Besonderheiten und benötigt $\mathcal{O}(N \cdot K \cdot D \cdot (\log h_B^{-1})^L)$.

4.4 Numerische Aspekte

In diesem Abschnitt diskutieren wir numerische Aspekte, die bei der Implementation des Sparse GTM relevant sind. Dazu gehören die Kondition der linken Seite des M-Schritt-LGS und Besonderheiten bei der *Responsibilities*-Berechnung durch die Verwendung negativer Quadraturgewichte.

4.4.1 Kondition der linken Seite

Wie in Abschnitt 4.3 dargestellt wurde, müssen im M-Schritt eines unregularisierten GTM D lineare Gleichungssysteme mit identischer linken Seite $\boldsymbol{A}$ gelöst werden.

Wir kennen die Konvergenzrate für den Quadraturfehler, wissen aber noch nicht, welche Eigenschaften die Quadraturpunkte mindestens erfüllen müssen, damit ein gut gestelltes lineares Gleichungssystem resultiert. Zur Analyse ist es sinnvoll, $\boldsymbol{A}$ als

$$\boldsymbol{A} = \boldsymbol{\Phi}^T \boldsymbol{G} \boldsymbol{\Phi}$$

darzustellen, wobei $\boldsymbol{\Phi}$ eine $K \times M$-Matrix mit Elementen $(\boldsymbol{\Phi})_{ij} = \phi_j(x_i)$ und $\boldsymbol{G}$ eine $K \times K$-Diagonalmatrix mit Elementen

$$(\boldsymbol{G})_{ii} = \frac{\omega_i}{N} \sum_{n=1}^{N} (\boldsymbol{R})_{in}$$

ist. Satz 4.3 zeigt, dass die Quadraturpunkte $\{\mathbf{x}_i\}_{i=1}^K$ so gewählt werden können, dass $\operatorname{rg} \boldsymbol{\Phi} = M$ gilt. Dies ist eine notwendige aber keine hinreichende Bedingung dafür, dass $\boldsymbol{A}$ vollen Rang hat. Wenn drei $M \times M$-Matrizen mit vollem Rang miteinander multipliziert werden, so hat das Produkt auch vollen Rang. Beispiel 4.4 beweist, dass dies nicht garantiert ist, wenn $\boldsymbol{\Phi}$ eine $K \times M$-Matrix mit Rang M und $\boldsymbol{G}$ eine $K \times K$-Diagonalmatrix ist.

Satz 4.3. *Seien $\{\phi_j\}_{j=1}^M$ linear unabhängige Basisfunktionen auf $[0,1]^D$. Dann lassen sich M Punkte $\{\mathbf{x}_i\}_{i=1}^M$ bestimmen, so dass $\boldsymbol{\Phi} \in \mathbb{R}^{M \times M}$ mit*

$$(\boldsymbol{\Phi})_{ij} = \phi_j(\mathbf{x}_i)$$

vollen Rang hat.

Beweis. Induktion nach z: Wir schreiben die Matrix $\mathbf{\Phi}$ als

$$\mathbf{\Phi} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_M \end{pmatrix}$$

mit den Zeilenvektoren $\mathbf{v}_i = (\phi_1(\mathbf{x}_i), \ldots, \phi_M(\mathbf{x}_i))$. $\mathbf{\Phi}_z$ bezeichne die Matrix, die aus den ersten z Zeilen von $\mathbf{\Phi}$ besteht. Wir konstruieren nun $\mathbf{\Phi}$ zeilenweise von oben nach unten.

- Induktionsanfang $z = 1$
  Wir wählen ein beliebiges $\mathbf{x}_1 \in [0,1]^D$ mit $\mathbf{v}_1 \neq 0$, wodurch $\mathbf{\Phi}_1$ definiert ist.

- Induktionsvoraussetzung für gegebenes $z < M$:

$$\operatorname{rg} \mathbf{\Phi}_z = z$$

- Induktionsschluss $z \to z + 1$
  Wir fassen $\mathbf{\Phi}_z$ als lineare Abbildung auf und wählen ein $\mathbf{v} \in \ker \mathbf{\Phi}_z \setminus \{0\}$. Wegen der linearen Unabhängigkeit der Basisfunktionen existiert ein $\mathbf{x} \in [0,1]^D$ mit

$$\mathbf{v}^T \cdot \begin{pmatrix} \phi_1(\mathbf{x}) \\ \vdots \\ \phi_M(\mathbf{x}) \end{pmatrix} \neq 0. \tag{4.7}$$

  Wir setzen $\mathbf{x}_{z+1} := \mathbf{x}$, wodurch $\mathbf{v}_{z+1}$ und $\mathbf{\Phi}_{z+1}$ definiert sind. Wegen Gleichung (4.7) gilt

$$\mathbf{v} \notin \ker \mathbf{\Phi}_{z+1},$$

  was bedeutet, dass $\ker \mathbf{\Phi}_{z+1}$ um eine Dimension kleiner geworden ist. Also gilt

$$\operatorname{rg} \mathbf{\Phi}_{z+1} = \operatorname{rg} \mathbf{\Phi}_z + 1 = z + 1.$$

Wegen $\mathbf{\Phi} = \mathbf{\Phi}_M$ gilt

$$\operatorname{rg} \mathbf{\Phi} = \operatorname{rg} \mathbf{\Phi}_M = M.$$

□

Beispiel 4.4. Sei

$$\mathbf{\Phi} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

und

$$\boldsymbol{G} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

Dann gilt

$$\boldsymbol{A} = \boldsymbol{\Phi}^T \boldsymbol{G} \boldsymbol{\Phi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Offenbar ist die $\boldsymbol{G}$-Matrix bedeutsam für den Rang von $\boldsymbol{A}$. In Satz 4.5 wird untersucht, wie das Spektrum von $\boldsymbol{A}$ mit dem von $\boldsymbol{\Phi}$ zusammenhängt, wenn $\boldsymbol{G}$ die Identität $\boldsymbol{I}$ ist. Dieser Fall ist instruktiv und tritt häufig bei Regressionsproblemen wie beispielsweise in [Gar04] auf.

Satz 4.5. *Es sei $\boldsymbol{G} = \boldsymbol{I}$, $K \geq M$, und $\sigma_1, \ldots, \sigma_M$ seien die Singulärwerte von $\boldsymbol{\Phi}$. Die Matrix $\boldsymbol{A}$ hat dann die Eigenwerte $\sigma_1^2, \ldots, \sigma_M^2$.*

Beweis. Eine Singulärwertzerlegung von $\boldsymbol{\Phi} \in \mathbb{R}^{K \times M}$ ergibt

$$\boldsymbol{\Phi} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{V}^T,$$

wobei U eine orthogonale $K \times K$-Matrix, V^T eine orthogonale $M \times M$-Matrix und $\boldsymbol{\Lambda}$ eine $K \times M$-Matrix

$$\boldsymbol{\Lambda} = \begin{pmatrix} \sigma_1 & 0 & \ldots & & & \\ 0 & \sigma_2 & 0 & \ldots & & \\ \ldots & 0 & \sigma_3 & 0 & \ldots & \\ & & \ddots & \ddots & \ddots & \\ & & \ldots & 0 & \sigma_{M-1} & 0 \\ & & & \ldots & 0 & \sigma_M \\ & & & & \ldots & 0 \\ & \vdots & & & & \vdots \end{pmatrix}$$

ist. Wegen

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{\Phi}^T \boldsymbol{\Phi} = \boldsymbol{V} \boldsymbol{\Lambda}^T \boldsymbol{U}^T \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{V}^T \\ &= \boldsymbol{V} \boldsymbol{\Lambda}^T \boldsymbol{\Lambda} \boldsymbol{V}^T \end{aligned}$$

sind $\boldsymbol{\Lambda}^T \boldsymbol{\Lambda}$ und $\boldsymbol{A}$ ähnlich, und beide Matrizen haben die gleichen Eigenwerte. Auf der Diagonalen von $\boldsymbol{\Lambda}^T \boldsymbol{\Lambda}$ stehen die Werte $\sigma_1^2, \ldots, \sigma_M^2$. □

Der Satz 4.5 demonstriert, wie das Spektrum von $\boldsymbol{A}$ aussieht, wenn $\boldsymbol{G} = \boldsymbol{I}$ ist. Beispiel 4.4 zeigt andererseits, dass das Spektrum durch die $\boldsymbol{G}$-Matrix beliebig degeneriert sein kann. Mit Voraussetzungen an $\boldsymbol{G}$ können wir sicherstellen, dass $\boldsymbol{A}$ vollen Rang behält, wie der folgende Satz 4.6 zeigt.

Satz 4.6. *Es seien alle $\boldsymbol{G}$-Diagonaleinträge > 0, und $\boldsymbol{\Phi}$ habe Rang M. Dann hat auch $\boldsymbol{A}$ vollen Rang.*

Beweis. Es sei $\mathbf{v} \in \mathbb{R}^M$ ein beliebiger Vektor $\neq 0$. Wegen $\operatorname{rg} \boldsymbol{\Phi} = M$, folgt $\boldsymbol{\Phi}\mathbf{v} \neq 0$. Es gilt

$$\mathbf{v}^T \boldsymbol{\Phi}^T \boldsymbol{G} \boldsymbol{\Phi} \mathbf{v} = (\boldsymbol{\Phi}\mathbf{v})^T \boldsymbol{G} (\boldsymbol{\Phi}\mathbf{v}) > 0.$$

Hieraus folgt, dass $\boldsymbol{\Phi}^T \boldsymbol{G} \boldsymbol{\Phi}$ positiv definit ist und somit vollen Rang hat. □

Wir fassen die Ergebnisse dieses Abschnitts zusammen:

- Es muss $\operatorname{rg} \boldsymbol{\Phi} = M$ gelten, damit $\boldsymbol{A}$ vollen Rang haben kann. Dies lässt sich durch ausreichend viele und gut platzierte Quadraturpunkte realisieren.
- Das Spektrum von $\boldsymbol{A}$ kann durch die $\boldsymbol{G}$-Matrix beliebig degeneriert sein.
- Mit Positivität der Einträge
$$(\boldsymbol{G})_{ii} = \frac{\omega_i}{N} \sum_{n=1}^{N} (\boldsymbol{R})_{in}$$
kann garantiert werden, dass die $\boldsymbol{A}$-Matrix vollen Rang behält, wenn $\operatorname{rg} \boldsymbol{\Phi} = M$ gilt.

Die $\boldsymbol{G}$-Einträge können sich aus folgenden Gründen ungünstig verhalten:

- Die Positivität von $\boldsymbol{G}$ ist nur für Quadraturregeln garantiert, deren Gewichte ω_i positiv sind. Dies ist bei der Dünngitterquadratur nicht der Fall. Eine andere Schwierigkeit mit negativen Quadraturgewichten wird in Unterabschnitt 4.4.2 diskutiert.
- Wenn sich keine Datenpunkte in der Nähe von $\boldsymbol{y}(\mathbf{x}_i)$ befinden, kann der dazugehörige $(\boldsymbol{G})_{ii}$-Eintrag beliebig klein werden, was das Spektrum von $\boldsymbol{A}$ negativ beeinflussen kann.

In der Praxis sind die genannten Punkte meist unproblematisch, denn

- durch eine geeignete Regularisierung bleibt das Problem gut gestellt.
- Die $(\boldsymbol{G})_{ii}$ werden nur in Ausnahmefällen negativ, was selbst dann keinen Effekt auf $\operatorname{rg} \boldsymbol{A}$ haben muss.
- Sollte die Kondition von $\boldsymbol{A}$ extrem groß werden und das Bild $\boldsymbol{y}([0,1]^D)$ degenerieren, bleibt dies für die niederdimensionale Projektion der Daten meist ohne Folgen, da die Degeneration nur dort auftritt, wo wenige Datenpunkte für einen großen $(\boldsymbol{G})_{ii}$-Eintrag sorgen können. Dies wird hier nicht im strengen mathematischen Sinn bewiesen, sondern reflektiert die Erfahrung aus einer Vielzahl von Experimenten mit Dünngitterbasis und -quadratur.

4.4.2 Normierungsfaktoren bei Responsibilities-Berechnung

Die Einträge der *Responsibilities*-Matrix aus Definition 4.2 werden mit

$$(\boldsymbol{R})_{in} = \frac{\exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}_i) - \mathbf{t}_n\|^2\right)}{\sum_{i=1}^{K} \omega_i \exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}_i) - \mathbf{t}_n\|^2\right)}$$

bestimmt. Wenn alle Abstände $\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2$ sehr groß sind, werden sehr kleine Zahlen durch einen beliebig kleinen Normierungsfaktor geteilt. Bei großem β oder Ausreißern in den Datenpunkten ist nicht ausgeschlossen, dass die Maschinengenauigkeit unterschritten wird. Um dieses Problem zu lösen, definieren für ein fixes $\mathbf{t}_n$ und alle $i = 1, \ldots, K$

$$s_i := -\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}_i) - \mathbf{t}_n\|^2$$

und schreiben unsere *Responsibilities* als

$$(\boldsymbol{R})_{in} = \frac{\exp(s_i)}{\sum_{i'=1}^{K} \omega_{i'} \exp(s_{i'})}. \tag{4.8}$$

Nun möchten wir den Ausdruck (4.8) so umformen, dass keine numerischen Probleme mehr auftreten. Eine Addition einer Konstanten auf alle s_i entspricht einer einfachen Erweiterung des Bruchs. Wir setzen also für alle $i = 1, \ldots, K$

$$s_i' := s_i - \max_i s_i.$$

Nun gilt

$$s_i' \leq 0 \quad \text{und} \quad \max_i s_i' = 0,$$

und damit

$$\exp\left(s_i'\right) \leq 1 \quad \text{und} \quad \max_i \exp\left(s_i'\right) = 1.$$

Bei einzelnen Summanden kann immer noch ein numerischer Unterlauf auftreten, diese sind dann jedoch nicht signifikant für das Ergebnis. Das Sparse GTM setzt diese Vorgehensweise ohne Änderung der Laufzeitkomplexität um.

Durch die Möglichkeit, auch sehr große Abstände zwischen Datenpunkten und dem Bild $\boldsymbol{y}([0,1]^L)$ handhaben zu können, entsteht ein neues Problem: Ein Punkt mit sehr großem Abstand übt bei gleichzeitig kleiner Varianz einen peak-artigen Einfluss auf die Struktur $y([0,1]^L)$ aus. Wenn dieser peak auf einem negativen Quadraturgewicht liegt, wird die numerische Approximation des Normierungsfaktors

$$\int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}_i) - \mathbf{t}_n\|^2\right) d\mathbf{x}$$

negativ. Dies ist unzulässig, da die *Responsibilities* grundsätzlich nicht-negativ sein müssen. Es gibt mehrere Möglichkeiten, dieses Problem zu beheben.

- Die Dünngitterquadraturregel wird so weit verfeinert, bis das negative Gewicht durch positive Nachbarn kompensiert wird.

- Wir verwenden eine Quadraturregel mit positiven Gewichten. Neben Quasi-Monte-Carlo kommt selbst das übliche volle Gitter in Betracht, denn wenn die Anzahl der Punkte proportional zur Anzahl der Dünngitterbasisfunktionen bleibt, wird die Komplexitätsordnung der dünnen Gitter nicht zerstört.

- Die einfachste Variante ist, die *Responsibilities* der Datenpunkte, die solche Grenzfälle erzeugen, auf Null zu setzen. Dies scheint zulässig, da sie aufgrund ihrer für gegebene inverse Varianz β atypisch hohen Entfernung als Ausreißer gewertet werden können. Diese Variante setzt das Sparse GTM gegenwärtig um.

4.5 Hilberträume mit reproduzierendem Kern

In diesem Abschnitt werden wir den Zusammenhang zwischen Regularisierungsterm und Hilberträumen mit reproduzierendem Kern behandeln. Die Dissertation [Gar04] enthält eine kompakte Darstellung, an der wir uns hier orientieren. Für eine umfangreichere Darstellung sei auf [SS01] verwiesen. Dieser Abschnitt erhebt keinen Anspruch auf Vollständigkeit, sondern konzentriert sich auf die Besonderheiten des GTM.

Definition 4.7 (Hilbertraum mit reproduzierendem Kern (RKHS))**.** Sei Ω eine nicht-leere Menge, und $\mathcal{H}$ ein Hilbertraum von Funktionen $f : \Omega \to \mathbb{R}$. $\mathcal{H}$ ist ein Hilbertraum mit reproduzierendem Kern, wenn eine Funktion $k : \Omega \times \Omega \to \mathbb{R}$ mit den folgenden Eigenschaften existiert:

1. k ist reproduzierend, also

$$\langle f, k(x, \cdot)\rangle_{\mathcal{H}} = f(x) \quad \text{für alle} \quad f \in \mathcal{H}, x \in \Omega$$

   und damit auch

$$\big\langle k(x, \cdot), k(x', \cdot)\big\rangle_{\mathcal{H}} = k(x, x') \quad \text{für alle} \quad x, x' \in \Omega.$$

2. k spannt $\mathcal{H}$ auf, also $\mathcal{H} = \overline{\operatorname{span}\{k(x, \cdot) \mid x \in \Omega\}}$, wobei $\overline{X}$ die Vervollständigung der Menge bezüglich der Hilbertraumnorm bezeichnet.

Wir können einen Hilbertraum mit reproduzierendem Kern auch als Hilbertraum von Funktionen auf Ω definieren, auf denen die Funktionale zur Punktauswertung $f \mapsto f(x')$ mit $x' \in \Omega$ stetig sind. Nach dem Satz von Riesz-Fischer existiert zu jedem x' eine Funktion $k(\cdot, x')$, so dass

$$f(x') = \big\langle f, k(\cdot, x')\big\rangle_{\mathcal{H}}$$

gilt. Eigenschaft 1 aus Definition 4.7 ist somit erfüllt. Wenn k' diese ebenfalls erfüllt, gilt für alle $x, x' \in \Omega$

$$k(x, x') = \big\langle k(x, \cdot), k'(x', \cdot)\big\rangle_{\mathcal{H}} = \big\langle k'(x', \cdot), k(x, \cdot)\big\rangle_{\mathcal{H}} = k'(x', x).$$

Die Symmetrie in den Argumenten führt zu $k = k'$ und zeigt die Eindeutigkeit des Kerns zu gegebenem Skalarprodukt.

Das sogenannte *Representer Theorem* sagt aus, dass sich in einem RKHS das Minimum eines regularisierten Kostenfunktionals als Summe von Kernfunktionen darstellen lässt, die ausschließlich in den vom Kostenfunktional ausgewerteten Punkten „befestigt“ sind. Statt der Anwendung dieses Satzes rechnen wir dies konkret beim GTM nach. Hierzu gehen wir zurück zur Definition 3.14 der M-Schritt-Minimierung

$$\boldsymbol{y}^{(s+1)} := \arg\min_{\boldsymbol{y}} \frac{\beta}{2} \int_{\mathbb{R}^D} \int_{[0,1]^L} R^{(s)}(\mathbf{t}, \mathbf{x}) \|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2 d\mathbf{x} d\mu(\mathbf{t}) + \lambda \cdot S(\boldsymbol{y}).$$

Wir diskretisieren das $d\mu(\mathbf{t})$-Integral durch die empirische Verteilung der Datenpunkte $\{\mathbf{t}_n\}_{n=1}^N$ und das $d\mathbf{x}$-Integral durch eine Dünngitterquadraturregel. Die korrekten *Responsibilities* seien

bereits berechnet und in der Matrix $\boldsymbol{R} \in \mathbb{R}^{K \times N}$ abgelegt. Da wir für den Regularisierungsterm voraussetzen, dass er additiv über die D Datenraumdimensionen ist, können wir die einzelnen Komponentenfunktionen von $\boldsymbol{y}$ unabhängig voneinander minimieren. Die folgenden Berechnungen führen wir mit der d-ten Komponentenfunktion $y_d(\mathbf{x})$ durch und kürzen aus Gründen der Übersichtlichkeit $(\mathbf{t}_n)_d$ mit t_n ab.

Wir nehmen an, dass sich der Regularisierungsterm in der Form

$$S(y_d) = \|Gy_d\|_{L^2}^2$$

darstellen lässt. Hierbei sei G ein linearer Operator. Wir bestimmen nun eine Orthonormalbasis aus Eigenvektoren $\{\phi_j(\mathbf{x})\}_{j=1}^{\infty}$ von G^*G mit den Eigenwerten $\{\gamma_j\}_{j=1}^{\infty}$. Wenn der Operator Null-Eigenwerte hat, muss der Nullraum als zusätzliche additive Komponente mitgeführt werden, wie in [Gar04] erwähnt wird. Aus Gründen der Übersichtlichkeit führen wir die folgenden Rechnungen ohne Null-Eigenwerte durch. Für eine mit

$$y_d(\mathbf{x}) = \sum_{j=1}^{\infty} \alpha_j \phi_j(\mathbf{x})$$

diskretisierte Funktion gilt nun

$$S(y_d) = \|Gy_d\|_{L^2}^2 = \sum_{j=1}^{\infty} \alpha_j^2 \gamma_j.$$

Wenn wir zusätzlich λ mit dem Faktor $\frac{2}{\beta}$ skalieren, hat die Minimierung von y die Gestalt

$$y_d^{(s+1)} := \arg\min_y \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{K} \omega_i (\boldsymbol{R})_{in} (y_d(\mathbf{x}_i) - t_n)^2 + \lambda \cdot \sum_{j=1}^{M} \alpha_j^2 \gamma_j.$$

Wir leiten nach α_j ab und setzen gleich 0. Dies ergibt

$$\frac{\partial}{\partial \alpha_j} \left(\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{K} \omega_i (\boldsymbol{R})_{in} (y_d(\mathbf{x}_i) - t_n)^2 + \lambda \sum_{j=1}^{M} \alpha_j^2 \gamma_j \right) = 0 \tag{4.9}$$

$$\Leftrightarrow \sum_{i=1}^{K} \frac{\omega_i}{N} \sum_{n=1}^{N} (\boldsymbol{R})_{in} (y_d(\mathbf{x}_i) - t_n) \phi_j(\mathbf{x}_i) + \lambda \alpha_j \gamma_j = 0 \tag{4.10}$$

$$\Leftrightarrow \sum_{i=1}^{K} \left(\left(\frac{\omega_i}{N} \sum_{n=1}^{N} (\boldsymbol{R})_{in} \right) y_d(\mathbf{x}_i) - \frac{\omega_i}{N} \sum_{n=1}^{N} (\boldsymbol{R})_{in} t_n \right) \phi_j(\mathbf{x}_i) + \lambda \alpha_j \gamma_j = 0. \tag{4.11}$$

Nun setzen wir

$$\eta_i := \frac{\omega_i}{N} \sum_{n=1}^{N} (\boldsymbol{R})_{in} \, t_n - \left(\frac{\omega_i}{N} \sum_{n=1}^{N} (\boldsymbol{R})_{in} \right) y_d(\mathbf{x}_i) \tag{4.12}$$

und lösen Zeile (4.11) nach α_j auf. Dies ergibt

$$\alpha_j = \frac{1}{\lambda\gamma_j} \sum_{i=1}^{K} \eta_i \phi_j(\mathbf{x}_i). \tag{4.13}$$

Also gilt

$$y_d(\mathbf{x}) = \sum_{j=1}^{\infty} \alpha_j \phi_j(\mathbf{x}) = \frac{1}{\lambda} \sum_{j=1}^{\infty} \frac{1}{\gamma_j} \sum_{i=1}^{K} \eta_i \phi_j(\mathbf{x}_i) \phi_j(\mathbf{x}) = \frac{1}{\lambda} \sum_{i=1}^{K} \eta_i k(\mathbf{x}_i, \mathbf{x}), \tag{4.14}$$

wobei die Kernfunktion k definiert ist als

$$k(\mathbf{x}, \mathbf{y}) := \sum_{j=1}^{\infty} \frac{1}{\gamma_j} \phi_j(\mathbf{x}) \phi_j(\mathbf{y}).$$

An dieser Stelle wird die Aussage des *Representer Theorem* deutlich, denn unsere Darstellung von y in Zeile (4.14) benötigt nur K Kernfunktionen, die in den Quadraturpunkten $\mathbf{x}_i$ befestigt sind. Es ist bemerkenswert, dass es sich hierbei um die exakte Lösung in einer unendlichdimensionalen Funktionenraumbasis $\{\phi_j(\mathbf{x})\}_{j=1}^{\infty}$ aus Eigenfunktionen des Regularisierungsoperators handelt. Nun setzen wir die Darstellung mit einer endlichen Summe in (4.12) ein und erhalten

$$\eta_i = \frac{\omega_i}{N} \sum_{n=1}^{N} (\boldsymbol{R})_{in} t_n - \frac{1}{\lambda} \left(\frac{\omega_i}{N} \sum_{n=1}^{N} (\boldsymbol{R})_{in} \right) \sum_{k=1}^{K} \eta_k k(\mathbf{x}_k, \mathbf{x}_i)$$

und damit das lineare Gleichungssystem

$$\left(\boldsymbol{I} + \frac{1}{\lambda} \boldsymbol{G}\boldsymbol{K} \right) \boldsymbol{\eta} = \tilde{\boldsymbol{R}}\mathbf{t}, \tag{4.15}$$

wobei $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_K)^T$, $\mathbf{t} = (t_1, \ldots, t_K)^T$, $\boldsymbol{K} \in \mathbb{R}^{K \times K}$ mit

$$(\boldsymbol{K})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j),$$

$\tilde{\boldsymbol{R}} \in \mathbb{R}^{K \times N}$ mit

$$(\tilde{\boldsymbol{R}})_{in} = \frac{\omega_i}{N} (\boldsymbol{R})_{in}$$

und $\boldsymbol{G} \in \mathbb{R}^{K \times K}$ mit

$$(\boldsymbol{G})_{ii} = \frac{\omega_i}{N} \sum_{n=1}^{N} (\boldsymbol{R})_{in}$$

gilt. Dieses Ergebnis ist interessant, weil die *Responsibilities* offensichtlich zu zusätzlichen $\boldsymbol{G}$- und $\tilde{\boldsymbol{R}}$-Matrizen in der Kerndarstellung führen, die in [Gar04] nicht entstehen.

Nun wollen wir noch einen interessanten Zusammenhang zwischen Funktionenraumdarstellung und Kerndarstellung herstellen. Sei hierzu $\boldsymbol{\Phi}$ eine $K \times \infty$-Matrix mit den Einträgen

$$(\boldsymbol{\Phi})_{ij} = \phi_j(\mathbf{x}_i).$$

Dann können wir y darstellen als

$$y_d(\mathbf{x}_i) = \sum_{j=1}^{\infty} \alpha_j \phi_j(\mathbf{x}_i) = \sum_{j=1}^{\infty} (\boldsymbol{\Phi})_{ij} \alpha_j = (\boldsymbol{\Phi}\boldsymbol{\alpha})_i$$

mit $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots)^T$. Wir definieren eine $\infty \times \infty$-Diagonalmatrix $\boldsymbol{C}$ mit

$$(\boldsymbol{C})_{jj} = \gamma_j$$

und schreiben (4.13) als

$$\lambda \cdot (\boldsymbol{C}\boldsymbol{\alpha})_j = \sum_{i=1}^{K} \eta_i \phi_j(\mathbf{x}_i) = (\boldsymbol{\Phi}^T \boldsymbol{\eta})_j.$$

Alle $j = 1, \dots, \infty$ Gleichungen in einem linearen Gleichungssystem zusammengefasst ergibt

$$\lambda \cdot \boldsymbol{C}\boldsymbol{\alpha} = \boldsymbol{\Phi}^T \boldsymbol{\eta}.$$

Die Gleichungen $i = 1, \dots, K$ aus (4.12) kombiniert ergeben

$$\boldsymbol{I}\boldsymbol{\eta} = \tilde{\boldsymbol{R}}\mathbf{t} - \boldsymbol{G}\boldsymbol{\Phi}\boldsymbol{\alpha}.$$

Diese linearen Gleichungssysteme fassen wir zusammen und erhalten

$$\begin{pmatrix} \boldsymbol{I} & \boldsymbol{G}\boldsymbol{\Phi} \\ \boldsymbol{\Phi}^T & -\lambda \cdot \boldsymbol{C} \end{pmatrix} \begin{pmatrix} \boldsymbol{\eta} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{R}}\mathbf{t} \\ \mathbf{0} \end{pmatrix}$$

Wenn wir das Schurkomplement nach $\boldsymbol{\eta}$ berechnen, erhalten wir die bereits bekannte Kerndarstellung aus Gleichung (4.15)

$$\left(\boldsymbol{I} - \frac{1}{\lambda} \boldsymbol{G}\boldsymbol{\Phi}\boldsymbol{C}^{-1}\boldsymbol{\Phi}^T \right) \boldsymbol{\eta} = \tilde{\boldsymbol{R}}\mathbf{t},$$

wobei offenbar $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{C}^{-1}\boldsymbol{\Phi}^T$ gilt. Nach $\boldsymbol{\alpha}$ aufgelöst, erhalten wir

$$\boldsymbol{\Phi}^T\boldsymbol{G}\boldsymbol{\Phi}\boldsymbol{\alpha} + \lambda\boldsymbol{C}\boldsymbol{\alpha} = \boldsymbol{\Phi}^T\tilde{\boldsymbol{R}}\mathbf{t}.$$

Dies entspricht dem üblichen M-Schritt-LGS wie es beispielsweise in Formel (4.6) mit

$$\boldsymbol{A} = \boldsymbol{\Phi}^T\boldsymbol{G}\boldsymbol{\Phi}$$

und

$$\mathbf{b}_d = \boldsymbol{\Phi}^T\tilde{\boldsymbol{R}}\mathbf{t}$$

zu finden ist.

4.6 Laufzeitaspekte

Die Formulierung des Sparse GTM erlaubt die Verwendung unterschiedlicher Basen und Quadraturverfahren. Wir wollen in diesem Abschnitt die Laufzeiten einiger Varianten behandeln.

4.6.1 Lokalität verschiedener Basen

In der Tabelle 4.1 stellen wir für verschiedene Basen die Anzahl der Funktionen und die Anzahl der $\phi_j(\mathbf{x}) \neq 0$ zu gegebenem $\mathbf{x} \in [0,1]^L$ gegenüber. Eine gitterartige Anordnung von Gauß-Funktionen wie in [BSW98b] unterliegt dem Fluch der Dimension, und die Basisfunktionen sind nicht lokal. Allerdings kann Lokalität durch Abschneiden der Gauß-Funktionen erzeugt werden, siehe beispielsweise [Wis08]. Die Dünngitterbasis unterliegt dem Fluch der Dimension nur mit dem $\log h^{-1}$-Term, ist jedoch weniger lokal als eine nodale Basis.

Basisfunktion	Gitter	Funktionsanzahl	Funktionsauswertung
Hütchen	voll	$\mathcal{O}(h^{-L})$	$\mathcal{O}(1)$
Hütchen	dünn	$\mathcal{O}(h^{-1} \cdot (\log h^{-1})^{L-1})$	$\mathcal{O}((\log h^{-1})^{L})$
Gauß	voll	$\mathcal{O}(h^{-L})$	$\mathcal{O}(h^{-L})$

Tabelle 4.1: Komplexität und Lokalität verschiedener Basen

4.6.2 Balancierung von Quadratur und Diskretisierung

Wir kennen die Fehlerrate des Quadraturverfahren und wissen aus Abschnitt 4.4, dass wir mindestens soviele Quadraturpunkte wie Basisfunktionen benötigen. Die Frage nach der Balancierung mit der Funktionsdiskretisierung ist noch offen. Als eine naheliegende Möglichkeit garantieren wir jeder Basisfunktion eine gewisse Anzahl von Quadraturpunkten innerhalb ihres Trägers. In [BSW98b] wird beispielsweise empfohlen, „$\mathcal{O}(100)$“ Quadraturpunkte im 2σ-Radius von jeder Gauß-Basisfunktion zu platzieren. Es bezeichne k_B den Diskretisierungslevel und k_Q den Quadraturlevel mit $h_B = 2^{-k_B}$ und $h_Q = 2^{-k_Q+1}$. Eine einfache Wahl von $q \in \mathbb{N}, q \geq 1$ mit $k_Q := k_B + q$ lässt die Anzahl der Quadraturpunkte je Basisfunktion mit feiner werdender Diskretisierung nicht sinken:

- Im Fall einer nodalen Basis und einer tensorierten Trapezregel liegt die Anzahl der Quadraturpunkte innerhalb des Trägers einer Basisfunktion in $\Theta(2^{L(k_Q-k_B)}) = \Theta(2^{Lq})$. Dies bedeutet, dass das Verhältnis von Quadraturpunkten zu Basisfunktionen unabhängig vom tatsächlichen Level ist.

- Im Fall einer Dünngitterbasis ist die Anzahl von Quadraturpunkten innerhalb des Trägers einer Basisfunktion schwieriger zu bestimmen, da die Träger unterschiedlich groß sind. Wenn wir nur die Punkte auf den Achsen berücksichtigen, lässt sich eine untere Schranke mit $\Omega(L \cdot 2^{k_Q-k_B}) = \Omega(L \cdot 2^q)$ angeben.

Dies motiviert die Daumenregel, ein um q höheres Quadraturlevel als Diskretisierungslevel zu verwenden.

4.6.3 Laufzeit

Ein Iterationsschritt des Sparse GTM besteht aus der *Responsibilities*-Berechnung, dem Aufstellen der linearen Gleichungssysteme, dem Lösen nach $\boldsymbol{y}$ und der Bestimmung von β. Wenn wir die Laufzeiten dieser Schritte aufaddieren, ergibt dies

$$\mathcal{O}(K \cdot N \cdot D \cdot A + K \cdot A^{2L} + M^3 + D \cdot M^2),$$

wobei A den Aufwand der Auswertung aller $\phi(\mathbf{x}) \neq 0$ zu einem $\mathbf{x}$ bezeichnet und die Dimension L fixiert ist. Um die Laufzeit einer GTM-Variante zu bestimmen, müssen die Angaben aus Tabelle 4.1 eingesetzt werden. Der Speicherverbrauch kann mit

$$\mathcal{O}(K \cdot N + M^2 + D \cdot M)$$

abgeschätzt werden.

Die folgenden Kombinationen reduzieren den Fluch der Dimension auf einen $\log h^{-1}$-Term:

- Eine Dünngitterdiskretisierung und eine Dünngitterquadraturregel mit $k_Q := k_B + q$,
- eine Dünngitterdiskretisierung und ein beliebiges Quadraturverfahren, wobei die Anzahl von Quadraturpunkten an die Anzahl der Basisfunktionen gekoppelt ist und
- eine RKHS-Diskretisierung mit Dünngitterquadraturregel.

In unseren Experimenten testen wir die ersten beiden GTM-Varianten, ein GTM mit RKHS-Diskretisierung wurde nicht implementiert. Wenn nichts Anderes angegeben ist, verwenden wir eine Dünngitterdiskretisierung und das niedrigste Vollgitter-Quadraturlevel, so dass die Anzahl der Quadraturpunkte dreimal größer als die Anzahl der Freiheitsgrade ist. Diese Vorgehensweise ist empfehlenswert, da so auch bei niedrigen Auflösungen keine Schwierigkeiten mit negativen Quadraturgewichten auftreten können, siehe auch Unterabschnitt 4.4.2.

5 Low-Rank GTM

Unsere Formulierung des GTM definiert eine Abbildung $\boldsymbol{y}(\mathbf{x}) = \boldsymbol{W}\Phi(\mathbf{x})$ aus dem Latent-Space in den Datenraum. Die Funktionen im Vektor $\Phi(\mathbf{x})$ sind frei wählbar, so dass beispielsweise eine Dünngitterdiskretisierung verwendet werden kann, um den Fluch der Dimension zu reduzieren.

Es existieren alternative Möglichkeiten, $\boldsymbol{y}$ zu diskretisieren. In diesem Kapitel verwenden wir eine Low-Rank Funktionsdarstellung, bei der nicht die Koeffizienten, sondern die Basisfunktionen die Unbekannten sind, siehe [BGM09]. Eine Funktion $f : [0,1]^L \to \mathbb{R}$ stellen wir mit

$$f(\mathbf{x}) = \sum_{r=1}^{P} s_r \prod_{i=1}^{L} g_{r,i}(x_i).$$

dar. Wäre $P = 1$, könnten wir nur separable $f(\mathbf{x})$ beschreiben. Da dies nur auf wenige Funktionen zutrifft, werden P separable Funktionen aufaddiert. Entscheidend für den Low-Rank Ansatz ist, dass die Anzahl der Summanden P gering gehalten und die fehlende Komplexität durch modifizierbare $g_{r,i}$ kompensiert wird. Bei einem P, das nicht exponentiell von der Latent-Space-Dimension L abhängt, hätten wir den Fluch der Dimension in den Freiheitsgraden unserer $\boldsymbol{y}$-Darstellung gebrochen.

5.1 Vektorwertige Low-Rank Darstellung

Es gibt zwei äquivalente Möglichkeiten, vektorwertige Low-Rank Darstellungen zu erzeugen. Zum einen können wir für jede der D Komponenten der Funktion $\boldsymbol{y} : [0,1]^L \to \mathbb{R}^D$ eine eigene Low-Rank Darstellung verwenden, also

$$y_d(\mathbf{x}) = \sum_{r=1}^{P} s_r^d \prod_{i=1}^{L} g_{r,i}^d(x_i) \tag{5.1}$$

mit skalaren $s_r^d \in \mathbb{R}$ und skalarwertigen $g_{r,i}^d : [0,1] \to \mathbb{R}$. Eine alternative Darstellung von $\boldsymbol{y}$ ist

$$\boldsymbol{y}(\mathbf{x}) = \sum_{r=1}^{P'} \mathbf{v}_r \prod_{i=1}^{L} h_{r,i}(x_i) \tag{5.2}$$

mit Vektoren $\mathbf{v}_r \in \mathbb{R}^D$ und skalarwertigen Funktionen $h_{r,i} : [0,1] \to \mathbb{R}$.

Beide Darstellungen sind gleichmächtig. Beim Übergang von Gleichung (5.2) zu (5.1) setzen wir

$$\begin{aligned} P &:= P', \\ g_{r,i}^d &:= h_{r,i} \qquad \text{für alle } r = 1,\dots,P, i = 1,\dots,L, d = 1,\dots,D \text{ und} \\ s_r^d &:= (\mathbf{v}_r)_d \quad \text{für alle } r = 1,\dots,P, d = 1,\dots,D. \end{aligned}$$

Beim Übergang von Gleichung (5.1) zu Gleichung (5.2) setzen wir

$$\begin{aligned} P' &:= D \cdot P, \\ h_{(d-1)\cdot D+r,i} &:= g^d_{r,i} \qquad \text{für alle } r = 1,\dots,P, i = 1,\dots,L, d = 1,\dots,D \text{ und} \\ \mathbf{v}_{(d-1)\cdot D+r} &:= \mathbf{e}_d \qquad \text{für alle } r = 1,\dots,P, d = 1,\dots,D, \end{aligned}$$

wobei $\mathbf{e}_d$ den d-ten Einheitsvektor im $\mathbb{R}^D$ bezeichnet. Diese Umformung ist nicht sonderlich kompakt, zeigt jedoch die Äquivalenz beider Darstellungsformen. Im Folgenden bleiben wir bei Darstellung (5.1) mit D voneinander unabhängigen Low-Rank Funktionen.

5.2 Separabilität der Exponentialfunktion

Ein Integral, das bei der *Responsibilities*-Berechnung als Normierungsfaktor auftritt und häufig berechnet werden muss, hat die Gestalt

$$\int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2\right) d\mathbf{x}, \tag{5.3}$$

wobei $\mathbf{t} \in \mathbb{R}^D$ ein vorgegebener Punkt im Datenraum ist. Nun setzen wir für $\boldsymbol{y}(\mathbf{x})$ eine Low-Rank Funktion ein, die den Latent-Space in den Datenraum abbildet. Es wäre wünschenswert, dass sich der Integrand als eine Summe von separablen Funktionen darstellen lässt, und die Integration wesentlich beschleunigt werden kann. Wir wollen einige elementare Umformungen durchführen:

$$\int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\sum_{d=1}^{D}\left(y_d(x) - (\mathbf{t})_d\right)^2\right) d\mathbf{x} \tag{5.4}$$

$$= \int_{[0,1]^L} \prod_{d=1}^{D}\left(\exp\left(-\frac{\beta}{2}y_d(x)^2\right)\cdot\exp\left(\beta y_d(x)(\mathbf{t})_d\right)\cdot\exp\left(-\frac{\beta}{2}(\mathbf{t})_d^2\right)\right) d\mathbf{x} \tag{5.5}$$

$$= \int_{[0,1]^L} \prod_{d=1}^{D}\left(\exp\left(-\frac{\beta}{2}\left(\sum_{r=1}^{R} s_r^d \prod_{i=1}^{L} g^d_{r,i}(x_i)\right)^2\right)\cdot\right. \tag{5.6}$$

$$\left.\cdot\exp\left(\beta\left(\sum_{r=1}^{R} s_r^d \prod_{i=1}^{L} g^d_{r,i}(x_i)\right)(\mathbf{t})_d\right)\cdot\exp\left(-\frac{\beta}{2}(\mathbf{t})_d^2\right)\right) d\mathbf{x} \tag{5.7}$$

$$= \int_{[0,1]^L} \prod_{d=1}^{D}\left(\prod_{r=1}^{P}\prod_{r'=1}^{P}\exp\left(-\frac{\beta}{2}s_r^d s_{r'}^d \prod_{i=1}^{L} g^d_{r,i}(x_i) g^d_{r',i}(x_i)\right)\cdot\right. \tag{5.8}$$

$$\left.\cdot\prod_{r=1}^{P}\exp\left(\beta(\mathbf{t})_d s_r^d \prod_{i=1}^{L} g^d_{r,i}(x_i)\right)\cdot\exp\left(-\frac{\beta}{2}(\mathbf{t})_d^2\right)\right) d\mathbf{x} \tag{5.9}$$

Der Übergang von den Zeilen (5.6) und (5.7) zu den Zeilen (5.8) und (5.9) verwendet die Umformung $\exp\left(\sum \dots\right) = \prod \exp(\dots)$. Die Produkte $\exp(\prod_{i=1}^{L} \dots)$ lassen sich hingegen nicht

mit der exp-Funktion vertauschen. Wegen der Darstellbarkeit als $\prod_{d=1}^{D}(\dots)$ ist unser Integrand

$$\exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2\right)$$

separabel bezüglich der Datenraumdimensionen. Wir integrieren jedoch nicht über den Datenraum mit $d\mathbf{t}$, sondern über den Latent-Space mit $d\mathbf{x}$. Obwohl wir $\boldsymbol{y}(\mathbf{x})$ als Summe separabler Funktionen darstellen können, überträgt sich diese Eigenschaft nicht auf den Integranden.

Die Darstellung mit vektorwertigen Koeffizienten aus Gleichung (5.2) kann das Auftreten der Produkte im Exponenten ebenfalls nicht verhindern. Da wegen der *Responsibilities* alle GTM-Integrale einen exp-Term enthalten, kann keines von einer Produktstruktur profitieren. Dies bedeutet, dass die numerische Quadratur L-dimensional ist und konventionell erfolgen muss.

5.3 M-Schritt

Da die M-Schritt-Minimierung der komplexeste Schritt der GTM-Funktionalminimerung ist, untersuchen wir in diesem Abschnitt die Auswirkungen eines Low-Rank Ansatzes. Die M-Schritt-Minimierung aus Definition 3.14 ohne Regularisierungsterm lautet

$$\arg\min_{\boldsymbol{y}} \int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x})\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2 d\mathbf{x}d\mu(\mathbf{t}).$$

Diese Berechnung minimiert das Funktional $\mathcal{K}$ in Richtung des zweiten Parameters auf Basis der bisher berechneten *Responsibilities* R. Durch Umformungen und Einsetzen der Low-Rank Darstellung von $\boldsymbol{y}$ erhalten wir

$$\int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x})\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2 d\mathbf{x}d\mu(\mathbf{t}) \tag{5.10}$$

$$= \sum_{d=1}^{D}\int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x})(y_d(\mathbf{x})-(\mathbf{t})_d)^2 d\mathbf{x}d\mu(\mathbf{t}) \tag{5.11}$$

$$= \sum_{d=1}^{D}\int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x})\left(\sum_{r=1}^{P} s_r^d \prod_{i=1}^{L} g_{r,i}^d(x_i)-(\mathbf{t})_d\right)^2 d\mathbf{x}d\mu(\mathbf{t}). \tag{5.12}$$

An Zeile (5.12) wird deutlich, dass wir die D Dimensionen unabhängig voneinander optimieren können. Dies ist insbesondere deshalb möglich, weil $R(\mathbf{t},\mathbf{x})$ nicht vom zu optimierenden $\boldsymbol{y}$ abhängt. Die nun folgenden Berechnungen gelten für ein d und müssen für $d=1,\dots,D$ durchgeführt werden.

Die Minimierung von

$$\int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x})\left(\sum_{r=1}^{P} s_r^d \prod_{i=1}^{L} g_{r,i}^d(x_i)-(\mathbf{t})_d\right)^2 d\mathbf{x}d\mu(\mathbf{t}) \tag{5.13}$$

in Richtung der $g_{r,i}^d$ erfordert nichtlineare Optimierung. Wir entscheiden uns für einen Alternating-Least-Squares-Algorithmus (ALS), wie er in [BGM09] beschrieben wird.

Zunächst bestimmen wir eine Koordinatenrichtung $i = m$ und minimieren in Richtung der Funktionen $\left\{g^d_{r,m}\right\}_{r=1}^P$. Die folgenden Berechnungen müssen für alle $m = 1, \ldots, L$ durchgeführt werden. Zur Minimierung werden die übrigen Richtungen fixiert, und wir definieren für $r = 1, \ldots, R$

$$p^d_{r,m}(\mathbf{x}) := s^d_r \prod_{\substack{i=1 \\ i \neq m}}^{L} g^d_{r,i}(x_i).$$

Es bleibt die Wahl einer Basis für die Funktionen $g^d_{r,m}$. Es ist theoretisch möglich, für jedes (d, r, m)-Tupel einen eigenen Basisfunktionensatz zu wählen, doch wir entscheiden uns für eine einheitliche Basis $\{\phi_k\}_{k=1}^M$ von eindimensionalen Funktionen. Wir fassen die Funktionen in einem Basisfunktionsvektor Φ zusammen. In Kombination mit einem Koeffizientenvektor $\mathbf{w}^d_{r,m} \in \mathbb{R}^M$ ermöglicht dies die kompakte Schreibweise $g^d_{r,m}(x_m) = \Phi(x_m)^T \mathbf{w}^d_{r,m}$.

Die einzelnen Komponenten des Vektors $\mathbf{w}^d_{c,m}$ bezeichnen wir mit $\left(\mathbf{w}^d_{c,m}\right)_j$ für $j = 1, \ldots, M$. Die notwendigen Bedingungen für die Minimierung des Ausdrucks in Zeile (5.13) sind bei bereits fixierten Parametern d und m, dass für alle $c = 1, \ldots, R$ und $j = 1, \ldots, M$ gilt

$$\begin{aligned} & \frac{\partial}{\partial\left(\mathbf{w}^d_{c,m}\right)_j} \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t}, \mathbf{x}) \left(\sum_{r=1}^P p^d_{r,m}(\mathbf{x}) \left(\Phi(x_m)^T \mathbf{w}^d_{r,m} \right) - (\mathbf{t})_d \right)^2 d\mathbf{x} d\mu(\mathbf{t}) && = 0 \\ \Leftrightarrow \quad & \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t}, \mathbf{x}) \left(\sum_{r=1}^P p^d_{r,m}(\mathbf{x}) \left(\Phi(x_m)^T \mathbf{w}^d_{r,m} \right) - (\mathbf{t})_d \right) p^d_{c,m}(\mathbf{x}) \phi_j(x_m) d\mathbf{x} d\mu(\mathbf{t}) && = 0. \end{aligned}$$

Offenbar müssen alle Koeffizientenvektoren für die Richtung m, also $\left\{\mathbf{w}^d_{c,m}\right\}_{c=1}^P$, in einem gemeinsamen linearen Gleichungssystem

$$\boldsymbol{A}^d_m \mathbf{w}^d_{*,m} = \mathbf{b}^d_m \tag{5.14}$$

bestimmt werden. Die Matrix $\boldsymbol{A}^d_m$ und die rechte Seite $\mathbf{b}^d_m$ haben eine Blockstruktur, die wir mit $c, c' = 1, \ldots, R$ indizieren. Es gilt, dass $\boldsymbol{A}(c, c')$ eine $M \times M$-Matrix mit den Einträgen

$$\left(\boldsymbol{A}^d_m(c, c')\right)_{jk} = \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t}, \mathbf{x}) p^d_{c',m}(\mathbf{x}) \phi_k(x_m) p^d_{c,m}(\mathbf{x}) \phi_j(x_m) d\mathbf{x} d\mu(\mathbf{t})$$

und $\mathbf{b}^d_m(c)$ ein M-dimensionaler Vektor mit den Einträgen

$$\left(\mathbf{b}^d_m(c)\right)_j = \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t}, \mathbf{x}) p^d_{c,m}(\mathbf{x}) \phi_j(x_m) (\mathbf{t})_d d\mathbf{x} d\mu(\mathbf{t})$$

ist.

5.4 Laufzeit

Die Matrix $\boldsymbol{A}^d_m$ hat die Größe $M \cdot P \times M \cdot P$. Um den Aufwand zur Aufstellung der Matrix zu bestimmen, untersuchen wir zunächst die Komplexität einer Auswertung des Integranden. Unter der Annahme einer $1d$-Hütchenbasis lässt sich die Auswertung eines $\phi_j(x_i)$ und auch eines $g^d_{r,i}(x_i)$ in konstanter Zeit realisieren. Die Bestimmung eines $p^d_{r,m}(\mathbf{x})$ liegt dann in $\mathcal{O}(L)$, die

Auswertung von $y_d(\mathbf{x})$ in $\mathcal{O}(P \cdot L)$. Damit benötigt ein $R(\mathbf{t}, \mathbf{x})$ zur Auswertung $\mathcal{O}(D \cdot P \cdot L)$. Die hiermit verbundene Annahme ist, dass der in R enthaltene Normierungsfaktor aus Formel (5.3) bereits bekannt ist. Insgesamt benötigt eine Integrandenauswertung also $\mathcal{O}(D \cdot P \cdot L^3)$.

Das äußere Integral entspricht nach Einsetzen der diskreten Samples einer Summierung über N Terme. Wenn wir annehmen, dass die Quadraturregel für das innere $d\mathbf{x}$-Integral I Auswertungen benötigt, hat die Berechnung eines Matrixeintrags den Aufwand $\mathcal{O}(N \cdot I \cdot D \cdot P \cdot L^3)$.

Erwähnenswert ist, dass im Fall einer eindimensionalen Hütchenbasis die einzelnen Matrixblöcke dünn besetzt sind. Für $\mathcal{O}(P^2 \cdot M)$ Einträge der Gesamtmatrix $\boldsymbol{A}_m^d$ bedeutet dies eine Gesamtkomplexität von $\mathcal{O}(N \cdot I \cdot D \cdot M \cdot P^3 \cdot L^3)$. Dies ist der benötigte Aufwand *einer* ALS-Richtung *einer* Dimension *eines* M-Schritts, was in Bezug auf das Gesamtproblem zusätzliche Komplexität bedeutet.

Beim Low-Rank Ansatz hängt bei fixem P die Anzahl der Freiheitsgrade nicht exponentiell von der Dimension des Latent-Space ab. Damit ist die Aussicht verbunden, höherdimensionale Latent-Spaces rechnen zu können. Die von uns ermittelte Laufzeit enthält jedoch nur auf den ersten Blick keine exponentielle Abhängigkeit von L: Die Anzahl der benötigten Quadraturpunkte I unterliegt dem Fluch der Dimension. Wie wir aus Abschnitt 5.2 wissen, sind die *Responsibilities* R nicht separabel bezüglich $\mathbf{x}$, so dass sich die Integrale nicht als Produkt von eindimensionalen Integralen darstellen lassen. Zwar würde eine Dünngitterquadratur die Integration beschleunigen, doch dann hätte der Low-Rank Ansatz in Hinblick auf die Komplexität keinen wesentlichen Vorteil gegenüber einer mächtigeren Dünngitterbasis. Dies ist der Grund, warum das Low-Rank GTM vorgestellt, aber nicht implementiert wurde.

Dieses Verfahren ist jedoch Ausgangspunkt für einen anderen Ansatz, welcher die exponentielle Abhängigkeit von der Dimension L vollständig eliminiert. Dieses Verfahren wird im folgenden Kapitel 6 dargestellt.

6 Low-ANOVA GTM

Wie in Abschnitt 5.2 beschrieben wurde, haben die beim GTM auftretenden Integranden keine Low-Rank Darstellung, selbst wenn $\boldsymbol{y}$ eine Low-Rank Funktion ist.

In diesem Kapitel beschreiben wir einen Ansatz für $\boldsymbol{y}$, bei dem die Integranden Produktstruktur haben und sich die L-dimensionalen Integrationsprobleme als Produkt von L eindimensionalen Integralen schreiben lassen. Auf diese Weise wird die exponentielle Abhängigkeit des GTM von L vollständig aufgehoben.

Der Ansatz beruht auf den niedrigsten Termen einer ANOVA-Reihe, und wird in Abschnitt 6.3 beschrieben. Zuvor stellen wir die ANOVA-Zerlegung einer Funktion und die Mayer Cluster Expansion vor.

6.1 ANOVA-Zerlegung

Diese allgemeine Darstellung der ANOVA-Zerlegung orientiert sich an [Hol08]. Sei $\Omega \subseteq \mathbb{R}$ ein Gebiet und

$$d\mu(\mathbf{x}) = \prod_{r=1}^{d} d\mu_r(x_r)$$

ein d-dimensionales Produktmaß auf Borel-Teilmengen von Ω^d. Hierbei ist $\mathbf{x} = (x_1, \ldots, x_d)$, und die μ_r sind Wahrscheinlichkeitsmaße auf Borel-Teilmengen von Ω. $V^{(d)}$ bezeichne den Hilbertraum von Funktionen $f : \Omega^d \to \mathbb{R}$ mit dem inneren Produkt

$$(f, g) := \int_{\Omega^d} f(\mathbf{x}) g(\mathbf{x}) d\mu(\mathbf{x}).$$

Für eine gegebene Teilmenge $\mathbf{u} \subseteq \mathcal{D}$, wobei $\mathcal{D} := \{1, \ldots, d\}$ die Koordinatenindizes bezeichnet, definieren wir die Projektion $P_{\mathbf{u}} : V^{(d)} \to V^{|\mathbf{u}|}$

$$P_{\mathbf{u}} f(\mathbf{x}_{\mathbf{u}}) := \int_{\Omega^{d-|\mathbf{u}|}} f(\mathbf{x}) d\mu_{\mathcal{D}\setminus\mathbf{u}}(\mathbf{x}).$$

Hierbei ist $\mathbf{x}_{\mathbf{u}}$ der $|\mathbf{u}|$-dimensionale Vektor, der die Komponenten von $\mathbf{x}$ enthält, deren Indizes in $\mathbf{u}$ enthalten sind, und es gilt $d\mu_{\mathcal{D}\setminus\mathbf{u}}(\mathbf{x}) := \prod_{r\notin\mathbf{u}} d\mu_r(x_r)$. Diese Projektionen erlauben eine eindeutige Zerlegung der Funktion $f \in V^{(d)}$ in eine endliche Summe

$$f(x_1, \ldots, x_d) = f_0 + \sum_{i=1}^{d} f_i(x_i) + \sum_{i=1, j>i}^{d} f_{i,j}(x_i, x_j) + \cdots + f_{1,\ldots,d}(x_1, \ldots, x_d),$$

welche in der kompakten Notation

$$f(\mathbf{x}) = \sum_{\mathbf{u} \subseteq \mathcal{D}} f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}) \tag{6.1}$$

geschrieben werden kann. Die 2^d Terme $f_{\mathbf{u}}$ beschreiben die Abhängigkeit der Funktion f von den Dimensionen $j \in \mathbf{u}$ bezüglich des Maßes μ. Sie werden rekursiv mit

$$f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}) := P_{\mathbf{u}} f(\mathbf{x}_{\mathbf{u}}) - \sum_{\mathbf{v} \subsetneq \mathcal{D}} f_{\mathbf{v}}(\mathbf{x}_{\mathbf{v}})$$

definiert und können auch explizit durch

$$f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}) = \sum_{\mathbf{v} \subseteq \mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} P_{\mathbf{v}} f(\mathbf{x}_{\mathbf{v}})$$

angegeben werden. Die Zerlegung (6.1) ist orthogonal im Sinne von

$$(f_{\mathbf{u}}, f_{\mathbf{v}}) = 0$$

für $\mathbf{u} \neq \mathbf{v}$. Diese Orthogonalität führt dazu, dass wir die Varianz der Funktion f ohne Kovarianzterme zerlegen können in

$$\sigma^2(f) = \sum_{\substack{\mathbf{u} \subseteq \mathcal{D} \\ \mathbf{u} \neq \emptyset}} \sigma^2(f_{\mathbf{u}}).$$

Wenn wir $\Omega = [0, 1]$ setzen und das Lebesgue-Maß $d\mu(\mathbf{x}) = d\mathbf{x}$ wählen, so wird $V^{(d)}$ zum Raum der quadratintegrierbaren Funktionen. Die ANOVA-Zerlegung bietet Vorteile, wenn ein Großteil der Varianz von f durch signifikant weniger als 2^d Terme rekonstruiert werden kann. Dieses Prinzip ähnelt der Dimensionsreduktion durch die PCA, wie in Unterabschnitt 2.2.2 nachzulesen ist. Die Superpositionsdimension einer Funktion f ist als die kleinste Zahl d_s definiert, so dass

$$\sum_{\substack{|\mathbf{u}| \leq d_s \\ \mathbf{u} \neq \emptyset}} \sigma^2(f_{\mathbf{u}}) \geq \alpha \sigma^2(f).$$

gilt, wobei beispielsweise $\alpha = 0.99$ ist.

Im folgenden Abschnitt werden wir anhand eines Beispiels aus der statistischen Physik nachvollziehen, welche Auswirkung die Anwendung der exp-Funktion auf die Superpositionsdimension einer Funktion hat.

6.2 Mayer Cluster Expansion

Die Mayer Cluster Expansion aus der statistischen Physik demonstriert, dass eine Funktion mit kleiner Superpositionsdimension im Argument der exp-Funktion nicht zu einer kleinen Superpositionsdimension des Gesamtausdrucks führt. Die folgende Darstellung orientiert sich an [LS95].

Die Hamilton-Funktion eines klassischen N-Partikel-Systems im Volumen V sei durch

$$H_N(\vec{p}^N, \vec{r}^N) = \frac{1}{2}\sum_{i=1}^{N} p_i^2 + \sum_{i=1, i<j}^{N} u_2(r_{ij})$$

gegeben, wobei $u_2(r_{ij})$ die paarweisen Potentiale zwischen den Partikeln i und j bezeichnet. Die kanonische Zustandssumme des Systems mit Temperatur T ist gegeben durch

$$Z_N(V,T) = \frac{1}{h^{3N}N!}\int_{\mathbb{R}^{3N}}\int_{V^N} \exp\left(-\frac{1}{2}\beta\sum_{i=1}^{N} p_i^2 - \beta\sum_{i=1, i<j}^{N} u_2(r_{ij})\right) d\vec{r}^N\, d\vec{p}^N,$$

wobei $\beta = \frac{1}{kT}$ und h eine Konstante zur Entdimensionalisierung von Z_N ist. Die Integration über die Impulse ergibt

$$Z_N(V,T) = \frac{1}{\lambda^{3N}N!}\int_{V^N} \exp\left(-\beta\sum_{i=1, i<j}^{N} u_2(r_{ij})\right) d\vec{r}^N \equiv \frac{1}{\lambda^{3N}N!} Q_N(V,T),$$

wobei $\lambda = \sqrt{2\pi h^2/kT}$ gilt. Die Idee der Cluster Expansion besteht in einem Variablenwechsel

$$\phi_{ij} = \exp\left(-\beta u_2(r_{ij})\right) - 1,$$

was die folgende Umformung des Konfigurationsintegrals

$$\begin{aligned} Q_N(V,T) &:= \int_{V^N} \exp\left(-\beta\sum_{i,i<j}^{N} u_2(r_{ij})\right) d\vec{r}^N = \int_{V^N}\prod_{i=1,i<j}^{N}(1+\phi_{ij})\, d\vec{r}^N \\ &= \int_{V^N}\left(1 + \sum_{i=1,i<j}^{N}\phi_{ij} + \sum_{i=1,i<j}^{N}\sum_{k=1,k<l}^{N}\phi_{ij}\phi_{kl} + \ldots\right) d\vec{r}^N \end{aligned}$$

ermöglicht. Die Wechselwirkungen zwischen den N Partikeln werden von der Cluster Expansion in einzelne Terme aufgespalten, deren Größe angibt, wieviele Partikel an diesem Potential beteiligt sind. Ob ein Abschneiden der Reihe eine gute Approximation ist, hängt von ihrem Abfallverhalten mit steigender Clustergröße zusammen. Es ist bemerkenswert, dass die exp-Funktion aus paarweisen Wechselwirkungen Terme erzeugt, die von allen $\vec{r}^N$ abhängen. Wenn wir dieses Ergebnis im ANOVA-Sinn interpretieren, hat der Ausdruck

$$\sum_{i=1, i<j}^{N} u_2(r_{ij})$$

Superpositionsdimension 2, und die Cluster Expansion demonstriert, dass diese Funktion im Argument der exp-Funktion ohne weitere Approximation zu Superpositionsdimension N führt. Dieses Resultat ist eine wichtige Beobachtung für die folgende Konstruktion des Low-ANOVA GTM.

6.3 Konstruktion

Zunächst stellen wir für $d = 1, \ldots, D$ die Komponentenfunktion $y_d(\mathbf{x})$ durch eine Funktion mit Superpositionsdimension 1 dar, also

$$y_d(\mathbf{x}) = \sum_{l=1}^{L} s_l^d g_l^d(x_l) \tag{6.2}$$

mit $s_l^d \in \mathbb{R}$ und $g_l^d(x) : [0,1] \to \mathbb{R}$ für $l = 1, \ldots, L$. Wie beim Low-Rank GTM sind die g_l^d selbst wieder Unbekannte. Wir betrachten nun, welche Auswirkung diese Funktionsdarstellung auf den Integranden

$$\int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2\right) d\mathbf{x}$$

aus Abschnitt 5.2 hat. Das Einsetzen von (6.2) ergibt

$$\begin{aligned}
& \exp\left(-\frac{\beta}{2}\sum_{d=1}^{D}\left(y_d(\mathbf{x}) - (\mathbf{t})_d\right)^2\right) \\
= & \prod_{d=1}^{D} \exp\left(-\frac{\beta}{2}\left(\sum_{l=1}^{L} s_l^d g_l^d(x_l) - (\mathbf{t})_d\right)^2\right) \\
= & \prod_{d=1}^{D} \exp\left(-\frac{\beta}{2}\sum_{l,l'=1}^{L} s_l^d s_{l'}^d g_l^d(x_l) g_{l'}(x_{l'}) + \beta\sum_{l=1}^{L} s_l^d g_l^d(x_l)(\mathbf{t})_d - \frac{\beta}{2}(\mathbf{t})_d\right).
\end{aligned}$$

Anhand der Doppelsumme $\sum_{l,l'=1}^{L}$ erkennen wir, dass die Argumente der exp-Funktionen Superpositionsdimension 2 haben. Offenbar führen die Quadrate aus der euklidischen Norm zu Wechselwirkungen zwischen Paaren von Latent-Space-Dimensionen. Wie wir im vorhergehenden Abschnitt gesehen haben, entstehen durch die Anwendung der exp-Funktion auf Funktionen mit Superpositionsdimension 2 Terme, die von allen Variablen abhängen. Es ist somit nicht möglich, durch eine Funktionsdiskretisierung mit niedrigen ANOVA-Termen eine Integrandendarstellung mit niedrigen ANOVA-Termen zu erhalten.

Wir können jedoch mit einer zusätzlichen Voraussetzung sicherstellen, dass der Integrand separabel wird. Hierfür setzen wir eine orthonormale Basis $\{\mathbf{v}_d\}_{d=1}^{D}$ des D-dimensionalen Datenraums voraus, und partitionieren diese in L Gruppen, von denen einige leer bleiben dürfen. Dies drücken wir durch eine Funktion

$$p : \{1, \ldots, D\} \to \{1, \ldots, L\}$$

aus. Da sie o.B.d.A. nicht injektiv ist, existiert kein eindeutiges Inverses. Wir verwenden daher

$$p^{-1} : \{1, \ldots, L\} \to \mathcal{P}\left(\{1, \ldots, D\}\right)$$

mit $i \in p^{-1}(j) \Leftrightarrow p(i) = j$. Hieraus folgt $p^{-1}(i) \cap p^{-1}(j) = \emptyset$ für $i \neq j$. Unsere Low-ANOVA

Funktion $\boldsymbol{y} : [0,1]^L \to \mathbb{R}^D$ stellen wir mit

$$\boldsymbol{y}(\mathbf{x}) = \sum_{d=1}^{D} \mathbf{v}_d g_d(x_{p(d)}) \tag{6.3}$$

dar. Hierbei sind die $g_d(x) : [0,1] \to \mathbb{R}$ skalare Funktionen und die $\mathbf{v}_d \in \mathbb{R}^D, d = 1, \ldots, D$ orthonormale Vektoren. Im Gegensatz zum Sparse GTM oder Low-Rank GTM diskretisieren wir die Komponentenfunktionen von $\boldsymbol{y}(\mathbf{x}) = (y_1(\mathbf{x}), \ldots, y_D(\mathbf{x}))^T$ nicht unabhängig voneinander. Bezüglich der exp-Funktion ist diese Darstellung günstig, wie die folgende Rechnung für ein fixes $\mathbf{t} \in \mathbb{R}^D$ zeigt:

$$\begin{aligned}
& \exp\left(-\frac{\beta}{2}\left\|\sum_{d=1}^{D} \mathbf{v}_d g_d(x_{p(d)}) - \mathbf{t}\right\|^2\right) \\
= \;& \exp\left(-\frac{\beta}{2}\left\langle \sum_{d=1}^{D} \mathbf{v}_d g_d(x_{p(d)}) - \mathbf{t}, \sum_{d=1}^{D} \mathbf{v}_d g_d(x_{p(d)}) - \mathbf{t} \right\rangle\right) \\
= \;& \exp\left(-\frac{\beta}{2} \sum_{d,d'=1}^{D} \underbrace{\langle \mathbf{v}_d, \mathbf{v}_{d'} \rangle}_{\delta_{dd'}} g_d(x_{p(d)}) g_{d'}(x_{p(d')}) + \beta \sum_{d=1}^{D} \langle \mathbf{v}_d, t \rangle \, g_d(x_{p(d)}) - \frac{\beta}{2} \langle \mathbf{t}, \mathbf{t} \rangle\right) \\
= \;& \exp\left(-\frac{\beta}{2} \sum_{d=1}^{D} g_d(x_{p(d)})^2 + \beta \sum_{d=1}^{D} \langle \mathbf{v}_d, \mathbf{t} \rangle \, g_d(x_{p(d)}) - \frac{\beta}{2} \langle \mathbf{t}, \mathbf{t} \rangle\right) \\
= \;& \exp\left(-\frac{\beta}{2} \langle \mathbf{t}, \mathbf{t} \rangle\right) \prod_{d=1}^{D} \exp\left(-\frac{\beta}{2} g_d(x_{p(d)})^2 + \beta \langle \mathbf{v}_d, \mathbf{t} \rangle \, g_d(x_{p(d)})\right).
\end{aligned}$$

Offenbar stellt unsere Funktionsdarstellung (6.3) eine Produktstruktur her, was sehr vorteilhaft bei der Integration ist. Im Folgenden sollte die Unbekannte d nicht mit dem Maß $d\mathbf{x}$ oder dx_l verwechselt werden. Es gilt

$$\begin{aligned}
& \int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\left\|\sum_{d=1}^{D} \mathbf{v}_d g_d(x_{p(d)}) - \mathbf{t}\right\|^2\right) d\mathbf{x} \\
= \;& \exp\left(-\frac{\beta}{2} \langle \mathbf{t}, \mathbf{t} \rangle\right) \int_{[0,1]^L} \prod_{d=1}^{D} \exp\left(-\frac{\beta}{2} g_d(x_{p(d)})^2 + \beta \langle \mathbf{v}_d, \mathbf{t} \rangle \, g_d(x_{p(d)})\right) d\mathbf{x} \\
= \;& \exp\left(-\frac{\beta}{2} \langle \mathbf{t}, \mathbf{t} \rangle\right) \prod_{l=1}^{L} \int_{[0,1]} \prod_{d \in p^{-1}(l)} \exp\left(-\frac{\beta}{2} g_d(x_l)^2 + \beta \langle \mathbf{v}_d, \mathbf{t} \rangle \, g_d(x_l)\right) dx_l.
\end{aligned}$$

Da das Integral in ein Produkt von L eindimensionalen Integralen zerfällt, haben wir den Fluch der Dimension auf Kosten der Darstellungsvielfalt von $\boldsymbol{y}(\mathbf{x})$ gebrochen.

Beispiel 6.1. Ein Beispiel für ein zulässiges Low-ANOVA Mapping ist

$$\boldsymbol{y}(\mathbf{x}) = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} \sin(1.5\pi x_1) + \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} \cos(1.5\pi x_1) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} x_2.$$

Hierbei gilt $D = 3$, $L = 2$ und $p(1) = 1$, $p(2) = 2$, $p(3) = 2$. In Abbildung 6.1 ist die Unabhängigkeit der Achse t_3 von den Achsen t_1 und t_2 erkennbar. Offenbar kann das Mapping $\boldsymbol{y}$ trotz der PCA-ähnlichen Orthogonalitätsbedingungen Nichtlinearitäten erfassen.

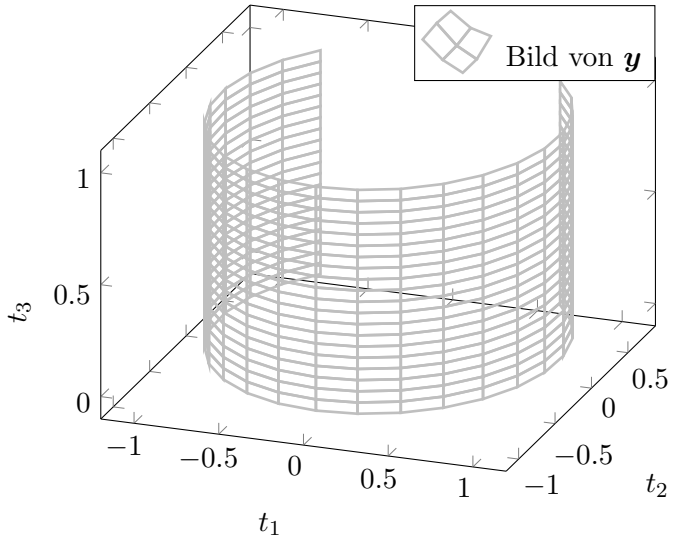

Abb. 6.1: Darstellung eines Low-ANOVA GTM Mappings.

Bemerkung. Eine Kugel ist ein Objekt, das wir mit dem Low-ANOVA GTM nicht darstellen können, da hierzu Basisfunktionen benötigt werden, die von mindestens zwei Veränderlichen abhängen.

Bemerkung. Von einer wahrscheinlichkeitstheoretischen Perspektive aus betrachtet können nur Daten modelliert werden, die keine Abhängigkeiten zwischen den verschiedenen Gruppen von Raumrichtungen $\{\mathbf{v}_d\}_{d \in p^{-1}(i)}$ und $\{\mathbf{v}_d\}_{d \in p^{-1}(j)}$ für $i \neq j$ haben. Diese Modellannahme ist weniger strikt als die der PCA, die Unabhängigkeit aller Richtungen ohne eine Gruppierung fordert.

Bemerkung. Prinzipiell kann ein Low-ANOVA GTM konstruiert werden, das mehrere Latent-Space-Dimensionen miteinander koppelt. Hierzu müssen die Datenraumdimensionen und die Latent-Space-Dimensionen partitioniert, und die Gruppen einander zugeordnet werden. Da die Darstellung eines solchen Verfahrens nicht instruktiver ist, und nicht alle Integrale zu Produkten zerfallen, haben wir uns auf eine Darstellung von $\boldsymbol{y}$ mit Superpositionsdimension 1 beschränkt.

6.4 Initialisierung

Zusätzliche Freiheitsgrade des Low-ANOVA GTM entstehen durch die Verteilung der D Raumdimensionen auf die L Latent-Space-Dimensionen. Der erste mögliche Unterschied nicht-trivialer Zuordnungen ergibt sich ab $L = 2$ und $D = 4$, wofür uns leider die Anschauung fehlt. Mögliche Strategien bei der Konstruktion von p sind,

- jeder Latent-Space-Dimension möglichst die gleiche Anzahl von Datenraumdimensionen zuzuordnen,
- oder ein 1:1-Mapping umzusetzen, wobei $D - L - 1$ Datenraumdimensionen der letzten Latent-Space-Dimension zugeordnet werden müssen.

Bei jedem Experiment mit einem Low-ANOVA GTM geben wir das verwendete Mapping p an.

Die Raumrichtungen $\{\mathbf{v}_d\}_{d=1}^D$ initialisieren wir mit einer klassischen PCA auf den Daten. Nun müssen die Funktionen g_d bestimmt werden. Es ist sinnvoll, aus jeder der L Gruppen $\left\{p^{-1}(l)\right\}_{l=1}^L$ eine Dimension d auszuwählen, und g_d mit $g_d(x) = x$ zu initialisieren. Die übrigen Funktionen werden auf Null gesetzt. Mit der entsprechenden Zuordnung p können wir auf diese Weise die ersten L Hauptachsen der PCA berücksichtigen. Alternativ können alle g_d auf $g_d(x) = x$ gesetzt werden. Welche Vorgehensweise zu besseren Ergebnissen führt, hängt letztlich von der Probleminstanz ab.

6.5 Responsibilities-Berechnung

Die Minimierung im ersten Paramter des GTM-Funktionals $\mathcal{K}$ erfordert die Berechnung der *Responsibilities* $R\colon \mathbb{R}^D \times [0,1]^L \to (0,\infty)$, wie sie in Definition 3.8 dargestellt werden. Sie entsprechen der zu jedem Datenpunkt $\mathbf{t} \in \mathbb{R}^D$ gehörigen Posterior-Verteilung im Latent-Space. Wir setzen in der folgenden Rechnung die Low-ANOVA Funktionsdarstellung ein, um zu demonstrieren, wie wir die L-dimensionalen *Responsibilities* in L Raumrichtungen aufspalten können.

$$
\begin{aligned}
& R(\mathbf{t}, \mathbf{x}) && (6.4)\\
= \; & \frac{\exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2\right)}{\int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}') - \mathbf{t}\|^2\right) d\mathbf{x}'} && (6.5)\\
= \; & \frac{\exp\left(-\frac{\beta}{2}\left\langle \mathbf{t}, \mathbf{t}\right\rangle\right) \prod_{d=1}^D \exp\left(-\frac{\beta}{2} g_d(x_{p(d)})^2 + \beta \left\langle \mathbf{v}_d, \mathbf{t}\right\rangle g_d(x_{p(d)})\right)}{\exp\left(-\frac{\beta}{2}\left\langle \mathbf{t}, \mathbf{t}\right\rangle\right) \prod_{l=1}^L \int_{[0,1]} \prod_{d\in p^{-1}(l)} \exp\left(-\frac{\beta}{2} g_d(x'_l)^2 + \beta \left\langle \mathbf{v}_d, \mathbf{t}\right\rangle g_d(x'_l)\right) dx'_l} && (6.6)\\
= \; & \prod_{l=1}^L \frac{\prod_{d\in p^{-1}(l)} \exp\left(-\frac{\beta}{2} g_d(x_l)^2 + \beta \left\langle \mathbf{v}_d, \mathbf{t}\right\rangle g_d(x_l)\right)}{\int_{[0,1]} \prod_{d\in p^{-1}(l)} \exp\left(-\frac{\beta}{2} g_d(x'_l)^2 + \beta \left\langle \mathbf{v}_d, \mathbf{t}\right\rangle g_d(x'_l)\right) dx'_l}. && (6.7)
\end{aligned}
$$

Dies motiviert die folgende Definition 6.2.

Definition 6.2 (Richtungs-*Responsibilities*)**.** Zu gegebenem Mapping $\boldsymbol{y}$ und inverser Varianz β definieren wir für die Latent-Space-Richtungen $l = 1, \ldots, L$ die Richtungs-*Responsibilities* $R_l \colon \mathbb{R}^D \times [0,1] \to \mathbb{R}$

$$R_l(\mathbf{t}, x) := \frac{\prod_{d \in p^{-1}(l)} \exp\left(-\frac{\beta}{2} g_d(x)^2 + \beta \langle \mathbf{v}_d, \mathbf{t} \rangle g_d(x)\right)}{\int_{[0,1]} \prod_{d \in p^{-1}(l)} \exp\left(-\frac{\beta}{2} g_d(x')^2 + \beta \langle \mathbf{v}_d, \mathbf{t} \rangle g_d(x')\right) dx'}.$$

Eine wichtige Eigenschaft der Richtungs-*Responsibilities* ist

$$\int_{[0,1]} R_l(\mathbf{t}, x) = 1 \quad \text{für alle } l = 1, \ldots, L \text{ und } \mathbf{t} \in \mathbb{R}^D. \tag{6.8}$$

Entsprechend der Gleichungszeile (6.7) können wir die $R(\mathbf{t}, \mathbf{x})$ aufspalten in

$$R(\mathbf{t}, \mathbf{x}) = \prod_{l=1}^{L} R_l(\mathbf{t}, x_l). \tag{6.9}$$

Im E-Schritt des Low-ANOVA GTM berechnen wir die L Richtungs-*Responsibilities* vor. Die eindimensionalen Integrale zur Normierung approximieren wir durch eine einfache Trapezregel mit K Punkten. Damit erfolgt das Aufstellen der Richtungs-*Responsibilities*-Matrizen $\boldsymbol{R}_l \in \mathbb{R}^{K \times N}$ in $\mathcal{O}(K \cdot N \cdot D)$ und wird analog zum Sparse GTM durchgeführt, siehe Unterabschnitt 4.3.2. Der entscheidende Unterschied zum Sparse GTM ist jedoch, dass die Anzahl der Quadraturpunkte K nicht von L abhängig ist, sondern $K = \mathcal{O}(h_Q^{-1})$ gilt. Die Produktdarstellung der Integrale setzt effektiv eine tensorierte Quadraturregel mit K^L Punkten auf $[0,1]^L$ in der linearen Laufzeit $\mathcal{O}(K)$ um.

In Unterabschnitt 4.4.2 haben wir beschrieben, wie bei der *Responsibilities*-Berechnung des Sparse GTM verhindert wird, dass durch beliebig kleine Zahlen geteilt werden muss. Beim Low-ANOVA GTM treten auch beliebig große Zahlen auf, denn obwohl

$$\begin{aligned} 1 &\geq \exp\left(-\frac{\beta}{2}\left\|\sum_{d=1}^{D} \mathbf{v}_d g_d(x_{p(d)}) - \mathbf{t}\right\|^2\right) \\ &= \exp\left(-\frac{\beta}{2}\langle \mathbf{t}, \mathbf{t} \rangle\right) \prod_{d=1}^{D} \exp\left(-\frac{\beta}{2} g_d(x_{p(d)})^2 + \beta \langle \mathbf{v}_d, \mathbf{t} \rangle g_d(x_{p(d)})\right) \end{aligned}$$

gilt, sind die Faktoren aus dem Produkt $\prod_{d=1}^{D} \ldots$ unbeschränkt. Dies ist darauf zurückzuführen, dass zwar

$$-\frac{\beta}{2}\left\|\sum_{d=1}^{D} \mathbf{v}_d g_d(x_{p(d)}) - \mathbf{t}\right\|^2 \leq 0$$

gilt, die Projektionen $\beta \langle \mathbf{v}_d, \mathbf{t} \rangle g_d(x_{p(d)})$ jedoch beliebig groß sein können. Im Argument der exp-Funktion führt dies schnell zu numerischen Überläufen.

Wenn wir für ein fixes l und $\mathbf{t}_n$ und alle $i = 1, \ldots, K$

$$s_i := \sum_{d \in p^{-1}(l)} -\frac{\beta}{2} g_d(x_{p(d)})^2 + \beta \langle \mathbf{v}_d, \mathbf{t} \rangle g_d(x_{p(d)})$$

setzen, können wir die diskretisierten Richtungs-*Responsibilities* $\boldsymbol{R}_l$ mit

$$(\boldsymbol{R}_l)_{in} = \frac{\exp(s_i)}{\sum_{i'=1}^{K} \omega_{i'} \exp(s_{i'})} \tag{6.10}$$

darstellen. Um hierbei den beschriebenen Überlauf zu vermeiden, gehen wir wie in Unterabschnitt 4.4.2 vor und setzen

$$s_i' := s_i - \max_i s_i.$$

Wegen

$$\exp(s_i') \leq 1 \quad \text{und} \quad \max_i \exp(s_i') = 1$$

können die $\boldsymbol{R}_l$-Matrizen auf diese Weise ohne numerische Schwierigkeiten aufgestellt werden. Da eindimensionale Quadraturregeln in der Regel ohne negative Gewichte auskommen, können keine Probleme mit negativen Normierungsfaktoren im Nenner auftreten.

6.6 M-Schritt

Im M-Schritt wird zu gegebenen *Responsibilities* ein optimales

$$\boldsymbol{y}(\mathbf{x}) = \sum_{d=1}^{D} \mathbf{v}_d g_d(x_{p(d)})$$

bestimmt. Die Optimierung kann jedoch nur dann mit linearen Gleichungssystemen stattfinden, wenn wir bei den Funktionenupdates die Vektoren $\mathbf{v}_d$ fixieren und bei den Vektorenupdates die Funktionen g_d fixieren. Diese Vorgehensweise wird in [BGM09] vorgeschlagen und folgt dem Prinzip des Alternating-Least-Squares-Algorithmus. Auf die Addition eines Regularisierungsterms verzichten wir, da die wenigen Freiheitsgrade von $\boldsymbol{y}$ wie eine starke Regularisierung wirken.

6.6.1 Funktionenupdates

Um die $g_d(x)$ zu modifizieren, benötigen wir eine Diskretisierung. Diese erfolgt wie in Abschnitt 5.3 über einen einheitlichen Basisfunktionsvektor $\Phi : [0,1] \to \mathbb{R}^M$ und Koeffizientenvektoren $\mathbf{w}_d \in \mathbb{R}^M$, so dass wir $g_d : [0,1] \to \mathbb{R}$ als

$$g_d(x) = \Phi(x)^T \mathbf{w}_d$$

schreiben können. Nun setzen wir unsere Funktionsdarstellung in das Integral der M-Schritt-Minimierung aus Definition 3.14 ein

$$\int_{\mathbb{R}^D}\int_{[0,1]^D} R(\mathbf{t},\mathbf{x}) \left\| \sum_{d=1}^{D} \mathbf{v}_d g_d(x_{p(d)}) - \mathbf{t} \right\|^2 d\mathbf{x}d\mu(\mathbf{t}) \tag{6.11}$$

$$= \int_{\mathbb{R}^D}\int_{[0,1]^D} R(\mathbf{t},\mathbf{x}) \left(\sum_{d=1}^{D} \left(g_d(x_{p(d)})^2 - 2\left\langle \mathbf{v}_d, \mathbf{t} \right\rangle g_d(x_{p(d)}) \right) + \left\langle \mathbf{t}, \mathbf{t} \right\rangle \right) d\mathbf{x}d\mu(\mathbf{t}) \tag{6.12}$$

$$= \int_{\mathbb{R}^D}\int_{[0,1]^D} R(\mathbf{t},\mathbf{x}) \left(\sum_{d=1}^{D} \left(\left\langle \Phi(x_{p(d)}), \mathbf{w}_d \right\rangle^2 - 2\left\langle \mathbf{v}_d, \mathbf{t} \right\rangle \left\langle \Phi(x_{p(d)}), \mathbf{w}_d \right\rangle \right) + \left\langle \mathbf{t}, \mathbf{t} \right\rangle \right) d\mathbf{x}d\mu(\mathbf{t}). \tag{6.13}$$

Wir erkennen, dass keine Koeffizientenvektoren $\mathbf{w}_d, \mathbf{w}_{d'}$ mit $d \neq d'$ miteinander interagieren. Die Minimierung kann somit für $d = 1, \ldots, D$ in D unabhängigen linearen Gleichungssystemen erfolgen. Wir leiten das Integral aus Zeile (6.13) nach der k-ten Komponente von $\mathbf{w}_d$ ab und setzen den Ausdruck gleich 0. Es gilt

$$\begin{aligned}
& \frac{\partial}{\partial(\mathbf{w}_d)_k} \int_{\mathbb{R}^D}\int_{[0,1]^D} R(\mathbf{t},\mathbf{x}) \cdot \\
& \cdot \left(\sum_{d=1}^{D} \left(\left\langle \Phi(x_{p(d)}), \mathbf{w}_d \right\rangle^2 - 2\left\langle \mathbf{v}_d, \mathbf{t} \right\rangle \left\langle \Phi(x_{p(d)}), \mathbf{w}_d \right\rangle \right) + \left\langle \mathbf{t}, \mathbf{t} \right\rangle \right) d\mathbf{x}d\mu(\mathbf{t}) = 0 \\
\Leftrightarrow \quad & \int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) \left(\left\langle \Phi(x_{p(d)}), \mathbf{w}_d \right\rangle - \left\langle \mathbf{v}_d, \mathbf{t} \right\rangle \right) \phi_k(x_{p(d)}) d\mathbf{x}d\mu(\mathbf{t}) = 0 \\
\Leftrightarrow \quad & \sum_{j=1}^{M} \left(\int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) \phi_j(x_{p(d)}) \phi_k(x_{p(d)}) d\mathbf{x}d\mu(\mathbf{t}) \right) (\mathbf{w}_d)_j \\
= \quad & \int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) \left\langle \mathbf{v}_d, \mathbf{t} \right\rangle \phi_k(x_{p(d)}) d\mathbf{x}d\mu(\mathbf{t}).
\end{aligned}$$

Wir schreiben dieses Ergebnis in einem linearen Gleichungssystem

$$\boldsymbol{A}_d \mathbf{w}_d = \mathbf{b}_d,$$

wobei $\boldsymbol{A}_d$ eine $M \times M$-Matrix mit den Einträgen

$$(\boldsymbol{A}_d)_{kj} = \int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) \phi_j(x_{p(d)}) \phi_k(x_{p(d)}) d\mathbf{x}d\mu(\mathbf{t})$$

und $\mathbf{b}_d$ ein M-dimensionaler Vektor mit den Einträgen

$$(\mathbf{b}_d)_k = \int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) \left\langle \mathbf{v}_d, \mathbf{t} \right\rangle \phi_k(x_{p(d)}) dx\mu(d\mathbf{t})$$

ist. Nun zeigen wir, wie wir die Separabilität der *Responsibilities* beim Aufstellen des linearen Gleichungssystem ausnutzen können. Nach Gleichung (6.9) gilt

$$\begin{aligned}
(\boldsymbol{A}_d)_{kj} &= \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t},\mathbf{x})\phi_j(x_{p(d)})\phi_k(x_{p(d)})d\mathbf{x}d\mu(\mathbf{t}) \\
&= \int_{\mathbb{R}^D} \int_{[0,1]^L} \left(\prod_{l=1}^{L} R_l(\mathbf{t},x_l)\right) \phi_j(x_{p(d)})\phi_k(x_{p(d)})d\mathbf{x}d\mu(\mathbf{t}) \\
&= \int_{\mathbb{R}^D} \prod_{\substack{l=1 \\ l\neq p(d)}}^{L} \underbrace{\int_{[0,1]} R_l(\mathbf{t},x_l)dx_l}_{=1} \cdot \int_{[0,1]} R_{p(d)}\phi_j(x_{p(d)})\phi_k(x_{p(d)})dx_{p(d)}d\mu(\mathbf{t}) \\
&= \int_{[0,1]} \left(\int_{\mathbb{R}^D} R_{p(d)}(\mathbf{t},x_{p(d)})d\mu(\mathbf{t})\right) \phi_j(x_{p(d)})\phi_k(x_{p(d)})dx_{p(d)}.
\end{aligned}$$

Die Separabilität der *Responsibilities* bewirkt offenbar, dass in der Matrix $\boldsymbol{A}_d$ keine Dimensionen des Latent-Space $\neq p(d)$ enthalten sind. Da ausserdem $\boldsymbol{A}_d = \boldsymbol{A}_{d'}$ für $p(d) = p(d')$ gilt, müssen nur L Matrizen aufgestellt werden. Die Initialisierung aller Matrizen benötigt $\mathcal{O}(L{\cdot}M^2)$, wenn sie nicht dünnbesetzt gespeichert werden. Zum Berechnen der Matrixeinträge iterieren wir wie in Unterabschnitt 4.3.3 in einer äußeren Schleife über die Quadraturpunkte und in einer inneren Schleife über die Basisfunktionen, die ungleich 0 ausgewertet werden. Da dies bei einer eindimensionalen Hütchenbasis maximal zwei sind, und die $\left(\int_{\mathbb{R}^D} R_{p(d)}(\mathbf{t},x_{p(d)})d\mu(\mathbf{t})\right)$ im E-Schritt bei gleicher Komplexität vorberechnet werden können, benötigt dieser Schritt $\mathcal{O}(L \cdot K)$. Auf der rechten Seite des linearen Gleichungssystems gilt analog

$$\begin{aligned}
(\mathbf{b}_d)_k &= \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) \left\langle \mathbf{v}_d,\mathbf{t}\right\rangle \phi_k(x_{p(d)})d\mathbf{x}d\mu(\mathbf{t}) \\
&= \int_{\mathbb{R}^D} \int_{[0,1]^L} \left(\prod_{l=1}^{L} R_l(\mathbf{t},x_l)\right) \left\langle \mathbf{v}_d,\mathbf{t}\right\rangle \phi_k(x_{p(d)})d\mathbf{x}d\mu(\mathbf{t}) \\
&= \int_{\mathbb{R}^D} \prod_{\substack{l=1 \\ l\neq p(d)}}^{L} \underbrace{\int_{[0,1]} R_l(\mathbf{t},x_l)dx_l}_{=1} \cdot \int_{[0,1]} R_{p(d)} \left\langle \mathbf{v}_d,\mathbf{t}\right\rangle \phi_k(x_{p(d)})dx_{p(d)}d\mu(\mathbf{t}) \\
&= \int_{\mathbb{R}^D} \int_{[0,1]} R_{p(d)}(\mathbf{t},x_{p(d)}) \left\langle \mathbf{v}_d,\mathbf{t}\right\rangle \phi_k(x_{p(d)})dx_{p(d)}d\mu(\mathbf{t}).
\end{aligned}$$

Die D rechten Seiten $\{\mathbf{b}_d\}_{d=1}^D$ benötigen zum Aufstellen $\mathcal{O}(D \cdot N \cdot K)$.

6.6.2 Vektorenupdates

Die Implementierung des Low-ANOVA GTM minimiert in Richtung der $\{\mathbf{v}_d\}_{d=1}^D$ zunächst ohne Nebenbedingung und orthonormalisiert die Ergebnisvektoren mit Gram-Schmidt. Dies ist nur eine der möglichen Verfahrensweisen, die Orthonormalitätsnebenbedingung umzusetzen. Warum der Vektorenupdate-Schritt bei keinem der Experimente verwendet wurde, erklären wir im Anschluss.

Minimierung

Wir bezeichnen die d-te Komponente von $\mathbf{v}_r$ mit $(\mathbf{v}_r)_d$. Wir formen das zu minimierende Integral zunächst um

$$\begin{aligned} & \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) \left\| \sum_{r=1}^{D} \mathbf{v}_r g_r(x_{p(r)}) - \mathbf{t} \right\|^2 d\mathbf{x} d\mu(\mathbf{t}) \\ = & \sum_{d=1}^{D} \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) \left(\sum_{r=1}^{D} (\mathbf{v}_r)_d g_r(x_{p(r)}) - (\mathbf{t})_d \right)^2 d\mathbf{x} d\mu(\mathbf{t}). \end{aligned}$$

Da wir erst im Anschluss die Orthonormalität wiederherstellen, kann die Minimierung für $d = 1, \ldots, D$ in D voneinander unabhängigen linearen Gleichungssystemen erfolgen. Für alle $s = 1, \ldots, D$ muss die Gleichung

$$\begin{aligned} & \frac{\partial}{\partial (\mathbf{v}_s)_d} \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) \left(\sum_{r=1}^{D} (\mathbf{v}_r)_d g_r(x_{p(r)}) - (\mathbf{t})_d \right)^2 d\mathbf{x} d\mu(\mathbf{t}) = 0 \\ \Leftrightarrow \quad & \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) \left(\sum_{r=1}^{D} (\mathbf{v}_r)_d g_r(x_{p(r)}) - (\mathbf{t})_r \right) g_s(x_{p(s)}) d\mathbf{x} d\mu(\mathbf{t}) = 0 \\ \Leftrightarrow \quad & \sum_{r=1}^{D} \left(\int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) g_r(x_{p(r)}) g_s(x_{p(s)}) d\mathbf{x} d\mu(\mathbf{t}) \right) (\mathbf{v}_r)_d \\ = & \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) (\mathbf{t})_d g_s(x_{p(s)}) d\mathbf{x} d\mu(\mathbf{t}) \end{aligned}$$

erfüllt sein. Dies führt zu einem linearen Gleichungssystem der Form

$$\boldsymbol{A} (\mathbf{v}_*)_d = \mathbf{b}_d$$

wobei $(\mathbf{v}_*)_d = ((\mathbf{v}_1)_d, \ldots, (\mathbf{v}_D)_d)^T$, $\boldsymbol{A}$ eine $D \times D$-Matrix mit den Einträgen

$$(\boldsymbol{A})_{sr} = \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) g_r(x_{p(r)}) g_s(x_{p(s)}) d\mathbf{x} d\mu(\mathbf{t}),$$

und $\mathbf{b}_d$ ein D-dimensionaler Vektor mit den Einträgen

$$(\mathbf{b}_d)_s = \int_{\mathbb{R}^D} \int_{[0,1]^L} R(\mathbf{t},\mathbf{x}) (\mathbf{t})_d g_s(x_{p(s)}) d\mathbf{x} d\mu(\mathbf{t})$$

ist. Nun nutzen wir die Separabilität der *Responsibilities* für das schnelle Berechnen der Matrix- und Vektoreinträge. Im Fall der $\boldsymbol{A}$-Matrix fällt auf, dass sie nicht von d abhängig und insofern in allen D Gleichungssystemen identisch sind. Für $p(r) \neq p(s)$ gilt

$$(\boldsymbol{A})_{sr} = \int_{\mathbb{R}^D} \int_{[0,1]} R_{p(r)}(\mathbf{t}, x_{p(r)}) g_r(x_{p(r)}) dx_{p(r)} \cdot \int_{[0,1]} R_{p(s)}(\mathbf{t}, x_{p(s)}) g_s(x_{p(s)}) dx_{p(s)} d\mu(\mathbf{t}).$$

Für $p(r) = p(s)$ gilt

$$(\boldsymbol{A})_{sr} = \int_{\mathbb{R}^D} \int_{[0,1]} R_{p(r)}(\mathbf{t}, x_{p(r)}) g_r(x_{p(r)}) g_s(x_{p(s)}) dx_{p(r)} d\mu(\mathbf{t}).$$

Auch die rechte Seite lässt sich mit eindimensionalen Integralen darstellen. Es gilt

$$(\mathbf{b}_d)_s = \int_{\mathbb{R}^D} \int_{[0,1]} R_{p(s)}(\mathbf{t}, x_{p(s)})(\mathbf{t})_d g_s(x_{p(s)}) dx_{p(s)} d\mu(\mathbf{t}).$$

Orthonormalisierung

Schließlich kommt die Gram-Schmidt Orthonormalisierung $\{\mathbf{v}_d\}_{d=1}^D \mapsto \{\mathbf{v}_d^*\}_{d=1}^D$ zum Einsatz. Wir wiederholen für $i = 1, \ldots, D$ folgende Schritte

$$\begin{aligned} \mathbf{v}_i^\dagger &:= \mathbf{v}_i - \sum_{j=1}^{i-1} \mathbf{v}_j^* \left\langle \mathbf{v}_j^*, \mathbf{v}_i \right\rangle \\ \mathbf{v}_i^* &:= \frac{\mathbf{v}_i^\dagger}{\|\mathbf{v}_i^\dagger\|} \end{aligned}$$

Diskussion

Das beschriebene Vektorenupdate ist im Low-ANOVA GTM implementiert, wurde jedoch in den Experimenten deaktiviert. Dies hat zwei Gründe:

- Die $\{\mathbf{v}_d\}_{d=1}^D$ verändern sich nur sehr geringfügig. Dies lässt sich mit dem zu minimierenden Integral
$$\int_{\mathbb{R}^D} \int_{[0,1]^D} R(\mathbf{t}, \mathbf{x}) \left\| \sum_{r=1}^D \mathbf{v}_r g_r(x_{p(r)}) - \mathbf{t} \right\|^2 d\mathbf{x} d\mu(\mathbf{t})$$
erklären. Die *Responsibilities* $R(\mathbf{t}, \mathbf{x})$ sind dort besonders groß, wo $\mathbf{t}$ und $\boldsymbol{y}(\mathbf{x})$ nah beieinander liegen. Dies bedeutet, dass durch R kleine Abstände stark und große Abstände schwach gewichtet werden, was zur Folge hat, dass das Verkleinern von großen Distanzen eine geringere Bedeutung hat als die Beibehaltung kleiner Distanzen. Dieses intuitive Argument erklärt, wieso die Vektorenupdates meist keine Veränderung bewirken. Zu dieser Sattelpunktproblematik siehe auch Unterabschnitt 3.4.4.

- Das Low-ANOVA GTM hat ein sehr restriktives Modell. Es ist realistisch anzunehmen, dass auf Datensätzen, die diesem Modell entsprechen, die Principal Component Analysis im Initialisierungsschritt die optimalen Raumrichtungen $\{\mathbf{v}_d\}_{d=1}^D$ identifiziert.

6.6.3 Beta-Berechnung

Auch im β-Berechnungsschritt muss kein L-dimensionales Integral bestimmt werden. Bei den folgenden Umformungen nutzen wir wie gewohnt die Darstellung $R(\mathbf{t}, \mathbf{x}) = \prod_{l=1}^L R_l(\mathbf{t}, x_l)$ mit $\int_{[0,1]} R_l(\mathbf{t}, x) dx_l = 1$.

$$
\begin{aligned}
\beta^{-1} &:= \frac{1}{D}\int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x})\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|^2 d\mathbf{x}d\mu(\mathbf{t}) \\
&= \frac{1}{D}\int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x})\left\|\sum_{d=1}^{D}\mathbf{v}_d g_d(x_{p(d)})-\mathbf{t}\right\|^2 d\mathbf{x}d\mu(\mathbf{t}) \\
&= \frac{1}{D}\int_{\mathbb{R}^D}\int_{[0,1]^L} R(\mathbf{t},\mathbf{x})\left(\sum_{d=1}^{D} g_d(x_{p(d)})^2 - 2\sum_{d=1}^{D}\langle\mathbf{v}_d,\mathbf{t}\rangle\, g_d(x_{p(d)}) + \langle\mathbf{t},\mathbf{t}\rangle\right) d\mathbf{x}d\mu(\mathbf{t}) \\
&= \frac{1}{D}\int_{\mathbb{R}^D}\langle\mathbf{t},\mathbf{t}\rangle + \sum_{d=1}^{D}\int_{[0,1]^L}\left(\prod_{l=1}^{L} R_l(\mathbf{t},x_l)\right)\left(g_d(x_{p(d)})^2 - 2\langle\mathbf{v}_d,\mathbf{t}\rangle\, g_d(x_{p(d)})\right) d\mathbf{x}d\mu(\mathbf{t}) \\
&= \frac{1}{D}\int_{\mathbb{R}^D}\langle\mathbf{t},\mathbf{t}\rangle + \sum_{l=1}^{L}\int_{[0,1]} R_l(\mathbf{t},x_l)\sum_{d\in p^{-1}(l)}\left(g_d(x_{p(d)})^2 - 2\langle\mathbf{v}_d,\mathbf{t}\rangle\, g_d(x_{p(d)})\right) dx_l d\mu(\mathbf{t})
\end{aligned}
$$

Die Umsetzung der β-Berechnung benötigt $\mathcal{O}(N \cdot D \cdot K)$.

6.7 Laufzeit

Wie in Unterabschnitt 6.6.1 beschrieben wurde, nutzt die Low-ANOVA GTM-Implementierung die Dünnbesetztheit der $M \times M$-Matrizen $\boldsymbol{A}_d$ nicht aus. Dies ist vertretbar, da die Anzahl der Freiheitsgrade M nicht mit der Latent-Space-Dimension L wächst. Es muss jedoch erwähnt werden, dass hier Spielraum für Verbesserungen ist.

Bei der folgenden Laufzeitabschätzung berücksichtigen wir den Vektorenupdate-Schritt nicht, da sich dieser in den meisten Experimenten als nicht wirksam erwiesen hat. Das Low-ANOVA GTM benötigt für eine Iteration aus Richtungs-*Responsibilities*-Berechnung, M-Schritt und β-Berechnung eine Laufzeit von

$$\mathcal{O}(L \cdot M^3 + N \cdot K \cdot D).$$

Diese Laufzeitabschätzung gilt in jeder Größe, das heißt es existieren keine Konstanten, die exponentiell mit der Dimension wachsen. Es ist bemerkenswert, dass die Laufzeit linear in der Latent-Space-Dimension L ist. Das Low-ANOVA Modell ist interessant, da seine Mächtigkeit zwischen der Hauptkomponentenanalyse und dem klassischen GTM liegt.

7 p-Gauß-Kern GTM

Bevor wir das p-Gauß-Kern GTM beschreiben, zeigen wir zunächst, warum die euklidische Norm und damit auch die Gauß-Kerne in hochdimensionalen Räumen an Aussagekraft verlieren. Dies ist relevant für das GTM, da wir mit gaußschem Rauschen aus dem L-dimensionalen Bild $\boldsymbol{y}([0,1]^L)$ eine D-dimensionale Wahrscheinlichkeitsverteilung erzeugen.

7.1 Effekte in hochdimensionalen Räumen

In hochdimensionalen Räumen treten einige unintuitive Effekte auf, die bei der Arbeit mit hochdimensionalen Daten berücksichtigt werden müssen. In diesem Abschnitt wollen wir einen kurzen Überblick geben. Eine ausführliche Darstellung findet sich in [Ver02].

7.1.1 Das Volumen der Hyperkugel

Das Volumen einer d-dimensionalen Hyperkugel mit Radius r ist durch den Ausdruck

$$V(d) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)} r^d$$

gegeben. Abbildung 7.1 stellt die Volumina für $r = 1$ in den ersten 30 Dimensionen dar. Wir erkennen, dass sie in niedrigen Dimensionen ansteigen, für größere d jedoch gegen 0 abfallen. Wenn sich die Hyperkugel in einem Hyperwürfel befindet, bedeutet dies, dass sich die Masse in den Ecken konzentriert. Eine intiuitive Erklärung ist, dass die Anzahl der Ecken exponentiell mit der Dimension ansteigt.

Die Masse im Inneren einer Kugel konzentriert sich in höheren Dimensionen an ihrer Oberfläche. Das Verhältnis der Volumina von zwei Hyperkugeln mit Radius 0.9 und 1 fällt mit steigender Dimension gegen 0, siehe Abbildung 7.2. In hohen Dimensionen befindet sich die Masse der größeren Kugel fast ausschließlich in der Kugelschale

$$\{\mathbf{x} \mid 0.9 \leq \|x\|_2 \leq 1\}\,. \tag{7.1}$$

Diese überraschenden Effekte demonstrieren, dass sich die Anschauung in niederdimensionalen Räumen nicht auf hochdimensionale Räume übertragen lässt. Die Beschreibung der Kugelschale (7.1) macht deutlich, dass die euklidische Norm in Zusammenhang mit diesen Beobachtungen steht. Im folgenden Unterabschnitt werden wir dies genauer untersuchen.

7.1.2 Concentration-of-Measure

Der Concentration-of-Measure-Effekt führt zu einer Konzentration der euklidischen Norm in hochdimensionalen Räumen. Die Norm eines Zufallsvektors beschreibt eine Zufallsvariable, de-

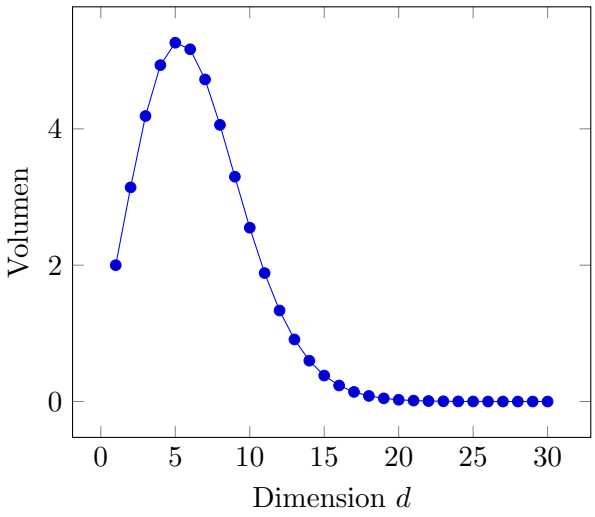

Abb. 7.1: Volumen der d-dimensionalen Hyperkugel mit Radius 1.

ren Erwartungswert mit der Dimension steigt, während die Varianz annähernd konstant bleibt. Der Satz 7.1 aus [Fra07] beschreibt dieses Verhalten.

Satz 7.1 (Demartines)**.** *Sei* $\mathbf{X} \in \mathbb{R}^d$ *ein Zufallsvektor mit unabhängig identisch verteilten Komponenten* $X_i \sim \mathcal{F}$*. Dann gilt*

$$\mathbb{E}\left(\|\mathbf{X}\|_2\right) = \sqrt{\alpha \cdot d - \beta} + \mathcal{O}(\frac{1}{d})$$

und

$$\sigma^2(\|\mathbf{X}\|_2) = \beta + \mathcal{O}(d^{-\frac{1}{2}}),$$

wobei α *und* β *Konstanten sind, die von der Verteilung* $\mathcal{F}$ *aber nicht von der Dimension abhängen.*

Da das Verhältnis von Varianz und Erwartungswert von $\|\mathbf{X}\|_2$ mit steigender Dimension immer kleiner wird, gilt dies auch für den relativen Fehler, den wir machen, wenn wir statt $\mathbf{X}$ den Erwartungswert $\mathbb{E}(\mathbf{X})$ verwenden. Dies bedeutet, dass sich die Datenpunkte in hohen Dimensionen auf der Kugeloberfläche mit Radius $\mathbb{E}(\|\mathbf{X}\|_2)$ konzentrieren. Da die Differenz zwischen zwei Zufallsvektoren wieder ein Zufallsvektor ist, erscheinen auch die Abstände zwischen allen Datenpunkten identisch. Dies ist problematisch für Verfahren, die wie die Nächste-Nachbarn-Suche auf Abständen basieren.

7.1.3 Lokalität von Gauß-Kernen

In [FWV05] wird die Lokalität der Gauß-Kerne

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\|\mathbf{x} - \mathbf{y}\|^2 / 2\sigma^2\right)$$

in hohen Dimensionen untersucht. Sie ist wichtig für die

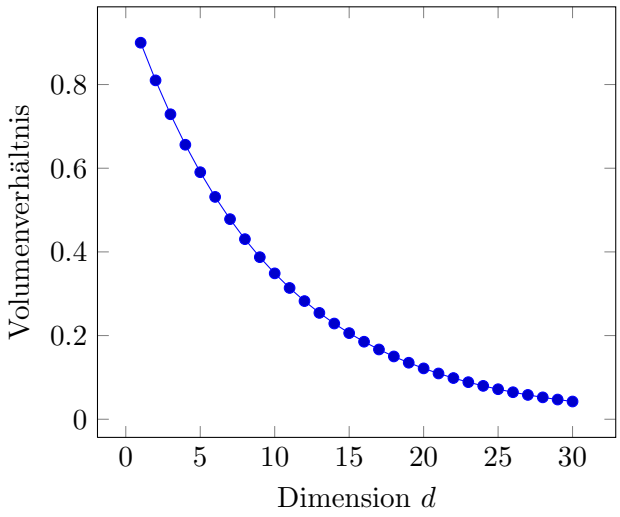

Abb. 7.2: Volumenverhältnis von zwei d-dimensionalen Hyperkugeln mit Radius 0.9 und 1.

- Interpretierbarkeit der Kerne als Ähnlichkeitsmaß und
- die numerische Stabilität von Rechnungen.

Wenn wir in hohen Dimensionen die Abstandsverteilung von Gauß-verteilten Punkten mit dem Abfallverhalten des Kerns vergleichen, liegt ein Großteil der Punkte in einem Bereich, die der Kern nicht unterscheiden kann. Dieses Prinzip wird in Abbildung 7.3 deutlich. In Dimension 1 liegen Abstandsverteilung der Gauß-verteilten Punkte und der Bereich des stärksten Abfalls des Kerns übereinander. Mit steigender Dimension wird diese Beziehung gestört. In Dimension 10 wird ein Großteil der Gauß-verteilten Punkte vom Kern als „unähnlich“ zum Ursprung eingestuft, obwohl sie mit der Dichte $k(0, \cdot)$ generiert wurden.

Für diese Degenerierung ist der Concentration-of-Measure-Effekt verantwortlich: In Dimension 2 liegt ein Großteil der Datenpunkte im Kreis

$$\{\mathbf{x} \mid \|\mathbf{x}\|_2 \leq 3\},$$

während sich in Dimension 10 die Masse in der Kugelschale

$$\{\mathbf{x} \mid 1 \leq \|\mathbf{x}\|_2 \leq 5\}$$

konzentriert. Somit haben wir experimentell die Aussage des Satzes 7.1 nachvollzogen.

7.2 p-Gauß-Kerne und p-Minkowki-Norm

In [FWV05] wird beschrieben, dass Modifikationen des Parameters σ^2 nicht ausreichend sind, um die Lokalität der Gauß-Kerne in hochdimensionalen Räumen herzustellen. Neben der Geschwindigkeit, mit der der Kern abfällt, muss kontrolliert werden, ab welcher Distanz das Ab-

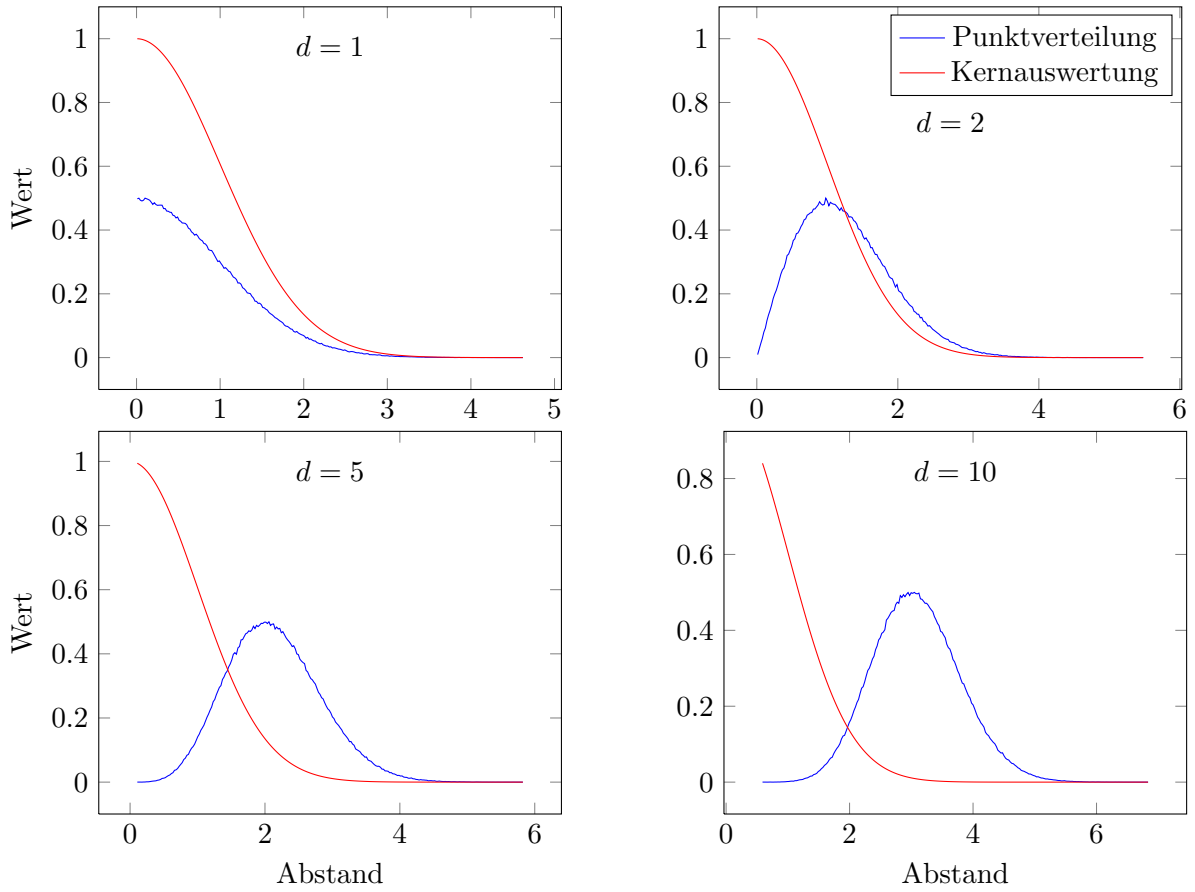

Abb. 7.3: Abfallverhalten des Gauß-Kerns mit $\sigma = 1$ und nicht-normierte Abstandverteilung von 10^6 Gauß-verteilten Punkten in den Dimensionen 1, 2, 5 und 10.

fallverhalten einsetzt. Hierfür wird der p-Gauß-Kern

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-d(\mathbf{x}, \mathbf{y})^p / \sigma^p\right)$$

vorgeschlagen, wobei σ und p Parameter sind, die an die Dimensionalität des Problems angepasst werden müssen.

Betrachten wir nun die unregularisierte M-Schritt-Minimierung aus Definition 3.14

$$\arg\min_{\boldsymbol{y}} \frac{\beta}{2} \int_{\mathbb{R}^D} \int_{[0,1]^L} \psi^{(s)}(\mathbf{t}, \mathbf{x}) \underbrace{\|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|^2}_{\equiv d(\boldsymbol{y}(\mathbf{x}), \mathbf{t})^p} d\mathbf{x} d\mu(\mathbf{t}).$$

Die Minimierung ist nur dann für jede der $\{y_d\}_{d=1}^D$ Komponentenfunktionen unabhängig voneinander durchführbar, wenn sich $d(\boldsymbol{y}(\mathbf{x}), \mathbf{t})^p$ als Summe über die Datenraumdimensionen darstellen lässt. Im Fall der euklidischen Norm $\|\mathbf{x}\|_2^2 = \sum_{d=1}^D x_d^2$ ist dies der Fall. Dass keine Mischterme über die Datenraumdimensionen entstehen, ist eine wichtige Bedingung dafür, dass die Laufzeit des GTM linear in D bleibt.

Dies bedeutet, dass wir für einen p-Gauß-Kern mit $p \neq 2$ die passende Metrik $d(\mathbf{x}, \mathbf{y})$ wählen müssen. Die Definition der Metrik über die p-Minkowski-Norm ist wegen

$$d(\mathbf{x}, \mathbf{y})^p := \|\mathbf{x} - \mathbf{y}\|_p^p = \sum_{d=1}^{D} |x_d|^p.$$

geeignet. Eine p'-Minkowski-Norm

$$\|\mathbf{x} - \mathbf{y}\|_{p'}^p = \left(\sqrt[p']{\sum_{d=1}^{D} |x_d|^{p'}} \right)^p = \left| \sum_{d=1}^{D} |x_d|^{p'} \right|^{p/p'}$$

führt zu einem gebrochenen Exponenten, wenn $p' \nmid p$ gilt. Auch ein ganzzahliges $\frac{p}{p'} \neq 1$ führt zu Mischtermen der einzelnen Koordinatenrichtungen und damit der Koordinatenfunktionen y_d in der M-Schritt-Minimierung. Dies muss vermieden werden, was die Wahl eines einheitlichen p für die p-Gauß-Kerne und die p-Minkowski-Norm begründet.

In [FWV05] wird gezeigt, dass die p-Gauß-Kerne mit der euklidischen Norm in hohen Dimensionen Lokalität herstellen können. Dass dies auch für die p-Minkowski-Norm gilt, weisen wir durch ein einfaches Experiment nach. In Abbildung 7.4 sind die p-Gauß-Kerne mit der Metrik $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p$ in 10 Dimensionen und die dazugehörigen Punktverteilungen dargestellt. Wir erkennen, dass sich die Masse der Punktverteilung und der Bereich des Kernabfalls für steigende p aufeinander zu bewegen.

7.3 Konstruktion

Im Folgenden werden wir ein p-Gauß-Kern GTM konstruieren. Wir werden das Problem so formulieren, dass für $p = 2$ die bisherige Formulierung entsteht. Wegen

$$\int_{-\infty}^{+\infty} e^{-\frac{x^2}{2}} dx = \sqrt{2\pi} < \infty$$

gilt für $p \geq 2$

$$c_p := \int_{-\infty}^{+\infty} e^{-\frac{|x|^p}{2}} dx < \infty. \tag{7.2}$$

Wegen

$$\int_{-\infty}^{+\infty} e^{-\frac{\beta}{2}|x|^p} dx = \beta^{-\frac{1}{p}} \int_{-\infty}^{+\infty} e^{-\frac{|x|^p}{2}} dx = \beta^{-\frac{1}{p}} c_p$$

folgt für das D-dimensionale Integral

$$\frac{\beta^{\frac{D}{p}}}{c_p^D} \int_{\mathbb{R}^D} e^{-\frac{\beta}{2}\|\mathbf{x}\|_p^p} d\mathbf{x} = 1. \tag{7.3}$$

In Zeile (7.3) steht die p-Gauß-Kern Wahrscheinlichkeitsdichte, mit der wir das Bild $\boldsymbol{y}([0,1]^L)$ stören werden, um eine Wahrscheinlichkeitsverteilung im D-dimensionalen Datenraum zu er-

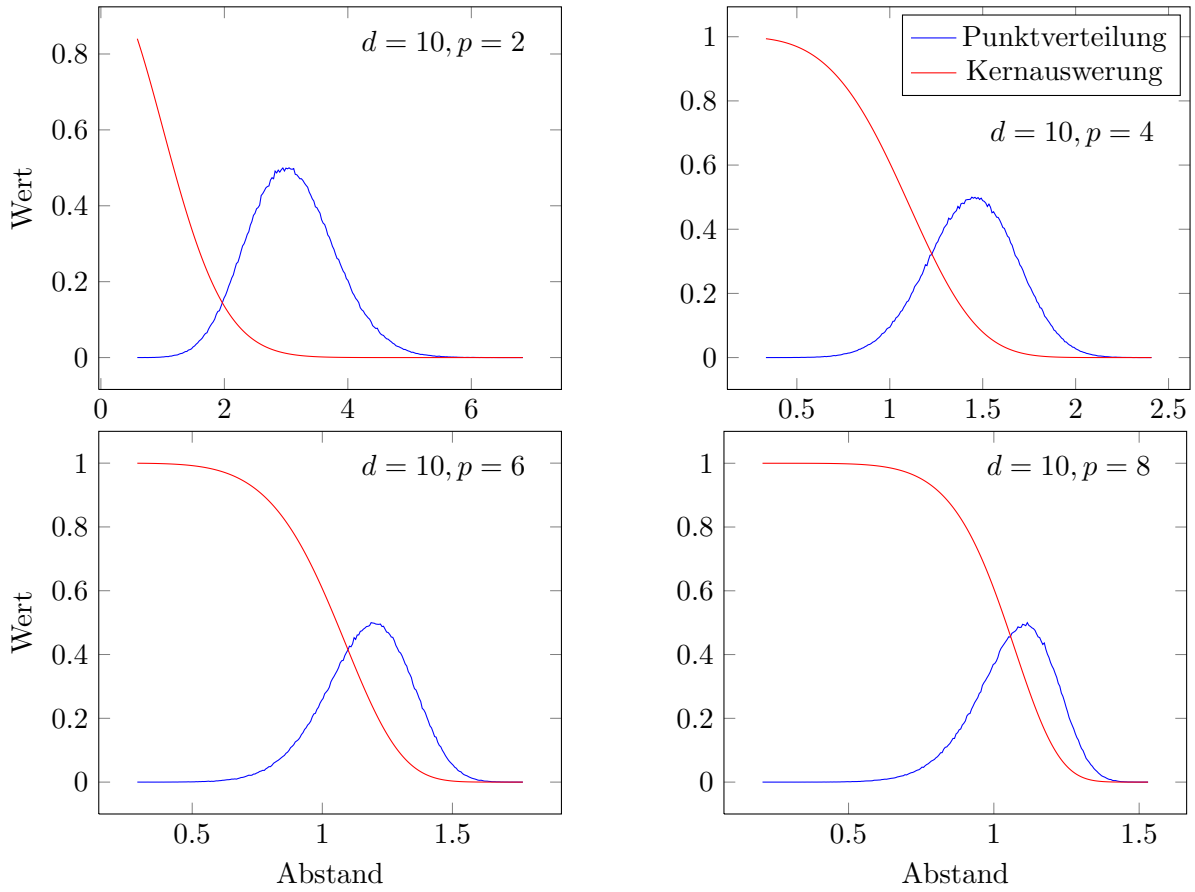

Abb. 7.4: Abfallverhalten der p-Gauß-Kerne mit der Metrik $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_p$ und $\sigma = 1$ in 10 Dimensionen und nicht-normierte Abstandverteilung von 10^6 entsprechend verteilten Punkten mit $p = 2, 4, 6, 8$.

halten. Für $p = 2$ entspricht sie der üblichen Gauß-Verteilung mit inverser Varianz β. Für größere p ist β nicht mehr exakt gleich der inversen Varianz, wird der Einfachheit halber dennoch so bezeichnet. Hiermit können wir das p-Gauß-Kern GTM-Funktional definieren.

Definition 7.2 (p-Gauß-Kern GTM-Funktional)**.** Wenn wir das GTM-Modell mit der Wahrscheinlichkeitsdichte aus Zeile (7.3) kombinieren, ergibt dies das p-Gauß-Kern GTM-Funktional

$$\begin{aligned}\mathcal{G}_p(\boldsymbol{y}, \beta) &:= -\int_{\mathbb{R}^D} \log \frac{\beta^{\frac{D}{p}}}{c_p^D} \int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\mathbf{t} - \boldsymbol{y}(\mathbf{x})\|_p^p\right) d\mathbf{x} d\mu(\mathbf{t}) + \lambda \cdot S(\boldsymbol{y}) \\ &= -\int_{\mathbb{R}^D} \log \int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\mathbf{t} - \boldsymbol{y}(\mathbf{x})\|_p^p\right) d\mathbf{x} d\mu(\mathbf{t}) \\ &\quad -\frac{D}{p} \log \frac{\beta}{c_p^p} + \lambda \cdot S(\boldsymbol{y}).\end{aligned}$$

Satz 7.3. *Das p-Gauß-Kern GTM-Funktional aus Definition 7.2 lässt sich umformen zu*

$$\begin{aligned}\mathcal{G}_p(\boldsymbol{y},\beta) &= \int_{\mathbb{R}^D}\int_{[0,1]^L} \frac{e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|_p^p}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-\mathbf{t}\|_p^p} d\mathbf{x}'} \log \frac{e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|_p^p}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-\mathbf{t}\|_p^p} d\mathbf{x}'} d\mathbf{x} d\mu(\mathbf{t}) \\ &+\frac{\beta}{2}\int_{\mathbb{R}^D}\int_{[0,1]^L} \frac{e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|_p^p}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-\mathbf{t}\|_p^p} d\mathbf{x}'} \|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|_p^p d\mathbf{x} d\mu(\mathbf{t}) \\ &-\frac{D}{p}\log\frac{\beta}{c_p^p} + \lambda\cdot S(\boldsymbol{y}).\end{aligned}$$

Beweis. Weitestgehend analog zum Beweis von Satz 3.9. □

Dieses Funktional minimieren wir wie in Abschnitt 3.4. Hierzu benötigen wir die Definition 7.4.

Definition 7.4 (p-Gauß-Kern Responsibilities)**.** Wir definieren eine Funktion $R : \mathbb{R}^D \times [0,1]^L \to (0,\infty)$, die jedem Datenpunkt $\mathbf{t} \in \mathbb{R}^D$ seine Posterior-Verteilung im Latent-Space zuordnet.

$$R_p(\mathbf{t},\mathbf{x}) := \frac{\exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|_p^p\right)}{\int_{[0,1]^L}\exp\left(-\frac{\beta}{2}\|\boldsymbol{y}(\mathbf{x}')-\mathbf{t}\|_p^p\right) d\mathbf{x}'} \tag{7.4}$$

Bis hierhin war der Übergang vom GTM zum p-Gauß-Kern GTM problemlos. Die Minimierung in Richtung von $\boldsymbol{y}$ führt jedoch nicht mehr zu einem linearen Gleichungssystem, wie der folgende Abschnitt zeigen wird.

7.4 M-Schritt

7.4.1 Funktionenupdates

Unsere M-Schritt-Minimierung hat die Form

$$\arg\min_{\boldsymbol{y}} \frac{\beta}{2}\int_{\mathbb{R}^D}\int_{[0,1]^L} R_p(\mathbf{t},\mathbf{x})\|\boldsymbol{y}(\mathbf{x})-\mathbf{t}\|_p^p d\mathbf{x} d\mu(\mathbf{t}) + \lambda\cdot S(\boldsymbol{y}).$$

Da unsere $\|\cdot\|_p^p$-Norm wie gefordert additiv über die Datenraumdimensionen ist, können wir die einzelnen Komponentenfunktionen von $\boldsymbol{y}$ unabhängig voneinander minimieren. Wir müssen für alle $d = 1,\ldots,D$

$$\arg\min_{y_d} \frac{\beta}{2}\int_{\mathbb{R}^D}\int_{[0,1]^L} R_p(\mathbf{t},\mathbf{x})\,|y_d(\mathbf{x})-(\mathbf{t})_d|^p\, d\mathbf{x} d\mu(\mathbf{t}) + \lambda\cdot S_d(y_d) \tag{7.5}$$

bestimmen. Für $p = 2$ entsteht ein lineares Gleichungssystem, was bereits hinreichend diskutiert wurde. Im Fall von $p > 2$ handelt es sich nicht mehr um eine quadratische Minimierung, aber noch um eine konvexe. Das Problem bleibt auch nach Diskretisierung der Integrale konvex,

wenn ausschließlich positive Quadraturgewichte verwendet werden. Wir diskretisieren

$$y_d(x) = \sum_{j=1}^{M} (\mathbf{w}_d)_j \, \phi_j(x)$$

mit dem Koeffizientenvektor $\mathbf{w}_d \in \mathbb{R}^M$, und nehmen an, dass sich der Regularisierungsterm in der Form

$$S_d(y_d) = \mathbf{w}_d^T \boldsymbol{C}_d \mathbf{w}_d$$

darstellen lässt. Zur Minimierung von (7.5) verwenden wir ein gradientenbasiertes Minimierungsverfahren wie die Broyden–Fletcher–Goldfarb–Shanno-Methode (BFGS) aus der GNU Scientific Library, siehe [GSL]. Der Gradient $\mathbf{g}_d \in \mathbb{R}^M$ hat dann zu gegebenem $\mathbf{w}_d$-Vektor die Gestalt

$$\begin{aligned} (\mathbf{g}_d)_k &= \frac{p\beta}{2} \int_{\mathbb{R}^D} \int_{[0,1]^L} R_p(\mathbf{t}, \mathbf{x}) \operatorname{sgn}(y_d(\mathbf{x}) - (\mathbf{t})_d) \, |y_d(\mathbf{x}) - (\mathbf{t})_d|^{p-1} \, \phi_k(\mathbf{x}) d\mathbf{x} d\mu(\mathbf{t}) \\ &\quad + 2 \left(\boldsymbol{C}_d \mathbf{w}_d\right)_k . \end{aligned}$$

Das BFGS-Verfahren balanciert die Genauigkeit der Schrittweitenbestimmung und die Häufigkeit der Neuberechnung des Gradienten. Wir brechen die Minimierung ab, wenn die Norm des Gradienten eine gewisse Genauigkeit unterschreitet oder die Anzahl der Iterationen eine feste Obergrenze überschreitet.

7.4.2 Beta-Berechnung

Das neue β lässt sich exakt bestimmen durch

$$\frac{1}{\beta} := \frac{p}{2D} \int_{\mathbb{R}^D} \int_{[0,1]^L} R_p(\mathbf{t}, \mathbf{x}) \|\boldsymbol{y}(\mathbf{x}) - \mathbf{t}\|_p^p d\mathbf{x} d\mu(\mathbf{t}).$$

8 Numerische Experimente

In Abschnitt 8.1 testen wir die beschriebenen GTM-Varianten an synthetischen Beispieldatensätzen. Im darauffolgenden Abschnitt 8.2 konstruieren wir einen GTM-Klassifikator und führen Experimente mit hochdimensionalem Rauschen auf Daten mit niedriger intrinsischer Dimension durch. Wir untersuchen die Auswirkungen von einer zu niedrigen oder zu hohen Latent-Space-Dimension in Abschnitt 8.3. In dem darauffolgenden Abschnitt 8.4 vergleichen wir das p-Gauß-Kern GTM mit verschiedenen p auf demselben Datensatz. Das Kapitel schließen wir mit Ergebnissen auf bekannten Anwendungsbeispielen in Abschnitt 8.5 ab.

8.1 Synthetische Beispieldatensätze

8.1.1 Open Box

Die Open Box wird in [LV07] als Benchmark verwendet und ist in Abbildung 8.1 dargestellt. Sie besteht aus 316 dreidimensionalen Punkten, die auf einer zweidimensionalen Mannigfaltigkeit angeordnet sind. Diese ist an den Rändern unglatt und im Gegensatz zu einem Würfel oder einer geschlossenen Box nicht kompakt. In der Darstellung sind die Datenpunkte in Abhängigkeit von ihrer z-Koordinate eingefärbt. Dies ist lediglich ein Visualisierungsaspekt, der bei einer zweidimensionalen Einbettung der Open Box Orientierung bietet.

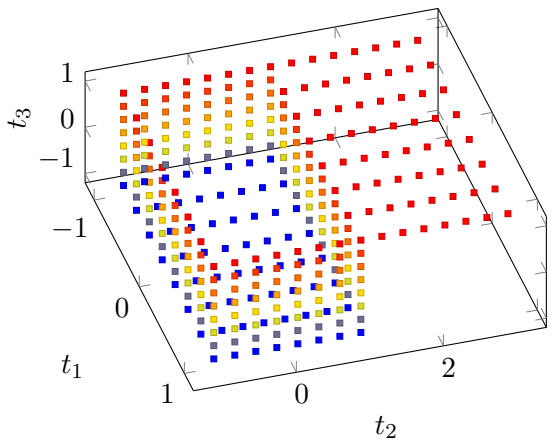

Abb. 8.1: 316 Datenpunkte auf einer Open Box

Zur Dimensionsreduktion benötigen wir ein nichtlineares Verfahren wie das GTM. Wir verwenden einen zweidimensionalen Latent-Space, eine Dünngitterbasis mit Level 6, regularisieren die Komponentenfunktionen mit $|\cdot|_{H^1}$ und $\lambda = 0.008$, und initialisieren β mit 1 und $\boldsymbol{y}$ mit

der PCA auf den Daten. In unseren Experimenten verwenden statt der H^1-Norm die Seminorm, da der L^2-Anteil nicht translationsinvariant ist. In der linken Spalte von Abbildung 8.2 ist $\boldsymbol{y}([0,1]^2)$ zum Initialisierungszeitpunkt und nach 30 Iterationsschritten dargestellt. In der rechten Spalte betten wir die Datenpunkte mit dem Erwartungswert ihrer Posterior-Verteilung in den Latent-Space ein, siehe Unterabschnitt 2.3.4. Wir sehen, dass das GTM die Topologie der Box korrekt erkennt, was die PCA nicht leistet.

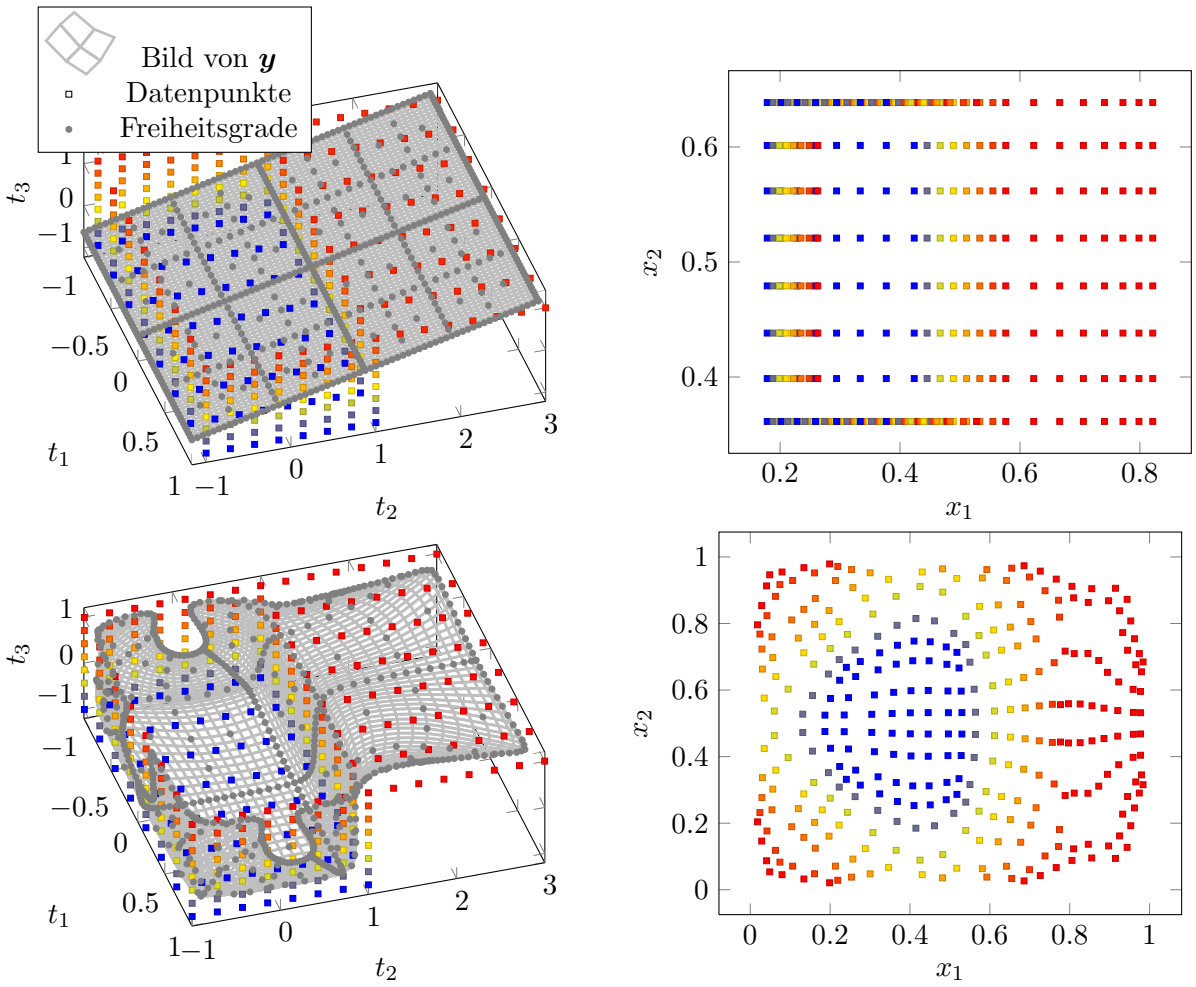

Abb. 8.2: In der linken Spalte ist das Bild $\boldsymbol{y}([0,1]^2)$ dargestellt, in der rechten die dazugehörige Latent-Space-Projektion. Die obere Zeile zeigt das GTM-Modell zum Initialisierungszeitpunkt, die untere nach 30 Iterationen.

Nun möchten wir die Wirkung des Regularisierungsterms untersuchen. Das GTM-Funktional besteht aus der Kreuzentropie und einem λ-gewichteten Regularisierungsterm, siehe Definition 3.6. Der Regularisierungsparameter λ muss so groß gewählt werden, dass kein Overfitting auftritt, wobei zu beachten ist, dass mit einer stärkeren Regularisierung die Approximation der Daten schlechter wird. Diesen Zusammenhang wollen wir am Beispiel der Open Box nachvollziehen. Die Kreuzentropie wird kleiner, je besser die Datenapproximation ist, womit sie sich

spiegelbildlich zur inversen Varianz β verhält. Abbildung 8.3 zeigt, dass sie schneller abfällt und somit das GTM-Modell die Daten besser approximiert, je weniger regularisiert wird.

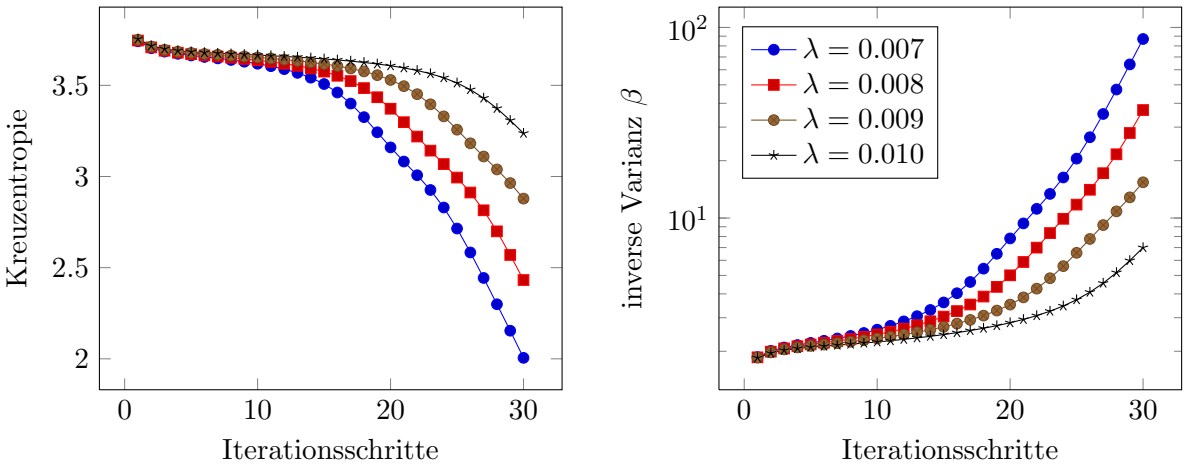

Abb. 8.3: Entwicklung der Kreuzentropie und der inversen Varianz beim Open Box-Experiment zu verschieden Regularisierungsparametern λ.

Bisher haben wir mit der H^1-Seminorm regularisiert, die im Gegensatz zur $H^{1,\text{mix}}$-Seminorm keine stetige Einbettbarkeit der Komponentenfunktionen $y_d : [0,1]^L \to \mathbb{R}$ garantiert, siehe Abschnitt 3.3. In Abbildung 8.4 stellen wir zwei mit den genannten Seminormen verhältnismäßig stark regularisierte Mannigfaltigkeiten gegenüber. Wir erkennen, wie die Regularisierung zu einer schlechteren Datenapproximation führt. Es werden jedoch auch Unterschiede zwischen den beiden Seminormen deutlich: Die t_1-Richtung der mix-regularisierten Mannigfaltigkeit ist fast unabhängig von den beiden anderen Richtungen. Offenbar führt eine zu starke mix-Regularisierung zu einem Modell, das dem Low-ANOVA GTM nahe kommt. Das Low-ANOVA Modell werden wir in dem folgenden Unterabschnitt mit einem anderen Datensatz testen.

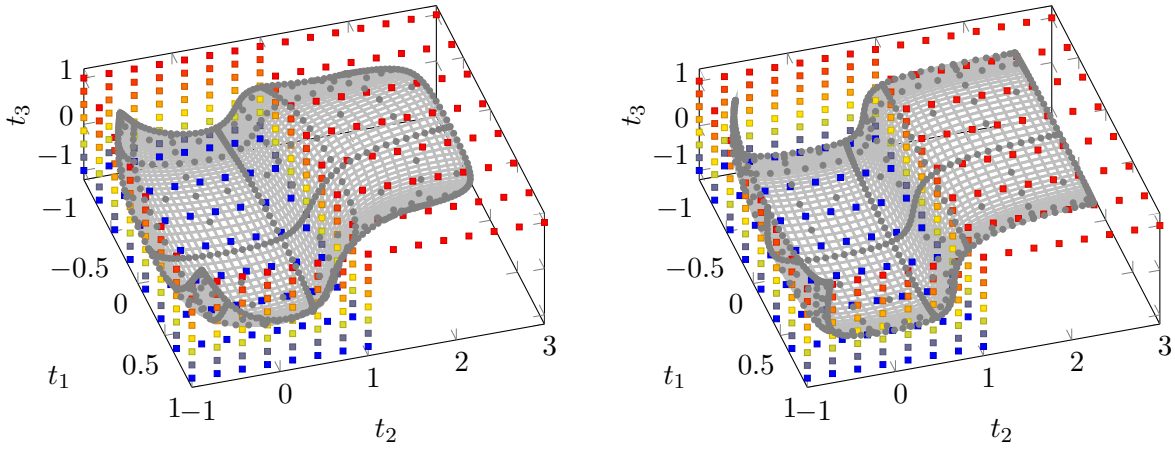

Abb. 8.4: Das linke GTM-Modell wurde mit der H^1-Seminorm und $\lambda = 0.01$ regularisiert, das rechte mit $H^{1,\text{mix}}$ und $\lambda = 0.009$.

8.1.2 Swiss Roll

Swiss Roll ist der englische Ausdruck für Biskuitrolle, also einem Biskuitteig, der dünn mit Füllung bestrichen und aufgerollt wird. In [LV07] wird die Füllung der Swiss Roll als eine Benchmark-Mannigfaltigkeit verwendet, die mit 350 Datenpunkten beschrieben wird. Die Mannigfaltigkeit ist nicht kompakt, im Gegensatz zur Open Box jedoch glatt. Die Herausforderung für ein Verfahren zur Dimensionsreduktion besteht darin, die Swiss Roll so „abzurollen“, dass die Topologie erhalten bleibt und ein bijektives Mapping entsteht.

Die erste Hauptachse der Swiss Roll ist die t_3-Richtung. Diese Richtung ist unabhängig von den Richtungen t_1 und t_2, was die Anwendung des Low-ANOVA GTM nahelegt. Hierzu mappen wir die erste Hauptachse auf die Latent-Space-Richtung x_2, die beiden anderen Achsen auf die Latent-Space-Richtung x_1. Wir wählen eine Diskretisierung auf Level 5 und ein initiales β von 1. Die Abbildung 8.5 zeigt das GTM-Modell nach 65 Iterationen. Wir sehen, dass die Swiss Roll erstaunlich gut erkannt aber nicht ganz korrekt abgerollt wird. Dieses Ergebnis lässt sich auch mit einer Dünngitterdiskretisierung erzielen, entscheidend ist der geringe Startwert für β.

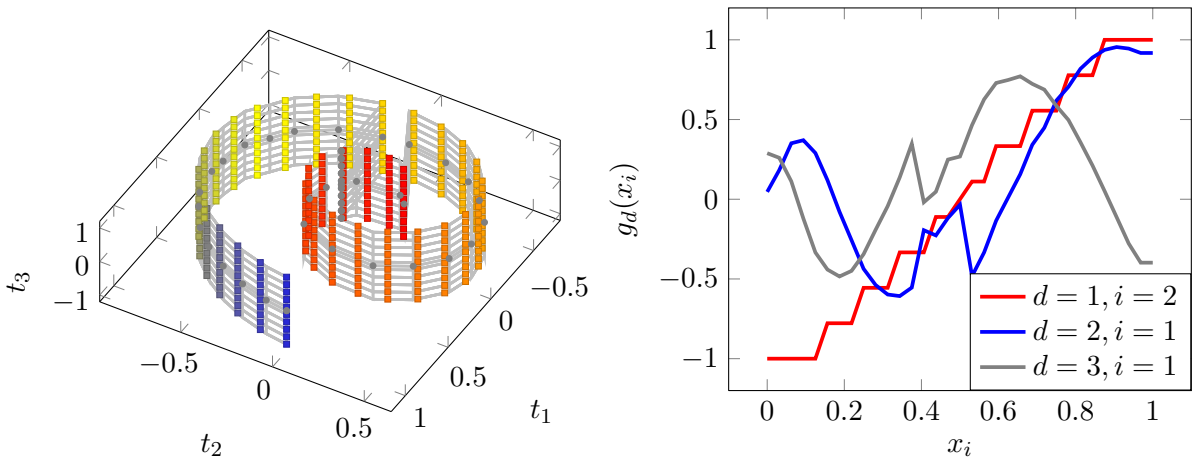

Abb. 8.5: Das linke Bild zeigt das Low-ANOVA GTM-Modell angwendet auf die Swiss Roll, das rechte die drei Komponentenfunktionen $g_d(x)$.

In [LV07] kann das GTM die Swiss Roll nicht einbetten, sondern erzeugt ein mit der PCA vergleichbares Ergebnis. Im Vergleich dazu ist die Einbettung in Abbildung 8.6 links gelungen. Dass die Swiss Roll nicht ganz korrekt abgerollt wird, erkennt man an einer kurzen Unterbrechung im Farbverlauf.

8.1.3 Sinusschwingung

In diesem Experiment wollen wir die Fähigkeiten der verschiedenen GTM-Varianten gegeneinander abgrenzen. Wir generieren 5000 Datenpunkte, die auf einer Sinusschwingung mit abfallender Amplitude und steigender Frequenz liegen, siehe Abbildung 8.7 links. Offenbar ist bei dieser Struktur die t_2-Richtung unabhängig von den übrigen Richtungen. Der rechte Teil der Abbildung zeigt die Datenpunkte des zweiten Experiments, bei denen eine leichte Schwingung

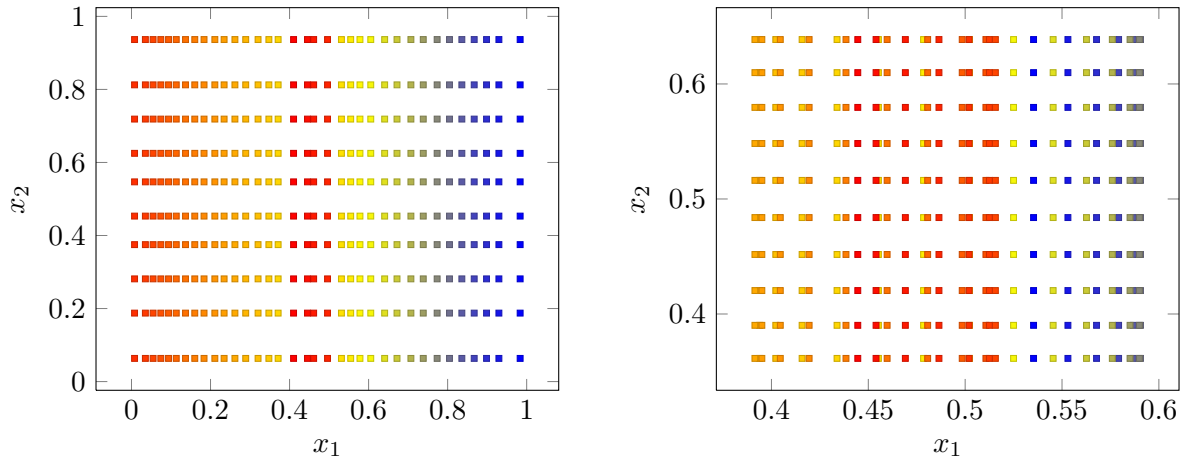

Abb. 8.6: Das linke Bild zeigt die Einbettung der Swiss Roll nach 65 Iterationen, das rechte Bild die Einbettung zum Initialisierungszeitpunkt.

in Abhängigkeit von t_2 hinzukommt.

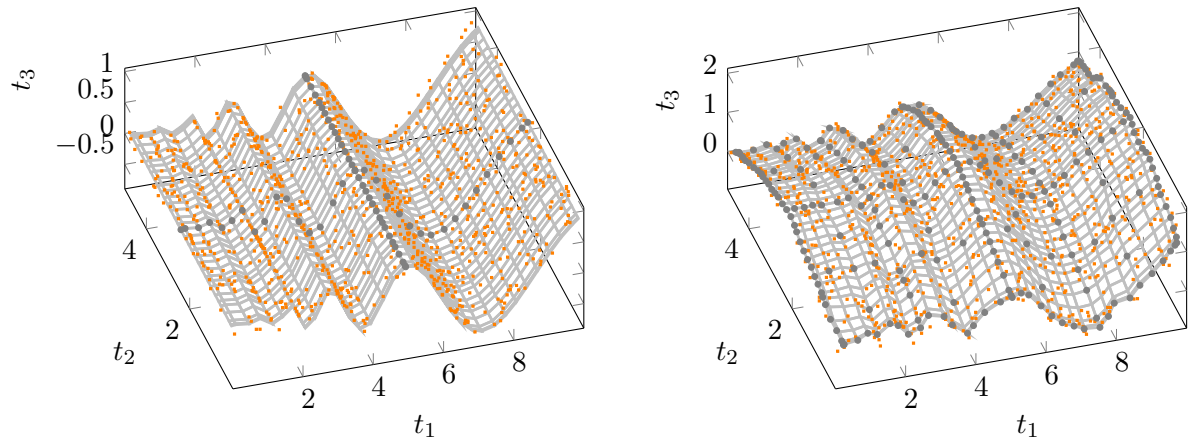

Abb. 8.7: Die Abbildung links zeigt das erste Sinusschwingung-Experiment mit Low-ANOVA GTM, die Abbildung rechts das zweite Experiment mit Sparse GTM-Modell.

Wir wenden das Low-ANOVA GTM, das Sparse GTM und das Vollgitter GTM auf die beiden Beispiele an, wobei wir die Güte der Datenapproximation nach fünf Iterationsschritten ohne Regularisierung messen. Wir testen die Diskretisierungslevel 1 bis 5 der jeweiligen Verfahren. Eine Aufstellung der benötigten Freiheitsgrade findet sich in Tabelle 8.1.

Die Ergebnisse der beiden Experimente sind in Abbildung 8.8 dargestellt. Im ersten Experiment benötigt das Low-ANOVA GTM am wenigsten Freiheitsgrade, weil die t_2-Richtung der Struktur unabhängig von den übrigen Richtungen ist. Diese restriktive Vorgabe ist im zweiten Experiment nicht mehr erfüllt, so dass eine Erhöhung der Freiheitsgrade bei dem Low-ANOVA GTM den Funktionalwert nicht weiter vermindern kann. In beiden Experimenten benötigt die Dünngitterdiskretisierung zur besseren Approximation der Daten weniger Freiheitsgrade als die

Level	Volles Gitter	Dünnes Gitter	Low-ANOVA
1	9	9	3
2	25	21	5
3	81	49	9
4	289	113	17
5	1,089	257	33

Tabelle 8.1: Freiheitsgrade in Abhängigkeit vom Diskretisierungslevel.

Vollgitterdiskretisierung.

Dass eine Punktauswertung in der Dünngitterbasis komplexer ist als auf einem vollen Gitter, fällt beim GTM asymptotisch gesehen nicht ins Gewicht. Dies liegt daran, dass die Lösung des linearen Gleichungssystems die Laufzeitabschätzung des GTM dominiert, und hier ausschließlich die Anzahl der Freiheitsgrade eingeht.

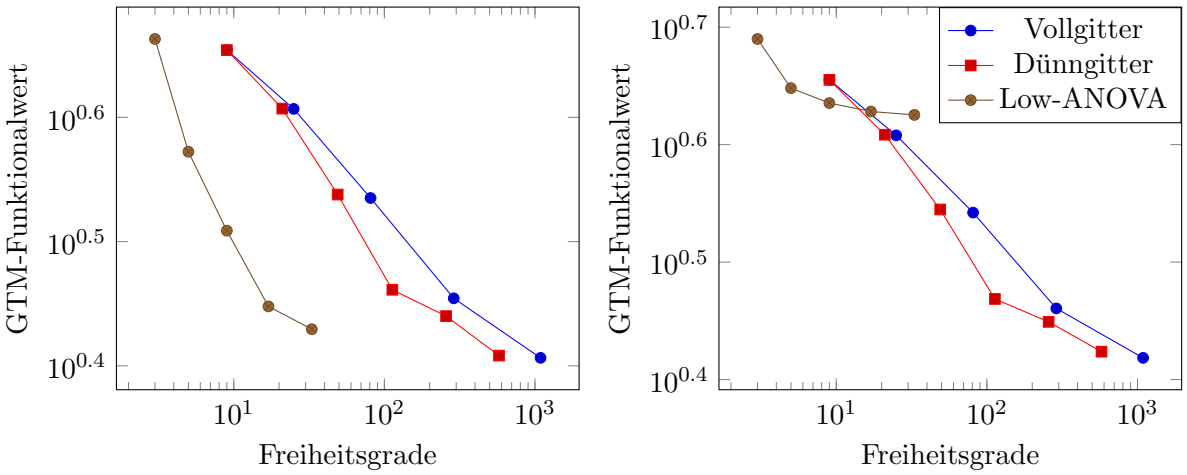

Abb. 8.8: In den Abbildungen ist der GTM-Funktionalwert gegen die Anzahl der benötigten Freiheitsgrade nach 5 Iterationen aufgetragen. Die linke Abbildung beschreibt das erste, die rechte das zweite Sinusschwingung-Experiment.

8.2 Klassifikation

Mit der Dichteschätzung des GTM lässt sich auf einfache Weise ein Klassifikator konstruieren. Es seien N Datenpunkte $\{\mathbf{t}_n\}_{n=1}^N$ mit den Klassenzugehörigkeiten $\{l_n\}_{n=1}^N$ gegeben, wobei

$$l_n \in \{-1, 1\}$$

für $n = 1, \ldots, N$ gilt. Wir fassen die Klassen oder Target-Variablen als $(D+1)$-te Dimension auf, indem wir

$$\mathbf{t}'_n := \begin{pmatrix} (\mathbf{t}_n)_1 \\ \vdots \\ (\mathbf{t}_n)_D \\ l_n \end{pmatrix}$$

setzen. Mit dem GTM können wir auf $\{\mathbf{t}'_n\}_{n=1}^N$ eine $(D+1)$-dimensionale Wahrscheinlichkeitsdichte g lernen. Da bei der Klassifikation von einem funktionalen Zusammenhang

$$l = f(\mathbf{t})$$

ausgegangen wird, können wir annehmen, dass die $(D+1)$-dimensionale Wahrscheinlichkeitsdichte die gleiche intrinsische Dimensionalität hat wie die Daten ohne Target-Variable. Das GTM-Modell gibt nach erfolgter Parameterbestimmung zu einer D-dimensionalen Position $\mathbf{t} = (t_1, \ldots, t_D)^T$ eine eindimensionale Wahrscheinlichkeitsdichte

$$g(l \mid t_1, \ldots, t_D) = \frac{g(t_1, \ldots, t_D, l)}{\int_{\mathbb{R}} g(t_1, \ldots, t_D, l) dl}$$

zurück. Wir bestimmen die zu gegebener Position wahrscheinlichere Klasse durch den Klassifikator

$$f(\mathbf{t}) := \begin{cases} 1, & \text{wenn } g(t_1, t_2, \ldots, t_D, 1) \geq g(t_1, t_2, \ldots, t_D, -1) \\ -1 & \text{sonst.} \end{cases}$$

Die Klassifikation mit dem GTM ist interessant, da die Laufzeit im Wesentlichen durch die intrinsische Dimension der Daten bestimmt wird. Die tatsächliche Datenraumdimension D geht nur linear in die Komplexität ein. Weiterhin ist bemerkenswert, dass das GTM nicht nur gegebene Datenpunkte klassifizieren kann, sondern ebenfalls die Dichte der Datenpunkte modelliert. Somit können zusätzliche *ähnliche* Punkte generiert werden.

8.2.1 Halbschale

Um die Arbeitsweise des GTM als Klassifikator zu illustrieren, verwenden wir ein einfaches Modellproblem mit dreidimensionalen Datenpunkten auf einer Halbschale. Ein Viertel dieser Schale ist der Klasse 1 zugeordnet, der Rest der Klasse -1. Um das GTM als Klassifikator einzusetzen, wird die Klassenzugehörigkeit als vierte Dimension aufgefasst. Für unser GTM-Modell verwenden wir eine Dünngitterdiskretisierung mit Level 4. Dies entspricht 113 Freiheitsgraden für jede der 4 Komponentenfunktionen. Die Dünngitterquadratur erfolgt auf Level 10 mit 3329 Quadraturpunkten. Als Regularisierungsterm wählen wir die H^1-Seminorm mit $\lambda = 0.01$.

Unser Klassifikator erzielt nach vier Schritten auf 500 Trainingsdatenpunkten eine Genauigkeit von 98.8% und auf 500 Testdatenpunkten eine Genauigkeit von 97.4%. In Tabelle 8.2 sind die Ergebnisse nach Klassen aufgeschlüsselt, und Abbildung 8.9 zeigt die Halbschale und ihre zweidimensionale Projektion auf das dünne Quadraturgitter. Es ist bemerkenswert, dass beide Klassen deutlich voneinander getrennt wurden. In Abbildung 8.10 sehen wir, wie sich die anfänglich durch die PCA initialisierte Struktur den Datenpunkten anpasst. Hierbei sind in der linken Spalte die ersten drei Dimensionen dargestellt, wohingegen in der rechten Spalte die ersten beiden und die vierte Dimension zu sehen sind.

		Trainingsdaten		Testdaten	
		1	−1	1	−1
GTM	1	20.6%	0.4%	23.2%	0.6%
	−1	0.8%	78.2%	2.0%	74.2%

Tabelle 8.2: Konfusionsmatrizen des GTM-Klassifikators auf Trainings- und Testdaten.

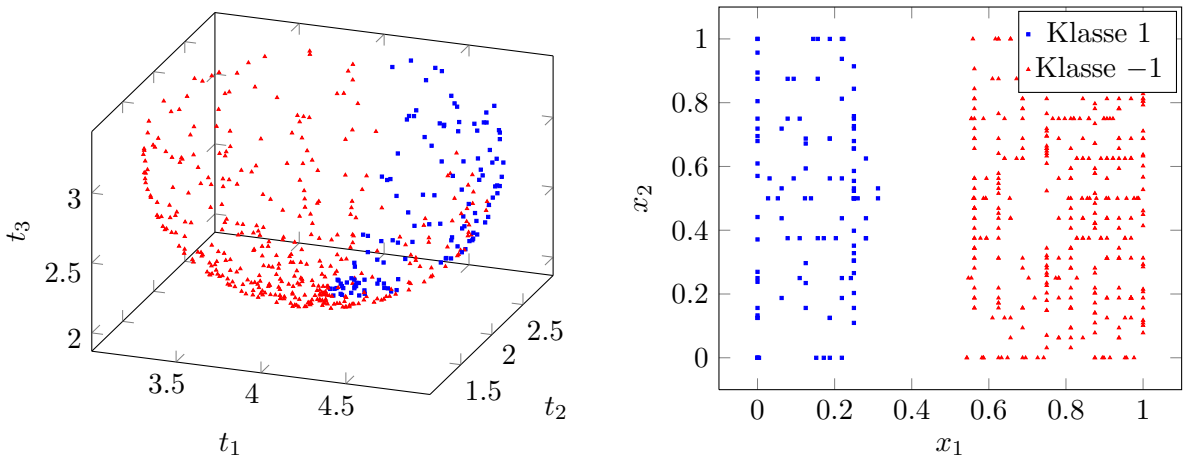

Abb. 8.9: Links ist die Halbschale dargestellt, rechts ihre zweidimensionale Projektion auf das dünne Quadraturgitter nach 4 Iterationsschritten.

8.2.2 Concentration-of-Measure und die intrinsische Dimension

In Unterabschnitt 7.1.2 wurde der Concentration-of-Measure-Effekt beschrieben: Während der Erwartungswert der euklidischen Norm eines Zufallsvektors mit der Dimension ansteigt, bleibt die Varianz der Norm weitestgehend konstant. Wir wollen in diesem Unterabschnitt untersuchen, welche Effekte hochdimensionales Rauschen auf die Klassifikationsrate des GTM hat, und welche Wechselwirkungen mit der intrinsischen Dimension bestehen.

Zunächst führen wir ein einfaches Experiment durch. Hierzu erzeugen wir 10^6 Samples, die gleichverteilt auf den Ecken $\{-1, 1\}^D$ eines D-dimensionalen Hyperwürfels liegen. Ob die Anzahl der Einsen in den Koordinaten gerade oder ungerade ist, entscheidet über die Klassenzuordnung der jeweiligen Ecke. Anschließend stören wir die Koordinaten mit unabhängig identisch

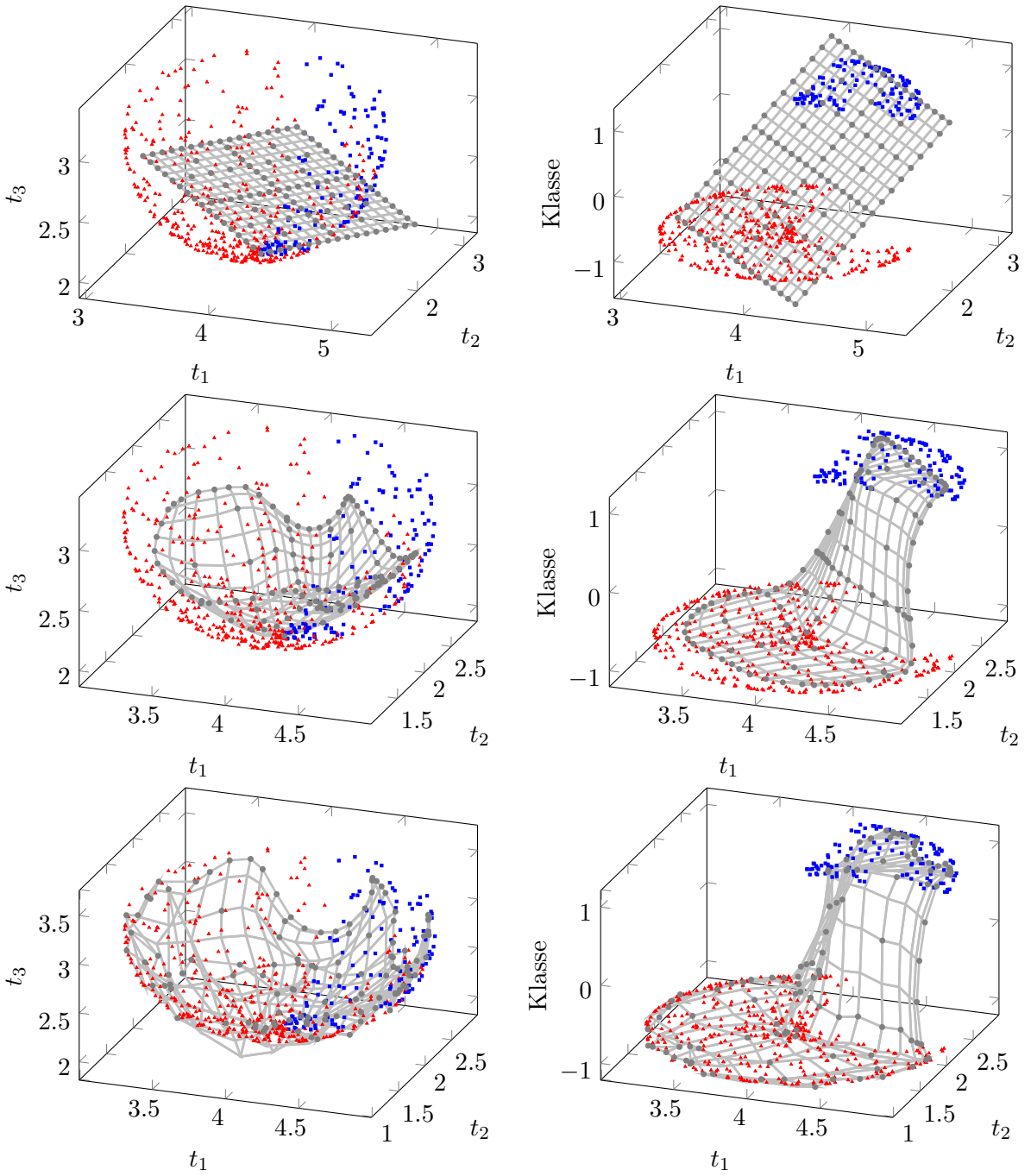

Abb. 8.10: In der linken Spalte sind die ersten drei Dimensionen der Halbschale dargestellt, rechts die ersten beiden Dimensionen und die Target-Variable. Von oben nach unten: GTM-Schritte 0, 1 und 4.

verteiltem $\mathcal{N}(0, \sigma^2)$-Rauschen. Abbildung 8.11 links stellt den dreidimensionalen Fall dar.

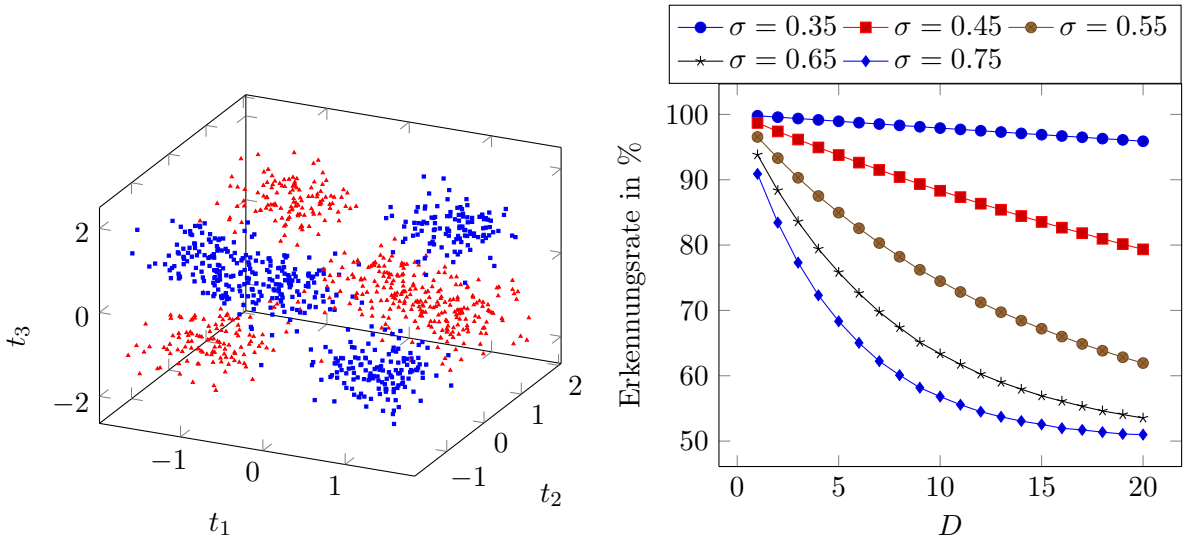

Abb. 8.11: Die linke Abbildung zeigt die Gauß-verrauschten Datenpunkte ($\sigma = 0.35$) auf den Ecken eines dreidimensionalen Hyperwürfels mit Klasseneinteilung, die rechte zeigt die Erkennungsraten auf den Ecken in Abhängigkeit von der Dimension D für verschiedene σ.

Nun reklassifizieren wir die Datenpunkte: Um einen Punkt einer Klasse zuzuordnen, zählen wir, ob die Anzahl positiver Koordinaten gerade oder ungerade ist. Die Ergebnisse für verschiedene Varianzen und Dimensionen sind in Abbildung 8.11 rechts graphisch aufbereitet. Wir erkennen einen Abfall der Erkennungsrate mit steigender Dimension, der umso deutlicher auftritt, je höher die Standardabweichung σ ist. Um falsch klassifiziert zu werden, muss sich ein Datenpunkt mindestens um den euklidischen Abstand 1 verschieben. Dass dies mit steigender Dimension häufiger auftritt, spiegelt den Concentration-of-Measure-Effekt wider.

Der D-dimensionale Hyperwürfel ist nicht intrinsisch niederdimensional, weil die Anzahl der Würfelecken mit 2^D exponentiell wächst. Nun führen wir Experimente mit einem eindimensionalen Halbkreis und einem zweidimensionalen Schachbrett durch. Diese Strukturen betten wir durch die Abbildung

$$P_2^D(\mathbf{x}) = \begin{pmatrix} 2 & 1 \\ 2 & -1 \\ \vdots & \vdots \\ 2 & 1 \\ 2 & -1 \end{pmatrix} \mathbf{x} \tag{8.1}$$

in den D-dimensionalen Raum ein. Anschließend werden die Datenpunkte des Halbkreises normalverteilt mit Varianz $\sigma^2 = 20$ gestört, die des Schachbretts mit $\sigma^2 = 1$. Der dreidimensionale Fall ist in Abbildung 8.12 dargestellt. Um zu verdeutlichen, wie das GTM klassifiziert, sind Halbkreis und Schachbrett aus $D = 2$ mit der Klasseneinteilung als dritter Dimension in Abbildung 8.13 dargestellt.

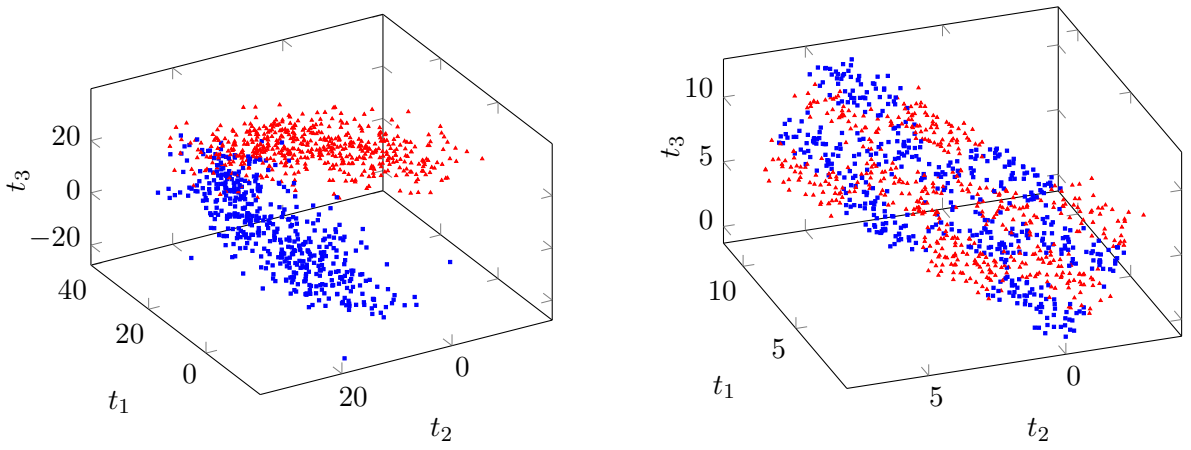

Abb. 8.12: Eindimensionaler Halbkreis und zweidimensionales Schachbrett in drei Dimensionen. Das Rauschen wurde zur besseren Erkennbarkeit vermindert.

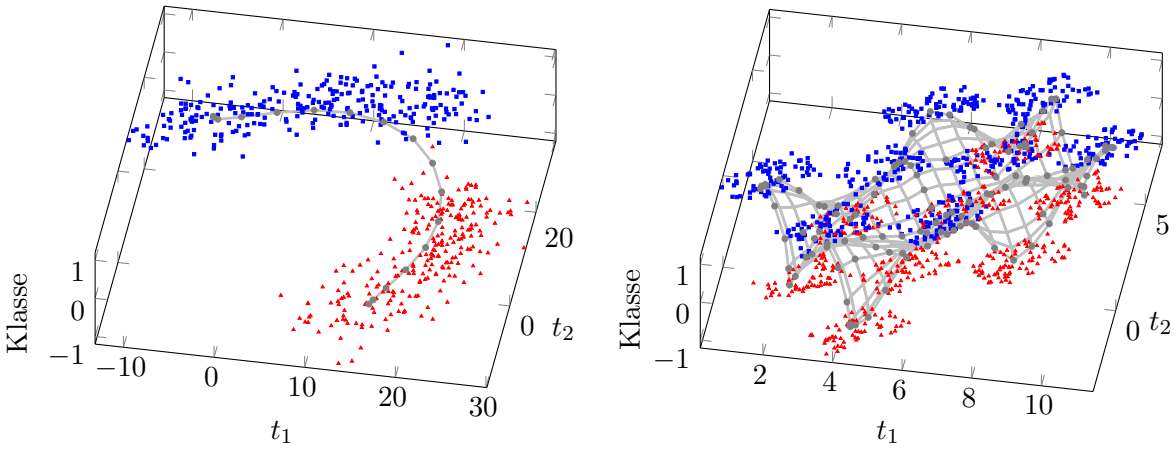

Abb. 8.13: GTM-Modell auf dem zweidimensional eingebetteten Halbkreis (links) und Schachbrett (rechts) mit Klasseneinteilung als dritter Dimension.

Für unseren GTM-Klassifikator wählen wir eine Dünngitterbasis mit Level 4, eine Quadraturregel auf einem regulären Gitter mit Level 5 und ein initiales β von 20. Wir regularisieren mit der H^1-Seminorm und $\lambda = 10^{-4}$ beim Halbkreis und $\lambda = 10^{-3}$ beim Schachbrett. Die Erkennungsraten werden nach 4 Iterationen auf den Trainings- und Testdaten bestimmt. In Abbildung 8.14 links sind die Raten in Abhängigkeit von der Raumdimension graphisch dargestellt.

Wir erkennen, dass mit der Dimension die Erkennungsraten ansteigen. Dies ist überraschend, da das hochdimensionale Rauschen in euklidischer Norm gemessen immer größer wird. Es tritt jedoch noch ein stärkerer, gegenläufiger Effekt auf: Redundanz. Eine zweidimensionale Struktur in einem 20-dimensionalen Raum kann lokal durch eine einfache Projektion

$$p(\mathbf{x}) = (x_i, x_j)$$

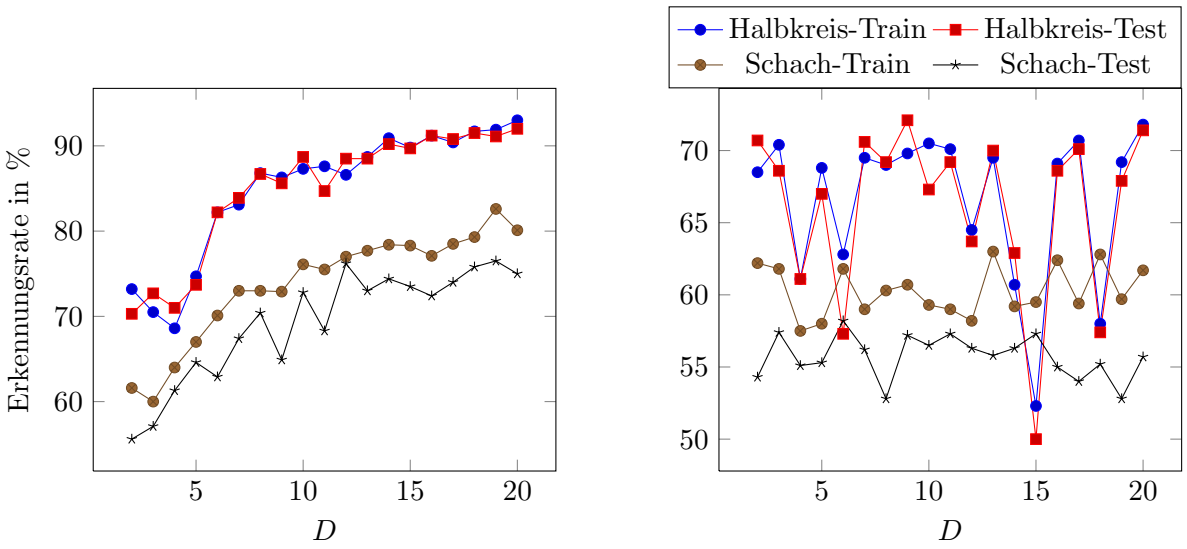

Abb. 8.14: Erkennungsraten auf dem eindimensionalen Halbkreis und dem zweidimensionalen Schachfeld mit steigender Raumdimension. Links wurden die niederdimensionalen Strukturen in alle Raumrichtungen projiziert, rechts nur in die ersten beiden.

dargestellt werden. Das Schachbrett lässt sich mit jeder Kombination aus einer geraden und einer ungeraden Dimension zweidimensional projizieren und erkennen, siehe hierzu auch die Projektion vom 2- in den D-dimensionalen Raum in Formel (8.1). Je größer die Dimensionsanzahl ist, desto mehr zweidimensionale Projektionen enthalten die Information, wie unsere intrinsisch niederdimensionale Struktur beschaffen ist. Diese Redundanz steigert die Erkennungsrate stärker, als sie durch Extremwerte beim Rauschen gestört wird.

Wenn die niederdimensionale Struktur orthogonal auf den Projektionsrichtungen steht, tritt dieser Effekt nicht auf. Verwenden wir beispielsweise die Projektion

$$P_2^D(\mathbf{x}) = \begin{pmatrix} 2 & 1 \\ 2 & -1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix} \mathbf{x},$$

so verbessern sich die Erkennungsraten auf den Testdaten nicht, siehe Abbildung 8.14 rechts.

8.3 Dimensionsschätzung

Wie in Unterabschnitt 2.1.1 erwähnt wird, stellt das GTM keinen internen Mechanismus zur Dimensionsschätzung zur Verfügung. In diesem Abschnitt wollen wir untersuchen, wie sich das GTM-Modell verhält, wenn die intrinsische Dimension der Daten unter- oder überschätzt wird. Hierzu betrachten wir die inverse Varianz β und den GTM-Funktionalwert. Unter gewissen Voraussetzungen sind Rückschlüsse auf die tatsächliche intrinsische Dimension der Daten möglich.

Betrachten wir zunächst 1000 dreidimensionale auf $\{(x, 0, 0) \mid 0 \leq x \leq 10\}$ gleichverteilte Datenpunkte, die $\mathcal{N}(0, 1)$-verrauscht werden. Diese sind in Abbildung 8.15 dargestellt. Das Rauschen ist im Verhältnis zur Länge der Linie so groß, dass man sie nicht eindeutig als eindimensionale Struktur identifizieren kann.

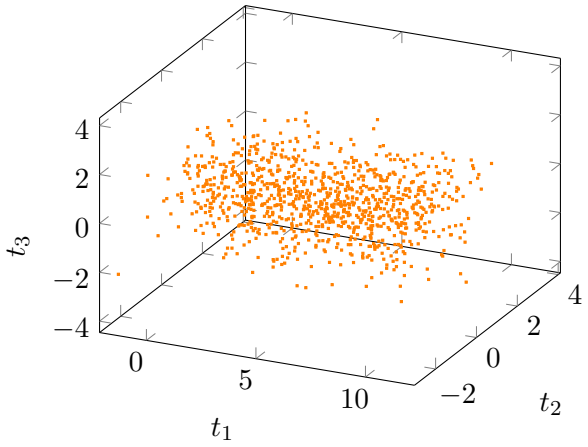

Abb. 8.15: 1000 verrauschte Datenpunkte auf einer Linie.

Mit dem GTM-Modell lernen wir die empirische Dichte im Datenraum. Hierzu verwenden wir eine Dünngitterdiskretisierung auf Level 4 und ein initiales β von 5. Um die Interpretation der Ergebnisse zu vereinfachen, wird das Mapping $\boldsymbol{y}$ nicht regularisiert. In Abbildung 8.16 sind die Resultate für verschiedene Latent-Space-Dimensionen L dargestellt. Wir erkennen, dass mit $L = 1$ die Varianz 1 der Datenpunkte korrekt erkannt wird. Für $L = 2$ und 3 wird die Varianz deutlich unterschätzt, da das Bild $\boldsymbol{y}([0, 1]^L)$ mehr Datenpunkte erreichen kann, und die räumliche Ausdehnung nicht mehr über das Rauschen modelliert werden muss. Wir beobachten ausserdem, dass ein Überschätzen der intrinsischen Dimension in Hinblick auf den Funktionalwert bessere Ergebnisse erzeugt. Allerdings wird hier das Rauschen nicht mehr mit dem korrekten β, sondern mit dem Mapping $\boldsymbol{y}$ modelliert. Insofern ist es empfehlenswert, eine Latent-Space-Dimension L zu wählen, bei der die inverse Varianz dem erwarteten Rauschlevel entspricht.

In einem zweiten Experiment werden wir die intrinsische Dimension der Datenpunkte unterschätzen. Hierfür generieren wir Datenpunkte auf halbierten Hypersphären im D-dimensionalen Raum mit $D = 2, 3$ und 4. Die Punkte sind nicht verrauscht, so dass die inverse Varianz theoretisch $+\infty$ ist. In Abbildung 8.17 sind die Ergebnisse dargestellt. Wir erkennen, dass erst ab einer Latent-Space-Dimension $L \geq D - 1$ die inverse Varianz mit jeder Iteration deutlich zunimmt.

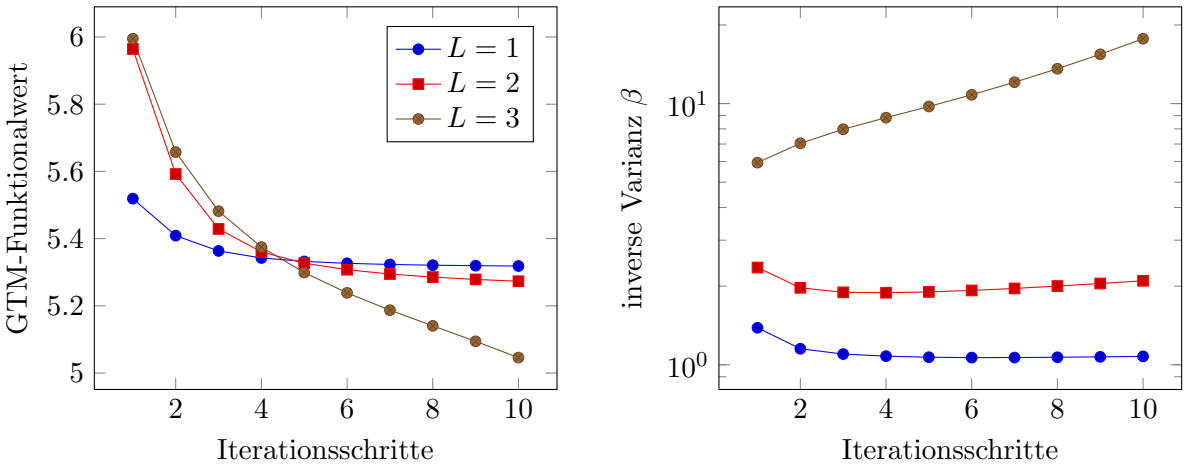

Abb. 8.16: GTM-Funktionalwerte und inverse Varianzen zu verschiedenen Latent-Space-Dimensionen L.

Dies ist der zum ersten Experiment gegenteilige Effekt: Mit einer zu geringen Latent-Space-Dimension wird die Varianz deutlich überschätzt. Dass mit steigender latenter Dimension der GTM-Funktionalwert abnimmt, gilt auch hier. Es ist bemerkenswert, dass die Darstellungen von Funktionalwert und logarithmierter inverser Varianz in Abbildung 8.17 annähernd achsensymmetrisch sind. Dies ist eine direkte Folge des simplen Settings ohne Regularisierungsterm. Weiterhin ist anzumerken, dass eine Überschätzung der Varianz allgemein auf eine schlechte Approximation hindeutet, die auch bei der passenden Latent-Space-Dimension auftreten kann. Dies betrifft jedoch einfache Strukturen wie Halbsphären nicht.

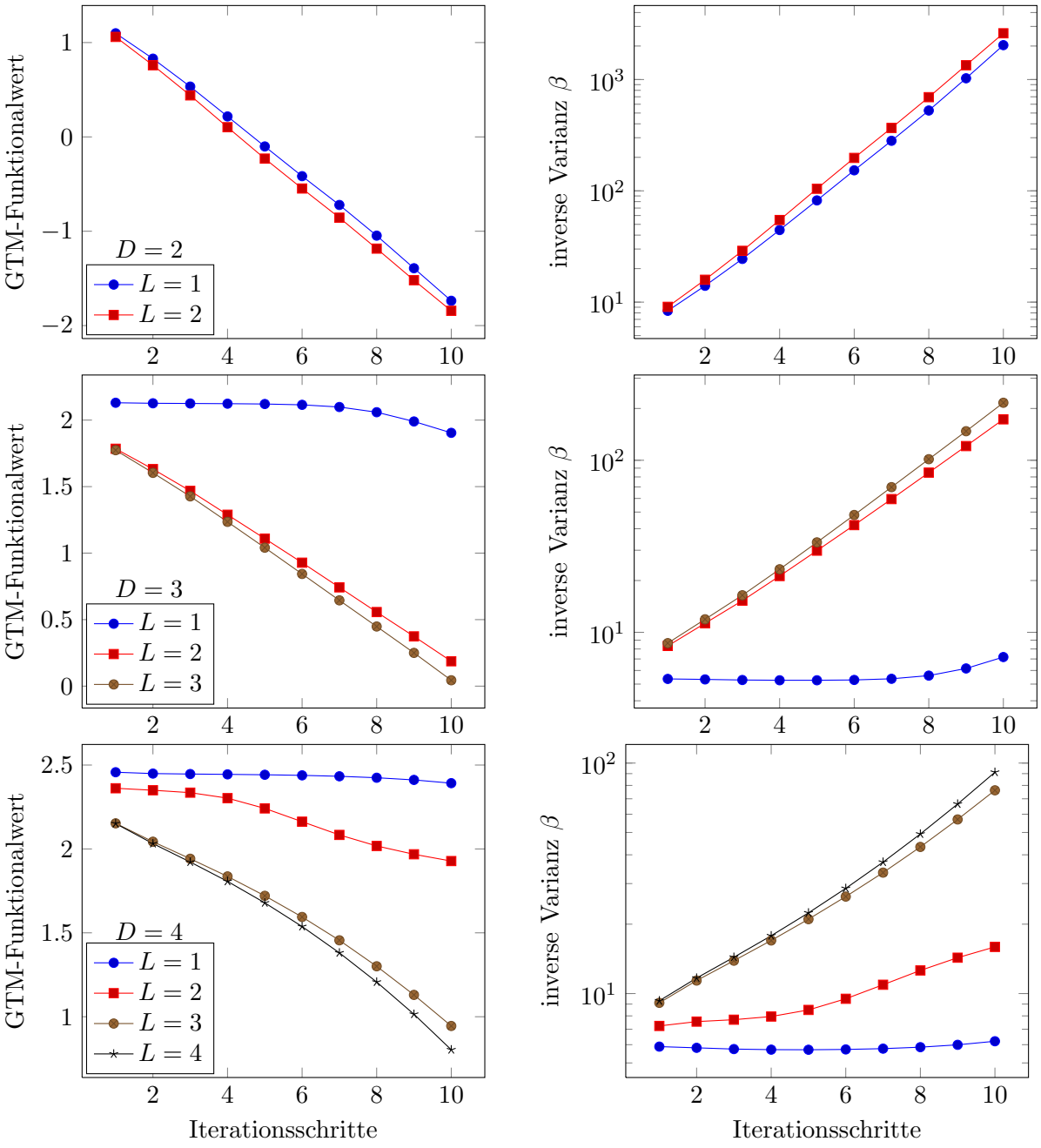

Abb. 8.17: GTM-Funktionalwerte (linke Spalte) und inverse Varianzen (rechte Spalte) von GTM-Modellen auf halbierten Hypersphären in den Dimensionen 2, 3 und 4 (von oben nach unten).

8.4 Varianzschätzung mit p-Gauß-Kernen

In diesem Abschnitt möchten wir das p-Gauß-Kern GTM an einem einfachen Beispiel testen. Wir konstruieren im zehndimensionalen Raum eine Kurve, die in t_1 linear und in den Richtungen $\{t_d\}_{d=2}^{10}$ leicht gekrümmt ist. Von dieser Kurve samplen wir 10^4 Datenpunkte, die zusätzlich p-Gauß-verrauscht werden im Sinne von Abschnitt 7.2 mit $p = 5$ und $\sigma = 1$. Die Abbildung 8.18 stellt einen Teil der Punkte in den ersten drei Dimensionen und ein GTM-Modell nach 10 Iterationsschritten dar.

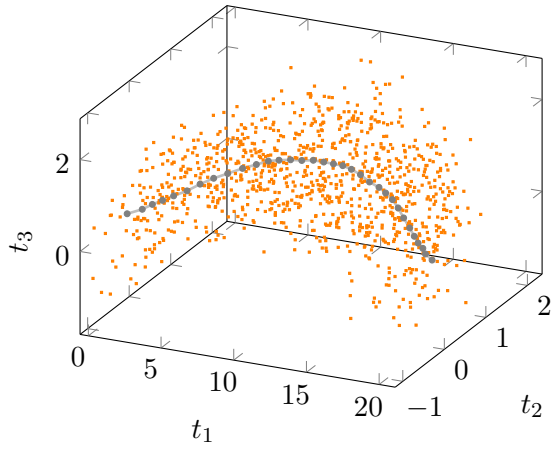

Abb. 8.18: p-Gauß-Kern-verrauschte Datenpunkte auf einer Kurve mit $p = 5$ und $\sigma = 1$.

Nun führen wir mehrere Experimente durch, in denen wir verschiedene p-Gauß-Kern GTM mit $p = 2, \ldots, 8$ die Struktur lernen lassen. Wir verwenden einen eindimensionalen Latent-Space, keine Regularisierung und ein Diskretisierungslevel von 5. In Abbildung 8.19 erkennen wir, dass nur die p-Gauß-Kern-GTM mit $p = 5$ die inverse Varianz richtig schätzt. Dieses Ergebnis war zu erwarten, da unsere Punkte mit $p = 5$ generiert wurden. Ein zu großes p führt zu einem Unterschätzen, ein zu kleines p zu einem Überschätzen. Offenbar müssen die üblichen Gauß-Kerne im 10-dimensionalen Raum die Varianz reduzieren, um eine vergleichbare Lokalität wie ein p-Gauß-Kern mit $p = 5$ zu erhalten, siehe auch Abschnitt 7.2.

Abbildung 8.20 zeigt, dass der GTM-Funktionalwert für ein GTM mit $p = 5$ am stärksten minimiert werden kann. Dies deckt sich mit unserer Erwartung, dass die beste Rekonstruktion der Datendichte mit dem passenden Rauschen erfolgt. Hieran erkennen wir, dass die Berechnung des Funktionalwerts auch in Hinblick auf Konstanten korrekt erfolgt und für verschiedene Kerne vergleichbar ist.

Anhand der Abbildungen ist erkennbar, dass bereits nach wenigen Iterationsschritten das Optimum erreicht wird. Wir haben ein so einfaches Beispiel gewählt, damit sich die hier beschriebenen Zusammenhänge deutlich beobachten lassen.

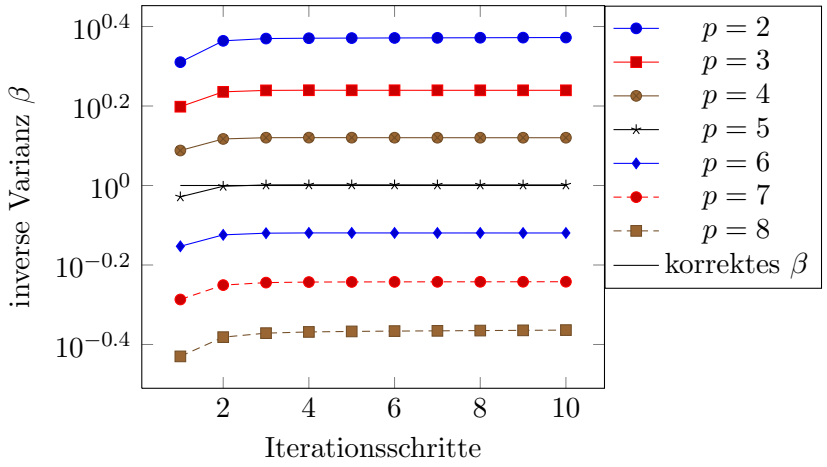

Abb. 8.19: Inverse Varianzen zu $p = 2, \ldots, 8$.

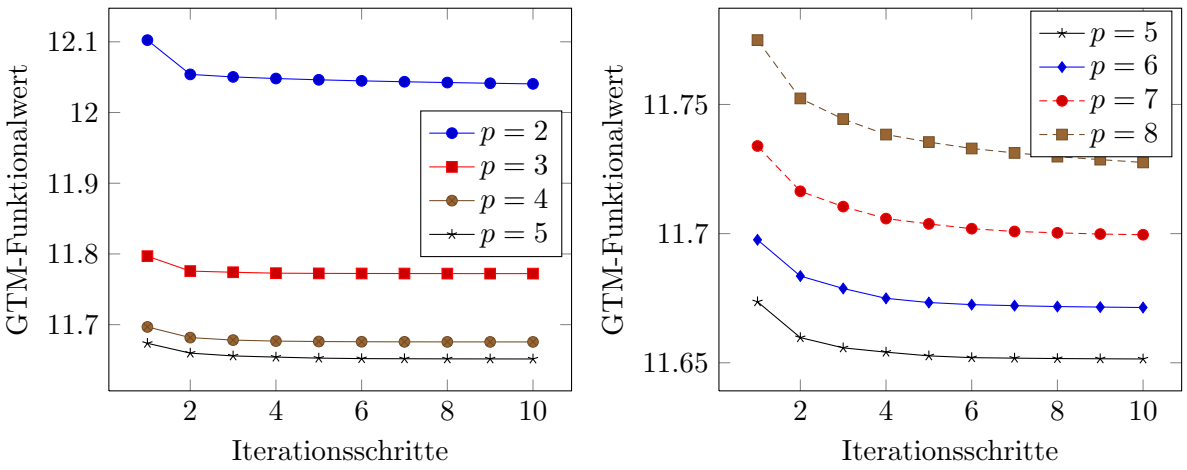

Abb. 8.20: Entwicklung der GTM-Funktionalwerte zu verschiedenen p.

8.5 Anwendungsbeispiele

8.5.1 Oilflow-Datensatz

Der Oilflow-Datensatz wird in [BSW98b] verwendet. Er steht in engem Zusammenhang mit dem Problem, den Anteil von Öl in einer Pipeline, die zudem noch Wasser und Gas führt, mit Gamma-Densitometrie zu bestimmen. Der synthetische Datensatz besteht aus 1000 Messpunkten und modelliert die Abschwächung der Gammastrahlung in der Pipeline sowie das Rauschen der Photonenstatistik. Die drei Phasen Öl, Wasser und Gas können unterschiedliche geometrische Konfigurationen einnehmen, nämlich die Ringströmung, die laminare und und die homogene Strömung.

Zur Dimensionsreduktion und Visualisierung setzen wir das GTM-Modell mit $H^{1,\text{mix}}$-Seminorm-Regularisierung, $\lambda = 0.004$, einem initialen β von 3 und einer Dünngitterdiskretisierung mit Level 3 ein. Abbildung 8.21 links zeigt die Einbettung mit einem zweidimensionalen Latent-Space, Abbildung 8.22 links mit einem dreidimensionalen Latent-Space. Auf der rechten Seite sind die Einbettungen durch die PCA dargestellt. Wir erkennen, dass das GTM als nichtlineares Verfahren im Gegensatz zu der PCA die drei Strömungskonfigurationen voneinander trennen kann.

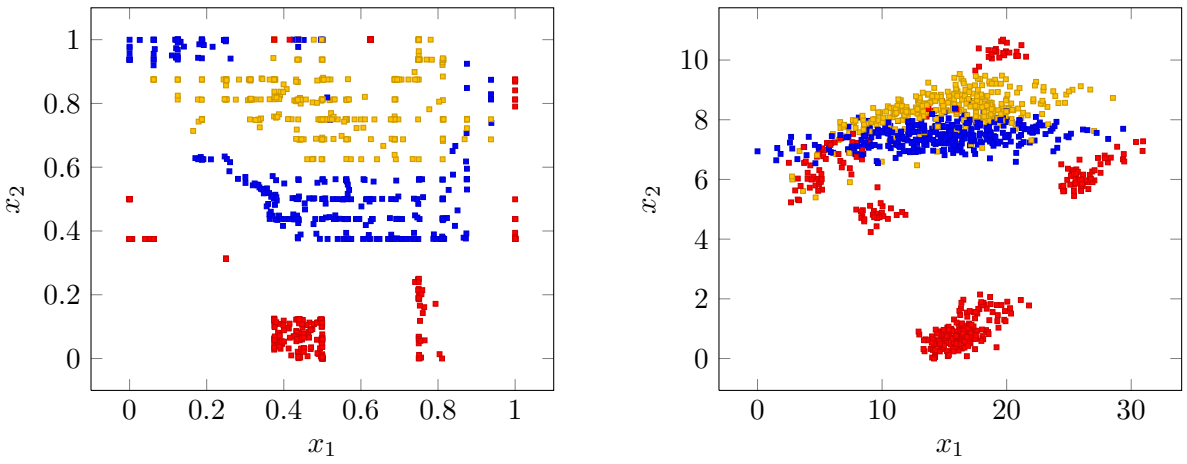

Abb. 8.21: Das linke Bild zeigt die zweidimensionale Projektion des GTM nach 10 Iterationsschritten, das rechte die Einbettung mit der PCA. Hierbei entsprechen die gelben Quadrate der Ringströmung, die roten der laminaren und die blauen der homogenen Strömung.

8.5.2 Faces

Der Faces-Datensatz wird in [TSL00] verwendet und besteht aus 689 Bildern eines Gesichts mit variierender Beleuchtung und Kopfdrehung. Die Bilder sind 64×64 Pixel groß und somit 4096-dimensional. Wenn die Dimension der Daten reduziert wird, entsprechen die ersten Einbettungsdimensionen im Idealfall der Drehung und Beleuchtung.

Wir wollen ein Sparse GTM mit dreidimensionalem Latent-Space anwenden, um eine Einbet-

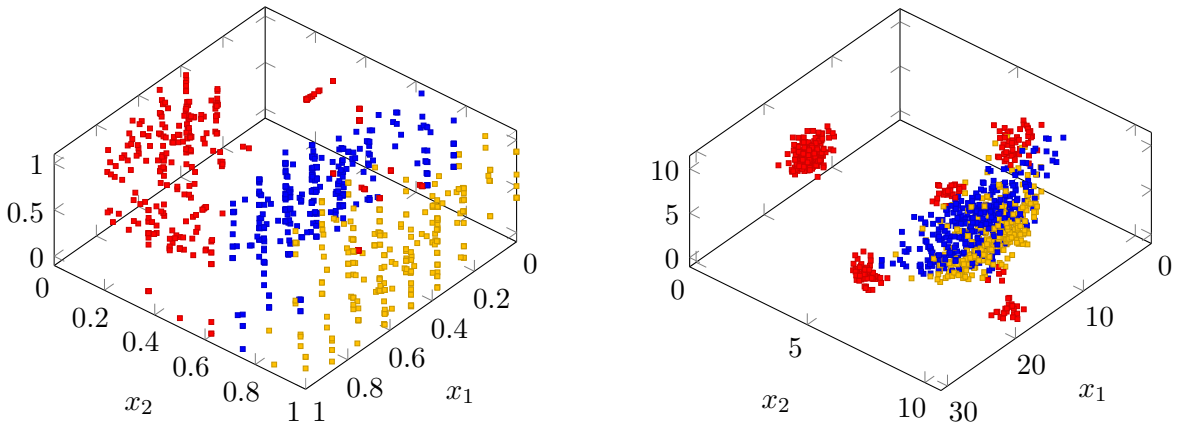

Abb. 8.22: Das linke Bild zeigt die dreidimensionale Projektion des GTM nach 10 Iterationsschritten, das rechte die Einbettung mit der PCA.

tung zu erhalten. Analog zu [LV07] reduzieren wir zunächst die Dimension der Daten mit einer PCA auf 240. Anschließend skalieren wir die Punkte linear auf $[0,1]^{240}$ und wenden das GTM mit H^1-Seminorm, $\lambda = 0.004$ und einem initialen β von 20 an. Wir wählen ein Dünngitterdiskretisierungslevel von 3, was 225 Freiheitsgraden in jeder der 240 Dimensionen entspricht. Um eine niederdimensionale Projektion mit guter Auflösung zu erhalten, verwenden wir ein volles Quadraturgitter auf Level 8 mit 35 937 Punkten.

In Abbildung 8.23 sind die Dimensionen 1 und 2 und die Dimensionen 2 und 3 der niederdimensionalen Projektion auf das Quadraturgitter nach 5 Iterationen dargestellt. An den Positionen der Datenpunkte wird das dazugehörige Gesicht abgebildet. Diese Art der Visualisierung erlaubt leider keine kombinierte Darstellung aller drei Latent-Space-Dimensionen. Im oberen Bild lassen sich folgende Strukturen erkennen: Links sind die Gesichter dunkler als rechts, in der oberen Hälfte schaut die Mehrheit nach rechts, in der unteren nach links. Im unteren Bild ist die Beleuchtung im linken oberen Viertel geringer, und insbesondere für Gesichter am Rand der Abbildung gilt, dass sie ähnlich orientiert sind.

Es ist erkennbar, dass das GTM im Gegensatz zu linearen Verfahren wie der PCA die Datenpunkte so einbetten kann, dass sie sich gleichmäßig im Latent-Space verteilen. Man kann sich durchaus eine bessere niederdimensionale Darstellung der Gesichter vorstellen als die hier gezeigte, es muss jedoch erwähnt werden, dass auch durch die zweidimensionale Darstellung des dreidimensionalen Latent-Space Informationen verloren gehen.

8.5.3 Sonar

In diesem Anwendungsbeispiel setzen wir das GTM zur Klassifikation ein. Der Sonar-Datensatz wurde in [GS88] untersucht und befindet sich im UCI Machine Learning Repository, siehe [UCI]. Es handelt sich hierbei um Sonar-Messungen eines Metallzylinders und eines Gesteinsbrockens aus verschiedenen Winkeln. Die Datenpunkte bestehen aus 60 Dimensionen, die jeweils die Energie in einem bestimmten Frequenzband beschreiben. Der Datensatz enthält 111 Metallzylinder-Messungen und 97 Gesteinsbrocken-Messungen. Wegen dieses Bias erwarten wir eine Klassifikationsrate von mindestens 53.3%.

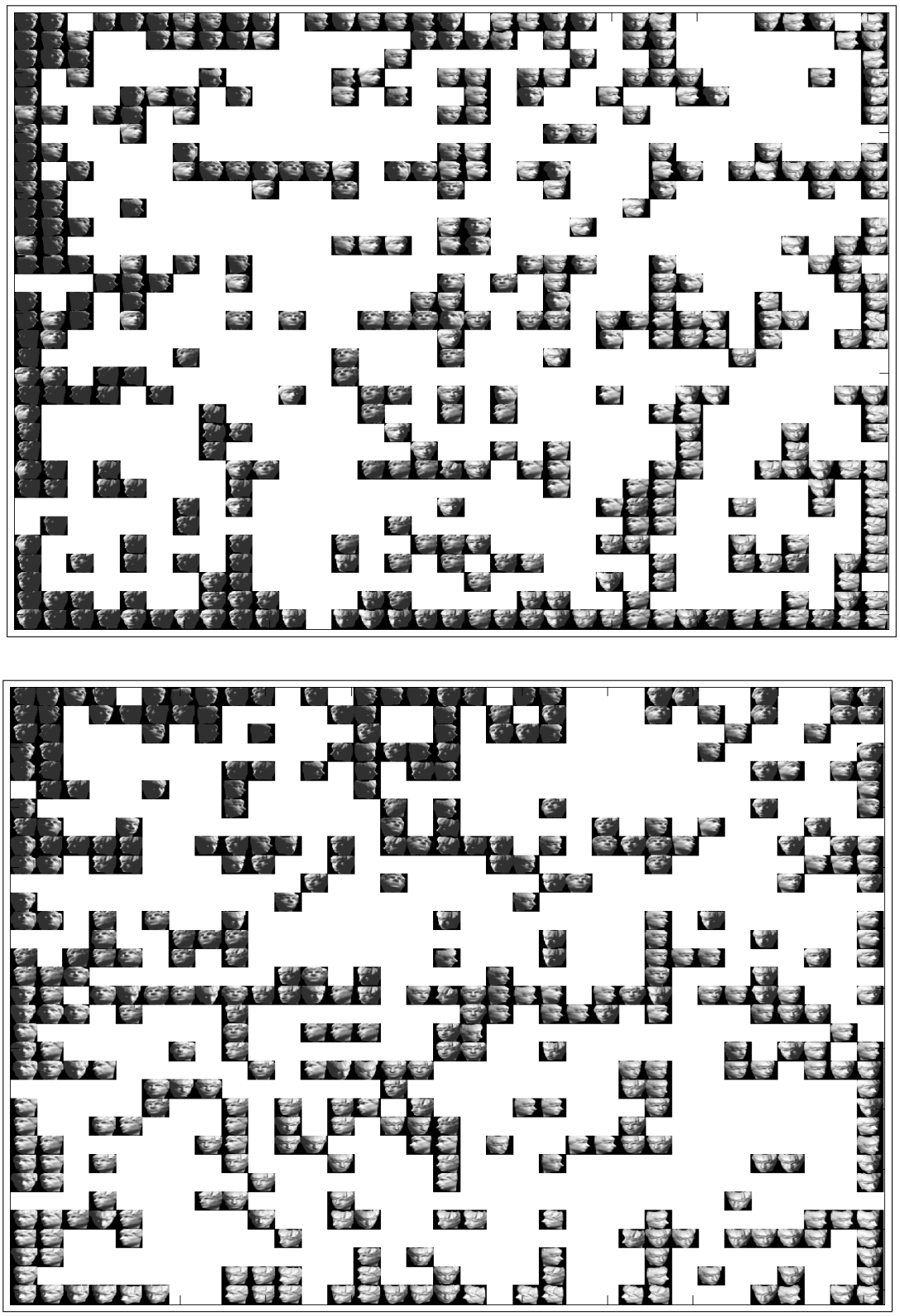

Abb. 8.23: Dimensionen 1 und 2 (oben) und Dimensionen 2 und 3 (unten) der dreidimensionalen GTM-Einbettung der Faces.

Um Robustheit in Hinblick auf das GTM-Klassifikationsergebnis zu erzielen, verwenden wir eine Resampling-Methode, wie sie in [GS88] beschrieben wird. Wir partitionieren unsere 208 Messpunkte zufällig in 13 Testdatensätze zu je 16 Punkten. Die jeweils 192 anderen Messpunkte dienen als dazugehörige Trainingsdatensätze. Wir verwenden ein $H^{1,\text{mix}}$-regularisiertes Mapping mit Dünngitterdiskretisierung und einem initialen β von 3. Die Abbildung 8.24 zeigt die durchschnittlichen Erkennungsraten auf den 13 Datensätzen mit einem eindimensionalen Latent-Space für verschiedene Diskretisierungslevel und Regularisierungsparameter. Abbildung 8.25 zeigt die durchschnittlichen Erkennungsraten mit einem zweidimensionalen Latent-Space.

Mit $L = 2$, einer schwachen Regularisierung und einem hohen Level erreichen wir annähernd 100% Erkennungsrate auf den Trainingsdaten. Dies ist ein klares Beispiel von Overfitting. Die Bandbreite der Ergebnisse auf den Testdaten liegt für $L = 1$ zwischen 61.0% und 78.8%, für $L = 2$ zwischen 72.6% und 84.6%. Die höchste Erkennungsrate auf den Testdaten von 84.6% ($\sigma = 2.6\%$) erzielen wir mit $L = 2$, Diskretisierungslevel 5 und $\lambda = 3.16 \cdot 10^{-5}$.

Bei diesem Experiment wird in [GS88] mit Neuronalen Netzen eine höchste durchschnittliche Erkennungsrate von 84.7% ($\sigma = 5.7\%$) erreicht. Der GTM-Klassifikator erzielt somit ein vergleichbar gutes Ergebnis.

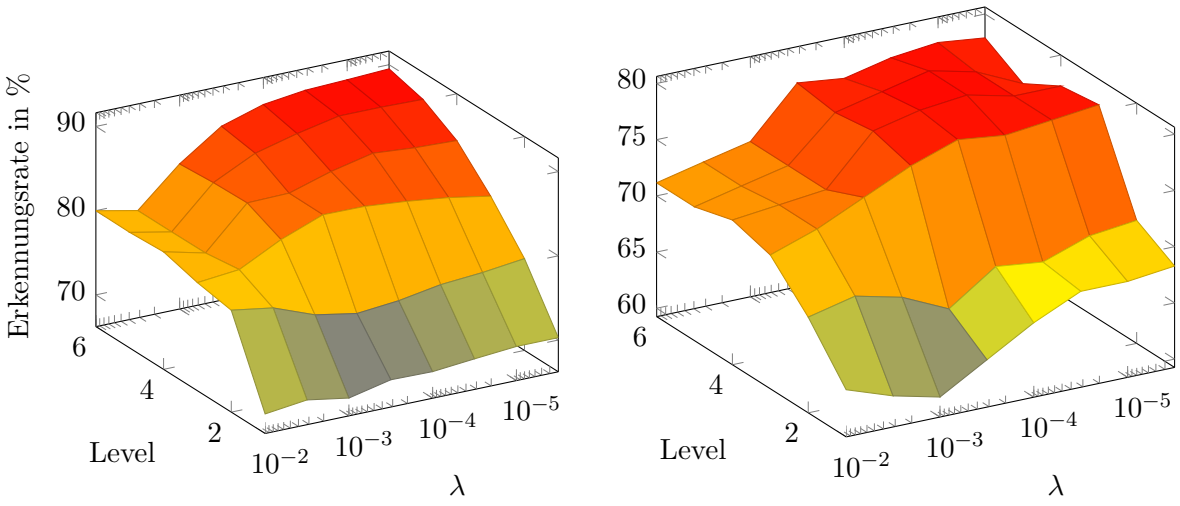

Abb. 8.24: Erkennungsraten mit $L = 1$ auf den Trainings- (links) und Testdaten (rechts).

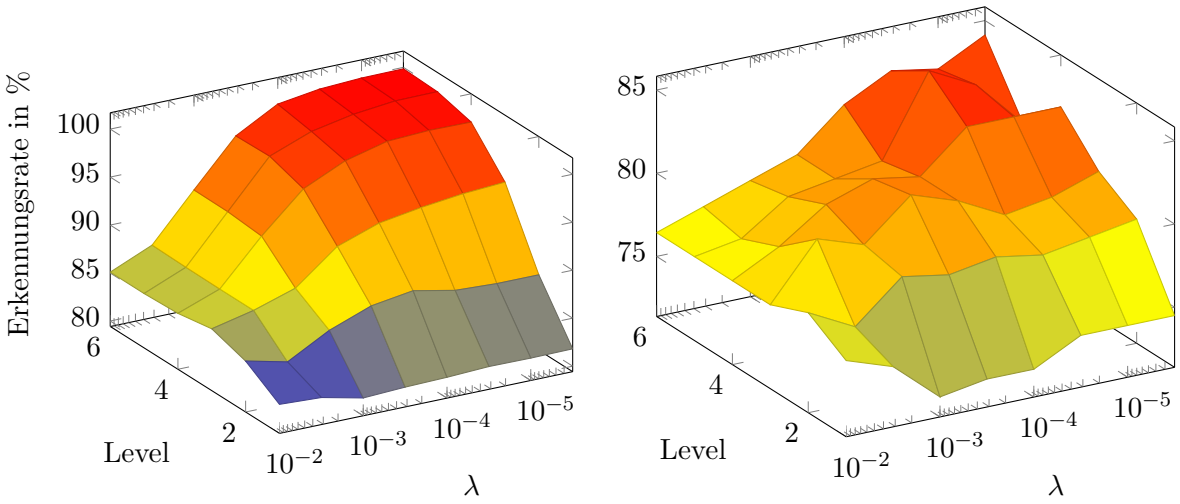

Abb. 8.25: Erkennungsraten mit $L = 2$ auf den Trainings- (links) und Testdaten (rechts).

9 Abschließende Bemerkungen

9.1 Zusammenfassung

In dieser Arbeit haben wir das Generative Topographic Mapping mit einem auf $[0,1]^L$ gleichverteilten Latent-Space-Prior formuliert. Dies hat dazu geführt, dass statt endlicher Summen Integrale über $[0,1]^L$ auftreten. Diese konnten wir mit optimierten Quadraturregeln bestimmen oder auch eine Produktstruktur des Integranden ausnutzen. Das klassische GTM geht mit der Wahl einer bestimmten Quadraturregel aus dieser Formulierung hervor.

Wir haben den Zusammenhang zwischen der Likelihood-Maximierung auf einer endlichen Menge von Samples und der Minimierung der Kullback-Leibler-Divergenz zwischen Datenraumdichte und GTM-Modell-Dichte hergestellt. Dies hat die Definition eines GTM-Funktionals motiviert, dessen Minimierung wir in der Art des EM-Algorithmus – jedoch ohne die Verwendung von stochastischen Begriffen – beschreiben konnten.

Wir haben ein Sparse GTM beschrieben und implementiert, das einen Dünngitteransatz zur Diskretisierung des Mappings $\boldsymbol{y}$ und zur Quadratur über den Latent-Space verwenden kann. Experimentell haben wir die Existenz von Beispielen gezeigt, auf denen die Dünngitterbasis mit weniger Freiheitsgraden eine bessere Datenapproximation erzielt als ein volles Gitter.

Wir konnten ein Low-ANOVA GTM entwerfen und umsetzen, das die in unserer Formulierung entstehenden Integrationsprobleme in das Produkt von eindimensionalen Integralen zerlegt. Hierdurch wird der Fluch der Dimension vollständig gebrochen. Allerdings entstehen durch die verwendete Diskretisierung starke Anforderungen an die Daten: Es wird eine Partitionierung von orthonormalen Richtungen im Datenraum verlangt, bei der die Richtungen über die Partitionen hinweg keine statistischen Abhängigkeiten haben dürfen. Innerhalb einzelner Partitionen kann das Low-ANOVA GTM Nichtlinearitäten erfassen. Somit ist hier ein Verfahren mit einer größeren Darstellungsvielfalt als das PCA-Modell und einer geringeren Laufzeitkomplexität als das klassische GTM entstanden.

Zusätzlich haben wir ein p-Gauß-Kern GTM beschrieben und implementiert, das statt normalverteiltem Rauschen das Rauschen aus p-Gauß-Kernen verwendet. Bei einem solchen GTM ist die M-Schritt-Minimierung ein konvexes Minimierungsproblem, dessen Lösung sich nicht mit einem linearen Gleichungssystem bestimmen lässt. Wir konnten experimentell zeigen, dass das p-Gauß-Kern GTM die Varianz von entsprechend verrauschten Daten richtig schätzt, und bei übereinstimmendem p die beste Datenapproximation möglich ist.

Aus der Dichteschätzung des GTM haben wir einen Klassifikator konstruiert. Mit diesem konnten wir in einem Experiment zeigen, dass die Redundanz von intrinsisch niederdimensionalen Daten in hochdimensionalen Räumen die Erkennungsrate trotz Rauschen verbessern kann. Dem haben wir ein intrinsisch hochdimensionales Klassifikationsproblem gegenübergestellt, in dem das Rauschen mit steigender Dimension die Erkennungsrate verschlechtert.

Schließlich haben wir das Sparse GTM auf in der Literatur gängige Beispiele angewendet und konnten Erkennungsraten wie in bereits bestehenden Veröffentlichungen nachweisen.

9.2 Ausblick

Wir haben in dieser Arbeit Varianten des Generative Topogaphic Mapping vorgestellt, die mit einer substanziellen Reduktion der Freiheitsgrade auf kleineren Problemklassen operieren können. Hierdurch lassen sich höhere Latent-Space Dimensionen verwenden, was beispielsweise für die Klassifikation von hochdimensionalen Daten mit einer intrinsischen Dimension von 4 oder 5 entscheidend sein kann. Das GTM als Klassifikator ist eine interessante Anwendung, da die Datenraumdimension nur linear in die Laufzeit eingeht. Eine Erweiterung der hier vorgestellten Varianten um kategorielle Variablen oder Regression könnte zusätzliche Anwendungsfelder eröffnen.

In [LV07] wird festgestellt, dass die unintuitiven Effekte in hohen Dimensionen nicht nur im Datenraum auftreten, sondern bei vielen Verfahren ab einer bestimmten intrinsischen Dimension auch die Einbettung betreffen. Inwieweit dies für den Latent-Space des GTM gilt, müsste untersucht werden. Des Weiteren wäre es interessant, Beispiele aus der Praxis zu identifizieren, bei denen sich die Vorteile des p-Gauß-Kern GTM gegenüber den üblichen Gauß-Kernen nachweisen lassen.

Literaturverzeichnis

[Aro50] Aronszajn, N.: *Theory of reproducing kernels.* Transactions of the American Mathematical Society, 68, 1950.

[AS06] Altun, Yasemin und Alex Smola: *Unifying Divergence Minimization and Statistical Inference via Convex Duality.* In: *Proc. of Conf. on Learning Theory (COLT*, 2006.

[Bel61] Bellman, R.: *Adaptive Control Processes: A Guided Tour.* Princeton University Press, 1961.

[BG04] Bungartz, Hans-Joachim und Michael Griebel: *Sparse grids.* Acta Numerica, 13:1–123, 2004.

[BGM09] Beylkin, Gregory, Jochen Garcke und Martin J. Mohlenkamp: *Multivariate Regression and Machine Learning with Sums of Separable Functions.* SIAM Journal on Scientific Computing, 31(3):1840–1857, 2009.

[BSW98a] Bishop, Christopher M., Markus Svensén und Christopher K. I. Williams: *Developments of the Generative Topographic Mapping.* Neurocomputing, 21:203–224, 1998.

[BSW98b] Bishop, Christopher M., Markus Svensén und Christopher K. I. Williams: *GTM: The Generative Topographic Mapping.* Neural Computation, 10:215–234, 1998.

[CA02] Cichocki, A und S Amari: *Adaptive Blind Signal and Image Processing.* Wiley, 2002.

[Caf98] Caflisch, RE: *Monte Carlo and quasi-Monte Carlo methods.* Acta Numerica, Seiten 1–49, 1998.

[CP97] Carreira-Perpinan, Miguel: *A Review of Dimension Reduction Techniques*, 1997.

[DR84] Davis, Philip J. und Ph. Rabinowitz: *Methods of numerical integration.* Academic Press, Inc., Orlando, US-FL, 2nd Auflage, 1984.

[Feu05] Feuersänger, C.: *Dünngitterverfahren für hochdimensionale elliptische partielle Differentialgleichungen.* Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2005.

[FG09] Feuersänger, Chr. und M. Griebel: *Principal Manifold Learning by Sparse Grids.* Computing, 85(4), August 2009. Also available as INS Preprint no 0801.

[Fra07] François, D.: *High-dimensional data analysis : optimal metrics and feature selection.* Dissertation, l'Université catholique de Louvain, Januar 2007.

[FWV05] François, D., V. Wertz und M. Verleysen: *About the locality of kernels in high-dimensional spaces.* In: *ASMDA 2005, International Symposium on Applied Stochastic Models and Data Analysis*, Seiten 238–245, Brest (France), 2005.

[Gar04] Garcke, J.: *Maschinelles Lernen durch Funktionsrekonstruktion mit verallgemeinerten dünnen Gittern.* Doktorarbeit, Institut für Numerische Simulation, Universität Bonn, 2004.

[GG98] Gerstner, T. und M. Griebel: *Numerical Integration using Sparse Grids.* Numer. Algorithms, 18:209–232, 1998. (also as SFB 256 preprint 553, Univ. Bonn, 1998).

[GH08] Griebel, M. und M. Hegland: *A finite element method for density estimation with gaussian priors.* SIAM Num ANAL, 2008. submitted. Also available as SFB 611 Preprint no. 424.

[GP83] Grassberger, P. und I. Procaccia: *Measuring the Strangeness of Strange Attractors.* Physica D, 9:189–208, 1983.

[Gri06] Griebel, M.: *Sparse grids and related approximation schemes for higher dimensional problems.* In: Pardo, L., A. Pinkus, E. Suli und M.J. Todd (Herausgeber): *Foundations of Computational Mathematics (FoCM05), Santander*, Seiten 106–161. Cambridge University Press, 2006.

[GS88] Gorman, R Paul und Terrence J Sejnowski: *Analysis of hidden units in a layered network trained to classify sonar targets.* Neural Networks, 1:75, 1988.

[GSL] *GSL - GNU Scientific Library.* http://www.gnu.org/software/gsl/.

[GSZ92] Griebel, M., M. Schneider und C. Zenger: *A combination technique for the solution of sparse grid problems.* In: Groen, P. de und R. Beauwens (Herausgeber): *Iterative Methods in Linear Algebra*, Seiten 263–281. IMACS, Elsevier, North Holland, 1992. also as SFB Bericht, 342/19/90 A, Institut für Informatik, TU München, 1990.

[HKO01] Hyvärinen, A, J Karhunen und E Oja: *Independent Component Analysis.* Wiley, 2001.

[HNS08] Huo, Xiaoming, Xuelei Ni und Andrew K. Smith: *A survey of manifold-based learning methods - Emerging nonparametric methodology*, 2008.

[Hol08] Holtz, M.: *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance.* Dissertation, Institut für Numerische Simulation, Universität Bonn, 2008.

[Hot33] Hotelling, H: *Analysis of a complex of statistical variables into principal components.* In: *Journal of Educational Psychology, 24:417–441 and 498–520*, 1933.

[Kar46] Karhunen, K: *Zur Spektraltheorie Stochastischer Prozesse. Annales Academiae Scientiarum Fennicae*, 1946.

[KB89] Kiers, Henk und Jos Berge: *Alternating least squares algorithms for simultaneous components analysis with equal component weight matrices in two or more populations.* Psychometrika, 54(3):467–473, September 1989.

[KL97] Kambhatla, Nandakishore und Todd K. Leen: *Dimension reduction by local principal component analysis.* Neural Comput., 9(7):1493–1516, 1997.

[Koh82] Kohonen, T.: *Self-organized formation of topologically correct feature maps.* Biological Cybernetics, 43:59–69, 1982.

[Loe48] Loeve, M: *Fonctions aleatoires du second ordre, Supplement to.* In: *Processus Stochastiques et Mouvement Brownien, Gauthier-Villars*, 1948.

[LS95] Lafferty, John D. und Bernhard Suhm: *Cluster Expansions and Iterative Scaling for Maximum Entropy Language Models.* CoRR, abs/cmp-lg/9509003, 1995.

[LV07] Lee, J. A. und M. Verleysen: *Nonlinear Dimensionality Reduction.* Springer, 2007.

[Mac95] MacKay, D. J. C.: *Bayesian Neural Networks and Density Networks.* Nuclear Instruments and Methods in Physics Research, Section A, 354(1):73–80, 1995.

[NH98] Neal, Radford und Geoffrey E. Hinton: *A View Of The Em Algorithm That Justifies Incremental, Sparse, And Other Variants.* In: *Learning in Graphical Models*, Seiten 355–368. Kluwer Academic Publishers, 1998.

[Pea01] Pearson, K: *On lines and planes of closest fit to systems of points in space.* Phil.Mag, Seiten 559–572, 1901.

[RM51] Robbins, Herbert und Sutton Monro: *A Stochastic Approximation Method.* The Annals of Mathematical Statistics, 22(3):400–407, 1951.

[Sha06] Shalizi, Cosma: *Introduction to information theory.* In: *Lecture Notes on Advanced Probability II.* Carnegie Mellon University, 2006.

[Smo63] Smolyak, S.A.: *Quadrature and interpolation formulas for tensor products of certain classes of functions.* In: *Dokl. Akad. Nauk SSSR 4*, Seiten 240–243, 1963.

[SS01] Scholkopf, Bernhard und Alexander J. Smola: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA, 2001.

[Sve98] Svensén, Markus: *GTM: The Generative Topographic Mapping.* Dissertation, Aston University, September 1998.

[Tem84] Temlyakov, N.: *Approximation of periodic functions.* Nova Science, New York, 1984.

[TN01] Tino, Peter und Ian Nabney: *Hierarchical GTM: constructing localized nonlinear projection manifolds in a principled way*, 2001.

[TSL00] Tenenbaum, Joshua B., Vin de Silva und John C. Langford: *A Global Geometric Framework for Nonlinear Dimensionality Reduction.* Science, 290(5500):2319–2323, 2000.

[TW88] Traub, Wasilkowski und Wozniakowski: *Information-Based Complexity.* Academic Press, London, 1988.

[UCI] *UCI Machine Learning Repository.* http://archive.ics.uci.edu/.

[Ver02] Verleysen, Michel: *Learning High-Dimensional Data.* In: *Limitations and Future Trends in Neural Computation 186*, Seiten 141–162, 2002.

[Wah90] Wahba, G.: *Spline models for observational data*, Band 59 der Reihe *CBMS-NSF Regional Conference Series in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.

[Wis08] Wissel, D.: *Die Diskrete Gauß-Transformation - schnelle Approximationsverfahren und Anwendungen in hohen Dimensionen.* Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, April 2008.