Semi-supervised learning with sparse grids

Jochen Garcke JOCHEN.GARCKE@ANU.EDU.AU
Mathematical Sciences Institute, Australian National University, Canberra ACT 0200, Australia

Michael Griebel GRIEBEL@INS.UNI-BONN.DE
Rheinische Friedrich-Wilhelms-Universität Bonn, Institut für Numerische Simulation, 53115 Bonn, Germany

Abstract

Sparse grids were recently introduced for classification and regression problems. In this article we apply the sparse grid approach to semi-supervised classification. We formulate the semi-supervised learning problem by a regularization approach. Here, besides a regression formulation for the labeled data, an additional term is involved which is based on the graph Laplacian for an adjacency graph of all, labeled and unlabeled data points. It reflects the intrinsic geometric structure of the data distribution. We discretize the resulting problem in function space by the sparse grid method and solve the arising equations using the so-called combination technique. In contrast to recently proposed kernel based methods which currently scale cubic in regard to the number of overall data, our method scales only linear, provided that a sparse graph Laplacian is used. This allows to deal with huge data sets which involve millions of points. We show experimental results with the new approach.

1. Introduction

In semi-supervised classification in addition to labeled data also unlabeled data is used by a machine learning algorithm in the training phase. Here, the aim is to improve the quality of generalization results in comparison to approaches which use the labeled data only. The semi-supervised ansatz is furthermore motivated by the observation that in many application areas, like remote sensing, text classification, or medical imaging, data observations can be obtained cheaply and easily, whereas a precise labeling of the samples is expensive and time consuming, often involves processing by human experts and needs sophisticated and expensive tests, or is sometimes even almost infeasible. Also, in the business domain of costumer relationship management, where often a large amount of labeled data is available, like current profitable and un-profitable customers, typically even more unlabeled data, like new potential customers, can be easily obtained. In all these applications there is the potential for a better performance of machine learning algorithms by the additional use of unlabeled data.

One approach in semi-supervised learning is to exploit the geometric structure of the data distribution. The underlying assumption, often utilized indirectly, is that nearby data points should belong to the same class. A stronger condition is the so-called cluster assumption, which states that the decision boundary should lie in regions of low data density. Then points which are connected by a path through regions with high data density have the same label. For example in (Bousquet et al., 2004; Chapelle & Zien, 2005) different algorithms based on the cluster assumption are proposed. The former uses the gradient weighted by an empirical estimate of the density of the data distribution as a regularization operator, the latter uses a transductive support vector machine (Vapnik, 1998) with a graph-distance derived kernel.

Manifold learning algorithms implicitly use the cluster assumption. They attain classifiers which vary little along the manifolds described by the data. When classes form different manifolds the decision boundary will therefore be in-between the manifolds, which indirectly implements the cluster assumption. In (Belkin et al., 2004; Belkin et al., 2005) the regularization network approach of (Girosi et al., 1995) is extended with ideas from manifold learning. There, an additional regularization term is introduced for the smoothness with respect to the intrinsic structure of the data distribution. Furthermore, they propose new kernel based algorithms which extend support vector machines (SVM) and regularized least squares in this respect. Their approach is formulated in a Reproducing

Appearing in *Proceedings of the 22^{nd} International Conference on Machine Learning*, Bonn, Germany, 2005.

Kernel Hilbert Space and a Representer theorem is proved which gives the theoretical foundation for the algorithms. The additional regularization term employed is the graph Laplacian of a weighted data adjacency graph. It resembles an empirical estimate of the Laplace operator defined on the manifold on which the data is situated.

In this article, we apply the ansatz of (Garcke et al., 2001; Garcke & Griebel, 2002) to this extended regularization network approach. In contrast to kernel based methods, where global ansatz functions associated to data points are employed, in our approach a point grid, independent of the data, with associated local ansatz functions is used to discretize the function space and represent the classifier. This is similar to the numerical treatment of partial differential equations by finite element methods.

Conventional grid-based techniques usually suffer from the curse of dimensionality, i.e., the complexity of the computation grows exponentially with the dimension. This is probably the reason why these methods were not used in machine learning until recently. However, a discretization using so-called sparse grids (Zenger, 1991) allows to cope with the curse of dimensionality to some extent. It is based on a hierarchical basis and a tensor product construction. This method has been originally developed for the solution of partial differential equations and is now successfully used for integral equations, interpolation and approximation, eigenvalue problems, and integration problems, see (Bungartz & Griebel, 2004) for an overview article. This ansatz is also known as 'hyperbolic cross points' and the idea can be traced back to (Smolyak, 1963).

We apply sparse grids in the form of the combination technique (Griebel et al., 1992) to the extended regularization network from (Belkin et al., 2004; Belkin et al., 2005). The problem is discretized and solved on a certain sequence of anisotropic grids, i.e., grids with in general different uniform mesh sizes in each coordinate direction. A linear combination of the partial solutions then gives the sparse grid representation. Thus the classifier is built with ansatz functions associated to grid points and not data points. The resulting method scales linear in the number of instances, i.e., the amount of labeled and unlabeled data (as long the graph Laplacian is sparse). This allows to deal with huge data sets with millions of points. Due to complexity reasons, the number of attributes is currently limited to 20 dimensions in practical applications, which however often can be reached by a suitable preprocessing of the data.

The remainder of this paper is organized as follows: In section 2 we state the semi-supervised learning problem in the regularization network formulation with an additional regularization term which involves the graph Laplacian derived from the data points. In section 3 we discuss the discretization and solution in a sparse grid function space where we employ the so-called sparse grid combination technique. In section 4 we comment on the computational complexities of our method. Section 5 describes experimental results of our new method for the two moons toy problem, two large 10-dimensional data sets, and a task from the Data-Mining-Cup 2000, where the profit of a direct mailing campaign has to be maximized. We give some final remarks in section 6.

2. Semi-supervised Learning

Semi-supervised learning, also called learning from partially classified examples, has been first explored in statistics, where it is modeled as a missing data problem. There, mixture models are used where mixture components are identified as classes. Most approaches adapt the EM algorithm and perform maximum likelihood estimation, see (McLachlan, 1992). An overview of more recent approaches to semi-supervised learning including co-training, Fisher kernels, and transductive interference can be found in the review paper (Seeger, 2000). In (Belkin et al., 2004) a list of recent developments for semi-supervised classification using graph based approaches is given. Furthermore, (Smola & Kondor, 2003) describe kernels and regularization operators on graphs in a more general framework, including diffusion kernels and the graph Laplacian as special cases.

We interpret classification of data as a scattered data approximation problem in a possibly high-dimensional space. Given is a set of already labeled data

$$S_l = \{(\underline{x}_i, y_i)\}_{i=1}^{m_l} \quad \underline{x}_i \in \mathbb{R}^d,\ y_i \in \{-1, 1\},$$

and an additional set of unlabeled data

$$S_u = \{\underline{x}_i\}_{i=m_l+1}^{m_l+m_u} \quad \underline{x}_i \in \mathbb{R}^d.$$

We denote in the following $m := m_l + m_u$ as the number of all, labeled and unlabeled data.

First, let us briefly consider the conventional case, where we only make use of the labeled data set S_l. Assume that the data has been obtained by sampling an unknown function f which belongs to some space V of functions defined over $\mathbb{R}^d$. The sampling process may be disturbed by noise. The aim is to recover the function f from the given data as good as possible. To achieve a well-posed (and uniquely solvable) problem to recover f from the given data Tikhonov-regularization theory (Tikhonov & Arsenin, 1977; Wahba, 1990) imposes a smoothness constraint on the solution. This leads to the variational problem

$$\min_{f \in V} R(f)$$

with

$$R(f) = \frac{1}{m_l} \sum_{i=1}^{m_l} (f(\underline{x}_i) - y_i)^2 + \lambda ||\mathcal{S} f||_2^2. \qquad (1)$$

Here, $\mathcal{S}$ is a linear operator. Typical examples are

$$\mathcal{S} f = \nabla f, \quad \text{or} \quad \mathcal{S} f = \Delta f,$$

where ∇ denotes the gradient and Δ the Laplace operator. The first term in (1) measures the error and therefore enforces closeness of f to the labeled data, the second term $||\mathcal{S} f||_2^2$ enforces smoothness of f, and the regularization parameter λ balances these two terms. This formulation was introduced for machine learning in (Girosi et al., 1995) under the name regularization network.

Here and in the following we only consider the squared error $\sum_{i=1}^{m_l} (f(\underline{x}_i) - y_i)^2$ as a cost function, but note that other terms, e.g. soft margin loss for SVM classification, can be used as well.

In (Belkin et al., 2004; Belkin et al., 2005) the formulation (1) is extended to the case of labeled and unlabeled data with an additional regularization term $\|f\|_I$, which controls the smoothness with respect to the intrinsic geometry. This is motivated by the assumption that two points which are nearby in the intrinsic geometry of the data distribution might have a higher likelihood to be in the same class. One then has to minimize

$$R(f) = \frac{1}{m_l} \sum_{i=1}^{m_l} (f(\underline{x}_i) - y_i)^2 + \lambda_A ||\mathcal{S} f||_2^2 + \lambda_I \|f\|_I^2.$$

There are now two regularization parameters λ_A, λ_I which control the amount of smoothing of the function in the ambient space and the amount of smoothing with respect to the data distribution, respectively.

Furthermore, (Belkin et al., 2004; Belkin et al., 2005) propose to choose the Laplace operator defined on the manifold $\mathfrak{M}$ which is formed by the data

$$\|f\|_I^2 = \int_{\mathfrak{M}} \langle \nabla_{\mathfrak{M}} f, \nabla_{\mathfrak{M}} f \rangle$$

as the intrinsic regularizer. This approach is based on ideas from manifold learning, where in recent years various algorithms were suggested. There, the aim is to obtain a lower-dimensional embedding of data which form a nonlinear manifold $\mathfrak{M} \subset \mathbb{R}^d$. Algorithms like Isomap, Locally Linear Embedding (LLE), Laplacian Eigenmap, Hessian LLE, or Semidefinite Embedding (SDE) achieve this roughly as follows: First, for each point a neighborhood in input space is computed and, depending on which kind of mapping is to be learned and which geometric signature is to be preserved, a matrix is derived from it. Then, the top or bottom eigenvectors of this matrix are determined for a spectral embedding, see the recent paper on SDE (Weinberger & Saul, 2004) for details and references. Note that Isomap, LLE, Laplacian Eigenmaps, and SDE can also be linked to kernel PCA (Bengio et al., 2004; Ham et al., 2004; Weinberger et al., 2004).

To achieve an empirical estimate for $\|f\|_I^2$ (Belkin et al., 2004; Belkin et al., 2005) now first compute an adjacency graph with k-nearest neighbors or, alternatively, ϵ-neighborhoods from the labeled and unlabeled data to approximate the manifold structure. The associated graph Laplacian, using appropriate weights, can then be taken as an approximation of $\int_{\mathfrak{M}} \langle \nabla_{\mathfrak{M}} f, \nabla_{\mathfrak{M}} f \rangle$. This results in the new objective function

$$\begin{aligned} R(f) &= \frac{1}{m_l} \sum_{i=1}^{m_l} (f(\underline{x}_i) - y_i)^2 + \lambda_A \|\mathcal{S} f\|_2^2 \\ &\quad + \frac{\lambda_I}{m^2} \sum_{i,j=1}^{m} \left(f(\underline{x}_i) - f(\underline{x}_j)\right)^2 \mathcal{W}_{i,j}, \end{aligned}$$

where $\mathcal{W}_{i,j}$ are edge weights in the data adjacency graph. Here, the weights are chosen as a function of the distance between $\underline{x}_i$ and $\underline{x}_j$. Possible choices are the heat kernel $\exp(-\|\underline{x}_i - \underline{x}_j\|^2/\sigma)$ or the inverted squared Euclidean distance. Alternatively binary weights might be used, i.e., $\mathcal{W}_{i,j} = 1$ if there is an edge between vertices i and j and 0 otherwise. In text classification angles are commonly used instead of distances. The fore-factor $1/m^2$ is usually suggested as the natural scale for the empirical estimate of the Laplace operator, it may be replaced by $\sum_{i,j}^{m} \mathcal{W}_{i,j}$ for a sparse adjacency graph.

This formulation can be rewritten as

$$\begin{aligned} R(f) &= \frac{1}{m_l} \sum_{i=1}^{m_l} (f(\underline{x}_i) - y_i)^2 + \lambda_A \|\mathcal{S} f\|_2^2 \qquad (2) \\ &\quad + \frac{\lambda_I}{m^2} \underline{f}^\top \mathcal{L} \underline{f}. \end{aligned}$$

Here $\underline{f} = (f(\underline{x}_1), \ldots, f(\underline{x}_m))^\top$ and $\mathcal{L}$ denotes the respective graph Laplacian $\mathcal{L} = \mathcal{D} - \mathcal{W}$ with the diagonal matrix $\mathcal{D}$ given by $\mathcal{D}_{i,i} = \sum_{j=1}^{m} \mathcal{W}_{i,j}$.

Note that the graph Laplacian $\mathcal{L}$ is the empirical counterpart of the Laplace operator defined on the manifold only if the data distribution is uniform on $\mathfrak{M}$. In the general case the continuous equivalent of $\underline{f}^\top \mathcal{L} \underline{f}$ is $\langle p\nabla f, p\nabla f\rangle$, where p denotes the density of the marginal distribution, see (Bousquet et al., 2004). There, in an approach called measure based regularization, the term $\langle p\nabla f, p\nabla f\rangle_{L_2}$ is used for regularization but a corresponding term for $||\mathcal{S}f||_2^2$ is missing. For two-dimensional examples this method implements the cluster assumption, but for real world experiments no successful results could be reported. Furthermore, in (Bousquet et al., 2004), a method is proposed which uses a density based change of geometry which is linked to Isomap. Its implementation is discussed in (Chapelle & Zien, 2005). It is also shown that for the standard L_2-norm of the gradient "modifying the measure [according to the density] and keeping the geometry, or modifying the geometry [to a density based one] and keeping the Lebesgue measure leads to the same regularizer".

Let us now assume that we have a basis of the function space V given by $\{\varphi_j(\underline{x})\}_{j=1}^\infty$. We can then represent every $f \in V$ as

$$f(\underline{x}) = \sum_{j=1}^{\infty} \alpha_j \varphi_j(\underline{x}). \qquad (3)$$

We plug (3) into (2) and obtain

$$\begin{aligned} R(f) &= \frac{1}{m_l} \sum_{i=1}^{m_l} \left(\sum_{j=1}^{\infty} \alpha_j \varphi_j(\underline{x}_i) - y_i \right)^2 \\ &\quad + \lambda_A \sum_{i,j=1}^{\infty} \alpha_i \alpha_j \langle \mathcal{S}\varphi_i, \mathcal{S}\varphi_j \rangle_2 \\ &\quad + \frac{\lambda_I}{m^2} \sum_{i,j=1}^{\infty} \alpha_i \alpha_j \sum_{k,l=1}^{m} \varphi_i(\underline{x}_k) \mathcal{L}_{k,l} \varphi_j(\underline{x}_l). \end{aligned}$$

Setting $\frac{\partial R(f)}{\partial \alpha_q} = 0$ gives $(q = 1, \ldots, \infty)$

$$\sum_{j=1}^{\infty} \alpha_j \left[\sum_{i=1}^{m_l} \varphi_j(\underline{x}_i) \cdot \varphi_q(\underline{x}_i) + \lambda_A m_l \langle \mathcal{S}\varphi_q, \mathcal{S}\varphi_j \rangle_{L^2} + \frac{\lambda_I}{m^2} m_l \sum_{k,l=1}^{m} \varphi_q(\underline{x}_k)\, \mathcal{L}_{k,l}\, \varphi_j(\underline{x}_l) \right] = \sum_{i=1}^{m_l} y_i \varphi_q(\underline{x}_i). \qquad (4)$$

This is equivalent to the system of linear equations

$$\left(\mathcal{B}_{m_l}^\top \mathcal{B}_{m_l} + \lambda_A m_l \cdot \mathcal{C} + \frac{\lambda_I m_l}{m^2} \mathcal{B}^\top \mathcal{L} \mathcal{B} \right) \alpha = \mathcal{B}_{m_l}^\top y, \qquad (5)$$

with infinite-dimensional matrices and vectors, where the data matrix $\mathcal{B}$ is a 'rectangular' matrix $(\mathcal{B}^\top)_{j,k} = (\mathcal{B}_{m_l}^\top\ \mathcal{B}_{m_u}^\top)_{j,k} = \varphi_j(\underline{x}_k)$, $j = 1, \ldots, \infty$, $k = 1, \ldots, M$, and the 'quadratic' regularization matrix $\mathcal{C}$ is positive semi-definite with entries $\mathcal{C}_{j,k} = \langle \mathcal{S}\varphi_j, \mathcal{S}\varphi_k \rangle_{L_2}$, $j, k = 1, \ldots, \infty$. This can be rewritten as

$$\left(\lambda_A m_l \cdot \mathcal{C} + \mathcal{B}^\top \left(\mathcal{I}_{m_l} + \frac{\lambda_I m_l}{m^2} \mathcal{L} \right) \mathcal{B} \right) \alpha = \mathcal{B}_{m_l}^\top y, \qquad (6)$$

where $\mathcal{I}_{m_l}$ consists of the identity matrix for the m_l labeled data and is zero everywhere else, i.e., $\mathcal{I}_{m_l} = \begin{pmatrix} \mathcal{I} & 0 \\ 0 & 0 \end{pmatrix}$ provided that the vector α and the other matrices are accordingly block-partitioned.

3. Discretization with Sparse Grids

In the following we restrict the problem explicitly to a finite dimensional subspace $V_N \subset V$ with an appropriate basis. Such an explicit restriction to a discrete space is fundamentally different to kernel approaches. There, a finite representation of the solution in an infinite dimensional space is given via the representer theorem (Wahba, 1990) as a sum over kernel functions associated to the data points. Thus kernel based methods can be seen as working in the data space. In our approach we work in the function space induced by the smoothing operator $\mathcal{S}$ and discretize the function space by means of sparse grids, as explained in more detail in the following. Note beforehand that any discretization involves additional regularization by projection (Natterer, 1977) and that there is an interplay between regularization by projection and Tikhonov-regularization, see (Binder et al., 2002).

To be precise we use sparse grids (Zenger, 1991), which are based on a hierarchical subspace splitting and a suitable limited tensor product construction. We apply this approach in the form of the combination technique (Griebel et al., 1992) to approximate functions $f \in V$. We discretize and solve the problem (5) on a suitable sequence of small anisotropic grids $\Omega_{\underline{l}} = \Omega_{l_1,\ldots,l_d}$ with uniform mesh sizes $h_t = 2^{-l_t}$ in the t-th coordinate direction. For ease of presentation we assume the domain $[0,1]^d$ here and in the following.

A finite element approach with piecewise d-linear functions

$$\phi_{\underline{l},\underline{j}}(\underline{x}) := \prod_{t=1}^{d} \phi_{l_t,j_t}(x_t)$$

on each grid $\Omega_{\underline{l}}$, where the one-dimensional basis functions $\phi_{l,j}(x)$ are defined as the so-called hat functions

$$\phi_{l,j}(x) = \begin{cases} 1 - |\frac{x}{h_l} - j|, & x \in [(j-1)h_l, (j+1)h_l] \\ 0, & \text{otherwise,} \end{cases}$$

now gives the representation

$$f_{\underline{l}}(\underline{x}) = \sum_{j_1=0}^{2^{l_1}} \dots \sum_{j_d=0}^{2^{l_d}} \alpha_{\underline{l},\underline{j}} \phi_{\underline{l},\underline{j}}(\underline{x}).$$

Each d-linear function $\phi_{\underline{l},\underline{j}}(\underline{x})$ is one at grid point $\underline{j}$ and zero at all other points of grid $\Omega_{\underline{l}}$.

The variational procedure (2) - (5) now results in the discrete system

$$\left(\mathcal{B}_{m_l,\underline{l}}^{\top}\mathcal{B}_{m_l,\underline{l}} + \lambda_A m_l \cdot \mathcal{C}_{\underline{l}} + \frac{\lambda_I m_l}{m^2}\mathcal{B}_{\underline{l}}^{\top}\mathcal{L}\mathcal{B}_{\underline{l}}\right)\alpha_{\underline{l}} = \mathcal{B}_{m_l,\underline{l}}^{\top} y,$$

Note that the matrices on the left hand side can be stored in one $N \times N$ matrix, where $N = \Pi_{t=1}^{d}(2^{l_t}+1)$, and on the right hand side the evaluation of the matrix is only needed once. We then solve these problems by a feasible method. Due to the use of local basis functions the combined matrix has a sparse structure as long as the graph Laplacian $\mathcal{L}$ is sparse, therefore we can use an iterative method, currently a diagonally preconditioned conjugate gradient algorithm. But also multigrid approaches or the algebraic multigrid method are in principle possible to achieve iteration numbers independent of the mesh size, resulting in an $O(N)$ method.

We now in particular consider all grids $\Omega_{\underline{l}}$ with

$$|\underline{l}|_1 := l_1 + \dots + l_d = n - q, \quad q = 0,..,d-1, \quad l_t \geq 0,$$

set up and solve the associated problems (5) and linearly combine the resulting discrete solutions $f_{\underline{l}}(\underline{x})$ from the partial grids $\Omega_{\underline{l}}$ according to the combination formula (Griebel et al., 1992)

$$f_n^c(\underline{x}) := \sum_{q=0}^{d-1}(-1)^q \binom{d-1}{q} \sum_{|\underline{l}|_1 = n-q} f_{\underline{l}}(\underline{x}). \qquad (7)$$

Note the varying sign of the fore-factor, which "offsets" the fact that some sparse grid points occur several times within the combination technique.

For the two-dimensional case, we display the grids needed in the combination formula of level 4 in Figure 1 and give the resulting sparse grid.

The resulting function f_n^c lives in the sparse grid space

$$V_n^s := \bigcup_{\substack{|\underline{l}|_1 = n-q \\ q=0,...,d-1 \quad l_t \geq 0}} V_{\underline{l}},$$

where

$$V_{\underline{l}} \ := \ \text{span}\{\phi_{\underline{l},\underline{j}}, j_t = 0,...,2^{l_t}, t = 1,...,d\}, \qquad (8)$$

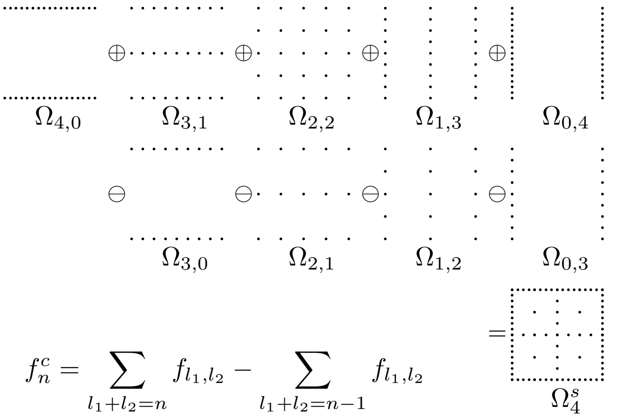

Figure 1. Grids involved for the combination technique of level $n = 4$ in two dimensions

is the space of piecewise d-linear functions on grid $\Omega_{\underline{l}}$. The V_n^s space has dimension of order $O(h_n^{-1}(\log(h_n^{-1}))^{d-1})$ in contrast to $O(h_n^d)$ for conventional grid based approaches. It is spanned by a piecewise d-linear hierarchical tensor product basis. Note that the summation of the discrete functions from different spaces $V_{\underline{l}}$ in (7) involves d-linear interpolation which resembles just the transformation to a representation in the hierarchical basis, see (Bungartz & Griebel, 2004).

Note that we never explicitly assemble the function f_n^c but keep instead the solutions $f_{\underline{l}}$ which arise in the combination formula. Therefore, if we now want to evaluate a newly given set of data points $\{\tilde{\underline{x}}_i\}_{i=1}^{m_n}$ by

$$\tilde{y}_i := f_n^c(\tilde{\underline{x}}_i), \quad i = 1, ..., m_n$$

we just form the combination of the associated values for $f_{\underline{l}}$ according to (7).

For second order elliptic PDE model problems it was proven that the combination solution f_n^c is almost as accurate as the standard full grid solution f_n, i.e., the discretization error satisfies

$$||e_n^{(c)}||_{L_p} := ||f - f_n^{(c)}||_{L_p} = O(h_n^2 \log(h_n^{-1})^{d-1})$$

provided that a slightly stronger smoothness requirement holds on f than for results with the full grid approach. One needs the seminorm

$$|f|_\infty := \left|\left| \frac{\partial^{2d} f}{\prod_{j=1}^{d} \partial x_j^2} \right|\right|_\infty$$

to be bounded. Furthermore, a series expansion of the error is necessary for formal convergence proofs of the combination technique for general problems. Its existence was shown for PDE model problems in (Bungartz et al., 1994), see also (Pflaum & Zhou, 1999) for an alternative convergence proof.

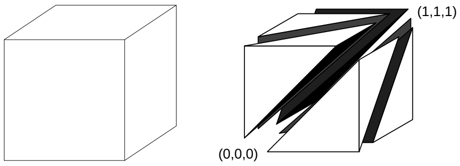

Figure 2. A simplicial discretization divides each rectangular block formed by grid points into d! simplices.

Note that the combination technique is only one of the various methods to solve problems on sparse grids. There exist also finite difference and Galerkin finite element approaches which work directly in the hierarchical product basis on the sparse grid, see (Bungartz & Griebel, 2004) for detailed references. But the combination technique is conceptually much simpler and easier to implement. Moreover, it allows to reuse standard solvers for its different subproblems.

4. Complexity

During the computation of the sparse grid solution we have to deal with $O(d \cdot n^{d-1})$ problems of size $\dim(V_{\underline{l}}) = O(2^{d-1} \cdot h_n^{-1}) = O(2^{d-1} \cdot 2^n)$. All these problems can be solved independently, which allows for a straightforward parallelization, for details see (Garcke et al., 2003). Note that the term 2^{d-1} in the above order complexity of $\dim(V_{\underline{l}})$ limits our approach in the number of attributes.

The original derivation of the combination technique is based on d-linear basis functions stemming from a tensor product approach. This way 2^d basis functions, associated to the nodes of a finite element cube, are non-zero for each data point inside a cube. In (Garcke & Griebel, 2002) linear basis functions for a simplicial discretization are used for the combination technique instead. With a simplicial discretization, see Figure 2, only $d + 1$ basis functions which are associated to the vertices of each simplex have to be evaluated for each data point. This reduces the number of operations needed for the processing of one data point during the computation of the entries of $\mathcal{B}_{m_l}^{\top}\mathcal{B}_{m_l}$ in (5) from costs which are exponential in d to costs which are only quadratic in d.

The situation is similar for the computation of the matrix entries which correspond to the additional regularization term

$$(\mathcal{B}^{\top}\mathcal{L}\mathcal{B})_{i,j} = \sum_{k,l=1}^{m} \varphi_i(\underline{x}_k) \cdot \mathcal{L}_{k,l} \cdot \varphi_j(\underline{x}_l).$$

For each data point $\underline{x}_k$ the values of all basis functions φ_i which are non-zero on it have to be computed. Furthermore, all basis functions φ_j which are non-zero on data points $\underline{x}_l$ which are connected to $\underline{x}_k$, i.e., with $\mathcal{L}_{k,l} \neq 0$, have to be evaluated. In the case of a sparse graph with degree k the work count for one data point is thus $O(k \cdot (d+1)^2)$ using the simplicial discretization. This results in a complexity of the order $O(k \cdot (d+1)^2 \cdot m)$ to compute all non-zero entries of $\mathcal{B}^{\top}\mathcal{L}\mathcal{B}$. As already mentioned, in this case the combined matrix $\mathcal{B}_{m_l}^{\top}\mathcal{B}_{m_l} + \lambda_A m_l \cdot \mathcal{C} + \frac{\lambda_I m_l}{m^2}\mathcal{B}^{\top}\mathcal{L}\mathcal{B}$ is sparse. Then, the cost of the actual solution of the linear equation system (5) has order $O(N)$, provided that an appropriate multigrid method is used. The cost is independent of m in any case.

For a full graph Laplacian the complexity for the computation of $\mathcal{B}^{\top}\mathcal{L}\mathcal{B}$ would be $O((d+1)^2 \cdot m^2)$ since all data points are connected. Furthermore the combined $N \times N$ matrix is now in general full which requires a different solution strategy. In the worst case $O(N^3)$ operations are needed. This can be reduced to $O(N^{2.376})$ operations if Strassen's approach or related techniques are employed. Nevertheless, for complexity reasons, it is better to a-priori approximate the full graph by a sparse one by e.g. choosing a similarity measure and a nearest neighbor cut-off which leads directly to a sparse matrix.

The complexity of recently proposed kernel based approaches (Belkin et al., 2004; Belkin et al., 2005; Chapelle & Zien, 2005) is currently of the order $O(m^3)$ which significantly limits the amount of data that can be handled by them. Related to these algorithms (Delalleau et al., 2005) introduce a semi-supervised approach using a heuristic subset selection and an approximate training algorithm with complexity $O(n^2(m-n))$, where n is the size of the subset.

Note that for the computation of a sparse adjacency graph from the data points in a preprocessing step, spatial indexing structures such as kd-trees can be used for the efficient computation of neighbors in vector spaces. But kd-trees suffer from the curse of dimensionality and degenerate to linear searches in high dimensions, see (Weber et al., 1998) for a quantitative study. There also exist various methods to compute *approximate* nearest neighbors (Berrani et al., 2003) which avoid these disadvantages, trading accuracy for computation time. In our context such an approach is most favorable for the economical construction of a sparse adjacency graph from the data.

It can be observed in practice and verified theoretically, that with high dimensions the distances between a point and its neighbors become almost constant, so that all data are close to a hypersphere around the point. The concept of nearest neighbor is no longer

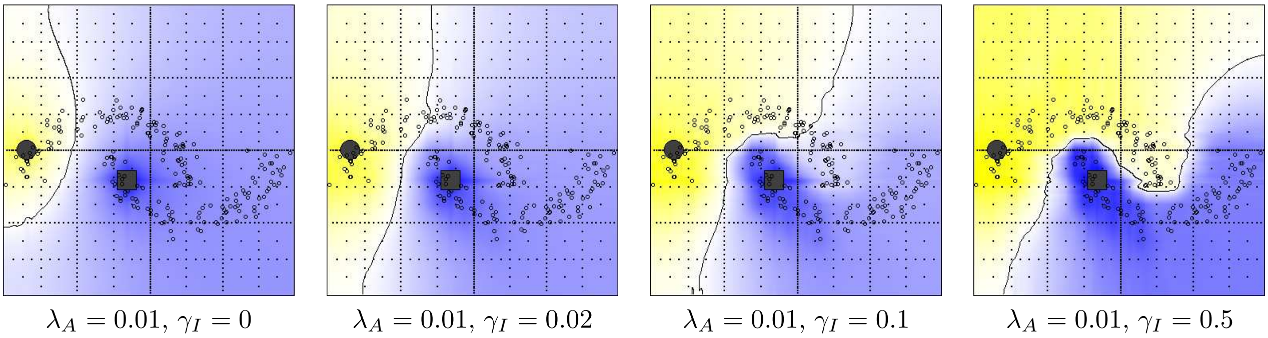

Figure 3. Two moons toy problem, 200 data points, one labeled sample for each class (the square and the circle). Sparse grid of level 8.

meaningful, see e.g. (Beyer et al., 1999). Already for 20 dimensions nearest neighbors can become unstable (Beyer et al., 1999). This is an effect of the concentration of measure phenomenon observed in higher dimensions, see e.g. (Ledoux, 2001). The applicability of the proposed geometry based approaches for truly high dimensional data has to be studied further under this aspect.

5. Examples

For the following experiments we compute the graph Laplacian for a k-nearest neighbor adjacency graph, which we symmetrize beforehand, i.e., if $\underline{x}_i$ is a k-nearest neighbor of $\underline{x}_j$, then this is also valid vice versa. We fix the number of nearest neighbors to $k = 7$ and we use binary weights in the graph. As smoothing operator we employ $S = \nabla$ due to the piecewise linear basis functions. Note that the qualitative behavior of the results is similar for slightly modified values of k and for the choice of the Euclidean distance instead of binary weights. In the following we use an intrinsic regularization parameter γ_I, where $\gamma_I := \frac{\lambda_I m_l}{(m)^2}$.

5.1. Two Moons

The two moons example is a common toy problem to visualize the behavior of semi-supervised learning algorithms in just two dimensions (Belkin et al., 2004; Bousquet et al., 2004). It contains 200 samples points with one labeled sample for each class.

In Figure 3 we show the results of our numerical experiments with the new sparse grid method for semi-supervised learning. We use a sparse grid of level 8, fix λ_A to 0.01, and vary the intrinsic regularization parameter γ_I. For the value $\gamma_I = 0$ we obtain just the standard classification problem where the geometric structure of the distribution of the labeled and unlabeled data has no effect. We now gradually turn the intrinsic regularization term on and set γ_I to $0.02, 0.1$ and 0.5. We clearly see that the two clusters get properly separated. Note that since we enforce Neumann boundary conditions, the layer separating the two classes is orthogonal to the boundary.

5.2. ndcHard data set

With the 10-dimensional synthetic ndcHard[1] data set we compare our numerical results with the approach of (Belkin et al., 2004). We use a subset for training of 2000 data with 200 labeled data and evaluate on the original test set of 20000 points. For the approach of (Belkin et al., 2004) we optimize the parameters (width of the RBF kernel, γ_I and λ_A) on the test set and achieve an error rate of 26.6%. Note that using a linear kernel and all 2-million labeled data an error rate of 30.1% was achieved in (Fung & Mangasarian, 2001) using the so-called proximal SVM, which is a variant of regularized least squares classification. For the supervised sparse grid approach using all data a 15.1% classification error is reported (Garcke & Griebel, 2002).

For our sparse grid approach we pick the parameters γ_I and λ_A using the labeled data of the training set and 5-fold cross-validation. Note that here and in the following we only use the grid Ω_0 with 2^d grid points, i.e., $n = 0$. A higher level sparse grid combination technique does not yield better results, this somewhat surprising result was observed before for higher dimensional supervised cases in (Garcke & Griebel, 2002; Garcke, 2004) and is being further investigated. Using the subset of 2000 data, i.e., 200 labeled and 1800 unlabeled data as above, we achieve a classification error of 28.9% on the test set. Using 20000 training data and the same 200 labeled data we get 27.8% classification error and if we go to 200 000 training data the clas-

[1] `www.cs.wisc.edu/dmi/svm/ndc/`, see (Fung & Mangasarian, 2001) for ndcHard

data (labeled)	AUC test	test error	time (sec.)
200 (200)	0.691	36.9 %	1.2
2000 (200)	0.695	36.6 %	1.9
20000 (200)	0.700	36.1 %	8.7
200000 (200)	0.703	36.2 %	73.1
2000 (2000)	0.754	30.6 %	1.3
20000 (2000)	0.755	30.6 %	8.8
200000 (2000)	0.764	29.7 %	73.7

Table 1. Results for the forest covertype data set. The time shown is for the computation of one parameter combination without the time for the computation of the graph Laplacian $\mathcal{L}$, which only needs to be computed once for all parameters.

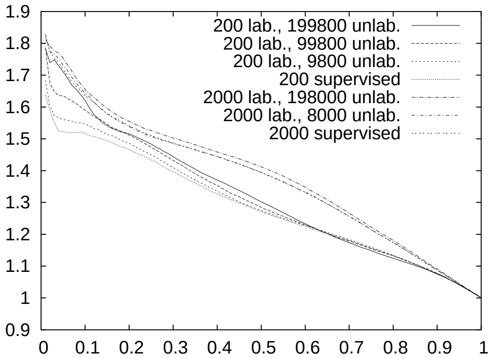

Figure 4. Lift-chart for the forest covertype data set

sifcation error reduces to 27.3%. The computation for the small size of 2000 data takes about 2 seconds for our approach in comparison to the approach of (Belkin et al., 2004) which takes about 5 seconds. For larger data set sizes the latter scales currently cubically (in time and memory usage) in the number of data and soon becomes intractable. Especially for $m > 15000$ the resulting kernel matrix cannot be stored anymore on current workstations, whereas our approach scales linearly in m for time and required memory.

5.3. Forest Covertype

We here use a subset of the Forest Covertype[2] data set, utilizing its 10 numerical attributes. Furthermore, we make the classification problem binary by learning class 2, roughly half of the data, against the rest.

We take a set of 200 000 points for training and parameter fitting. We learn with 200, 2000, 20 000, and 200 000 data where only the first 200 are used as labeled, the rest as unlabeled data. We solve for a range of λ_A and γ_I and pick good parameters with 5-fold cross-validation over the labeled data.

To be able to compare over different sizes of unlabeled data we now compute results on a separate test set of 200 000 points, but use the data from the training set as unlabeled data, the experimental results are given in Table 1. Using the unlabeled data increases the area under ROC curve (AUC) and reduces the test error for about 2%. The Lift-chart presented in Figure 4, which gives the ratio between the result predicted by our model and the result using no model, shows a more significant improvement for the high ranked data.

Using more labeled data, i.e., 2000 instead of 200, improves the results substantially. Here the use of unlabeled data gives results with even less change than before, furthermore the gain now seems to be equally distributed over the ranking of the data.

We also give the computational time for computing the result for one parameter set in Table 1. Here we do not include the time for computing the graph Laplacian, which we compute once with an outside tool, store and load for each computation of $\mathcal{B}^\top \mathcal{L} \mathcal{B}$. We see that the method scales only linear in the number of data (labeled and unlabeled).

5.4. Data-Mining-Cup 2000

We now apply our method to a real life data set which was used for a data mining competition[3]. The task is to identify the most likely responders to a direct mailing campaign, the aim is to maximize the profit of the marketing activity. The cost for a non-responder is 12 DEM, the average profit for a responder is 185 DEM. The data set consists of 10,000 training data with 40 attributes. The evaluation set contains 34,820 data points.

First, we perform a PCA on the training data to reduce the number of attributes to the 18 most important ones for which still 93.5% of the variance is captured. The same transformation is applied to the evaluation set. We then split the training data, using 6000 points for the actual learning and 4000 points as a test data set for the model selection of the method. Our model involves the free parameters λ_A, γ_I and *cut-off* %. The latter is the top percentage of the data points $\underline{x}_i$ sorted in descending order according to the value of $f(\underline{x}_i)$, which are then chosen for the direct mailing campaign. Again it turned out that a sparse grid discretization of the coarsest level, i.e., level $n = 0$, was already sufficient.

[2] UCI KDD archive, `http://kdd.ics.uci.edu`

[3] `http://www.data-mining-cup.de/2000/`

The winner of the original Data-Mining-Cup 2000 competition achieved a profit of 67,038 DEM on the evaluation set, non-competing data mining specialists achieved 84,995 DEM. Using a sparse grid approach for conventional supervised classification, i.e., $\gamma_I = 0$, we achieve 87,705 DEM (Garcke, 2004).

We now additionally use the evaluation data set in the semi-supervised ansatz. To this end the graph Laplacian is computed for all 44,820 data points, but the parameter selection still takes place over the same 6000:4000 split of the training data. We now achieve a profit of 93,812 DEM on the evaluation set. This is a gain of 7% in comparison to our supervised results and a gain of 10% in comparison to the results of the data mining specialists.

As a consistency check we also computed the optimal results, i.e., we picked the maximum profit attained on the evaluation set over a wide range of parameters, instead of using the proper model parameter selection on the training data as above. Again the maximum profit achieved with the semi-supervised approach is higher than with the supervised one.

6. Conclusions

In this article we presented a sparse grid method for semi-supervised learning problems using a graph-based approach for the unlabeled data. In contrast to most kernel based approaches which show a $O(m^3)$ complexity, its complexity only scales linear in the number of data. This allows to treat huge data sets which involve millions of points.

The number of attributes our method can handle is limited since the sparse grid based approach involves at least 2^d grid points. But the method can be improved to deal with higher dimensional problems as well. To this end, generalized sparse grids (Gerstner & Griebel, 1998) or dimension-adaptive sparse grids (Gerstner & Griebel, 2003; Hegland, 2003) can be used. For the grid selection in the dimension-adaptive approach, suitable refinement criteria still have to be developed, though.

Also the incorporation of other intrinsic regularization operators from manifold learning into our approach, coming from Isomap, LLE, or SDE, has to be investigated in more detail. Finally, the relation to the concept of diffusion maps and distances (Coifman & Lafon, 2004), which are constructed from Markov processes defined on data sets, is surely worth further studies.

Acknowledgments

JG acknowledges the support of the Australian Research Council. We would like to thank Olivier Chapelle for helpful comments and discussions.

References

Belkin, M., Niyogi, P., & Sindhwani, V. (2004). *Manifold regularization: A geometric framework for learning from examples* (Technical Report TR-2004-06). Univ. Chicago, Dept. of Computer Science.

Belkin, M., Niyogi, P., & Sindhwani, V. (2005). On manifold regularization. *Artificial Intelligence and Statistics AISTATS'05* (pp. 17–24).

Bengio, Y., Delalleau, O., Roux, N. L., Paiement, J.-F., Vincentand, P., & Ouimet, M. (2004). Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Comp.*, *16*, 2197–2219.

Berrani, S.-A., Amsaleg, L., & Gros, P. (2003). Approximate searches: k-neighbors + precision. *CIKM* (pp. 24–31).

Beyer, K. S., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "nearest neighbor" meaningful? *ICDT* (pp. 217–235). Springer.

Binder, T., Blank, L., Dahmen, W., & Marquardt, W. (2002). On the regularization of dynamic data reconciliation problems. *J. Proc. Cont.*, *12*, 557–567.

Bousquet, O., Chapelle, O., & Hein, M. (2004). Measure based regularization. *Advances in Neural Information Processing Systems*, *16*.

Bungartz, H.-J., & Griebel, M. (2004). Sparse grids. *Acta Numerica*, *13*, 147–269.

Bungartz, H.-J., Griebel, M., Röschke, D., & Zenger, C. (1994). Pointwise convergence of the combination technique for the Laplace equation. *East-West J. Numer. Math.*, *2*, 21–45.

Chapelle, O., & Zien, A. (2005). Semi-supervised classification by low density separation. *Artificial Intelligence and Statistics AISTATS'05* (pp. 57–64).

Coifman, R., & Lafon, S. (2004). Diffusion maps. Preprint, Yale Univ., Dept. of Mathematics.

Delalleau, O., Bengio, Y., & Roux, N. L. (2005). Efficient non-parametric function induction in semi-supervised learning. *Artificial Intelligence and Statistics AISTATS'05* (pp. 96–103).

Fung, G., & Mangasarian, O. (2001). Proximal support vector machine classifiers. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 77–86).

Garcke, J. (2004). *Maschinelles Lernen durch Funktionsrekonstruktion mit verallgemeinerten dünnen Gittern*. Doktorarbeit, Institut für Numerische Simulation, Rheinische Friedrich-Wilhelms-Universität Universität Bonn.

Garcke, J., & Griebel, M. (2002). Classification with sparse grids using simplicial basis functions. *Intelligent Data Analysis*, *6*, 483–502. (shortened version appeared in *KDD 2001*, (pp. 87-96)).

Garcke, J., Griebel, M., & Thess, M. (2001). Data mining with sparse grids. *Computing*, *67*, 225–253.

Garcke, J., Hegland, M., & Nielsen, O. (2003). Parallelisation of sparse grids for large scale data analysis. *Proc. ICCS 2003, Melbourne, Australia* (pp. 683–692). Springer.

Gerstner, T., & Griebel, M. (1998). Numerical integration using sparse grids. *Numer. Algorithms*, *18*, 209–232.

Gerstner, T., & Griebel, M. (2003). Dimension–adaptive tensor–product quadrature. *Computing*, *71*, 65–87.

Girosi, F., Jones, M., & Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation*, *7*, 219–265.

Griebel, M., Schneider, M., & Zenger, C. (1992). A combination technique for the solution of sparse grid problems. *Iterative Methods in Linear Algebra* (pp. 263–281). IMACS, Elsevier, North Holland.

Ham, J., Lee, D. D., Mika, S., & Schölkopf, B. (2004). A kernel view of the dimensionality reduction of manifolds. *ICML '04*. Banff, Canada: ACM Press.

Hegland, M. (2003). Adaptive sparse grids. *Proc. of CTAC-2001* (pp. C335–C353).

Ledoux, M. (2001). *The concentration of measure phenomenon*, vol. 89 of *Mathematical Surveys and Monographs*. AMS.

McLachlan, G. J. (1992). *Discriminant analysis and statistical pattern recognition.* John Wiley & Sons, New-York.

Natterer, F. (1977). Regularisierung schlecht gestellter Probleme durch Projektionsverfahren. *Numer. Math.*, *28*, 329–341.

Pflaum, C., & Zhou, A. (1999). Error analysis of the combination technique. *Numer. Mathematik*, *84*, 327–350.

Seeger, M. (2000). *Learning with labeled and unlabeled data* (Technical Report). Institute for ANC, Edinburgh, UK. `http://www.cs.berkeley.edu/~mseeger/papers/review.pdf`.

Smola, A. J., & Kondor, R. (2003). Kernels and regularization on graphs. *COLT* (pp. 144–158).

Smolyak, S. A. (1963). Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR*, *148*, 1042–1043. Russian, Engl. Transl.: Soviet Math. Dokl. 4:240–243, 1963.

Tikhonov, A. N., & Arsenin, V. A. (1977). *Solutions of ill-posed problems.* Washington D.C.: W.H. Winston.

Vapnik, V. N. (1998). *Statistical learning theory.* Wiley.

Wahba, G. (1990). *Spline models for observational data*, vol. 59 of *Series in Applied Mathematics.* Philadelphia: SIAM.

Weber, R., Schek, H.-J., & Blott, S. (1998). A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. *VLDB '98* (pp. 194–205). Morgan Kaufmann Publishers.

Weinberger, K. Q., & Saul, L. K. (2004). Unsupervised learning of image manifolds by semidefinite programming. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-04)* (pp. 988–995). Washington D.C.

Weinberger, K. Q., Sha, F., & Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. *ICML* (pp. 839–846). Banff, Canada.

Zenger, C. (1991). Sparse grids. *Parallel Algorithms for Partial Differential Equations, Proc. 6th GAMM-Seminar, Kiel, 1990* (pp. 241–251). Vieweg-Verlag.