Graph-based transformation manifolds for invariant pattern recognition with kernel methods

Alexei Pozdnoukhov and Samy Bengio
IDIAP Research Institute
Swiss Federal Institute of Technology
Martigny, CH-1920, Switzerland
{pozd, bengio}@idiap.ch

Abstract

We present here an approach for applying the technique of modeling data transformation manifolds for invariant learning with kernel methods. The approach is based on building a kernel function on the graph modeling the invariant manifold. It provides a way for taking into account nearly arbitrary transformations of the input samples. The approach is verified experimentally on the task of optical character recognition, providing state-of-the-art performance on harder problem settings.

1. Introduction

The idea of using the inner geometric structure of the data for better data processing algorithms is of increasing attention in Machine Learning. This trend arises from unsupervised methods such as clustering and dimensionality reduction techniques. A clever use of geometrical structure should improve the performance of any learning algorithm. In particular, semi-supervised methods are under rapid development recently. These methods exploit unlabeled data, i.e. those data samples which consist of the input values only, while the desired output value is unknown. In fact, most real-life learning problems are actually semi-supervised. For example, this is the situation when a huge amount of images are available, but only a part of them is annotated, i.e. labeled.

The information one obtains from the unlabeled part of the dataset can be of different nature. A reasonable assumption to make is the following. Assume the data lies on some lower-dimensional manifold in the original input space. Using some properties of the manifold, data analysis methods can be improved, as shown in recent developments devoted to the exploration of such an approach (see [1] and references therein for instance). Furthermore, given the explosive growth of interest in the field of kernel methods, nonparametric data-dependent kernels which reflect the inner geometry of the data are of particular interest. A general approach was recently proposed in [5].

In this paper, we apply this framework for another important problem, namely invariant learning. The methodology developed for semi-supervised learning is adapted to model the manifolds induced by the desired invariant transformations. Afterward, a kernel classifier is applied to the task. The kernel is constructed in a way to produce smooth decision functions on the modeled invariant manifolds, therefore preserving the class membership on these manifolds. We introduce manifold learning in Section 2, present the way to adapt this framework to invariant learning in Section 3, and introduce the corresponding kernels in Section 4. We provide some practical issues and experimental results on a real Optical Character Recognition (OCR) task in Section 5, and conclude the paper in Section 6.

2. Learning on Manifolds

The supervised learner aims at estimating the input-output relationship (dependency or function) $f(\mathbf{x})$ by using a training data set $\{\mathbf{x}_i, y_i\}$, $i = 1, \dots, N$ where the inputs $\mathbf{x}$ are n-dimensional vectors and the labels (or system responses) y are continuous values for regression tasks and discrete (e.g., boolean) for classification problems.

However, the situation where some labeled patterns are provided together with unlabeled ones, arises frequently. This is called *semi-supervised learning*.

Recently several approaches to semi-supervised learning were proposed. Low Density Separation (LDS) algorithms [2], Transductive SVMs, Graph and Gradient Transductive SVMs [3], and a group of Manifold Learning methods [1] are the core of those recently developed techniques.

Here we give a basic idea for the last group of methods, namely, Manifold Learning. The so-called *manifold* *assumption* is accepted in this framework. This implies that data actually belong to some lower dimensional manifold in high dimensional input space. Thus, it is reasonable to build models which exploit regularization on the manifolds.

Usually the only information about the manifold is the finite set of (unlabeled) samples, $\{\mathbf{x}_i\}$, $i = N+1, \ldots, M$. Thus, the model has to be smooth (regularized) on the corresponding graph, whose nodes are data samples and edges are constructed to preserve the geometrical properties (geodesic distances) on the graph. Let an edge connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ have some weight w_{ij}; zero value means that the nodes are not connected.

A nice property here arises from the notion of graph Laplacian. It is defined as

$$\mathcal{L} = D - W \tag{1}$$

where W is the matrix with elements $w_{ij} = \exp(-\delta(\mathbf{x}_i - \mathbf{x}_j)^2)$, and D is a diagonal matrix with $d_{ii} = \sum_j w_{ij}$.

It can be shown that eigenvectors of $\mathcal{L}$ provide a natural basis on the graph, giving rise to regularization by penalizing the complexity. This can be done by minimizing the norm in the space of functions defined on graphs. Please refer to [1] for details and solid justification behind this technique.

3. Invariant Manifolds

One of the well-known approaches to invariant learning is the Tangent Distance method [8]. It proposes to replace the Euclidean distance between data samples with a distance between the corresponding linear tangent manifolds defined by tangent vectors of the desired invariance transformation. This method was successfully applied to Optical Character Recognition (OCR) tasks. A restriction of these methods is that they are suited for distance-based kernels only. The proposed method, on the other hand, does not suffer this restriction.

The decision function, which is smooth on the corresponding manifold, provides invariant classification. This smoothness guarantees that the decision for class membership is unchanged as one considers samples from the invariant manifold (i.e. transformed samples).

The direct approach to enforce smoothness in the direction of tangent vectors is considered in [4]. This method, however, leads to complicated optimization and appeared to be impractical in real-life tasks.

3.1. Graph-based Invariant Manifolds

Given a training sample $\mathbf{x}_i$, consider a set of corresponding virtual samples, generated by applying the desired (and, virtually, arbitrary) transformation $G(\mathbf{x}, \alpha)$:

$$\{\mathbf{x}_i^k\} = G(\mathbf{x}_i, \alpha),\ \alpha \in \Lambda, \tag{2}$$

where α is a vector of parameters from some finite set Λ (such as a set of rotation angles). Then, a graph is built for every training sample by connecting and setting weights for the nodes $\mathbf{x}_i^k$ sharing the same original sample $\mathbf{x}_i$. The weights w_{ij} are set to $\exp(-\delta(\mathbf{x}_i - \mathbf{x}_j)^2)$ if nodes are connected, zero otherwise. Considering the graph-based manifold models and enforcing smoothness of the model on the graph, we constrain it to be invariant to the transformation which generated the manifold.

Next, we introduce a kernel, adapted from [5], to apply a kernel classifier such as Support Vector Machine to graph based manifolds.

4. Kernel Methods

A semi-positive definite function which satisfies Mercer conditions is called a kernel. This implies that it corresponds to a dot product in some space (Reproducing Kernel Hilbert Space, RKHS), sometimes referred to as a feature space. Generally, given a (linear) algorithm, which includes data samples in the form of dot products only, one can obtain a (non-linear) kernel version of it by substituting the dot products with kernel functions [6]. The choice of the kernel function is an open issue. Hereafter, we briefly present a method, adapted from [5] for constructing non-parametric semi-supervised kernels which deals with graph-modeled invariant manifolds.

We will follow the notation of [5]. Given data points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, and some RKHS H, consider the evaluation map $S(\mathbf{f}) = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$; $S : H \to \mathbf{R}^n$. The semi-norm on $\mathbf{R}^n$ is given by a symmetric semi-definite matrix $\mathbf{M}$,

$$\|S(\mathbf{f})\|^2 = \mathbf{f}^T \mathbf{M} \mathbf{f}, \tag{3}$$

where we denoted $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$ and T means transpose.

The exact explicit form of the corresponding reproducing kernel $\tilde{k}(\mathbf{x}, \mathbf{x}')$ was derived in [5] and is given by:

$$\tilde{k}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - k_{\mathbf{x}}^T (\mathbf{I} + \mathbf{M}\mathbf{K})^{-1} \mathbf{M} k_{\mathbf{x}'} \tag{4}$$

where $\mathbf{K}$ is the complete kernel matrix of $k(\cdot, \cdot)$, $k_{\mathbf{x}}$ represents one row of $\mathbf{K}$ and $\mathbf{I}$ is the identity matrix. In the presence of unlabeled data, the choice of $\mathbf{M}$ implements the smoothness assumption with respect to its geometric structure. As shown in [1], this is achieved by taking $\mathbf{M} = \gamma\mathcal{L}$, $\mathcal{L}$ being the Laplacian matrix of the graph built on unlabeled samples, and γ a regularization parameter which defines the extent of kernel deformation. By setting γ=0 one obtains the original kernel, as it is clearly seen with (4), and no invariance information is used in the model.

This kernel can be plugged into any classification algorithm. We will use the widely known standard form of soft margin SVMs [9]. This method provides a classifier of the form:

$$f(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^{N} \alpha_i \tilde{k}(\mathbf{x}_i, \mathbf{x}) + b\right), \qquad (5)$$

where the weights α_i are obtained from solving a QP optimization problem.

The advantage of the method in computational speed is that the size of this QP is the same as the size of the original optimization problem. In virtual samples methods, however, the training set size and, correspondingly, the optimization problem dimension is increased [7]. However, each kernel computation becomes more expensive.

5. Experiments

The experiments described below deal with global rotational invariance as an example. We start with an illustration of modeling the invariant manifold and kernel construction with graphs, using character images. Finally, we test our approach on a real-world handwritten digits dataset, commonly used in machine learning for benchmarking different algorithms and known as USPS digits.

5.1. Practical Issues

We first start with the discussion on some issues which arise while implementing the described method.

Manifold modeling. There are two basics to take into account while constructing the graph: the smoothness of the transformation, which is modeled as locally linear between the adjacent nodes; and the number of nodes, which has to be sufficient to model the manifold reliably.

A workaround is to build the graph by generating a sufficient number of virtual samples as nodes, and connect the K nearest neighbors. The rest of the procedure remains unchanged. This approach will capture some intra-class similarity in the data. However, the influence of noise in data such as mislabeling or outliers will probably be increased.

Choice of parameters. There are several parameters which influence the final classification model. A general way to tune them would be to carry out cross-validation on the training data. However, this is complicated since the parameter space is of high dimension. Here we consider several heuristics to simplify the choice.

There are two groups of parameters. The first group corresponds to the manifold-modeling graph. These are δ and γ. The parameter δ is taken such that the bandwidth of RBF function in graph Laplacian is equal to the average distance between the graph nodes. The influence of the γ parameter is explored experimentally below.

The second group contains the hyper-parameters of the learning algorithm, which include the trade-off parameter C of the SVM and the kernel parameters. In this paper, we used the kernel described in Section 4, using the standard Gaussian RBF kernel with bandwidth σ as a base kernel. These parameters are tuned using the standard cross-validation technique on the training data.

5.2. Global Rotation

The purpose of this section is to provide empirical evidence for the method, and particularly manifold modeling. We consider the problem of kernel construction for character image classification.

Figure 1 presents a contour plot of the kernel function centered at image A. Since the basic kernel is an RBF one, this value can be considered as a measure of similarity. The angle at the polar plane corresponds to the rotation angle of the image, and radius corresponds to the lag of vertical translation of the image before rotation. Black dots are the unlabeled rotated images used as graph nodes.

Figure 1. Kernel centered at image 'A'.

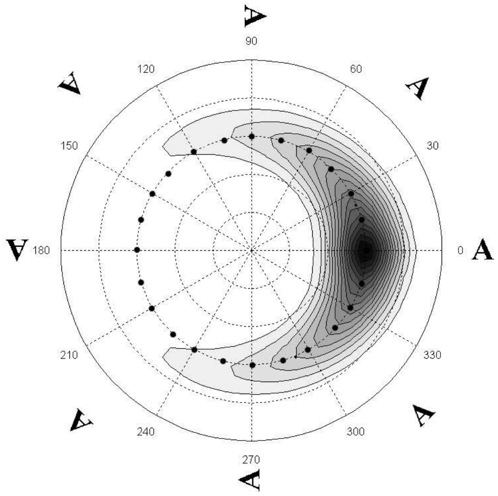


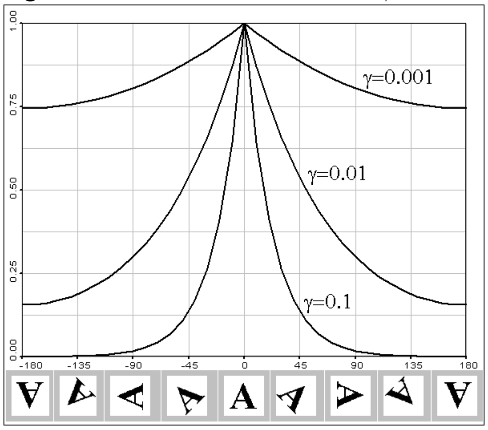
Figure 2 presents the value of kernel function centered at the original image of the letter, as a function of the angle the image was rotated. The values were normalized. As one can see, the parameter γ controls the amount of invariance information introduced by virtual inputs.

5.3. USPS digits

In this section we carry out experiments on a widely used benchmark: the USPS dataset of handwritten digits. It consists of 7291 training and 2007 testing samples. These are grey scaled images of 16x16 pixels.

The data were modified by applying rotation to each character image. The rotation angle is random in the range 0-360 degrees (see Figure 3). In this modified setting, this classification problem becomes extremely difficult. The proposed algorithm was applied to binary classification.

Figure 2. Kernels for different γ values.



The task was to classify digits "$0-4$" against "$5-9$". Graph nodes were constructed by consequently rotating the original images on 15 degrees. This results in 23 virtual samples per image. The training set was split into 36 subsets of 200 samples. The results averaged over splits are presented in Table 1. Standard SVM obviously fails to classify the rotated digits.

Figure 3. Some rotated USPS digits.

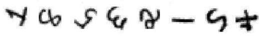


Table 1. Testing errors on USPS data.

	Algorithm	Testing Error, %	Time, s
	SVM (unrotated data)	12.0	8.5
	SVM (rotated data)	34.0	8.5
	VSV SVM	13.1	2160
	Graph SVM	12.5	480

6. Discussion and Conclusions

The universal approach to invariant learning is the virtual samples approach. Given unlimited computational resources and an ability to add enough virtual samples to the training set, all the information on invariances can be learned directly from data.

An alternative to this approach, proposed in this paper, consists in modeling the invariant manifolds instead in order to obtain improvements in training time without lack of precision. We adapted the recently developed method to model manifolds defined by samples and we built a kernel classifier which enforced smoothness on these manifolds. We thus obtained an invariance property of the classifier.

The method provides a way to model nearly arbitrary invariances. It requires additional computations to build the kernel. At the same time, the size of the optimization problem is unchanged. The amount of invariant information used in the algorithm can be tuned by the choice of parameter γ.

Promising classification performance on a real OCR task was observed. Other applications, such as dealing with specific invariances or matching problems are of particular interest for further developments.

7. Acknowledgments

This research has been supported by the IST Programme of the EC, PASCAL, IST-2002-506778, funded in part by the Swiss OFES. It was also partially funded by the Swiss NCCR project (IM)2.

References

[1] Belkin, M. Problems of Learning on Manifolds. Ph.D. dissertation, 2003.

[2] Chapelle, O., Zien, A. Semi-supervised Classification by Low Density Separation. In Proc. of AI&Statistics, 2005.

[3] Joachims, T. Transductive Learning via Specral Graph Partitioning In Proc. of ICML, 2003.

[4] Chapelle, O. and Scholkopf, B. Incorporating invariances in nonlinear SVMs. In: T.G. Dietterich, S. Becker and Z. Ghahramani, (eds.), *Advances in Neural Information Processing Systems*, vol. 14, pp. 609-616. MIT Press, Cambridge, MA, USA, 2002.

[5] Sindhwani, V., Niyogi, P., Belkin, M. Beyond the Point Cloud: from Transductive to Semi-supervised Learning In Proc. of ICML'05, Bonn, Germany.

[6] Scholkopf, B., Smola, A.J. Learning with Kernels. MIT press, Cambridge, MA, 2002.

[7] B. Scholkopf, C. Burges, and V. Vapnik. Incorporating invariances in support vector learning machines. In C. von der Malsburg, W. von Seelen, J. C. Vorbruggen, and B. Sendhoff, (eds.), ICANN'96, pp. 47-52, Berlin, 1996.

[8] P. Simard, Y. LeCun, J. Denker, B. Victorri. Transformation invariance in pattern recognition, tangent distance and tangent propagation. In G. Orr and K. Muller, (eds.), *Neural Networks: Tricks of the trade.* Springer, 1998.

[9] V. Vapnik. Statistical Learning Theory. J.Wiley, NY, 1998.