Graph-based Classification for Multiple Observations of Transformed Patterns

Effrosyni Kokiopoulou, Stefanos Pirillos and Pascal Frossard*
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Laboratory - LTS4
CH - 1015 Lausanne, Switzerland
{effrosyni.kokiopoulou,stefanos.pirillos,pascal.frossard}@epfl.ch

Abstract

We consider the problem of classification when multiple observations of a pattern are available, possibly under different transformations. We view this problem as a special case of semi-supervised learning where all the unlabelled samples belong to the same unknown class. We build on graph-based methods for semi-supervised learning and we optimize the graph construction in order to exploit the special structure of the problem. In particular, we assume that the optimal adjacency matrix is a linear combination of all possible class-conditional ideal adjacency matrices. We formulate the construction of the optimal adjacency matrix as a linear program (LP) on the weights of the linear combination. We provide experimental results that show the effectiveness and the validity of the proposed methodology.

1 Introduction

Recent years have witnessed a dramatic growth of multimedia data that need to be effectively processed and analyzed in order to cover the various information needs of diverse users and applications. It commonly happens that multiple observations of an object have been captured at different time instants or under different geometric transformations. For instance, a moving object may be observed over a time interval by a surveillance camera or under different viewing angles by a network of vision sensors. This typically produces a large volume of multimedia content that lends itself as a valuable source of information for effective knowledge discovery and content analysis. The problem of pattern classification with multiple observations thus becomes increasingly important. Classification methods that are able to exploit the diversity of the multiple observations in order to provide increased classification accuracy, are of particular interest in this context.

In this work, we focus on the pattern classification problem with multiple observations. We further assume that each observation is produced from the same object under a certain transformation (see also [**?**] for a similar case study). This problem can be seen as a particular case of semi-supervised learning [**?**]. Semi-supervised learning refers to the type of learning where the test unlabelled data are available in the training phase; the challenge is to exploit this extra information in order to increase the classification performance. In our problem all unlabelled samples typically belong to the same unknown class. Graph-based methods *represent state-of-the-art solutions* for semi-supervised learning problems. They typically assume that the data lie on a manifold in a high dimensional space and the main idea is to build a graph which captures the geometry of this manifold. *The label propagation algorithm [**?**] is a very popular representative from this family of methods.* Although the graph-based methods have been successfully applied in various classification tasks, the problem of graph construction is however not well studied, with the exception of [**?**]. Usually, the k nearest neighbor (NN) graph is employed, where two nodes are considered as adjacent if and only if one is among the k NNs of the other. However, the k-NN graph is far from being optimal, since the Euclidean distance may be misleading and the impact of the parameter k is not well understood. Hence, the k-NN graph has trouble to capture the real data geometry. This becomes even more problematic in the presence of geometric transformations of the data samples of interest.

This paper introduces a graph construction method-

*This work has been partly supported by the Swiss National Science Foundation, under grant NCCR IM2.

ology that exploits the special structure of the classification problem with multiple observations of the same pattern, possibly under different transformations. First we observe that, when the correct class of the multiple observations is known, an ideal adjacency matrix can be constructed where nodes that share the same label are made adjacent. We call such a matrix a class-conditional adjacency matrix and we propose to build the optimal adjacency matrix of the graph by a linear combination of all possible class-conditional adjacency matrices. The weights of the linear combination are optimized by linear programming, such that the optimal matrix is as close as possible to a realistic similarity matrix defined from the data. We provide experimental results on handwritten digits databases, which show that the proposed method outperforms the traditional k-NN graph methodology.

2 Label propagation overview

We first review the basics of the label propagation algorithm [**?**] for semi-supervised learning. Assume that we are given a data set $X = \{X_l, X_u\}$, where $X_l = \{x_1, x_2, \ldots, x_l\} \subset \mathbb{R}^d$ and $X_u = \{x_{l+1}, \ldots, x_n\} \subset \mathbb{R}^d$, as well as a label set $\mathcal{L} = \{1, \ldots, c\}$, where $n = l+m$ and c is the number of classes. The samples in X_l are labelled $\{y_1, y_2, \ldots, y_l\},\ y_i \in \mathcal{L}$, and the m samples in X_u are unlabelled. Denote by $\mathcal{M}$ the set of matrices with nonnegative entries, of size $n \times c$. Notice that any matrix $M \in \mathcal{M}$ provides a labelling of the data set by applying the following rule: $y_i = \max_{j=1,\ldots,c} M_{ij}$. We denote the initial label matrix as $Y \in \mathcal{M}$ where $Y_{ij} = 1$ if x_i belongs to class j and 0 otherwise.

The label propagation algorithm first forms the k nearest neighbor (NN) graph defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the vertices $\mathcal{V}$ correspond to the data samples X. An edge $e_{ij} \in \mathcal{E}$ is drawn if and only if x_i is among the k nearest neighbors of x_j or vice versa. The edge weights $W_{ij} = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$, when $(i,j) \in \mathcal{E}$ and 0 otherwise. The W_{ij}'s are usually called Gaussian weights. The similarity matrix $S \in R^{n \times n}$ is further defined as $S = D^{-1/2} W D^{-1/2}$, where D is a diagonal matrix with entries $d_i = \sum_{j=1}^{n} W_{ij}$. Next, the algorithm computes a real valued $M^* \in \mathcal{M}$ based on which the final classification is performed using the rule $y_i = \max_{j=1,\ldots,c} M^*_{ij}$. This is done via a regularization framework where the cost function is defined as,

$$\mathcal{Q}(M) = \frac{1}{2}\Big(\sum_{i,j=1}^{n} W_{ij} \| \frac{1}{\sqrt{D_{ii}}} M_i - \frac{1}{\sqrt{D_{jj}}} M_j \|^2 + \mu \sum_{i=1}^{n} \|M_i - Y_i\|^2 \Big). \quad (1)$$

The computation of M^* is done by solving the quadratic optimization problem $M^* = \arg\min_{M \in \mathcal{M}} \mathcal{Q}(M)$. Intuitively, we are seeking an M^* that is smooth along the edges of similar pairs (x_i, x_j) and at the same time close to Y when evaluated on the labelled data X_l. The first term in (1) is the *smoothness* term and the second is the *fitness* term. Notice that when two samples x_i and x_j are similar (i.e., the weight W_{ij} is large) minimizing the smoothness term in (1) results in M being smooth across similar samples. Thus, similar data samples will likely share the same class label. It can be shown [**?**] that the solution to problem (1) is given by

$$M^* = \beta(I - \alpha S)^{-1} \mu Y, \quad (2)$$

where $\alpha = \frac{1}{1+\mu}$ and $\beta = \frac{\mu}{1+\mu}$.

3 Classification of multiple observations

We now define formally the particular problem of classification of multiple observations of the test pattern s. We assume that we have m transformed observations of s of the following form $x_i = U(\eta_i)s,\ i = 1, \ldots, m$, where $U(\eta)$ denotes the geometric transformation with parameters η, which is applied on the pattern s. The problem is to classify s in the right class using the multiple observations $x_i,\ i = 1, \ldots, m$. We view this problem as a special case of semi-supervised learning, where the unlabelled data X_u represent the multiple observations. In particular, all unlabelled data samples belong to the same (unknown) class.

In this context, the label propagation method shall be robust to transformations. Transformation invariance can be introduced into graph-based methods by augmenting the graph vertices with the so-called *virtual samples*, denoted hereby as X_{vs} (see [**?**] for a similar approach). The virtual samples are essentially data samples that are generated artificially, by applying transformations to the original data samples. They are given the class labels of the original samples that they have been generated from and are treated as labelled data. By including the virtual samples in the graph, the graph-based algorithm becomes more robust to transformations of the test samples. We therefore adopt this strategy and we include n_{vs} virtual samples X_{vs} in our original data set that is finally written as $X = \{X_l, X_{vs}, X_u\}$.

The problem now resides in the construction of the weight matrix W for label propagation. We have observed in practice that the performance of the label propagation algorithm significantly depends on the structure of the graph. Moreover, we have seen that the

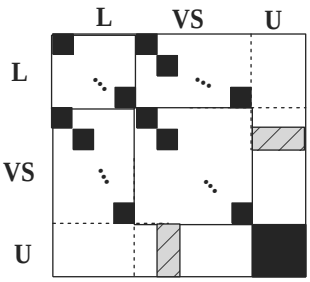


Figure 1. Sparsity pattern of a certain class-conditional adjacency matrix A_i.

role of adjacency information is more important than the weights themselves. We therefore propose an optimization problem about forming an effective adjacency matrix A, which is eventually used in the label propagation method.

4 Graph construction algorithm

In order to build an effective adjacency matrix A, we first assume that it corresponds to a linear combination of all possible c class-conditional adjacency matrices $A_i,\ i = 1, \ldots, c$. *Each A_i corresponds to the adjacency matrix that would be ideally obtained if the correct unknown class was the i-th and if samples from the same class are only allowed to be adjacent.* Assuming that the data samples are ordered as $X = \{X_l, X_{vs}, X_u\}$ and that the data samples of the same class are grouped together, all matrices A_i have similar structure and they only differ in the $\{X_{vs}$,$X_u\}$ part. Figure 1 shows the general structure of a certain class-conditional adjacency matrix, where the black entries denote the nonzero entries. Note that the A_i's differ only in the gray part.

Recall that the classification problem under study here consists in finding a single class for all unlabelled data. Ideally, the optimal adjacency matrix is therefore one of the matrices A_i. Alternatively, the optimal adjacency matrix can be written as

$$A = \sum_{i=1}^{c} \lambda_i A_i, \tag{3}$$

where the weights $\lambda_i \in \{0, 1\}$ and the vector λ has only one nonzero entry. At the same time, the optimal adjacency matrix should be as "realistic" as possible i.e., as close as possible to a similarity matrix built from the data samples X directly. Denote by P such a data-driven similarity matrix. In order to measure the closeness between the two similarity matrices X and P we will use the Frobenius inner product defined as $\langle X, P\rangle_F = \sum_{i,j=1}^{n} X(i,j)P(i,j)$.

In order to compute A, we finally need to solve the following linear program (LP):

Optimization problem: **OPT**
$\max_\lambda\ \langle A, P\rangle_F$
subject to
$A = \sum_{i=1}^{c} \lambda_i A_i$,
$\|\lambda\|_1 \leq 1$,

The constraint $\|\lambda\|_1 \leq 1$ encourages the sparsity of λ and especially the case where only one entry is nonzero. Intuitively, the magnitude of λ_i indicates the contribution of the i-th class-conditional matrix A_i to the construction of A. By encouraging the sparsity of λ we essentially limit the number of classes that can contribute to the optimal adjacency matrix.

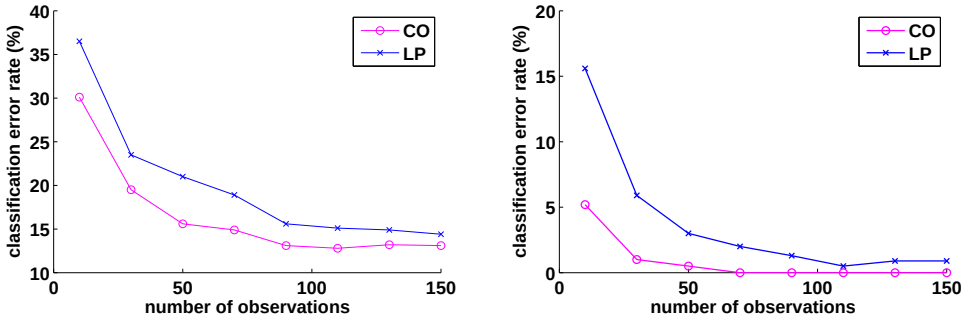
Once the adjacency matrix A has been obtained from the solution of OPT, we compute the similarity matrix as $S = D^{-1/2}AD^{-1/2}$ and then we employ the label propagation algorithm in order to get the estimated class labels on X_u. Finally, we perform majority voting on the labels of X_u in order to estimate the unknown class. We call the proposed algorithm CO since the target is to optimize the connectivity structure of the graph. Note that in some cases one could do the recognition based on λ solely and skip the label propagation step. This is an issue for further investigation.

5 Experimental results

We use two different data sets for our experimental evaluation; (i) a handwritten digit image collection[1]. and (ii) the USPS handwritten digit image collection. The first collection contains 20×16 bit binary images of "0" through "9", where each class contains 39 samples. The USPS collection contains 16×16 grayscale images of digits and each class contains 1100 samples.

In our experiments, we have chosen P to be the k-NN adjacency matrix with Gaussian weights, since it provides a kind of similarity matrix that is data-driven. However, this is by no means an optimal choice for selecting P and this is an issue to be further investigated. We will compare the proposed CO method with the Label Propagation (LP), which employs the k-NN graph in combination with the Gaussian weights in order to build the similarity matrix S. Note that both methods rely on label propagation followed by majority voting. *The only difference between LP and CO is the construction of the similarity matrix S, which is driven by the*

[1] http://www.cs.toronto.edu/∼roweis/data.html



(a) Binary digits (b) USPS digits

Figure 2. Classification results measured on two different data sets.

different graph construction methodologies i.e., k-NN versus optimized (see Section 4) respectively.

In the experiments that follow, first we split the data sets into training and test sets by including 2 samples per class in the training set and the remaining are assigned to the test set. Each training sample is augmented by 4 virtual samples generated by successive rotations of it, where each rotation angle is sampled regularly in $[-40°, 40°]$. This interval has been chosen to be sufficiently small in order to avoid the confusion of digits '6' and '9'. Next, in order to build the the unlabelled set X_u (i.e., multiple observations) of a certain class, we choose randomly a sample from the test set of this class and then we apply a random rotation on it by a random (uniformly sampled) angle $\theta \in [-40°, 40°]$. The number of nearest neighbors was set to $k = 10$ for the binary digit collection and $k = 5$ for the USPS data set, for both methods. These values of k were obtained by the best performance of LP on the test set, which gives it an unfair advantage over our method. We try different sizes of the unlabelled set (i.e., multiple observations), namely $m = [10 : 20 : 150]$ (in MATLAB notation). For each value of m, we report the average classification error rate across 100 random realizations of X_u generated from each one of the 10 classes. Thus, each point in the plot is an average over 1000 random experiments. Figures 2(a) and 2(b) show the results over the binary digits and the USPS digits image collections, respectively. Observe first that increasing the number of observations gradually improves the classification error rate of both methods. *This is expected, since all unlabelled samples belong to the same class, and more observations provide the algorithm with more evidence for estimating the unknown class.* Next, observe that the proposed CO algorithm outperforms LP. This indicates that the graph structure is very important for the effectiveness of the label propagation algorithm and can improve significantly its classification performance.

6 Conclusions

In this paper we have proposed a method for classification of multiple observations of a transformed pattern, which builds on graph-based methods for semi-supervised learning. The main idea is to form the graph in such a way that it exploits the specificities of the problem. We formulate the construction of the graph structure as a linear program, which can be solved efficiently. We provide experimental results that show that the proposed method outperforms the label propagation method in the context of our problem.

References