Efficiently Learning the Metric with Side-Information

Tijl De Bie[1], Michinari Momma[2], and Nello Cristianini[3]

[1] Department of Electrical Engineering ESAT-SCD, Katholieke Universiteit Leuven,
Kasteelpark Arenberg 10,
3001 Leuven, Belgium
tijl.debie@esat.kuleuven.ac.be

[2] Department of Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute,
Troy, NY 12180, USA
mommam@rpi.edu

[3] Department of Statistics, University of California, Davis
Davis, CA 95616, USA
nello@support-vector.net

Abstract. A crucial problem in machine learning is to choose an appropriate representation of data, in a way that emphasizes the relations we are interested in. In many cases this amounts to finding a suitable metric in the data space. In the supervised case, Linear Discriminant Analysis (LDA) can be used to find an appropriate subspace in which the data structure is apparent. Other ways to learn a suitable metric are found in [6] and [11]. However recently significant attention has been devoted to the problem of learning a metric in the semi-supervised case. In particular the work by Xing et al. [15] has demonstrated how semi-definite programming (SDP) can be used to directly learn a distance measure that satisfies constraints in the form of side-information. They obtain a significant increase in clustering performance with the new representation. The approach is very interesting, however, the computational complexity of the method severely limits its applicability to real machine learning tasks. In this paper we present an alternative solution for dealing with the problem of incorporating side-information. This side-information specifies pairs of examples belonging to the same class. The approach is based on LDA, and is solved by the efficient eigenproblem. The performance reached is very similar, but the complexity is only $O(d^3)$ instead of $O(d^6)$ where d is the dimensionality of the data. We also show how our method can be extended to deal with more general types of side-information.

1 Introduction

Machine learning algorithms rely to a large extent on the availability of a good representation of the data, which is often the result of human design choices. More specifically, a 'suitable' distance measure between data items needs to be specified, so that a meaningful notion of 'similarity' is induced. The notion of 'suitable' is inevitably task dependent, since the same data might need very different representations for different learning tasks.

This means that automatizing the task of choosing a representation will necessarily need utilization of some type of information (e.g. some of the labels, or less refined forms of information about the task at hand). Labels may be too expensive, while a less refined and more readily available source of information can be used (known as side-information). For example, one may want to define a metric over the space of movies descriptions, using data about customers associations (such as sets of movies liked by the same customer in [9]) as side-information.

This type of side-information is commonplace in marketing data, recommendation systems, bioinformatics and web data. Many recent papers have dealt with these and related problems; some by imposing extra constraints without learning a metric, as in the constrained K-means algorithm [5], others by implicitly learning a metric, like [9], [13] or explicitly by [15]. In particular, [15] provides a conceptually elegant algorithm based on semi-definite programming (SDP) for learning the metric in the data space based on side-information, an algorithm that unfortunately has complexity $O(d^6)$ for d-dimensional data[4].

In this paper we present an algorithm for the problem of finding a suitable metric, using the side-information that consists of n example pairs $(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}), i = 1, \ldots, n$ belonging to the *same but unknown* class. Furthermore, we place our algorithm in a general framework, in which also the methods described in [14] and [13] would fit. More specifically, we show how these methods can all be related with Linear Discriminant Analysis (LDA, see [8] or [7]).

For reference, we will first give a brief review of LDA. Next we show how our method can be derived as an approximation for LDA in case only side-information is available. Furthermore, we provide a derivation similar to the one in [15] in order to show the correspondence between the two approaches. Empirical results include a toy example, and UCI data sets also used in [15].

Notation. All vectors are assumed to be column vectors. With $\mathbf{I}_d$ the identity matrix of dimension d is meant. With $\mathbf{0}$, we denote a matrix or a vector of appropriate size, containing all zero elements. The vector $\mathbf{1}$ is a vector of appropriate dimension containing all 1's. A prime $'$ denotes a transpose.

[4] The authors of [15] see this problem, and they try to circumvent it by developing a gradient descent algorithm instead of using standard Newton algorithms for solving SDP problems. However, this may lead to convergence problems, especially for data sets in large dimensional spaces.

To denote the side-information that consists of n pairs $(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})$ for which is known that $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)} \in \mathcal{R}^d$ belong to the same class, we will use the matrices $\mathbf{X}^{(1)} \in \mathcal{R}^{n \times d}$ and $\mathbf{X}^{(2)} \in \mathcal{R}^{n \times d}$. These contain $\mathbf{x}_i^{(1)'}$ and $\mathbf{x}_i^{(2)'}$ as their ith rows: $\mathbf{X}^{(1)} = \begin{pmatrix} \mathbf{x}_1^{(1)'} \\ \mathbf{x}_2^{(1)'} \\ \cdots \\ \mathbf{x}_n^{(1)'} \end{pmatrix}$ and $\mathbf{X}^{(2)} = \begin{pmatrix} \mathbf{x}_1^{(2)'} \\ \mathbf{x}_2^{(2)'} \\ \cdots \\ \mathbf{x}_n^{(2)'} \end{pmatrix}$. This means that for any $i = 1, \ldots, n$, it is known that the samples at the ith rows of $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ belong to the same class. For ease of notation (but without loss of generality) we will construct the full data matrix[5] $\mathbf{X} \in \mathcal{R}^{2n \times d}$ as $\mathbf{X} = \begin{pmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \end{pmatrix}$. When we want to denote the sample corresponding to the ith row of $\mathbf{X}$ without regarding the side-information, it is denoted as $\mathbf{x}_i \in \mathcal{R}^d$ (without superscript, and $i = 1, \ldots, 2n$). The data matrix should be *centered*, that is $\mathbf{1}'\mathbf{X} = \mathbf{0}$ (the mean of each column is zero). We use $\mathbf{w} \in \mathcal{R}^d$ to denote a weight vector in this d-dimensional data space.

Although the labels for the samples are *not* known in our problem setting, we will consider the label matrix $\mathbf{Z} \in \mathcal{R}^{2n \times c}$ corresponding to $\mathbf{X}$ in our derivations. (The number of classes is denoted by c.) It is defined as (where $\widetilde{\mathbf{Z}}_{i,j}$ indicates the element at row i and column j):

$$\widetilde{\mathbf{Z}}_{i,j} = \begin{cases} 1 \text{ when the class of the sample } \mathbf{x}_i \text{ is } j \\ 0 \text{ otherwise} \end{cases},$$

followed by a centering to make all column sums equal to zero: $\mathbf{Z} = \widetilde{\mathbf{Z}} - \frac{\mathbf{1}\mathbf{1}'}{n}\widetilde{\mathbf{Z}}$. We use $\mathbf{w}_{\mathbf{Z}} \in \mathcal{R}^c$ to denote a weight vector in the c-dimensional label space.

The matrices $\mathbf{C}_{\mathbf{ZX}} = \mathbf{C}'_{\mathbf{XZ}} = \mathbf{Z}'\mathbf{X}, \mathbf{C}_{\mathbf{ZZ}} = \mathbf{Z}'\mathbf{Z}, \mathbf{C}_{\mathbf{XX}} = \mathbf{X}'\mathbf{X}$ are called *total scatter matrices* of $\mathbf{X}$ or $\mathbf{Z}$ with $\mathbf{X}$ or $\mathbf{Z}$. The total scatter matrices for the subset data matrices $\mathbf{X}^{(k)},\ k = 1, 2$, are indexed by integers: $\mathbf{C}_{kl} = \mathbf{X}^{(k)'}\mathbf{X}^{(l)}$.

Again if the labels were known, we could identify the sets $\mathcal{C}_i = \{\text{all } \mathbf{x}_j \text{ in class } i\}$. Then we could also compute the following quantities for the samples in $\mathbf{X}$: the number of samples in each class: $n_i = |\mathcal{C}_i|$; the class means $\mathbf{m}_i = \frac{1}{n_i}\sum_{j:\mathbf{x}_j \in \mathcal{C}_i} \mathbf{x}_j$; the between class scatter matrix $\mathbf{C}_B = \sum_{i=1}^{c} n_i \mathbf{m}_i \mathbf{m}_i'$. The within class scatter matrix $\mathbf{C}_W = \sum_{i=1}^{c} \sum_{j:\mathbf{x}_j \in \mathcal{C}_i} (\mathbf{x}_j - \mathbf{m}_i)(\mathbf{x}_j - \mathbf{m}_i)'$. Since the labels are not known in our problem setting, we will only use these quantities in our derivations, not in our final results.

2 Learning the Metric

In this section, we will show how the LDA formulation which requires labels can be adapted for cases where no labels but only side-information is available.

[5] In all derivations, the only data samples involved are the ones that appear in the side-information. It is not until the empirical results section that also data not involved in the side-information is dealt with: the side-information is used to learn the metric, and only subsequently, this metric is used to cluster any other available sample. We also assume no sample appears twice in the side-information.

The resulting formulation can be seen as an approximation of LDA with labels available. This will lead to an efficient algorithm to learn a metric: given the side-information, solving just a generalized eigenproblem is sufficient to maximize the expected separation between the clusters.

2.1 Motivation

Canonical Correlation Analysis (CCA) formulation of Linear Discriminant Analysis (LDA) for classification. When given a data matrix $\mathbf{X}$ and a label matrix $\mathbf{Z}$, LDA [8] provides a way to find a projection of the data that has the largest between class variance to within class variance ratio. This can be formulated as a maximization problem of the Rayleigh quotient $\rho(\mathbf{w}) = \frac{\mathbf{w}'\mathbf{C}_B\mathbf{w}}{\mathbf{w}'\mathbf{C}_W\mathbf{w}}$. In the optimum $\nabla_{\mathbf{w}}\rho = \mathbf{0}$, $\mathbf{w}$ is the eigenvector corresponding to the largest eigenvalue of the generalized eigenvalue problem $\mathbf{C}_B\mathbf{w} = \rho\,\mathbf{C}_W\mathbf{w}$. Furthermore, it is shown that LDA can also be computed by performing CCA between the data and the label matrix ([3],[2],[12]). In other words, LDA maximizes the correlation between a projection of the coordinates of the data points and a projection of their class labels. This means the following CCA generalized eigenvalue problem formulation can be used:

$$\begin{pmatrix} \mathbf{0} & \mathbf{C}_{\mathbf{XZ}} \\ \mathbf{C}_{\mathbf{ZX}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \mathbf{w}_{\mathbf{Z}} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{C}_{\mathbf{XX}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathbf{ZZ}} \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \mathbf{w}_{\mathbf{Z}} \end{pmatrix}$$

The optimization problem corresponding to CCA is (as shown in e.g. [4]):

$$\max_{\mathbf{w},\mathbf{w}_{\mathbf{Z}}} \mathbf{w}'\mathbf{X}'\mathbf{Z}\mathbf{w}_{\mathbf{Z}} \quad \text{s.t.} \quad \|\mathbf{X}\mathbf{w}\|^2 = 1 \text{ and } \|\mathbf{Z}\mathbf{w}_{\mathbf{Z}}\|^2 = 1 \tag{1}$$

This formulation for LDA will be the starting point for our derivations.

Maximizing the expected LDA cost function. In the problem setting at hand however, we do not know the label matrix $\mathbf{Z}$. Thus we can not perform LDA in its basic form. However, the side-information that given pairs of samples $(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})$ belong to the same class (and thus have the *same –but unknown– label*) is available. (This side-information is further denoted by splitting $\mathbf{X}$ into two matrices $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ as defined in the notation paragraph.)

Using a parameterization of the label matrix $\mathbf{Z}$ that explicitly realizes these constraints given by the side-information, we derive a cost function that is equivalent to the LDA cost function but that is written in terms of this parameterization. Then *we maximize the expected value of this LDA cost function*, where the expectation is taken over these parameters under a reasonable symmetry assumption. The derivation can be found in Appendix A. Furthermore it is shown in Appendix A that this expected LDA cost function is maximized by solving for the dominant eigenvector of:

$$(\mathbf{C}_{12} + \mathbf{C}_{21})\mathbf{w} = \lambda(\mathbf{C}_{11} + \mathbf{C}_{22})\mathbf{w} \tag{2}$$

where $\mathbf{C}_{kl} = \mathbf{X}^{(k)\prime}\mathbf{X}^{(l)}$.

In Appendix B we provide an alternative derivation leading to the same eigenvalue problem. This derivation is based on a cost function that is close to the cost function used in [15].

2.2 Interpretation and Dimensionality Selection

Interpretation. Given the eigenvector $\mathbf{w}$, the corresponding eigenvalue λ is equal to $\frac{\mathbf{w}'(\mathbf{C}_{12}+\mathbf{C}_{21})\mathbf{w}}{\mathbf{w}'(\mathbf{C}_{11}+\mathbf{C}_{22})\mathbf{w}}$. The numerator $\mathbf{w}'(\mathbf{C}_{12}+\mathbf{C}_{21})\mathbf{w}$ is twice the covariance of the projections $\mathbf{X}^{(1)}\mathbf{w}$ with $\mathbf{X}^{(2)}\mathbf{w}$ (up to a factor equal to the number of samples in $\mathbf{X}^{(k)}$). The denominator normalizes with the sum of their variances (up to the same factor). This means λ is very close to the correlation between $\mathbf{X}^{(1)}\mathbf{w}$ and $\mathbf{X}^{(2)}\mathbf{w}$ (it becomes equal to their correlation when the variances of $\mathbf{X}^{(1)}\mathbf{w}$ and $\mathbf{X}^{(2)}\mathbf{w}$ are equal, which will often be close to true as both $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are drawn from the same distribution). This makes sense: we want $\mathbf{X}\mathbf{w} = \begin{pmatrix}\mathbf{X}^{(1)}\\ \mathbf{X}^{(2)}\end{pmatrix}\mathbf{w}$ and thus both $\mathbf{X}^{(1)}\mathbf{w}$ and $\mathbf{X}^{(2)}\mathbf{w}$ to be strongly correlated with a projection $\mathbf{Z}\mathbf{w}_{\mathbf{Z}}$ of their (common but unknown) labels in $\mathbf{Z}$ on $\mathbf{w}_{\mathbf{Z}}$ (see equation (1); this is what we actually wanted to optimize, but could not do exactly since $\mathbf{Z}$ is not known). Now, when we want $\mathbf{X}^{(1)}\mathbf{w}$ and $\mathbf{X}^{(2)}\mathbf{w}$ to be strongly correlated with the *same* labels, they necessarily have to be strongly correlated with each other.

Some of the eigenvalues may be negative however. This means that along these eigenvectors, samples that should be co-clustered according to the side-information are *anti*-correlated. This can only be caused by features in the data that are irrelevant for the clustering problem at hand (which can be seen as noise).

Dimensionality selection. As with LDA, one will generally not only use the dominant eigenvector, but a dominant eigen*space* to project the data on. The number of eigenvectors used should depend on the signal to noise ratio along these components: when it is too low, noise effects will cause poor performance of a subsequent clustering. So we need to make an estimate of the noise level.

This is provided by the negative eigenvalues: they allow us to make a good estimate of the noise level present in the data, thus motivating the strategy adopted in this paper: only retain the k directions corresponding to eigenvalues larger than the largest absolute value of the negative eigenvalues.

2.3 The Metric Corresponding to the Subspace Used

Since we will project the data onto the k dominant eigenvectors $\mathbf{w}$, this finally boils down to using the distance measure

$$d^2(\mathbf{x}_i,\mathbf{x}_j) = \left(\mathbf{W}'(\mathbf{x}_i-\mathbf{x}_j)\right)'\left(\mathbf{W}'(\mathbf{x}_i-\mathbf{x}_j)\right) = \|\mathbf{x}_i-\mathbf{x}_j\|^2_{\mathbf{W}\mathbf{W}'}.$$

where $\mathbf{W}$ is the matrix containing the k eigenvectors as its columns.

Normalization of the different eigenvectors could be done so as to make the variance equal to 1 along each of the directions. However as can be understood from the interpretation in 2.2, along directions with a high eigenvalue λ a better separation can be expected. Therefore, we applied the heuristic to scale each of the eigenvectors by multiplying them with their corresponding eigenvalue. In doing that, a subsequent clustering like K-means will preferentially find cluster separations orthogonal to directions that will probably separate well (which is desirable).

2.4 Computational Complexity

Operations to be carried out in this algorithm are the computation of the $d \times d$ scatter matrices, and solving a symmetric generalized eigenvalue problem of size d. The computational complexity of this problem is thus $O(d^3)$. Since the approach in [15] is basically an SDP with d^2 parameters, its complexity is $O(d^6)$. Thus a massive speedup can be achieved.

3 Remarks

3.1 Relation with Existing Literature

Actually, $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ do not have to belong to the same space, they can be of a different kind: it is sufficient when corresponding samples in $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ belong to the same class to do something similar as above. Of course then we need different weight vectors in both spaces: $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$. Following a similar reasoning as above, in Appendix C we provide an argumentation that solving the CCA eigenproblem

$$\begin{pmatrix} \mathbf{0} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w}^{(1)} \\ \mathbf{w}^{(2)} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{w}^{(1)} \\ \mathbf{w}^{(2)} \end{pmatrix}$$

is closely related to LDA as well.

This is exactly what is done in [14] and [13] (in both papers in a kernel induced feature space).

3.2 More General Types of Side-Information

Using similar approaches, also general types of side-information may be utilized. We will only briefly mention them:

- When the groups of samples for which is known they belong to the same class is larger than 2 (let us call them $\mathbf{X}^{(i)}$ again, but now i is not restricted to only 1 or 2). This can be handled very analogously to our previous derivation. Therefore we just state the resulting generalized eigenvalue problem:

$$\left(\sum_k \mathbf{X}^{(k)\prime} \sum_k \mathbf{X}^{(k)} \right) \mathbf{w} = \lambda \left(\sum_k \mathbf{X}^{(k)\prime} \mathbf{X}^{(k)} \right) \mathbf{w}$$

– Also in case we are dealing with more than 2 data sets that are of a different nature (eg analogous to [14]: we could have more than 2 data sets, each consisting of a text corpus in a different language), but for which is known that corresponding samples belong to the same class (as described in the previous subsection), the problem is easily shown to reduce to the extension of CCA towards more data spaces, as is e.g. used in [1]. Space restrictions do not permit us to go into this.

– It is possible to keep this approach completely general, allowing for any type of side-information of the form of constraints that express for any number of samples they belong to the same class, or on the contrary do not to belong to the same class. Also knowledge of some of the labels can be exploited. For doing this, we have to use a different parameterization for $\mathbf{Z}$ than used in this paper. In principle also any prior distribution on the parameters can be taken into account. However, sampling techniques will be necessary to estimate the expected value of the LDA cost function in these cases. We will not go into this in the current paper.

3.3 The Dual Eigenproblem

As a last remark, the dual or *kernelized* version of the generalized eigenvalue problem can be derived as follows. The solution $\mathbf{w}$ can be expressed in the form $\mathbf{w} = \left(\mathbf{X}^{(1)'} \, \mathbf{X}^{(2)'} \right) \boldsymbol{\alpha}$ where $\boldsymbol{\alpha} \in \mathcal{R}^{2n}$ is a vector containing the dual variables. Now, with Gram matrices $\mathbf{K}_{kl} = \mathbf{X}^{(k)}\mathbf{X}^{(l)'}$, and after introducing the notation

$$\mathbf{G}_1 = \begin{pmatrix} \mathbf{K}_{11} \\ \mathbf{K}_{21} \end{pmatrix} \text{ and } \mathbf{G}_2 = \begin{pmatrix} \mathbf{K}_{12} \\ \mathbf{K}_{22} \end{pmatrix}$$

the $\boldsymbol{\alpha}$'s corresponding to the weight vectors $\mathbf{w}$ are found as the generalized eigenvectors of

$$(\mathbf{G}_1\mathbf{G}_2' + \mathbf{G}_2\mathbf{G}_1')\boldsymbol{\alpha} = \lambda(\mathbf{G}_1\mathbf{G}_1' + \mathbf{G}_2\mathbf{G}_2')\boldsymbol{\alpha}.$$

This motivates that it will be possible to extend the approach to learning non-linear metrics with side-information as well.

4 Empirical Results

The empirical results reported in this paper will be for clustering problems with the type of side-information described above. Thus, with our method we learn a suitable metric based on a set of samples for which the side-information is known, i.e. $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. Subsequently a K-means clustering of *all* samples (including those that are not in $\mathbf{X}^{(1)}$ or $\mathbf{X}^{(2)}$) is performed, making use of the metric that is learned.

4.1 Evaluation of Clustering Performance

We use the same measure of accuracy as is used in [15], namely, defining $I(\mathbf{x}_i, \mathbf{x}_j)$ as the function being 1 when $\mathbf{x}_i$ and $\mathbf{x}_j$ are clustered in the same cluster by the algorithm,

$$\mathrm{Acc} = \frac{\sum_k \sum_{i,j>i;\mathbf{x}_i,\mathbf{x}_j \in \mathcal{C}_k} I(\mathbf{x}_i, \mathbf{x}_j)}{2 \sum_k \sum_{i,j>i;\mathbf{x}_i,\mathbf{x}_j \in \mathcal{C}_k} 1} + \frac{\sum_{i,j>i;\neg\exists k:\mathbf{x}_i,\mathbf{x}_j \in \mathcal{C}_k} (1 - I(\mathbf{x}_i, \mathbf{x}_j))}{2 \sum_{i,j>i;\neg\exists k:\mathbf{x}_i,\mathbf{x}_j \in \mathcal{C}_k} 1}.$$

4.2 Regularization

To deal with inaccuracies, numerical instabilities and influences of finite sample size, we apply regularization to the generalized eigenvalue problem. This is done in the same spirit as for CCA in [1], namely by adding a diagonal to the scatter matrices $\mathbf{C}_{11}$ and $\mathbf{C}_{22}$. This is justified thanks to the CCA-based derivation of our algorithm. To train the regularization parameter, a cost function described below is minimized via 10-fold cross validation.

In choosing the right regularization parameter, there are two things to consider: firstly, we want the clustering to be good. This means that the side-information should be reflected as well as possible by the clustering. Secondly we want this clustering to be informative. This means, we don't want one very large cluster, the others being very small (the probability to fulfil the side-information would be too easy then). Therefore, the cross-validation cost minimized here, is the probability for the measured performance on the test set side-information, given the sizes of the clusters found. (More exactly, we maximized the difference of this performance with its expected performance, divided by its standard deviation.) This approach incorporates both considerations in a natural way.

4.3 Performance on a Toy Data Set

The effectiveness of the method is illustrated by using a toy example, in which each of the clusters consists of two parts lying far apart (figure (1)). Standard K-means has an accuracy of 0.50 on this data set, while the method developed here gives an accuracy of 0.92.

4.4 Performance on some UCI Data Sets

The empirical results on some UCI data sets, reported in table 1, are comparable to the results in [15]. The first column contains the K-means clustering accuracy without any side-information and preprocessing, averaged over 30 different initial conditions. In the second column, results are given for small side-information leaving 90 percent of the connected components[6], in the third column for *large*

[6] We use the notion *connected component* as defined in [15]. That is, for given side-information, a set of samples makes up one connected component, if between each pair of samples in this set, there exists a path via edges corresponding to pairs given in the side-information. For no side-information given, the number of connected components is thus equal to the total number of samples.

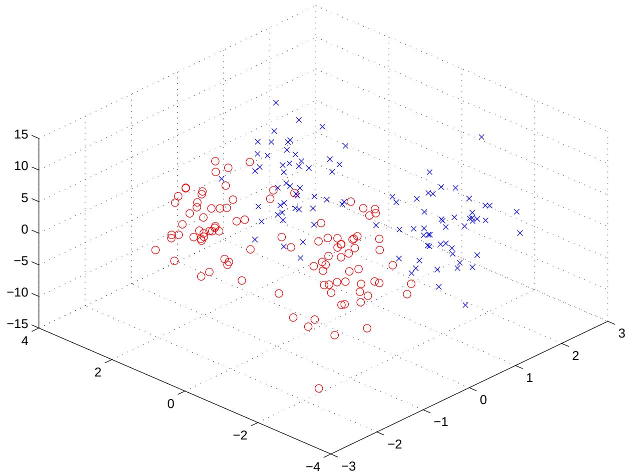

Fig. 1. A toy example whereby the two clusters each consist of two distinct clouds of samples, that are widely separated. Ordinary K-means obviously has a very low accuracy of 0.5, whereas when some side-information is taken into account as described in this paper, the performance goes up to 0.92.

side-information leaving 70 percent of the connected components. For these two columns, averages over 30 randomizations are shown. The side-information is generated by randomly picking pairs of samples belonging to the same cluster. The number between brackets indicates the standard deviation over these 30 randomizations.

Table 2 contains the accuracy on the UCI wine data set and on the protein data set, for different amounts of side-information. To quantize the amount of side-information, we used (as in [15]) the number of pairs in the side-information, divided by the total number of pairs of samples belonging to the same class (the *ratio of constraints*.)

These results are comparable with those reported in [15]. Like in [15], constrained K-means [5] will allow for a further improvement. (It is important to note that constrained K-means on itself does not learn a metric, ie, the side-information is not used for learning which directions in the data space are important in the clustering process. It rather imposes constraints assuring the clustering result does not contradict the side-information.)

5 Conclusions

Finding a good representation of the data is of crucial importance in many machine learning tasks. However, without any assumptions or side-information, there is no way to find the 'right' metric for the data. We thus presented a way

Table 1. Accuracies for on UCI data sets, for different numbers of connected components. (The more side-information, the less connected components. The fraction f is the number of connected components divided by the total number of samples.)

Data set	$f = 1$	$f = 0.9$	$f = 0.7$
wine	0.69 (0.00)	0.92 (0.05)	0.95 (0.03)
protein	0.62 (0.02)	0.71 (0.04)	0.72 (0.06)
ionosphere	0.58 (0.02)	0.69 (0.09)	0.75 (0.05)
diabetes	0.56 (0.02)	0.60 (0.02)	0.61 (0.02)
balance	0.56 (0.02)	0.66 (0.01)	0.67 (0.03)
iris	0.83 (0.06)	0.92 (0.03)	0.92 (0.04)
soy	0.80 (0.08)	0.85 (0.09)	0.91 (0.1)
breast cancer	0.83 (0.01)	0.89 (0.02)	0.91 (0.02)

Table 2. Accuracies on the wine and the protein data sets, as a function of the ratio of constraints.

ratio of	accuracy	ratio of	accuracy
constr.	for wine	constr.	for protein
0	0.69 (0.00)	0	0.62 (0.03)
0.0015	0.73 (0.08)	0.012	0.59 (0.04)
0.0023	0.78 (0.11)	0.019	0.60 (0.05)
0.0034	0.87 (0.08)	0.028	0.62 (0.04)
0.0051	0.91 (0.05)	0.041	0.67 (0.05)
0.0075	0.93 (0.05)	0.060	0.71 (0.05)
0.011	0.96 (0.05)	0.099	0.75 (0.05)
0.017	0.97 (0.017)	0.14	0.77 (0.05)
0.025	0.97 (0.018)	0.21	0.79 (0.06)
0.037	0.98 (0.015)	0.31	0.78 (0.07)

to learn an appropriate metric based on examples of co-clustered pairs of points. This type of side-information is often much less expensive or easier to obtain than full information about the label.

The proposed method is justified in two ways: as a maximization of the expected value of a Rayleigh quotient corresponding to LDA, and another way showing connections with previous work on this type of problems. The result is a very efficient algorithm, being much faster than, while showing similar performance as the algorithm derived in [15].

Importantly, the method is put in a more general context, showing it is only one example of a broad class of algorithms that are able to incorporate different forms of side-information. It is pointed out how the method can be extended to deal with basically any kind of side-information.

Furthermore, the result of the algorithm presented here is a lower dimensional representation of the data, just like in other dimensionality reduction methods such as PCA (Principal Component Analysis), PLS (Partial Least Squares), CCA and LDA, that try to identify interesting subspaces for a given task. This often comes as an advantage, since algorithms like K-means and constrained K-means will run faster on lower dimensional data.

Acknowledgements. TDB is a Research Assistant with the Fund for Scientific Research - Flanders (F.W.O.-Vlaanderen). MM is supported by the NSF grant IIS-9979860. This paper was written during a scientific visit of TDB and MM at U.C.Davis. The authors would like to thank Roman Rosipal, Pieter Abbeel and Eric Xing for useful discussions.

References

1. F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.
2. M. Barker and W.S. Rayens. Partial least squares for discrimination. *Journal of Chemometrics*, 17:166–173, 2003.
3. M. S. Bartlett. Further aspects of the theory of multiple regression. *Proc. Camb. Philos. Soc.*, 34:33–40, 1938.
4. M. Borga, T. Landelius, and H. Knutsson. A Unified Approach to PCA, PLS, MLR and CCA. Report LiTH-ISY-R-1992, ISY, SE-581 83 Linköping, Sweden, November 1997.
5. P. Bradley, K. Bennett, and Ayhan Demiriz. Constrained K-means clustering. Technical Report MSR-TR-2000-65, Microsoft Research, 2000.
6. N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.
7. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley & Sons, Inc., 2nd edition, 2000.
8. R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, Part II:179–188, 1936.
9. T. Hofmann. What people don't want. In *European Conference on Machine Learning (ECML)*, 2002.
10. R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, 1991.
11. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. Technical Report CSD-02-1206, Division of Computer Science , University of California, Berkeley, 2002.
12. R. Rosipal, L.J. Trejo, and B. Matthews. Kernel PLS-SVC for linear and nonlinear classification. In *(to appear) Proceedings of the Twentieth International Conference on Machine Learning*, 2003.
13. J.-P. Vert and M. Kanehisa. Graph-driven features extraction from microarray data using diffusion kernels and cca. In *Advances in Neural Information Processing Systems 15*, Cambridge, MA, 2003. MIT Press.
14. A. Vinokourov, N. Cristianini, and J. Shawe-Taylor. Inferring a semantic representation of text via cross-language correlation analysis. In *Advances in Neural Information Processing Systems 15*, Cambridge, MA, 2003. MIT Press.
15. E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, Cambridge, MA, 2003. MIT Press.

Appendix A: Derivation Based on LDA

Parameterization. As explained before, the side-information is such that we get pairs of samples $(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})$ which have the same class label. Using this side-information we stack the corresponding vectors $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ at the same row in their respective matrices $\mathbf{X}^{(1)} = \begin{pmatrix} \mathbf{x}_1^{(1)'} \\ \mathbf{x}_2^{(1)'} \\ \cdots \\ \mathbf{x}_n^{(1)'} \end{pmatrix}$ and $\mathbf{X}^{(2)} = \begin{pmatrix} \mathbf{x}_1^{(2)'} \\ \mathbf{x}_2^{(2)'} \\ \cdots \\ \mathbf{x}_n^{(2)'} \end{pmatrix}$. The full matrix containing all samples for which side-information is available, is then equal to $\mathbf{X} = \begin{pmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \end{pmatrix}$. Now, since we know each row of $\mathbf{X}^{(1)}$ has the same label as the corresponding row of $\mathbf{X}^{(2)}$, a parameterization of the label matrix $\mathbf{Z}$ is easily found to be $\mathbf{Z} = \begin{pmatrix} \mathbf{L} \\ \mathbf{L} \end{pmatrix}$. Note that $\mathbf{Z}$ is centered iff $\mathbf{L}$ is centered. The matrix $\mathbf{L}$ is in fact just the label matrix of both $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ on themselves. (We want to stress $\mathbf{L}$ is not known, but used in the equations as an unknown matrix parameter for now.)

The Rayleigh quotient cost function that incorporates the side-information. Using this parameterization we apply LDA on the matrix $\begin{pmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \end{pmatrix}$ with the label matrix $\begin{pmatrix} \mathbf{L} \\ \mathbf{L} \end{pmatrix}$ to find the optimal directions for separation of the classes. For this we use the CCA formulation of LDA. This means we want to solve the CCA optimization problem (1) where we substitute the values for $\mathbf{Z}$ and $\mathbf{X}$:

$$\max_{\mathbf{w},\mathbf{w_Z}} \mathbf{w}' \begin{pmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \end{pmatrix}' \begin{pmatrix} \mathbf{L} \\ \mathbf{L} \end{pmatrix} \mathbf{w_Z} = \max_{\mathbf{w},\mathbf{w_Z}} \mathbf{w}'\mathbf{X}^{(1)'}\mathbf{L}\mathbf{w_Z} + \mathbf{w}'\mathbf{X}^{(2)'}\mathbf{L}\mathbf{w_Z} \tag{3}$$

$$\text{s.t.} \quad \left\| \begin{pmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \end{pmatrix} \mathbf{w} \right\|^2 = \|\mathbf{X}^{(1)}\mathbf{w}\|^2 + \|\mathbf{X}^{(2)}\mathbf{w}\|^2 = 1 \tag{4}$$

$$\|\mathbf{L}\mathbf{w_Z}\|^2 = 1$$

The Lagrangian of this constrained optimization problem is:

$$\mathcal{L} = \mathbf{w}'\mathbf{X}^{(1)'}\mathbf{L}\mathbf{w_Z} + \mathbf{w}'\mathbf{X}^{(2)'}\mathbf{L}\mathbf{w_Z} - \widehat{\lambda}\mathbf{w}'(\mathbf{X}^{(1)'}\mathbf{X}^{(1)} + \mathbf{X}^{(2)'}\mathbf{X}^{(2)})\mathbf{w} - \mu\mathbf{w}_\mathbf{Z}'\mathbf{L}'\mathbf{L}\mathbf{w_Z}$$

Differentiating with respect to $\mathbf{w_Z}$ and $\mathbf{w}$ and equating to $\mathbf{0}$ yields

$$\nabla_{\mathbf{w_Z}}\mathcal{L} = 0 \Rightarrow \mathbf{L}'(\mathbf{X}^{(1)} + \mathbf{X}^{(2)})\mathbf{w} = 2\mu\mathbf{L}'\mathbf{L}\mathbf{w_Z} \tag{5}$$

$$\nabla_{\mathbf{w}}\mathcal{L} = 0 \Rightarrow (\mathbf{X}^{(1)} + \mathbf{X}^{(2)})'\mathbf{L}\mathbf{w_Z} = 2\widehat{\lambda}(\mathbf{X}^{(1)'}\mathbf{X}^{(1)} + \mathbf{X}^{(2)'}\mathbf{X}^{(2)})\mathbf{w} \tag{6}$$

From (5) we find that $\mathbf{w_Z} = \frac{1}{2\mu}(\mathbf{L}'\mathbf{L})^\dagger\mathbf{L}'(\mathbf{X}^{(1)}+\mathbf{X}^{(2)})\mathbf{w}$. Filling this into equation (6) and choosing $\widetilde{\lambda} = 4\widehat{\lambda}\mu$ gives that

$$(\mathbf{X}^{(1)} + \mathbf{X}^{(2)})' \left[\mathbf{L}(\mathbf{L}'\mathbf{L})^\dagger\mathbf{L}'\right] (\mathbf{X}^{(1)} + \mathbf{X}^{(2)})\mathbf{w} = \widetilde{\lambda}(\mathbf{X}^{(1)'}\mathbf{X}^{(1)} + \mathbf{X}^{(2)'}\mathbf{X}^{(2)})\mathbf{w}.$$

It is well known that solving for the dominant generalized eigenvector is equivalent to maximizing the Rayleigh quotient:

$$\frac{\mathbf{w}'(\mathbf{X}^{(1)} + \mathbf{X}^{(2)})' \left[\mathbf{L}(\mathbf{L}'\mathbf{L})^\dagger\mathbf{L}'\right] (\mathbf{X}^{(1)} + \mathbf{X}^{(2)})\mathbf{w}}{\mathbf{w}'(\mathbf{X}^{(1)'}\mathbf{X}^{(1)} + \mathbf{X}^{(2)'}\mathbf{X}^{(2)})\mathbf{w}}. \tag{7}$$

Until now, for the given side-information, *there is still an exact equivalence between LDA and maximizing this Rayleigh quotient.* The important difference between the standard LDA cost function and (7) however, is that in the latter the side-information is imposed explicitly by using the reduced parameterization for $\mathbf{Z}$ in terms of $\mathbf{L}$.

The expected cost function. As pointed out, we do not know the term between $[\cdot]$. What we will do then is compute the expected value of the cost function (7) by averaging over all possible label matrices $\mathbf{Z} = \begin{pmatrix}\mathbf{L}\\ \mathbf{L}\end{pmatrix}$, possibly weighted with any symmetric[7] a priori probability for the label matrices. Since the only part that depends on the label matrix is the factor between $[\cdot]$, and since it appears linearly in the cost function, we just need to compute the expectation of this factor. This expectation is proportional to $\mathbf{I} - \frac{\mathbf{11}'}{n}$. To see this we only have to use symmetry arguments (all values on the diagonal should be equal to each other, and all other values should be equal to each other), and the observation that $\mathbf{L}$ is centered and thus $\left[\mathbf{L}(\mathbf{L}'\mathbf{L})^\dagger\mathbf{L}'\right]\mathbf{1} = \mathbf{0}$. Now, since we assume that the data matrix $\mathbf{X}$ containing the samples in the side-information is centered too, $(\mathbf{X}^{(1)}+\mathbf{X}^{(2)})'\frac{\mathbf{11}'}{n}(\mathbf{X}^{(1)}+\mathbf{X}^{(2)})$ is equal to the null matrix. Thus the expected value of $(\mathbf{X}^{(1)}+\mathbf{X}^{(2)})' \left[\mathbf{L}(\mathbf{L}'\mathbf{L})^\dagger\mathbf{L}'\right] (\mathbf{X}^{(1)}+\mathbf{X}^{(2)})$ is proportional to $(\mathbf{X}^{(1)}+\mathbf{X}^{(2)})'(\mathbf{X}^{(1)}+\mathbf{X}^{(2)})$. The expected value of the LDA cost function in equation (7), where the expectation is taken over all possible label assignments $\mathbf{Z}$ constrained by the side-information, is then shown to be

$$\frac{\mathbf{w}'(\mathbf{C}_{11} + \mathbf{C}_{12} + \mathbf{C}_{22} + \mathbf{C}_{21})\mathbf{w}}{\mathbf{w}'(\mathbf{C}_{11} + \mathbf{C}_{22})\mathbf{w}} = 1 + \frac{\mathbf{w}'(\mathbf{C}_{12} + \mathbf{C}_{21})\mathbf{w}}{\mathbf{w}'(\mathbf{C}_{11} + \mathbf{C}_{22})\mathbf{w}}$$

The vector $\mathbf{w}$ maximizing this cost is the dominant generalized eigenvector of

$$(\mathbf{C}_{12} + \mathbf{C}_{21})\mathbf{w} = \lambda(\mathbf{C}_{11} + \mathbf{C}_{22})\mathbf{w}$$

[7] That is, the a priori probability of a label assignment $\mathbf{L}$ is the same as the probability of the label assignment $\mathbf{PL}$ where $\mathbf{P}$ can be any permutation matrix. Remember every row of $\mathbf{L}$ corresponds to the label of a pair of points in the side-information. Thus, this invariance means we have no discriminating prior information on which pair belongs to which of the classes. Using this ignorant prior is clearly the most reasonable we can do, since we assume only the side-information is given here.

where $\mathbf{C}_{kl} = \mathbf{X}^{(k)'}\mathbf{X}^{(l)}$.

(Note that the side-information is symmetric in the sense that one could replace an example pair $(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})$ with $(\mathbf{x}_i^{(2)}, \mathbf{x}_i^{(1)})$ without losing any information. However this operation does not change $\mathbf{C}_{12} + \mathbf{C}_{21}$ nor $\mathbf{C}_{11} + \mathbf{C}_{22}$, so that the eigenvalue problem to be solved does not change either, which is of course a desirable property.)

Appendix B: Alternative Derivation

More in the spirit of [15], we can derive the algorithm by solving the constrained optimization problem (where $\dim(\mathbf{W}) = k$ means that the dimensionality of $\mathbf{W}$ is k, that is, $\mathbf{W}$ has k columns):

$$\begin{aligned} \max_{\mathbf{W}} \; & \operatorname{trace}(\mathbf{X}^{(1)'}\mathbf{W}\mathbf{W}'\mathbf{X}^{(2)}) \\ \text{s.t.} \; & \dim(\mathbf{W}) = k \\ & \mathbf{W}' \begin{pmatrix} \mathbf{X}^{(1)} & \mathbf{X}^{(2)} \end{pmatrix} \begin{pmatrix} \mathbf{X}^{(1)'} \\ \mathbf{X}^{(2)'} \end{pmatrix} \mathbf{W} = \mathbf{I}_k \end{aligned}$$

so as to find a subspace of dimension k that optimizes the correlation between samples belonging to the same class.

This can be reformulated as

$$\begin{aligned} \max_{\mathbf{W}} \; & \operatorname{trace}(\mathbf{W}'(\mathbf{C}_{12} + \mathbf{C}_{21})\mathbf{W}) \\ \text{s.t.} \; & \dim(\mathbf{W}) = k \\ & \mathbf{W}'(\mathbf{C}_{11} + \mathbf{C}_{22})\mathbf{W} = \mathbf{I}_k \end{aligned}$$

Solving this optimization problem amounts to solving for the eigenvectors corresponding to the k largest eigenvalues of the generalized eigenvalue problem described above (2).

The proof involves the following theorem by Ky Fan (see eg [10]):

Theorem 5.01 *Let $\mathbf{H}$ be a symmetric matrix with eigenvalues $\lambda_1 > \lambda_2 > \ldots > \lambda_n$, and the corresponding eigenvectors $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n)$. Then*

$$\lambda_1 + \ldots + \lambda_k = \max_{\mathbf{P}'\mathbf{P}=\mathbf{I}} \operatorname{trace}(\mathbf{P}'\mathbf{H}\mathbf{P}).$$

Moreover, the optimal $\mathbf{P}^*$ is given by $\mathbf{P}^* = (\mathbf{u}_1, \ldots, \mathbf{u}_k)\mathbf{Q}$ where $\mathbf{Q}$ is an arbitrary orthogonal matrix.

Since $(\mathbf{C}_{11} + \mathbf{C}_{22})$ is positive definite, we can take $\mathbf{P} = (\mathbf{C}_{11} + \mathbf{C}_{22})^{1/2}\mathbf{W}$, so that the constraint $\mathbf{W}'(\mathbf{C}_{11} + \mathbf{C}_{22})\mathbf{W} = \mathbf{I}_k$ becomes $\mathbf{P}'\mathbf{P} = \mathbf{I}_k$. Also put $\mathbf{H} = (\mathbf{C}_{11} + \mathbf{C}_{22})^{-1/2}(\mathbf{C}_{12} + \mathbf{C}_{21})(\mathbf{C}_{11} + \mathbf{C}_{22})^{-1/2}$, so that the objective function (8) becomes $\operatorname{trace}(\mathbf{P}'\mathbf{H}\mathbf{P})$. Applying the Ky Fan theorem and choosing $\mathbf{Q} = \mathbf{I}_k$, leads to the fact that $\mathbf{P}^* = (\mathbf{u}_1, \ldots, \mathbf{u}_k)$, with $\mathbf{u}_1, \ldots, \mathbf{u}_k$ the k eigenvectors corresponding of the k largest eigenvalues of $\mathbf{H}$. Thus, the optimal $\mathbf{W}^* = (\mathbf{C}_{11} + \mathbf{C}_{22})^{-1/2}\mathbf{P}^*$. For $\mathbf{P}^*$ an eigenvector of $\mathbf{H} = (\mathbf{C}_{11} + \mathbf{C}_{22})^{-1/2}(\mathbf{C}_{12} + \mathbf{C}_{21})(\mathbf{C}_{11} + \mathbf{C}_{22})^{-1/2}$, this $\mathbf{W}^*$ is exactly the generalized eigenvector (corresponding to the same eigenvalue) of (2). The result is thus exactly the same as obtained in the derivation in Appendix A.

Appendix C: Connection to Literature

If we replace $\mathbf{w}$ in optimization problem (3) subject to (4) once by $\mathbf{w}^{(1)}$ and one by $\mathbf{w}^{(2)}$:

$$\begin{aligned}
&\max_{\mathbf{w}^{(1)},\mathbf{w}^{(2)}} \mathbf{w}^{(1)'}\mathbf{X}^{(1)'}\mathbf{L}\mathbf{w}_{\mathbf{Z}} + \mathbf{w}^{(2)'}\mathbf{X}^{(2)'}\mathbf{L}\mathbf{w}_{\mathbf{Z}} \\
\text{s.t.}\quad &\|\mathbf{X}^{(1)}\mathbf{w}^{(1)}\|^2 + \|\mathbf{X}^{(2)}\mathbf{w}^{(2)}\|^2 = 1 \\
&\|\mathbf{L}\mathbf{w}_{\mathbf{Z}}\|^2 = 1
\end{aligned}$$

where $\mathbf{L}$ corresponds to the common label matrix for $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ (both centered). In a similar way as previous derivation, this can be shown to amount to solving the eigenvalue problem:

$$\begin{pmatrix} \mathbf{0} & \mathbf{X}^{(1)'}\left[\mathbf{L}(\mathbf{L}'\mathbf{L})^{-1}\mathbf{L}'\right]\mathbf{X}^{(2)} \\ \mathbf{X}^{(2)'}\left[\mathbf{L}(\mathbf{L}'\mathbf{L})^{-1}\mathbf{L}'\right]\mathbf{X}^{(1)} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w}^{(1)} \\ \mathbf{w}^{(2)} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{w}^{(1)} \\ \mathbf{w}^{(2)} \end{pmatrix}$$

which again corresponds to a Rayleigh quotient. Since also here in fact we do not know the matrix $\mathbf{L}$, we again take the expected value (as in Appendix A). This leads to an expected Rayleigh quotient that is maximized by solving the generalized eigenproblem corresponding to CCA:

$$\begin{pmatrix} \mathbf{0} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w}^{(1)} \\ \mathbf{w}^{(2)} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{w}^{(1)} \\ \mathbf{w}^{(2)} \end{pmatrix}.$$