6 Semi-Supervised Learning using Semi-Definite Programming

Tijl De Bie  
Nello Cristianini

We discuss the problem of support vector machine (SVM) transduction, which is a combinatorial problem with exponential computational complexity in the number of unlabeled samples. Different approaches to such combinatorial problems exist, among which are exact integer programming approaches (only feasible for very small sample sizes, e.g. [1]) and local search heuristics starting from a suitably chosen start value such as the approach explained in Chapter 5, Transductive Support Vector Machines, and introduced in [2] (scalable to large problem sizes, but sensitive to local optima).

In this chapter, we discuss an alternative approach introduced in [3], which is based on a convex relaxation of the optimization problem associated to support vector machine transduction. The result is a semi-definite programming (SDP) problem which can be optimized in polynomial time, the solution of which is an approximation of the optimal labeling as well as a bound on the true optimum of the original transduction objective function. To further decrease the computational complexity, we propose an approximation that allows to solve transduction problems of up to 1000 unlabeled samples.

Lastly, we extend the formulation to more general settings of semi-supervised learning, where equivalence and inequivalence constraints are given on labels of some of the samples.

6.1 Relaxing SVM-transduction

In transduction problems, we are provided with a set of labeled data points (training set), as well as a set of unlabeled data points (test set). Our interest is to find suitable labels for the second set, with no immediate ambition to make predictions
for yet unseen data points that may become available later on. The way the SVM-transduction problems handles this, is by finding those test set labels for which, after training an SVM on the combined training and test set, the margin on the full data set is maximal. This involves optimizing over all labelings of the test set, an integer programming problem with exponential cost.

Primal  Let us recall the primal soft-margin SVM problem (see e.g. [4] and [5] for an introduction to SVM’s and kernel methods):

\[
\min_{\xi, w} \frac{1}{2} w^T w + C \sum_{i=1}^{t} \xi_i \\
\text{s.t.} \quad y_i w^T x_i \geq 1 - \xi_i \\
\xi_i \geq 0.
\]

We omitted the bias term here, as we will do through the entire chapter. This is not a problem, as argued in [6]. Only the labeled data points are involved in this optimization problem. Then, the transductive SVM can be written as:

\[
\min_{\xi, w, Y_u} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \\
\text{s.t.} \quad y_i w^T x_i \geq 1 - \xi_i \\
\xi_i \geq 0 \\
Y_u \in \{-1, 1\}^u,
\]  \hspace{1cm} (6.1)

where we used the notation \(Y_u = (y_{l+1}, \ldots, y_n)\) for the set of test set labels, a column vector containing the labels for the test points, and \(n = l + u\) for the total number of training and test points. It is the combinatorial constraint (6.1) that makes this optimization problem very hard to solve exactly.

Dual  Very often it is more interesting to focus on the dual problem, as it allows us to use the kernel trick for nonlinear classification and for classification of non-vectorial data. The standard soft-margin SVM problem is given by:

\[
\max_{\alpha_i} \quad 2\alpha_i^T 1 - \alpha_i^T (K_l \odot Y_l Y_l^T) \alpha_l \\
\text{s.t.} \quad C \geq \alpha_i \geq 0,
\]

where \(\alpha_l = (\alpha_1, \ldots, \alpha_l)\) is a column vector of dual variables \(\alpha_i\), and \(K_l\) is the kernel matrix for the training set. With \(\odot\), the element-wise matrix product is meant. The optimum of this optimization problem is equal to the inverse square of the margin (plus an additional cost term in the soft margin formulation). Hence, since we want to maximize the margin, the dual formulation of the transductive SVM can be written as:

\[
\max_{\alpha_i} \quad 2\alpha_i^T 1 - \alpha_i^T (K_l \odot Y_l Y_l^T) \alpha_l \\
\text{s.t.} \quad C \geq \alpha_i \geq 0,
\]

Dual transductive SVM formulation
\[
\begin{align*}
\min_Y \max_\alpha & \quad 2\alpha^T 1 - \alpha^T (K \odot YY^T) \alpha \\
\text{s.t.} & \quad C \geq \alpha_i \geq 0 \\
& \quad Y = \begin{pmatrix} Y_l \\ Y_u \end{pmatrix} \\
& \quad Y_u \in \{-1, 1\}^u.
\end{align*}
\]

Here, \(\alpha = (\alpha_1, \ldots, \alpha_l, \alpha_{l+1}, \ldots, \alpha_n)\) is a vector containing the dual variables for both the training and the test set, \(K\) is the complete kernel matrix, and \(Y\) is the complete label vector. Without loss of generality, we assume that the first \(l\) rows and columns of \(K\) correspond to training points, the last \(u\) to test points. Again, it is the same combinatorial constraint that makes finding an exact solution infeasible for reasonably sized problems.

Without affecting the solution, we slightly reformulate the optimization problem by introducing the matrix variable \(\Gamma = YY^T\) to which we will refer as the label matrix. The dual formulation then becomes:

\[
\begin{align*}
\min_{\Gamma} \max_\alpha & \quad 2\alpha^T 1 - \alpha^T (K \odot \Gamma) \alpha \\
\text{s.t.} & \quad C \geq \alpha_i \geq 0 \\
& \quad \Gamma = YY^T = \begin{pmatrix} Y_l Y_l^T & Y_l Y_u^T \\ Y_u Y_l^T & Y_u Y_u^T \end{pmatrix} \\
& \quad Y_u \in \{-1, 1\}^u.
\end{align*}
\]

All constraints are now linear (matrix) inequalities, and the objective is linear in \(\Gamma\) and concave in \(\alpha\). However, the problem is still an integer program due to constraint (6.3) and hence the overall problem is not convex.

### 6.1.1 Relaxation to an SDP-problem

We will write the label matrix \(\Gamma\) as a block matrix using the notation:

\[
\Gamma = \begin{pmatrix} \Gamma_{ll} & \Gamma_{lu} \\ \Gamma_{ul} & \Gamma_{uu} \end{pmatrix} = \begin{pmatrix} Y_l Y_l^T & Y_l Y_u^T \\ Y_u Y_l^T & Y_u Y_u^T \end{pmatrix}.
\]

Symmetry constraints such as \(\Gamma_{uu} = \Gamma_{uu}^T\) and \(\Gamma_{lu} = \Gamma_{ul}^T\) are understood and we will never mention them explicitly. Now, observe that any matrix of rank 1 with ones on the diagonal can be written as an outer product of a vector with itself where this vector only contains 1 and -1 as its elements. Thus, the following proposition holds:

**Proposition 6.1** We can reformulate the constraints (6.2) and (6.3) by the equiva-
allent set of constraints:

\[
\begin{align*}
\text{diag}(\Gamma) &= 1 \\
\text{rank}(\Gamma) &= 1 \\
\Gamma &= \begin{pmatrix}
Y_l Y_l^T & \Gamma_{lu} \\
\Gamma_{ul} & \Gamma_{uu}
\end{pmatrix}.
\end{align*}
\]

These constraints are linear in the parameters, except for the rank constraint, which is clearly non-convex (indeed, a convex combination of two matrices of rank 1 will generally be of rank 2). To deal with this problem, in this chapter we propose to relax the constraint set, by extending the feasible region to a convex set over which optimization can be accomplished in a reasonable computation time. To retain a good performance, it should not be much larger than the non-convex set specified by the constraints above.

Note that the constraints imply that the matrix \( \Gamma \) is positive semi-definite (PSD). So, we can add \( \Gamma \succeq 0 \) as an additional constraint without modifying the problem. The relaxation then consists in simply dropping the rank constraint. The resulting relaxed optimization problem is:

\[
\begin{align*}
\min_{\Gamma} \max_\alpha & \quad 2\alpha^T 1 - \alpha^T (K \odot \Gamma) \alpha \\
\text{s.t.} & \quad C \geq \alpha_i \geq 0 \\
& \quad \text{diag}(\Gamma) = 1 \\
& \quad \Gamma \succeq 0 \\
& \quad \Gamma = \begin{pmatrix}
Y_l Y_l^T & \Gamma_{lu} \\
\Gamma_{ul} & \Gamma_{uu}
\end{pmatrix}.
\end{align*}
\]

Of course, the rank of the resulting optimal matrix \( \Gamma \) will not necessarily be equal to 1 anymore, and its entries not equal to 1 and \(-1\). However, we can see that each entry of \( \Gamma \) will still lie in the interval \([-1, 1]\). Indeed, since all principal submatrices of a PSD matrix have to be PSD as well, every \( 2 \times 2 \) principal submatrix has to be PSD, which for a matrix containing ones on its diagonal can only be achieved for off-diagonal elements in \([-1, 1]\). Furthermore:

1. Ideally, we should relax the constraints so as to extend the feasible region to just the convex hull of the constraints, which is the smallest convex set containing the feasible region of the original problem. For a label matrix \( YY^T \) with \( Y \in \{-1, 1\}^n \), this convex hull is referred to as the cut polytope. However, no efficient description of the cut polytope is known. Hence, one has to resort to convex relaxations of the cut polytope itself, such as the elliptope, which is essentially the relaxation used in this paper. Other relaxations of the cut polytope are known (such as the metric polytope), and they can be used alternatively or in addition. Tighter relaxations tend to be computationally more challenging, though, and for brevity we will not consider these here. For more information we refer the reader to [7, 8].
Theorem 6.2 The above optimization problem is convex. More specifically, it is a SDP problem.

Proof  By introducing the notation

\[
f(\Gamma) = \max_\alpha 2\alpha^T 1 - \alpha^T (K \odot \Gamma) \alpha
\]

\[
s.t. \quad C \geq \alpha_i \geq 0,
\]

we can rewrite this optimization problem as:

\[
\min_\Gamma f(\Gamma)
\]

\[
s.t. \quad \text{diag} (\Gamma) = 1
\]

\[
\Gamma \succeq 0
\]

\[
\Gamma = \begin{pmatrix} Y_l Y_l^T & \Gamma_{lu} \\ \Gamma_{ul} & \Gamma_{uu} \end{pmatrix}
\]

Let us first concentrate on \( f(\Gamma) \). For a given \( \Gamma \succeq 0 \), the objective is concave and the constraints are all linear, i.e. we have a convex optimization problem. One can easily verify Slater’s constraint qualification (the existence of a strictly feasible point in the constraint set, see e.g. [8]), showing that strong duality holds. Let us now write the dual optimization problem by using Lagrange multipliers \( 2\mu \geq 0 \) and \( 2\nu \geq 0 \) for the inequality constraints \( C \geq \alpha_i \) and \( \alpha_i \geq 0 \) respectively (the factor 2 in front of \( \mu \) and \( \nu \) is used for notational convenience). By invoking strong duality which states that the dual optimum is equal to the primal optimum, we can now write \( f(\Gamma) \) as:

\[
f(\Gamma) = \min_{\mu, \nu} \max_\alpha 2\alpha^T (1 - \mu + \nu) - \alpha^T (K \odot \Gamma) \alpha + 2C \mu^T 1
\]

\[
s.t. \quad \mu \geq 0
\]

\[
\nu \geq 0
\]

We note in passing that the optimal value for \( 1 - \mu + \nu \) will be orthogonal to the null space of \( K \odot \Gamma \), since otherwise the solution could grow to infinity by increasing the component of \( \alpha \) along this null space. Now, the maximization with respect to \( \alpha \) can be carried out explicitly: the optimum is reached for \( \alpha = (K \odot \Gamma)^\dagger (1 - \mu + \nu) + \alpha_0 \), where \( \alpha_0 \) is a term in the null space of \( K \odot \Gamma \). Here \( \dagger \) is used to denote the Moore-Penrose inverse. Plugging this in gives:

\[
f(\Gamma) = \min_{\mu, \nu} (1 - \mu + \nu)^T (K \odot \Gamma)^\dagger (1 - \mu + \nu) + 2C \mu^T 1
\]

\[
s.t. \quad \mu \geq 0
\]

\[
\nu \geq 0
\]
Note that \( \alpha_0 \) has vanished. After introducing an additional variable \( t \):

\[
f(\Gamma) = \min_{\mu, \nu, t} t \\
\text{s.t. } \mu \geq 0 \\
\nu \geq 0 \\
t \geq (1 - \mu + \nu)^T (K \odot \Gamma)^\dagger (1 - \mu + \nu) + 2C\mu^T 1.
\]

Using the extended Schur complement lemma (see Appendix), we can rewrite the latter constraint as:

\[
\begin{pmatrix}
K \odot \Gamma & (1 - \mu + \nu) \\
(1 - \mu + \nu)^T & t - 2C\mu^T 1
\end{pmatrix} \succeq 0
\]

which is a PSD constraint on a matrix that is a linear function of the variables. We can thus rewrite the entire optimization problem as a linear optimization problem subject to linear (matrix) inequalities:

\[
\min_{\Gamma, \mu, \nu, t} t \\
\text{s.t. } \mu \geq 0 \\
\nu \geq 0 \\
\text{diag}(\Gamma) = 1 \\
\Gamma \succeq 0 \\
\begin{pmatrix}
K \odot \Gamma & (1 - \mu + \nu) \\
(1 - \mu + \nu)^T & t - 2C\mu^T 1
\end{pmatrix} \succeq 0 \\
\Gamma = \begin{pmatrix} Y_l Y_l^T & \Gamma_{lu} \\ \Gamma_{ul} & \Gamma_{uu} \end{pmatrix}.
\] (6.5)

This is a convex optimization problem that is solvable in polynomial time (see e.g. [9, 10]).

### 6.1.2 Some simplifications

We can simplify the problem using the following two propositions.

**Proposition 6.3** The optimal value for \( \Gamma \) will be of the form

\[
\Gamma = \begin{pmatrix} Y_l Y_l^T & Y_l \gamma_u^T \\ \gamma_u Y_l^T & \Gamma_{uu} \end{pmatrix}
\]

**Proof** From the extended Schur complement lemma it follows that the column space of \( \Gamma_{lu} \) should be orthogonal to the null space of \( Y_l Y_l^T \). This can only be if \( \Gamma_{lu} = Y_l \gamma_u^T \) for some vector \( \gamma_u \).
Proposition 6.4 The constraint $\Gamma \succeq 0$ is equivalent with
\[
\begin{pmatrix}
1 & \gamma_u^T \\
\gamma_u & \Gamma_{uu}
\end{pmatrix} \succeq 0.
\]

Proof We use the fact that a principal submatrix of a PSD matrix is PSD as well [11]. By taking a principal submatrix of $\Gamma$ containing exactly one row and corresponding column among the first $l$, and all of the last $u$ rows and columns, we can see that $\Gamma \succeq 0$ implies
\[
\begin{pmatrix}
1 & \gamma_u^T \\
\gamma_u & \Gamma_{uu}
\end{pmatrix} \succeq 0.
\]
On the other hand, from
\[
\begin{pmatrix}
1 & \gamma_u^T \\
\gamma_u & \Gamma_{uu}
\end{pmatrix} \succeq 0
\]
we get:
\[
\begin{pmatrix}
Y_l & 0 \\
0 & I
\end{pmatrix}
\begin{pmatrix}
1 & \gamma_u^T \\
\gamma_u & \Gamma_{uu}
\end{pmatrix}
\begin{pmatrix}
Y_l & 0 \\
0 & I
\end{pmatrix}^T
= \begin{pmatrix}
Y_l Y_l^T & Y_l \gamma_u^T \\
\gamma_u Y_l^T & \Gamma_{uu}
\end{pmatrix} = \Gamma \succeq 0.
\]

Thus, the final formulation of the relaxed SVM transduction problem is given by:
\[
\begin{align*}
\min_{\Gamma_{uu}, \gamma_u, \mu, \nu, t} & \quad t \\
\text{s.t.} & \quad \mu \geq 0 \\
& \quad \nu \geq 0 \\
& \quad \text{diag}(\Gamma_{uu}) = 1 \\
& \quad \begin{pmatrix}
1 & \gamma_u^T \\
\gamma_u & \Gamma_{uu}
\end{pmatrix} \succeq 0 \\
& \quad \begin{pmatrix}
K \odot \begin{pmatrix}
Y_l Y_l^T & Y_l \gamma_u^T \\
\gamma_u Y_l^T & \Gamma_{uu}
\end{pmatrix}
(1 - \mu + \nu)
\end{pmatrix} \succeq 0.
\end{align*}
\]
Here we would like to point out that the equality constraint on the diagonal can also be turned into an inequality constraint without affecting the solution: \(\text{diag}(\Gamma_{uu}) \leq 1\). Indeed, if the diagonal were lower than 1, we could simply increase it without affecting the constraints or increasing the objective. We will use this fact later in this chapter.

Computational complexity The total number of variables is equal to \(O(l + u^2)\), the number of linear inequality constraints being \(O(l + u)\), and we have an SDP constraint of size \(O(l + u)\) and one of size \(O(u)\). This implies a worst case computational complexity of \(O((l + u^2)^2(l + u)^2)\) (see [10] for a computational study of SDP problems).
6.1.3 Estimation of the label vector

As noted earlier, the optimal value for $\Gamma$ may have a rank different from 1. So it does not provide us with a direct estimate for the label vector $Y_u$. However, the previous section gives us a hint of what a suitable estimate for it can be: it is given by simply one of the columns of $\Gamma$ corresponding to a positively labeled training point. In other words, we propose to take the (thresholded) vector $\gamma_u$ as an estimate for the optimal test label vector.

Other approaches are possible, such as taking the dominant eigenvector of $\Gamma$, or using a randomized approach. For more information on such methods, see [7].

6.1.4 A bound on the performance of the transductive SVM

The minimum of a relaxed minimization problem is always smaller than the minimum of the unrelaxed problem. Therefore, our method immediately provides a lower bound on the squared inverse margin (plus a cost term for the soft margin formulation), and hence an upper bound on the (soft) margin that can be achieved. On the other hand, the (soft) margin of the SVM trained on the training and test set with estimated test labels, provides us with a lower bound. If both bounds are close to each other, we can be confident that the global optimum has been found.

6.2 An approximation for speed-up

In most practical cases, the computational complexity of this relaxation is still too high. In this section we present an approximation technique that will allow for a considerable speed-up of the method at the cost of a reasonable performance loss. It is notable that this technique may have wider applicability to speed up convex relaxations of combinatorial problems, such as for the max-cut problem (see e.g. [7]).

6.2.1 The subspace trick

Let us assume for a moment that we can come up with a $d$-dimensional subspace of $\mathbb{R}^n$ that contains the optimal label vector $Y$. We represent this subspace by the columns of the matrix $V \in \mathbb{R}^{n \times d}$ which form a basis for it. Then the optimal label matrix $\Gamma = YY^T$ can be represented as $\Gamma = VMV^T$, with $M \in \mathbb{R}^{d \times d}$, a symmetric matrix of rank 1. Our relaxation of the rank constraint on $\Gamma$ to an SDP constraint then translates in an analogous relaxation on $M$. The resulting optimization problem can be obtained by simply replacing all occurrences of $\Gamma$
6.2 An approximation for speed-up

with $VMV^T$ in (6.4)–(6.5) and optimizing over $M$ instead of over $\Gamma$:

$$\min_{M, \mu, \nu, t} \quad t$$
$$\text{s.t.}$$
$$\mu \geq 0$$
$$\nu \geq 0$$
$$\text{diag} (VMV^T) = 1$$
$$M \succeq 0$$
$$(K \odot (VMV^T) (1 - \mu + \nu) \quad (1 - \mu + \nu)^T t - 2C \mu^T 1) \succeq 0$$

6.2.2 Finding a subspace close to the label vector

In practice, it seems impossible to come up with exactly such a subspace $V$. However, there are several techniques to approximate it, which are based on fast eigenvalue problems [12, 13, 14]. Here we choose to use the method proposed in [12], which is based on a spectral relaxation of the normalized graph cut cost function (see e.g. [15] for an introduction to spectral clustering and other eigenvalue problems in pattern recognition). Let us first briefly recapitulate the basic spectral clustering method (without label constraints). Subsequently we will show how it is possible to constrain the result to satisfy the training label information.

The basic spectral clustering problem is solved by the following generalized eigenvalue problem:

$$(\text{diag} (K1) - K)v = \lambda \text{diag} (K1) v,$$

and the generalized eigenvectors belonging to the small eigenvalues capture the cluster structure in the data (which means that a label vector corresponding to a good clustering of the data is likely to be close to the space spanned by these generalized eigenvectors).

In order to ensure that the given training label information is respected by this solution, additional constraints should be imposed on $v$. This can be achieved constructively by making use what we call the label constraint matrix $L$, defined by:

$$L = \begin{pmatrix} 1_{l+} & 1_{l+} & 0 \\ 1_{l-} & -1_{l-} & 0 \\ 1_u & 0 & I \end{pmatrix},$$

where $1_{l+}$ and $1_{l-}$ are vectors containing as many ones as there are positively, respectively negatively, labeled training points, and $1_u$ contains $u$ ones. Using $L$, we can constrain $v$ to respect the training label information by parameterizing it as $v = Lz$. Then the generalized eigenvalue problem to be solved is:

$$L'((\text{diag} (K1) - K)Lz = \lambda L'\text{diag} (K1)Lz,$$  \hspace{1cm} (6.6)
and the corresponding constrained solution is $v = Lz$. For more details about this method we encourage the reader to consult [12].

A good subspace to which the label vector is likely to be close is then spanned by the vectors $v_i = Lz_i$ with $z_i$ the generalized eigenvectors of Eq. (6.6) corresponding to the $d$ smallest eigenvalues (except for the one equal to zero). Hence, we can construct a good matrix $V$ by stacking these $v_i$ next to each other.

We are interested in solutions $\Gamma = VMV^T$ for which oppositely labeled training points $x_i$ and $x_j$ have entries $\Gamma(i, j) = \Gamma(j, i) = -\Gamma(i, i) = -\Gamma(j, j)$ in the label matrix $\Gamma$. This could be ensured by imposing additional constraints to our optimization problem. However, it is easy to see that it can be ensured constructively as well, by ignoring the contribution of the constant column in $L$ to $v_i$, the $i$th column of $V$ (i.e., by equating the first entry of $z_i$ to 0 before computing $v_i$ as $v_i = Lz_i$).

The constraint on the diagonal $\text{diag}(\Gamma) = 1$ will in general be infeasible when using the subspace trick. However, as noted above, we can turn it into an inequality constraint $\text{diag}(\Gamma) \leq 1$ without fundamentally changing the problem. In fact, if the dimensionality $d$ (i.e. the number of columns) of $V$ were equal to $u$, there would be no difference between the optimal solutions obtained with or without the subspace approximation, as then the entire feasible region of $\Gamma$ is the same. The diagonal would then be equal to 1, even if only an inequality constraint is specified.

**Computational complexity** The number of constraints remains roughly the same as in the unapproximated optimization problem. However, we potentially gain a lot in terms of number of free variables, which is now $O(d^2 + n)$. Therefore, for fixed $d$, the worst case computational complexity is $O(n^{4.5})$. The actual value for $d$ can be chosen as large as can be handled by the available computational resources.

### 6.3 General semi-supervised learning settings

Thus far we have discussed the transductive setting, which is just one of the semi-supervised learning task described in chapter 1. We will briefly point out however how the technology in this chapter can straightforwardly be extended to deal with more general settings.

#### 6.3.1 Equivalence and inequivalence label constraints

As in [12] and in Chapter 3 of this book, we are able to handle more general semi-supervised learning settings (see also [16] and [17] where similar constraints are exploited for doing dimensionality reduction and in computing a gaussian mixture model respectively). Imagine the situation where we given *grouplets* of points for which a label vector $Y_i$ is specified. If we allow such grouplets to contain only
one data point, we can assume without loss of generality that each point belongs to exactly one grouplet. The label vector \( Y_i \) indicates which points within the grouplet are given to be in the same class (an equivalence constraint), namely those with the same entry 1 or \(-1\) in \( Y_i \), and which ones are given to belong to opposite classes (when their entry in \( Y_i \) is different, an inequivalence constraint). In between different grouplets no information is given. This means that the overall sign of such a grouplet label vector \( Y_i \) is arbitrary.

Then, using similar techniques as we used above, one can show that the label matrix \( \Gamma \) should be a block matrix, with on the diagonal blocks equal to \( Y_i Y_i^T \), and on the off-diagonal blocks \( (i,j) \) we have \( \gamma_{i,j} Y_j Y_i^T \):

\[
\Gamma = \begin{pmatrix}
Y_1 Y_1^T & \gamma_{1,2} Y_1 Y_2^T & \cdots & \gamma_{1,k} Y_1 Y_k^T \\
\gamma_{2,1} Y_2 Y_1^T & Y_2 Y_2^T & \cdots & \gamma_{2,k} Y_2 Y_k^T \\
\vdots & \vdots & \ddots & \vdots \\
\gamma_{k,1} Y_k Y_1^T & \gamma_{k,2} Y_k Y_2^T & \cdots & Y_k Y_k^T
\end{pmatrix}
\]

where \( \gamma_{i,j} = \gamma_{j,i} \) are the variables over which we have to optimize. Clearly, the label matrix as in the transduction scenario explained in the beginning of this chapter is a special case hereof. Now we can also see that the sign of the label vectors \( Y_i \) is irrelevant: upon changing the sign of \( Y_i \), the optimal solution will simply change accordingly by reversing the signs of \( \gamma_{i,j} \) and \( \gamma_{j,i} \) for all \( j \).

We want to point out that this method makes it possible to tackle the transductive SVM problem in a hierarchical way. First one can perform a crude clustering of the data points into many small clusters (grouplets) that respect the training data. Then, at a second stage, the semi-supervised SVM approach outlined above can be employed. This may greatly reduce the computational cost of the overall algorithm.

### 6.3.2 The subspace trick

Also here the subspace trick can be applied in a very analogous way. Again we can rely on the method described in [12], which is also able to deal with equivalence and inequivalence constraints.

### 6.4 Empirical results

For all implementations we used SeDuMi, a general purpose primal-dual interior point solver [18] for Matlab. We only used the hard margin SVM versions, which are obtained from the soft margin formulations by equating \( \mu \) to 0. Comparisons with SVMlight [2] are reported, with default parameter settings.
Semi-Supervised Learning using Semi-Definite Programming

Figure 6.1 The result of the basic SDP relaxation (left) and of SVM$^{\text{light}}$ (right) on an artificially constructed transduction problem. The boldface ‘o’ and ‘x’ signs represent the negatively and positively labeled training points. The other data points are labeled by the algorithms. The contour lines are drawn for the SVM as trained on the complete set of data points with labels as determined by the transduction algorithms. The SDP relaxation yields the desired result, while apparently SVM$^{\text{light}}$ got stuck in a local optimum.

6.4.1 The basic SDP relaxation

The kernel used in all experiments in this subsection is the RBF kernel, and the width is set to the average over all data points of the distance to their closest neighbor. Figure 6.1 shows an artificially constructed example of a transduction problem solved by the basic SDP relaxation of the transductive SVM. Only two data points were labeled, one for each of both classes. Clearly, a standard inductive SVM would fail in this extreme case.

Furthermore, transductive optimum is so far from the inductive optimum that a greedy strategy such as SVM$^{\text{light}}$ is bound to get stuck in a local optimum. Indeed, the norm of the SVM weight vector at the optimal labeling found by the SDP relaxation is 5.7, and for the SVM$^{\text{light}}$ local optimum it is 7.3. Thus, the labeling found by the SDP relaxation achieves a larger margin. Furthermore, it is notable that the optimum of the relaxed optimization problem is $35.318608$ while the (inductive) SVM optimum when using the predicted labels for the unlabeled data points is only slightly larger: $35.318613$. This indicates that most likely the optimal labeling has been found, since for the optimal labeling of the SVM optimum has to lie in between these values (see Section 6.1.4).

In the Figures 6.2 we show another artificial example, where the data seems to consist of 5 clusters. We labeled 6 samples, at least one in each of the clusters. Both the SDP relaxation and SVM$^{\text{light}}$ clearly succeed in assigning the same label to

---

2. There is a catch in the comparison of both optima: the SDP method does not use an offset parameter $b$, whereas SVM$^{\text{light}}$ does include such an offset. The numbers reported are the weight vector norms when including an offset, hence favoring SVM$^{\text{light}}$. 
6.4 Empirical results

Figure 6.2 The result of the basic SDP relaxation on an artificially constructed transduction problem (left), and the result of SVM$^{light}$ (right). Here we organized the data points in a few small clusters. In each of the clusters, 1 or 2 samples are labeled (in total there are 6 training points). For both methods, the training label determines the test labels of all data points within the cluster, as is desirable in most applications.

all data points that are within the same cluster, and consistent with the training label in that cluster. Figure 6.3 shows the same data set with a different labeling of the training points. The transductive optimum found by the SDP relaxation is slightly imbalanced: 38 data points in one class, and 42 in the other. For this reason SVM$^{light}$ seems to classify two data points differently, as, by default, it tries to find a solution with the same proportion of positively versus negatively labeled test points as in the training set. The norm of the SVM weight vector for the optimal labeling as found by the SDP relaxation is equal to 5.92, which is slightly smaller than 5.96, the weight vector norm for the SVM$^{light}$ solution. Hence also here the SDP approach achieves a larger margin.

Again, in both cases the lower bound provided by the optimum of the SDP relaxation supports the conclusion that the optimal labeling has been found. For the first problem, the optimum of the SDP relaxation is 35.338, while the SVM optimum for the predicted labels is 35.341. For the second problem, those optima are 32.3934 and 32.3937 respectively.

6.4.2 The subspace approximation

We conducted a few experiments on the constitution data set used in [19]. This data set contains 780 articles, an equal number in German, French, Italian and English, that are translations of each other. Furthermore, the articles are organized in so-called ‘Titles’. In our experiments, we solved two different problems: one is the classification of English+French texts versus Italian+English texts, and the other is the classification of the largest Title (roughly containing half of all articles) versus the smaller Titles. We tested the SDP relaxation as well as SVM$^{light}$ on both problems for different training set sizes, and plot the results in Figures 6.4. The kernel used is the normalized bag of words kernel, and $d = 4$. 
Figure 6.3 SDP transduction (left) and SVM\textsuperscript{light} (right) are applied to the same data set as in Figure 6.2, now with a different labeling of the training points. If we label the data points according to the labeled point in the cluster they (visually) belong to, this transduction problem is slightly unbalanced: one class of 38 points, the other of 42 points. Since SVM\textsuperscript{light} fixes the fraction of positively and negatively labeled data points to their fraction in the training set (by default), two data points are split off the cluster left above to satisfy this constraint.

Figure 6.4 The ROC score evaluated on the test set, as a function of the size of the training set for both classification problems. The bold lines are for the easy classification problem classifying languages, and the faint lines are for the harder classification problem classifying articles according to their ‘Title’. The performance of the approximated-SDP relaxation is shown in solid lines, the SVM\textsuperscript{light} performance in dotted lines. Bars indicate the standard deviation over 3 randomizations.
Apparently, SVM\textsuperscript{light} outperforms the approximated-SDP relaxation on the difficult problem of classifying articles according to the ‘Title’ they belong to. This is most likely due to the fact that the 4-dimensional subspace is too small to capture the fine cluster structure due to the different ‘Titles’. Only a subspace dimensionality $d$ larger than 4 would solve the problem. However, even though the computational cost is polynomial in $d$, this quickly becomes computationally demanding.

On the other hand, the approximated-SDP relaxation outperforms an already good performance of SVM\textsuperscript{light} for the easier problem, indicating that here the spectral transduction method finds a subspace sufficiently close to the correct label vector.

### 6.5 Summary and outlook

In this chapter we have presented an alternative approach to the transductive SVM as a combinatorial problem. Whereas early approaches to transduction are based on learning a suitable metric [20, 21], and other methods tackled the problem using exact integer programming approaches [1] (with very limited scalability) or using a local search heuristic [2], our approach consists of a relaxation of the combinatorial problem to a convex optimization problem. More specifically the resulting optimization problem is an SDP, which can be solved in a worst case polynomial time. The application of SDP and other convex optimization techniques seems a very promising line of current research in machine learning (see e.g. also [22, 23]).

While the empirical results for the relaxation are generally better than with SVM\textsuperscript{light}, unfortunately the scalability is still limited. To solve this problem, we introduced an approximation technique of general applicability in relaxations of combinatorial problems. The performance of this approximation strongly depends on the quality of the approximation, and mixed empirical results in comparison with SVM\textsuperscript{light} are reported.

Future work includes the investigation if the problem structure can be exploited to speed up the optimization problem. An important theoretical question that remains unanswered is whether the relaxation allows to find a solution with a margin that is provably within a fixed constant factor of the unrelaxed optimum. As we pointed out, the relaxation does provide us with an interval within which the true optimal solution must lie. However, the size of this interval is not known \textit{a priori} as is the case for e.g. the relaxation of the max-cut problem (see e.g. [7]). Lastly, it would be interesting to investigate theoretically what the influence is of the subspace approximation on the optimum.
Appendix: The extended Schur complement lemma

We state the Schur complement lemma without proof (see e.g. [7]):

**Lemma 6.5 (Schur complement lemma)** For symmetric matrices $A \succ 0$ and $C \succeq 0$:

$$C \succeq B^T A^{-1} B \iff \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \succeq 0.$$ 

When the matrix $A$ may be rank deficient, the following extended Schur complement lemma should be used. It is a generalization of the standard Schur complement lemma. We provide it here with a proof:

**Lemma 6.6 (Extended Schur complement lemma)** For symmetric matrices $A \succeq 0$ and $C \succeq 0$:

The column space of $B \perp$ the null space of $A$

$$C \succeq B^T A^+ B \iff \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \succeq 0.$$

**Proof** We write the singular value decomposition (SVD) of $A$ as

$$A = \begin{pmatrix} V & V_0 \end{pmatrix} \begin{pmatrix} \Lambda & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V & V_0 \end{pmatrix}^T = V \Lambda V^T$$

where $V_0$ denotes the singular vectors for the null space of $A$, $V$ the other singular vectors, and $\Lambda$ is a diagonal matrix containing the nonzero singular values of $A$, i.e. $\Lambda \succ 0$. The blocks are assumed to be compatible. Similarly, we write the SVD of $C$ as

$$C = \begin{pmatrix} W & W_0 \end{pmatrix} \begin{pmatrix} \Delta & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} W & W_0 \end{pmatrix}^T = W \Delta W^T.$$

(⇒) If the column space of $B \perp V_0$, we can write $B$ as $B = VB_V$ for some matrix $B_V$. Then also $B^T A^+ B = B_V^T \Lambda^{-1} B_V$. So, from $C \succeq B^T A^+ B = B_V^T \Lambda^{-1} B_V$ and from $\Lambda \succ 0$ and $C \succeq 0$, it follows from the Schur complement lemma that

$$\begin{pmatrix} \Lambda & B_V \\ B_V^T & C \end{pmatrix} \succeq 0.$$ 

Left multiplication of both sides of this inequality with

$$\begin{pmatrix} V & 0 \\ 0 & I \end{pmatrix}$$

and on the right with its transpose, yields

$$\begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \succeq 0.$$

(⇐) We will prove the orthogonality of the column space of $B$ with the null space of $A$ by contradiction. So, assume that the column space of $B$ is not orthogonal to the null space $V_0$ of $A$. Then, there exists a vector $v_0$ in the span of $V_0$ for which
\( B^T v_0 = b \neq 0 \). Now, we have that 
\[
\begin{pmatrix}
A & B \\
B^T & C
\end{pmatrix}
= 
\begin{pmatrix}
V \Lambda V^T & B \\
B^T & C
\end{pmatrix}
\succeq 0.
\]
Thus, for any vector \( w \), multiplying this matrix with \( \begin{pmatrix} v_0 \\ w \end{pmatrix} \) on the right and on the left with its transpose must result in a nonnegative number: 
\[
2 b^T w + w^T C w \geq 0.
\]
However, plugging in \( w = -C^T b - W_0 W_0^T b \) yields 
\[
2 b^T w + w^T C w = -2 b^T W_0 W_0^T b - b^T C^T b < 0,
\]
and thus we reached a contradiction. So we have established that the column space of \( B \) is orthogonal to the span of \( V_0 \).

This means that we can write \( B \) as \( B = VB_V \) for some particular \( B_V \), and:
\[
\begin{pmatrix}
A & B \\
B^T & C
\end{pmatrix}
= 
\begin{pmatrix}
V \Lambda V^T & VB_V \\
B_V^T V^T & C
\end{pmatrix}
= 
\begin{pmatrix}
V & 0 \\
0 & I
\end{pmatrix}
\begin{pmatrix}
\Lambda & B_V \\
B_V^T & C
\end{pmatrix}
\begin{pmatrix}
V & 0 \\
0 & I
\end{pmatrix}^T,
\]
so that:
\[
\begin{pmatrix}
A & B \\
B^T & C
\end{pmatrix}
\succeq 0 \Rightarrow \begin{pmatrix}
\Lambda & B_V \\
B_V^T & C
\end{pmatrix} \succeq 0.
\]
Since \( \Lambda \succ 0 \) and \( C \succeq 0 \) we can invoke the Schur complement lemma, which gives 
\[
C \succeq B_V^T \Lambda^{-1} B_V \succeq 0.
\]
However from the orthonormality of singular vectors: 
\( V^T V = I \), and thus 
\[
B_V^T \Lambda^{-1} B_V = B_V^T V \Lambda^{-1} V^T V B_V = B^T A^T B \text{, meaning that } B^T A^T B \succeq 0.
\]
References


