Appendix B. Mathematical Formulations of Binary and Multicategory SVMs

Below we summarize mathematical formulations of binary and multicategory Support Vector Machines. Refer to papers cited in text for full descriptions of the algorithms.

1. Binary SVMs

Given *n* training instances of dimension *m*: $x_i \in \Re^m$ and corresponding class labels $y_i \in \{-1,+1\}$ ($i = 1,2,..,n$), the margin between two classes is optimized via a solution of the following quadratic constrained optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C\sum_{i=1}^{n} \xi_i \text{ subject to}$$

$$y_i((w^T \Phi(x_i)) + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \text{ for } i = 1,2,..,n$$

where $w \in \Re^m$ is a vector of weights of training instances; *b* is a constant; *C* is a real-valued tradeoff (cost) parameter; ξ_i is a penalty parameter (slack variable); and Φ is a map from the input space $\Re^m$ to the typically much larger dimensional feature space $\Re^r$ ([Vapnik1998] and [Burges1998]). This optimization problem is quadratic in *n* variables and *n* constraints and depends on the data only through dot products $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$. Hence, there is no need to know mapping Φ explicitly, since one has to use only the function *K* (called a *kernel function*) to solve the optimization problem. Among commonly used kernel function are:

1. Linear kernel: $K(x_i, x_j) = x_i^T x_j$;
2. Polynomial kernel: $K(x_i, x_j) = (\gamma \cdot x_i^T x_j + r)^p$, where $\gamma, r \in \Re$;
3. Radial-basis kernel: $K(x_i, x_j) = \exp\left(-\sigma \left\| x_i - x_j \right\|^2\right)$, where $\sigma \in \Re$.

Given *w* and *b*, one can classify an instance *x* using the decision function:

$$f(x) = \text{sgn}[w^T \Phi(x) + b].$$

2. Multicategory SVMs

In formulations of multiclass SVM methods described below adopt the following notation: $x_i \in \Re^m$ are *m*-dimensional training instances and $y_i \in \{1,2,...,k\}$ ($i = 1,2,..,n$) are corresponding class labels.

One-vs-rest (OVR). The margins between each of *k* classes and the remaining classes are optimized via solution of the following constrained QP optimization problem:

$$\min_{w_p, b_p, \xi^p} \frac{1}{2} w_p^T w_p + C\sum_{i=1}^{n} \xi_i^p \text{ subject to}$$

$$w_p^T \Phi(x_i) + b_p \geq 1 - \xi_i^p, \text{ if } y_i = p,$$

$$w_p^T \Phi(x_i) + b_p \leq -1 + \xi_i^p, \text{ if } y_i \neq p,$$

$$\text{and } \xi_i^p \geq 0 \text{ for } i = 1,2,..,n$$

where $p \in \{1,2,...,k\}$; $w_p \in \Re^m$ is a vector of weights of training instances; $b_p \in \Re$; *C* is a real-valued tradeoff (cost) parameter; ξ_i^p is a penalty parameter (slack variable); and $\Phi : \Re^m \rightarrow \Re^r$ ([Kressel1999]). In total one needs to solve *k* constrained QP problems (for *k* values of *p*) with *n* variables and *n* constraints. Given optimal weights *w* and *b*, the following decision function is used for classification of an instance *x*:

$$f(x) = \arg\max_{p=1,...,k} [w_p^T \Phi(x) + b_p].$$

Note, that for the case *k=2*, this technique is equivalent (i.e. its hyperplane is identical) to the binary SVMs.

One-vs-one (OVO). The margins between each pair of *k* classes are optimized via solution of the following constrained QP optimization problem:

$$\min_{w_{pq},b_{pq},\xi^{pq}} \frac{1}{2} {w_{pq}}^T w_{pq} + C\sum_{i=1}^{n} \xi_i^{pq} \text{ subject to}$$

$${w_{pq}}^T \Phi(x_i) + b_{pq} \geq 1 - \xi_i^{pq}, \text{ if } y_i = p,$$

$${w_{pq}}^T \Phi(x_i) + b_{pq} \leq -1 + \xi_i^{pq}, \text{ if } y_i = q,$$

$$\text{and } \xi_i^{pq} \geq 0 \text{ for } i = 1,2,..,n_1$$

where $p \in \{1,2,...,k\}$; $q \in \{1,2,...,k\} \setminus p$; $n_1 \leq n$ is the number of training instances with class labels *p* and *q*; $w_{pq} \in \Re^m$ is a vector of weights of training instances; $b_{pq} \in \Re$; *C* is a real-valued tradeoff (cost) parameter; ξ_i^{pq} is a penalty parameter (slack variable); and $\Phi : \Re^m \to \Re^r$ ([Kressel1999]). In total one needs to solve $\binom{k}{2} = \frac{k(k-1)}{2}$ constrained QP problems (for all distinct pairs of *p* and *q*) with n_1 variables and n_1 constraints. If different classes have the same priors in the training dataset, then $n_1 = \frac{2n}{k}$. Given optimal weights *w* and *b*, individual decision functions for all distinct pairs of *p* and *q* are computed for an instance *x*:

$$f_{pq}(x) = \text{sgn}[w_{pq}^T \Phi(x) + b_{pq}].$$

There are various methods to combine votes of the individual decisions functions into a final decision. A common approach, so-called *Max Wins strategy*, is to assign an instance to a class which has the largest number of votes [Friedman1996].

Note, that for the case *k=2*, this technique is equivalent (i.e. its hyperplane is identical) to the binary SVMs.

DAGSVM. The margins between each pair of *k* classes are optimized via solution of the constrained QP optimization problem, same as for OVO. When all QP problems are solved, a new instance *x* is classified using DDAG, a rooted binary decision directed acyclic graph, constructed on the basis of $\binom{k}{2}$ individual classifiers (nodes) and *k* leaves corresponding to the classification decisions [Platt2000]. The choice of the class order in the DDAG list can be arbitrary as shown empirically in [Platt2000].

Method by Weston and Watkins (WW). The margins between all *k* classes are optimized via solution of the following constrained QP optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} \sum_{p=1}^{k} w_p^T w_p + C\sum_{i=1}^{n} \sum_{p \neq y_i} \xi_i^p \text{ subject to}$$

$$w_{y_i}^T \Phi(x_i) + b_{y_i} \geq w_p^T \Phi(x_i) + b_p + 2 - \xi_i^p \text{ and } \xi_i^p \geq 0$$

$$\text{for } i = 1,2...,n \text{ and } p \in \{1,2,...,k\} \setminus y_i,$$

where $w_p \in \Re^m$ is a vector of weights of training instances; $b \in \Re^k$; *C* is a real-valued tradeoff (cost) parameter; ξ_i^p is a penalty parameter (slack variable); and $\Phi : \Re^m \to \Re^r$ [Weston1999]. This optimization problem is quadratic in *(k-1)n* variables and *(k-1)n* constraints. Given optimal weights *w* and *b*, the following decision function is used for classification of an instance *x*:

$$f(x) = \arg\max_{p=1,...,k} [w_p^T \Phi(x) + b_p].$$

Note, that for the case *k=2*, this technique is equivalent (i.e. its hyperplane is identical) to the binary SVMs.

In our experiments we used a modified formulation of this algorithm, called *bounded formulation*, which is obtained by adding a term $\sum_{m=1}^{k} b_m^2$ to the objective function of the optimization problem stated above. By doing so, the dual formulation of

SVM problem is simplified, which can lead to easier optimization problem solvable by robust decomposition techniques (see [Hsu2002] for details).

Method by Crammer and Singer (CS). The margins between all *k* classes are optimized via solution of the following constrained QP optimization problem:

$$\min_{w,\xi} \frac{1}{2}\sum_{p=1}^{k} w_p^T w_p + C\sum_{i=1}^{n}\xi_i \quad \text{subject to}$$

$$w_{y_i}^T \Phi(x_i) - w_p^T \Phi(x_i) \geq e_i^p - \xi_i \text{ , and } \xi_i \geq 0$$

$$\text{for } i = 1,2...,n \text{ and } p \in \{1,2,...,k\} \setminus y_i ,$$

where $w_p \in \Re^m$ is a vector of weights of training instances; *C* is a real-valued tradeoff (cost) parameter; ξ_i is a penalty parameter (slack variable); $\Phi : \Re^m \rightarrow \Re^r$; and $e_i^p = \begin{cases} 0 \; if \; y_i = p \\ 1 \; if \; y_i \neq p \end{cases}$ ([Hsu2002] and [Crammer2000]). This optimization problem is quadratic in *(k-1)n* variables and only *n* constraints. Given optimal weights *w*, the following decision function is used for classification of an instance *x*:

$$f(x) = \arg\max_{p=1,...,k} w_p^T \Phi(x) .$$

Similarly to WW, in our experiments we used a bounded formulation, which can lead to a significant speed-up in the solution of the optimization problem (see [Hsu2002] for details).

References

[Burges1998] Burges, C. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):955-974, 1998.

[Crammer2000] Crammer, K. and Y. Singer. "On the Learnability and Design of Output Codes for Multiclass Problems", Proceedings of the Thirteen Annual Conference on Computational Learning Theory (COLT), 2000.

[Friedman1996] Friedman, J. "Another approach to polychotomous classification", Technical report, Stanford Univeristy, 1996.

[Hsu2002] Hsu, Chih-Wei and Chih-Jen Lin. "A Comparison of Methods for Multi-class Support Vector Machines", IEEE Transactions in Neural Networks 13(2) 415-425, 2002.

[Kressel1999] Kressel, U. "Pairwise classification and support vector machines", In Advances in Kernel Methods: Support Vector Learning (Chapter 15), MIT Press, 1999.

[Platt2000] Platt, J., N. Cristianini, and J. Shawe-Taylor. "Large margin dags for multiclass classification", Advances in Neural Information Processing Systems 12, pages 547-553. MIT Press, 2000.

[Vapnik1998] Vapnik, V. "Statistical Learning Theory", Wiley-Interscience, 1998.

[Weston1999] Weston, J. and C. Watkins. "Support Vector Machines for Multi-Class Pattern Recognition", Proceedings of the Seventh European Symposium On Artificial Neural Networks, 1999.