we cannot give an exhaustive listing of all successful SVM applications. We thus conclude the list with some of the more exotic applications, such as in High-Energy-Physics [19, 558], in the monitoring of household appliances [390], in protein secondary structure prediction [249], and, with rather intriguing results, in the design of decision feedback equalizers (DFE) in telephony [105].

7.9 Summary

This chapter introduced SV pattern recognition algorithms. The crucial idea is to use kernels to reduce a complex classification task to one that can be solved with separating hyperplanes. We discussed what kind of hyperplane should be constructed in order to get good generalization performance, leading to the idea of large margins. It turns out that the concept of large margins can be justified in a number of different ways, including arguments based on statistical learning theory, and compression schemes. We described in detail how the optimal margin hyperplane can be obtained as the solution of a quadratic programming problem. We started with the linear case, where the hyperplane is constructed in the space of the inputs, and then moved on to the case where we use a kernel function to compute dot products, in order to compute the hyperplane in a feature space.

Two further extensions greatly increase the applicability of the approach. First, to deal with noisy data, we introduced so-called slack variables in the optimization problem. Second, for problems that have more than just two classes, we described a number of generalizations of the binary SV classifiers described initially.

Finally, we reported applications and benchmark comparisons for the widely used USPS handwritten digit task. SVMs turn out to work very well in this field, as well as in a variety of other domains mentioned briefly.

7.10 Problems

7.1 (Weight Vector Scaling •) *Show that instead of the "1" on the right hand side of the separation constraint (7.11), we can use any positive number $\gamma > 0$, without changing the optimal margin hyperplane solution. What changes in the soft margin case?*

7.2 (Dual Perceptron Algorithm [175] ••) *Kernelize the perceptron algorithm described in footnote 1. Which of the patterns will appear in the expansion of the solution?*

7.3 (Margin of Optimal Margin Hyperplanes [62] ••) *Prove that the geometric margin ρ of the optimal margin hyperplane can be computed from the solution $\boldsymbol{\alpha}$ via*

$$\rho^{-2} = \sum_{i=1}^{m} \alpha_i. \tag{7.68}$$

Also prove that

$$\rho^{-2} = 2W(\boldsymbol{\alpha}) = \|\mathbf{w}\|^2. \tag{7.69}$$

Note that for these relations to hold true, $\boldsymbol{\alpha}$ needs to be the solution of (7.29).

7.4 (Relationship Between $\|\mathbf{w}\|$ and the Geometrical Margin •) *(i) Consider a separating hyperplane in canonical form. Prove that the margin, measured perpendicularly to the hyperplane, equals $1/\|\mathbf{w}\|$, by considering two opposite points which precisely satisfy $|\langle \mathbf{w}, \mathbf{x}_i\rangle + b| = 1$.*

(ii) How does the corresponding statement look for the case of ν-SVC? Use the constraint (7.41), and assume that all slack variables are 0.

7.5 (Compression Bound for Large Margin Classification ○○○) *Formalize the ideas stated in Section 7.2: Assuming that the data are separable and lie in a ball of radius R, how many bits are necessary to encode the labels of the data by encoding the parameters of a hyperplane? Formulate a generalization error bound in terms of the compression ratio by using the analysis of Vapnik [561, Section 4.6]. Compare the resulting bound with Theorem 7.3. Take into account the eigenvalues of the Gram matrix, using the ideas of from [604] (cf. Section 12.4).*

7.6 (Positive Definiteness of the SVC Hessian •) *From Definition 2.4, prove that the matrix $Q_{ij} := (y_i y_j k(x_i, x_j))_{ij}$ is positive definite.*

7.7 (Geometric Interpretation of Duality in SVC [38] ••) *Prove that the programming problem (7.10), (7.11) has the same solution as (7.22), provided the threshold b is adjusted such that the hyperplane bisects the shortest connection of the two convex hulls. Hint: Show that the latter is the dual of the former. Interpret the result geometrically.*

7.8 (Number of Points Required to Define a Hyperplane •) *From (7.22), argue that no matter what the dimensionality of the space, there can always be situations where two training points suffice to determine the optimal hyperplane.*

7.9 (In-Bound SVs in Soft Margin SVMs •) *Prove that in-bound SVs lie exactly on the margin. Hint: Use the KKT conditions, and proceed analogously to Section 7.3, where it was shown that in the hard margin case, all SVs lie exactly on the margin.*

Argue, moreover, that bound SVs can lie both on or in the margin, and that they will "usually" lie in the margin.

7.10 (Pattern-Dependent Regularization •) *Derive a version of the soft margin classification algorithm which uses different regularization constants C_i for each training example. Start from (7.35), replace the second term by $\frac{1}{m}\sum_{i=1}^{m} C_i \xi_i$, and derive the dual. Discuss both the mathematical form of the result, and possible applications (cf. [462]).*

7.11 (Uncertain Labels ••) *In this chapter, we have been concerned mainly with the case where the patterns are assigned to one of two classes, i.e., $y \in \{\pm 1\}$. Consider now the* *case where the assignment is not strict, i.e., $y \in [-1, 1]$. Modify the soft margin variants of the SV algorithm, (7.34), (7.35) and (7.41), (7.40), such that*

- *whenever $y = 0$, the corresponding pattern has effectively no influence*
- *if all labels are in $\{\pm 1\}$, the original algorithm is recovered*
- *if $|y| < 1$, then the corresponding pattern has less influence than it would have for $|y| = 1$.*

7.12 (SVMs vs. Parzen Windows ○○○) *Develop algorithms that approximate the SVM (soft or hard margin) solution by starting from the Parzen Windows algorithm (Figure 1.1) and sparsifying the expansion of the solution.*

7.13 (Squared Error SVC [111] ●●) *Derive a version of the soft margin classification algorithm which penalizes the errors quadratically. Start from (7.35), replace the second term by $\frac{1}{m}\sum_{i=1}^{m} \xi_i^2$, and derive the dual. Compare the result to the usual C-SVM, both in terms of algorithmic differences and in terms of robustness properties. Which algorithm would you expect to work better for Gaussian-like noise, which one for noise with longer tails (and thus more outliers) (cf. Chapter 3)?*

7.14 (C-SVC with Group Error Penalty ●●) *Suppose the training data are partitioned into ℓ groups,*

$$\begin{aligned} &(\mathbf{x}_1^1, y_1^1), \ldots, (\mathbf{x}_1^{m_1}, y_1^{m_1}) \\ &\quad\vdots \qquad\qquad \vdots \\ &(\mathbf{x}_\ell^1, y_\ell^1), \ldots, (\mathbf{x}_\ell^{m_\ell}, y_\ell^{m_\ell}), \end{aligned} \tag{7.70}$$

where $\mathbf{x}_i^j \in \mathcal{H}$ and $y_i^j \in \{\pm 1\}$ (it is understood that the index i runs over $\{1, 2, \ldots, \ell\}$ and the index j runs over $\{1, 2, \ldots, m_i\}$).

Suppose, moreover, that we would like to count a point as misclassified already if one point belonging to the same group is misclassified.

Design an SV algorithm where each group's penalty equals the slack of the worst point in that group.

Hint: Use the objective function

$$\frac{1}{2}\|\mathbf{w}\|^2 + \sum_i C_i \xi_i, \tag{7.71}$$

and the constraints

$$y_i^j \cdot (\left\langle \mathbf{w}, \mathbf{x}_i^j \right\rangle + b) \geq 1 - \xi_i, \tag{7.72}$$

$$\xi_i \geq 0. \tag{7.73}$$

Show that the dual problem consists of maximizing

$$W(\boldsymbol{\alpha}) = \sum_{i,j} \alpha_i^j - \frac{1}{2} \sum_{i,j,i',j'} \alpha_i^j \alpha_{i'}^{j'} y_i^j y_{i'}^{j'} \left\langle \mathbf{x}_i^j, \mathbf{x}_{i'}^{j'} \right\rangle, \tag{7.74}$$

subject to

$$0 = \sum_{i,j} \alpha_i^j y_i^j, \ 0 \leq \alpha_i^j, \ \textit{and} \ \sum_j \alpha_i^j \leq C_i. \tag{7.75}$$

Argue that typically, only one point per group will become an SV.

Show that C-SVC is a special case of this algorithm.

7.15 (ν-SVC with Group Error Penalty ●●●) *Derive a ν-version of the algorithm in Problem 7.14.*

7.16 (C-SVC vs. ν-SVC ●●) *As a modification of ν-SVC (Section 7.5), compute the dual of $\tau(\mathbf{w}, \boldsymbol{\xi}, \rho) = \|\mathbf{w}\|^2/2 + C(-\nu\rho + (1/m)\sum_{i=1}^m \xi_i)$ (note that in ν-SVC, $C = 1$ is used). Argue that due to the homogeneity of the objective function, the dual solution gets scaled by C, however, the decision function will not change. Hence we may set $C = 1$.*

7.17 (Multi-class vs. Binary SVC [593] ●●) *(i) Prove that the multi-class SVC formulation of (7.59) specializes to the binary C-SVC (7.35) in the case $k = 2$, by using $\mathbf{w}_1 = -\mathbf{w}_2$, $b_1 = -b_2$, and $\xi_i = \frac{1}{2}\xi_i^r$ for pattern $\mathbf{x}_i$ in class r. (ii) Derive the dual of (7.59).*

7.18 (Multi-Class ν-SVC ○○○) *Derive a ν-version of the approach described in Section 7.6.4.*

7.19 (LPM with Constrained Signs ●) *Modify the LPM algorithm such that it is guaranteed that each expansion pattern will have a coefficient v_i whose sign equals the class label y_i. Hint: Do not introduce additional constraints, but eliminate the α_i^* variables and use a different ansatz for the solution.*

7.20 (Multi-Class LPM [593] ●●) *In analogy to Section 7.6.4, develop a multi-class version of the LP machine (Section 7.7).*

7.21 (Version Space [368, 239, 451, 238] ●●●) *Consider hyperplanes passing through the origin, $\{\mathbf{x} | \langle \mathbf{w}, \mathbf{x} \rangle = 0\}$, with weight vectors $\mathbf{w} \in \mathcal{H}$, $\|\mathbf{w}\| = 1$. The set of all such hyperplanes forms a unit sphere in weight space. Each training example $(\mathbf{x}, y) \in \mathcal{H} \times \{\pm 1\}$ splits the sphere into two halves: one that correctly classifies $(\mathbf{x}, y)$, i.e., $\operatorname{sgn} \langle \mathbf{w}, \mathbf{x} \rangle = y$, and one that does not. Each training example thus corresponds to a hemisphere (or, equivalently, an oriented great circle) in weight space, and a training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ corresponds to the intersection of m hemispheres, called the* version space.

1. *Discuss how the distances between the training example and the hyperplane in the two representations are related.*

2. *Discuss the relationship to the idea of the Hough transform [255]. The Hough transform is sometimes used in image processing to detect lines. In a nutshell, each point gets to cast votes in support for all potential lines that are consistent with it, and at the end, the lines can be read off the histogram of votes.*

3. *Prove that if all $\mathbf{x}_i$ have the same norm, the maximum margin weight vector corresponds to the center of the largest $m - 1$-dimensional sphere that fits into version space.*

4. *Construct situations where the center of the above largest sphere will generalize poorly, and compare it to the center of mass of version space, called the* Bayes point.

5. *If you disregard the labels of the training examples, there is no longer a single area on the unit sphere which is distinguished from the others due to its corresponding to the correct labelling. Instead, the sphere is split into a number of cells. Argue that the expectation of the natural logarithm of this number equals the VC entropy (Section 5.5.6).*

7.22 (Kernels on Sets ○○○) *Use the construction of Proposition 2.19 to define a kernel that compares two points* $\mathbf{x}, \mathbf{x}' \in \mathcal{H}$ *by comparing the version spaces (see Problem 7.21) of the labelled examples* $(\mathbf{x}, 1)$ *and* $(\mathbf{x}', 1)$*. Define a prior distribution* P *on the unit sphere in* $\mathcal{H}$*, and discuss the implications of its choice for the induced kernel. What can you say about the connection between this kernel and the kernel* $\langle \mathbf{x}, \mathbf{x}' \rangle$*?*

7.23 (Training Algorithms for ν-SVC ○○○) *Try to come up with efficient training algorithms for* ν*-SVC, building on the material presented in Chapter 10.*

(i) Design a simple chunking algorithm that gradually removes all non-SVs.

(ii) Design a decomposition algorithm.

(iii) Is it possible to modify the SMO algorithm such that it deals with the additional equality constraint that ν*-SVC comes with? What is the smallest set of patterns that you can optimize over without violating the two equality constraints? Can you design a generalized SMO algorithm for this case?*

7.24 (Prior Class Probabilities ●●) *Suppose that it is known a priori that* π_+ *and* π_- *are the probabilities that a pattern belongs to the class* ± 1*, respectively. Discuss ways of modifying the simple classification algorithm described in Section 1.2 to take this information into account.*

7.25 (Choosing C ●●) *Suppose that* R *is a measure for the range of the data in feature space that scales like the length of the points in* $\mathcal{H}$ *(cf. Section 7.8.1). Argue that C should scale like* $1/R^2$.[15] *Hint: consider scaling the data by some* $\gamma > 0$*. How do you have to scale C such that* $f(x) = \langle \mathbf{w}, \Phi(x_j) \rangle + b$ *(where* $\mathbf{w} = \sum_i \alpha_i y_i \Phi(x_i)$*) remains invariant* $(j \in [m])$*?*[16] *Discuss measures* R *that can be used. Why does* $R := \max_j k(x_j, x_j)$ *not make sense for the Gaussian RBF kernel?*

Moreover, argue that in the asymptotic regime, the upper bound on the α_j *should scale with* $1/m$*, justifying the use of* m *in (7.38).*

7.26 (Choosing C, Part II ○○○) *Problem 7.25 does not take into account the class labels, and hence also not the potential overlap of the two classes. Note that this is different in the* ν*-approach, which automatically scales the margin with the noise. Can you modify the recommendation in Problem 7.25 to get a selection criterion for C which takes into account the labels, e.g., in the form of prior information on the noise level?*

[15] Thanks to Olivier Chapelle for this suggestion.

[16] Note that in the ν-parametrization, this scale invariance comes for free.