Support Vector Machines

Charlie Frogner [1]

MIT

2010

[1] Slides mostly stolen from Ryan Rifkin (Google).

Plan

- Regularization derivation of SVMs.
- Geometric derivation of SVMs.
- Practical issues.

The Regularization Setting (Again)

We are given n examples $(x_1, y_1), \ldots, (x_n, y_n)$, with $x_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$ for all i. As mentioned last class, we can find a classification function by solving a regularized learning problem:

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(y_i, f(x_i)) + \lambda ||f||_{\mathcal{H}}^2.$$

Note that in this class we are specifically consider **binary classification**.

The Hinge Loss

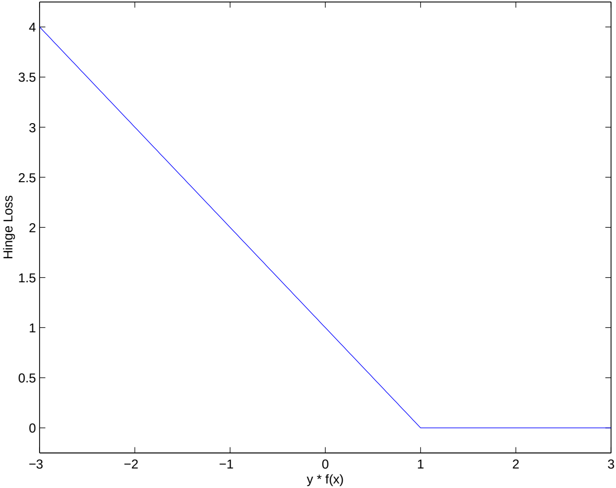
The classical SVM arises by considering the specific loss function

$$V(f(x,y)) \equiv (1 - yf(x))_+,$$

where

$$(k)_+ \equiv \max(k,0).$$

The Hinge Loss



Substituting In The Hinge Loss

With the hinge loss, our regularization problem becomes

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (1 - y_i f(x_i))_+ + \lambda ||f||_{\mathcal{H}}^2.$$

Note that we don't have a $\frac{1}{2}$ multiplier on the regularization term.

Slack Variables

This problem is non-differentiable (because of the “kink” in V), so we introduce slack variables ξ_i, to make the problem easier to work with:

$$\begin{array}{lll} \min\limits_{f \in \mathcal{H}} & \frac{1}{n}\sum_{i=1}^{n} \xi_i + \lambda ||f||_{\mathcal{H}}^2 & \\ \text{subject to :} & y_i f(x_i) \geq 1 - \xi_i & i = 1, \ldots, n \\ & \xi_i \geq 0 & i = 1, \ldots, n \end{array}$$

Applying The Representer Theorem

Substituting in:

$$f^*(x) = \sum_{i=1}^{n} c_i K(x, x_i),$$

we arrive at a constrained quadratic programming problem:

$$\begin{array}{lll}
\min\limits_{c\in\mathbb{R}^n, \xi\in\mathbb{R}^n} & \frac{1}{n}\sum_{i=1}^{n}\xi_i + \lambda c^T K c & \\
\text{subject to}: & y_i \sum_{j=1}^{n} c_j K(x_i, x_j) \geq 1 - \xi_i & i = 1, \ldots, n \\
 & \xi_i \geq 0 & i = 1, \ldots, n
\end{array}$$

Adding A Bias Term

If we add an unregularized bias term b, which presents some theoretical difficulties to be discussed later, we arrive at the “primal” SVM:

$$\begin{array}{rll} \min\limits_{c\in\mathbb{R}^n, b\in\mathbb{R}, \xi\in\mathbb{R}^n} & \frac{1}{n}\sum_{i=1}^{n}\xi_i + \lambda c^T K c & \\ \text{subject to :} & y_i(\sum_{j=1}^{n} c_j K(x_i, x_j) + b) \geq 1 - \xi_i & i = 1, \ldots, n \\ & \xi_i \geq 0 & i = 1, \ldots, n \end{array}$$

Standard Notation

In most of the SVM literature, instead of the regularization parameter λ, regularization is controlled via a parameter C, defined using the relationship

$$C = \frac{1}{2\lambda n}.$$

Using this definition (after multiplying our objective function by the constant $\frac{1}{2\lambda}$, the basic regularization problem becomes

$$\min_{f \in \mathcal{H}} \; C \sum_{i=1}^{n} V(y_i, f(x_i)) + \frac{1}{2} ||f||_{\mathcal{H}}^2.$$

Like λ, the parameter C also controls the tradeoff between classification accuracy and the norm of the function. The primal problem becomes . . .

The Reparametrized Problem

$$\begin{array}{lll}\min\limits_{c\in\mathbb{R}^n, b\in\mathbb{R}, \xi\in\mathbb{R}^n} & C\sum_{i=1}^n \xi_i + \frac{1}{2}c^TKc & \\ \text{subject to :} & y_i(\sum_{j=1}^n c_jK(x_i,x_j)+b) \geq 1-\xi_i & i=1,\dots,n \\ & \xi_i \geq 0 & i=1,\dots,n\end{array}$$

How to Solve?

$$\begin{array}{lll} \min\limits_{c\in\mathbb{R}^n, b\in\mathbb{R}, \xi\in\mathbb{R}^n} & C\sum_{i=1}^n \xi_i + \frac{1}{2}c^T K c & \\ \text{subject to :} & y_i(\sum_{j=1}^n c_j K(x_i, x_j) + b) \geq 1 - \xi_i & i = 1, \dots, n \\ & \xi_i \geq 0 & i = 1, \dots, n \end{array}$$

- This is a constrained optimization problem. The general approach:
  - Form the *primal* problem – we did this.
  - *Lagrangian* from primal – just like Lagrange multipliers.
  - *Dual* – one dual variable associated to each primal constraint in the Lagrangian.

The Reparametrized Lagrangian

We derive the dual from the primal using the Lagrangian:

$$\begin{aligned} L(c,\xi,b,\alpha,\zeta) &= C\sum_{i=1}^{n}\xi_i + c^T Kc \\ &\quad - \sum_{i=1}^{n}\alpha_i(y_i\{\sum_{j=1}^{n}c_j K(x_i,x_j) + b\} - 1 + \xi_i) \\ &\quad - \sum_{i=1}^{n}\zeta_i\xi_i \end{aligned}$$

The Reparametrized Dual, I

$$
\begin{aligned}
\frac{\partial L}{\partial b} \quad &\Longrightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\
\frac{\partial L}{\partial \xi_i} = 0 \quad &\Longrightarrow \quad C - \alpha_i - \zeta_i = 0 \\
&\Longrightarrow \quad 0 \leq \alpha_i \leq C
\end{aligned}
$$

The reduced Lagrangian:

$$L^R(c, \alpha) = c^T K c - \sum_{i=1}^{n} \alpha_i (y_i \sum_{j=1}^{n} c_j K(x_i, x_j) - 1)$$

The relation between c and α:

$$\frac{\partial L}{\partial c} = 0 \Longrightarrow c_i = \alpha_i y_i$$

The Primal and Dual Problems Again

$$
\begin{array}{rll}
\min\limits_{c\in\mathbb{R}^n, b\in\mathbb{R}, \xi\in\mathbb{R}^n} & C\sum_{i=1}^n \xi_i + \frac{1}{2}c^T K c & \\
\text{subject to :} & y_i(\sum_{j=1}^n c_j K(x_i, x_j) + b) \geq 1 - \xi_i & i = 1, \ldots, n \\
 & \xi_i \geq 0 & i = 1, \ldots, n
\end{array}
$$

$$
\begin{array}{rll}
\max\limits_{\alpha\in\mathbb{R}^{\mathbf{n}}} & \sum_{i=1}^n \alpha_i - \frac{1}{2}\alpha^T Q \alpha & \\
\text{subject to :} & \sum_{i=1}^n y_i \alpha_i = 0 & \\
 & 0 \leq \alpha_i \leq C & i = 1, \ldots, n
\end{array}
$$

SVM Training

- Basic idea: solve the dual problem to find the optimal α's, and use them to find b and c:

$$c_i = \alpha_i y_i$$

$$b = y_i - \sum_{j=1}^{n} c_j K(x_i, x_j)$$

  (We showed c_i several slides ago, will show b in a bit.)

- The dual problem is easier to solve the primal problem. It has simple box constraints and a single inequality constraint, and the problem can be decomposed into a sequence of smaller problems (see appendix).

Optimality conditions: complementary slackness

The dual variables are associated with the primal constraints as follows:

$$
\begin{aligned}
\alpha_i &\implies y_i\{\sum_{j=1}^{n} c_j K(x_i, x_j) + b\} - 1 + \xi_i \\
\zeta_i &\implies \xi_i \geq 0
\end{aligned}
$$

Complementary slackness: at optimality, either the primal inequality is satisfied with equality or the dual variable is zero. I.e. if c, ξ, b, α and ζ are optimal solutions to the primal and dual, then

$$
\begin{aligned}
\alpha_i(y_i\{\sum_{j=1}^{n} c_j K(x_i, x_j) + b\} - 1 + \xi_i) &= 0 \\
\zeta_i \xi_i &= 0
\end{aligned}
$$

Optimality Conditions: all of them

All optimal solutions must satisfy:

$$\sum_{j=1}^{n} c_j K(x_i, x_j) - \sum_{j=1}^{n} y_i \alpha_j K(x_i, x_j) = 0 \qquad i = 1, \ldots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$C - \alpha_i - \zeta_i = 0 \qquad i = 1, \ldots, n$$

$$y_i (\sum_{j=1}^{n} y_j \alpha_j K(x_i, x_j) + b) - 1 + \xi_i \geq 0 \qquad i = 1, \ldots, n$$

$$\alpha_i [y_i (\sum_{j=1}^{n} y_j \alpha_j K(x_i, x_j) + b) - 1 + \xi_i] = 0 \qquad i = 1, \ldots, n$$

$$\zeta_i \xi_i = 0 \qquad i = 1, \ldots, n$$

$$\xi_i, \alpha_i, \zeta_i \geq 0 \qquad i = 1, \ldots, n$$

Optimality Conditions, II

The optimality conditions are both necessary and sufficient. If we have *c*, ξ, *b*, α and ζ satisfying the above conditions, we know that they represent optimal solutions to the primal and dual problems. These optimality conditions are also known as the Karush-Kuhn-Tucker (KKT) conditons.

Toward Simpler Optimality Conditions — Determining b

Suppose we have the optimal α_i's. Also suppose (this happens in practice) that there exists an i satisfying $0 < \alpha_i < C$. Then

$$\begin{aligned}
\alpha_i < C &\implies \zeta_i > 0 \\
&\implies \xi_i = 0 \\
&\implies y_i(\sum_{j=1}^{n} y_j \alpha_j K(x_i, x_j) + b) - 1 = 0 \\
&\implies b = y_i - \sum_{j=1}^{n} y_j \alpha_j K(x_i, x_j)
\end{aligned}$$

So if we know the optimal α's, we can determine b.

Towards Simpler Optimality Conditions, I

Defining our classification function $f(x)$ as

$$f(x) = \sum_{i=1}^{n} y_i \alpha_i K(x, x_i) + b,$$

we can derive “reduced” optimality conditions. For example, consider an i such that $y_i f(x_i) < 1$:

$$\begin{aligned} y_i f(x_i) < 1 \quad &\Longrightarrow \quad \xi_i > 0 \\ &\Longrightarrow \quad \zeta_i = 0 \\ &\Longrightarrow \quad \alpha_i = C \end{aligned}$$

Towards Simpler Optimality Conditions, II

Conversely, suppose $\alpha_i = C$:

$$\begin{aligned} \alpha_i = C \quad &\Longrightarrow \quad y_i f(x_i) - 1 + \xi_i = 0 \\ &\Longrightarrow \quad y_i f(x_i) \leq 1 \end{aligned}$$

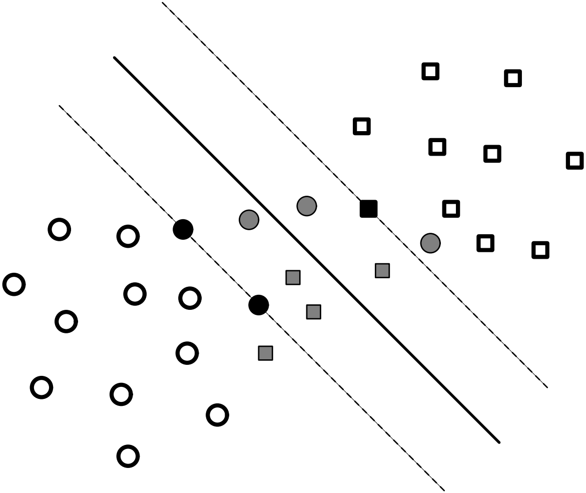
Reduced Optimality Conditions

Proceeding similarly, we can write the following “reduced” optimality conditions (full proof: homework):

$$
\begin{aligned}
\alpha_i = 0 \quad &\Longrightarrow \quad y_i f(x_i) \geq 1 \\
0 < \alpha_i < C \quad &\Longrightarrow \quad y_i f(x_i) = 1 \\
\alpha_i = C \quad &\Longleftarrow \quad y_i f(x_i) < 1
\end{aligned}
$$

$$
\begin{aligned}
\alpha_i = 0 \quad &\Longleftarrow \quad y_i f(x_i) > 1 \\
\alpha_i = C \quad &\Longrightarrow \quad y_i f(x_i) \leq 1
\end{aligned}
$$

Geometric Interpretation of Reduced Optimality Conditions



Summary so far

- The SVM is a Tikhonov regularization problem, with the hinge loss:

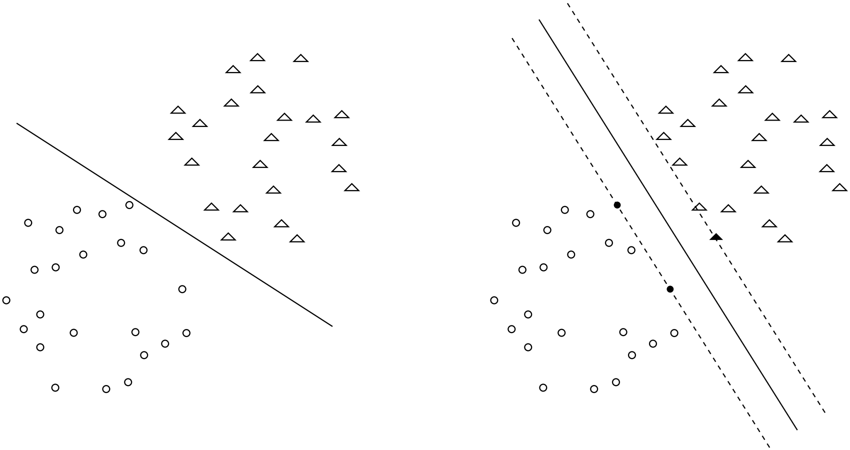
$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (1 - y_i f(x_i))_+ + \lambda ||f||_{\mathcal{H}}^2.$$

- Solving the SVM means solving a constrained quadratic program.
- Solutions can be *sparse* – few non-zero coefficients. This is where the "support vector" in SVM comes from.

The Geometric Approach

The “traditional” approach to developing the mathematics of SVM is to start with the concepts of *separating hyperplanes* and *margin*. The theory is usually developed in a linear space, beginning with the idea of a perceptron, a linear hyperplane that separates the positive and the negative examples. Defining the margin as the distance from the hyperplane to the nearest example, the basic observation is that intuitively, we expect a hyperplane with larger margin to generalize better than one with smaller margin.

Large and Small Margin Hyperplanes



(a) (b)

Classification With Hyperplanes

We denote our hyperplane by w, and we will classify a new point x via the function

$$f(x) = \mathrm{sign}\,(w \cdot x). \tag{1}$$

Given a separating hyperplane w we let x be a datapoint closest to w, and we let x^w be the unique point on w that is closest to x. Obviously, finding a maximum margin w is equivalent to maximizing $||x - x^w||$...

Deriving the Maximal Margin, I

For some k (assume $k > 0$ for convenience),

$$\begin{aligned} w \cdot x &= k \\ w \cdot x^w &= 0 \\ \implies \quad w \cdot (x - x^w) &= k \end{aligned}$$

Deriving the Maximal Margin, II

Noting that the vector $x - x^w$ is parallel to the normal vector w,

$$\begin{aligned}
w \cdot (x - x^w) &= w \cdot \left(\frac{||x - x^w||}{||w||} w \right) \\
&= ||w||^2 \frac{||x - x^w||}{||w||} \\
&= ||w|| \; ||x - x^w|| \\
\Longrightarrow & \quad ||w|| \; ||(x - x^w)|| = k \\
\Longrightarrow & \quad ||x - x^w|| = \frac{k}{||w||}
\end{aligned}$$

Deriving the Maximal Margin, III

k is a “nuisance parameter”. WLOG, we fix k to 1, and see that maximizing $||x - x^w||$ is equivalent to maximizing $\frac{1}{||w||}$, which in turn is equivalent to minimizing $||w||$ or $||w||^2$. We can now define the margin as the distance between the hyperplanes $w \cdot x = 0$ and $w \cdot x = 1$.

The Linear, Homogeneous, Separable SVM

$$
\begin{aligned}
\min_{w \in \mathbb{R}^n} & \quad ||w||^2 \\
\text{subject to}: & \quad y_i(w \cdot x) \geq 1 \quad i = 1, \ldots, n
\end{aligned}
$$

Bias and Slack

The SVM introduced by Vapnik includes an unregularized bias term b, leading to classification via a function of the form:

$$f(x) = \text{sign}\,(w \cdot x + b).$$

In practice, we want to work with datasets that are not linearly separable, so we introduce slacks ξ_i, just as before. We can still define the margin as the distance between the hyperplanes $w \cdot x = 0$ and $w \cdot x = 1$, but this is no longer particularly geometrically satisfying.

The New Primal

With slack variables, the primal SVM problem becomes

$$\begin{aligned} \min_{w\in\mathbb{R}^n,\xi\in\mathbb{R}^n,b\in\mathbb{R}} \quad & C\sum_{i=1}^n \xi_i + \frac{1}{2}||w||^2 & \\ \text{subject to} : \quad & y_i(w\cdot x + b) \geq 1 - \xi_i & i = 1,\dots,n \\ & \xi_i \geq 0 & i = 1,\dots,n \end{aligned}$$

Historical Perspective

Historically, most developments begin with the geometric form, derived a dual program which was identical to the dual we derived above, and only then observed that the dual program required only dot products and that these dot products could be replaced with a kernel function.

More Historical Perspective

In the linearly separable case, we can also derive the separating hyperplane as a vector parallel to the vector connecting the closest two points in the positive and negative classes, passing through the perpendicular bisector of this vector. This was the “Method of Portraits”, derived by Vapnik in the 1970’s, and recently rediscovered (with non-separable extensions) by Keerthi.

Summary

- The SVM is a Tikhonov regularization problem, with the hinge loss:

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (1 - y_i f(x_i))_+ + \lambda ||f||_{\mathcal{H}}^2.$$

- Solving the SVM means solving a constrained quadratic program.
  - It's better to work with the dual program.
- Solutions can be *sparse* – few non-zero coefficients. This is where the “support vector” in SVM comes from.
- There is alternative, geometric interpretation of the SVM, from the perspective of “maximizing the margin.”

Practical issues

- We can also use RLS for classification. What are the tradeoffs?
- SVM possesses sparsity: can have parameters set to zero in the solution. This enables potentially faster training and faster prediction than RLS.
- SVM QP solvers tend to have many parameters to tune.
- SVM can scale to very large datasets, unlike RLS – for the moment (active research topic!).

Good Large-Scale SVM Solvers

- SVM Light: `http://svmlight.joachims.org`
- SVM Torch: `http://www.torch.ch`
- libSVM: `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

Appendix

(Follows.)

SVM Training

Our plan will be to solve the dual problem to find the α's, and use that to find *b* and our function *f*. The dual problem is easier to solve the primal problem. It has simple box constraints and a single inequality constraint, even better, we will see that the problem can be *decomposed* into a sequence of smaller problems.

Off-the-shelf QP software

We can solve QPs using standard software. Many codes are available. Main problem — the Q matrix is dense, and is n-by-n, so we cannot write it down. Standard QP software requires the Q matrix, so is not suitable for large problems.

Decomposition, I

Partition the dataset into a *working set W* and the remaining points R. We can rewrite the dual problem as:

$$\begin{array}{cc}
\max\limits_{\alpha_W \in \mathbb{R}^{|W|},\ \alpha_R \in \mathbb{R}^{|R|}} & \sum\limits^{n}_{\substack{i=1\\ i\in W}} \alpha_i + \sum\limits_{\substack{i=1\\ i\in R}} \alpha_i \\
 & -\frac{1}{2}[\alpha_W \ \alpha_R] \left[\begin{array}{cc} Q_{WW} & Q_{WR} \\ Q_{RW} & Q_{RR} \end{array} \right] \left[\begin{array}{c} \alpha_W \\ \alpha_R \end{array} \right] \\
\text{subject to :} & \sum_{i \in W} y_i \alpha_i + \sum_{i \in R} y_i \alpha_i = 0 \\
 & 0 \leq \alpha_i \leq C, \ \forall i
\end{array}$$

Decomposition, II

Suppose we have a feasible solution α. We can get a better solution by treating the α_W as variable and the α_R as constant. We can solve the reduced dual problem:

$$\begin{array}{rc} \max\limits_{\alpha_W \in \mathbb{R}^{|W|}} & (\mathbf{1} - Q_{WR}\alpha_R)\alpha_W - \frac{1}{2}\alpha_W Q_{WW} \alpha_W \\ \text{subject to :} & \sum_{i \in W} y_i \alpha_i = -\sum_{i \in R} y_i \alpha_i \\ & 0 \leq \alpha_i \leq C, \ \forall i \in W \end{array}$$

Decomposition, III

The reduced problems are fixed size, and can be solved using a standard QP code. Convergence proofs are difficult, but this approach seems to always converge to an optimal solution in practice.

Selecting the Working Set

There are many different approaches. The basic idea is to examine points not in the working set, find points which violate the reduced optimality conditions, and add them to the working set. Remove points which are in the working set but are far from violating the optimality conditions.