Support Vector Machines

Charlie Frogner [1]

MIT

2011

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

Plan

- Regularization derivation of SVMs.
- Analyzing the SVM problem: optimization, duality.
- Geometric derivation of SVMs.
- Practical issues.

The Regularization Setting (Again)

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.
We can find a classification function by solving a regularized learning problem:

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

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

The Hinge Loss

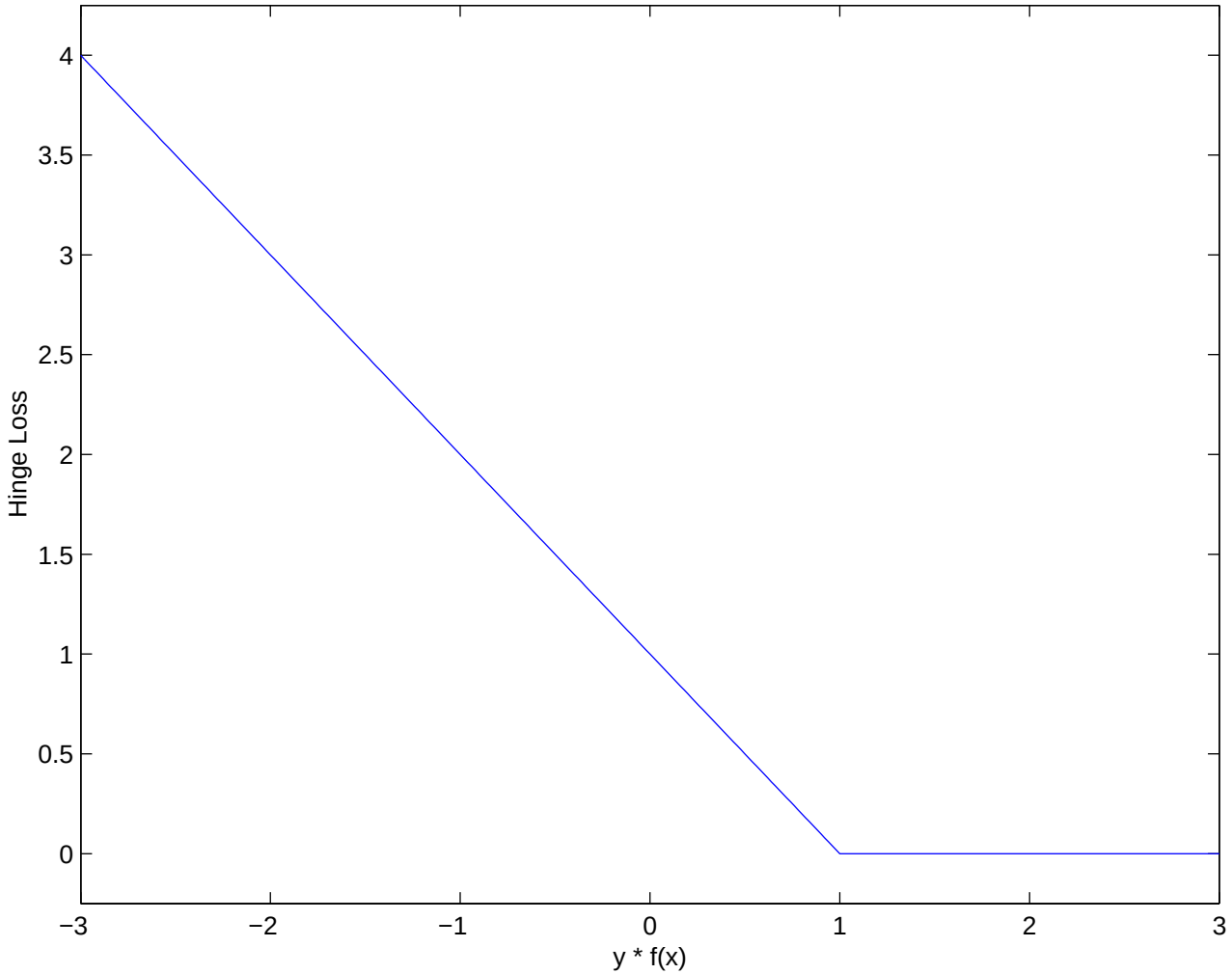
The classical SVM arises by considering the specific loss function

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

where

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

The Hinge Loss



Substituting In The Hinge Loss

With the hinge loss, our regularization problem becomes

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

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 rewrite the "max" function using slack variables ξ_i.

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

Applying The Representer Theorem

Substituting in:

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

we get a constrained quadratic programming problem:

$$\begin{aligned} \underset{c \in \mathbb{R}^n, \xi \in \mathbb{R}^n}{\text{argmin}} \quad & \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \xi_i + \lambda c^T \mathbf{K} c \\ \text{subject to :} \quad & \xi_i \geq 1 - y_i \textstyle\sum_{j=1}^{n} c_j K(x_i, x_j) \quad i = 1, \ldots, n \\ & \xi_i \geq 0 \qquad\qquad\qquad\qquad\quad i = 1, \ldots, n \end{aligned}$$

Adding A Bias Term

Adding an unregularized bias term b (which presents some theoretical difficulties) we get the "primal" SVM:

$$\begin{aligned} \underset{c\in\mathbb{R}^n, b\in\mathbb{R}, \xi\in\mathbb{R}^n}{\text{argmin}} \quad & \tfrac{1}{n}\sum_{i=1}^{n}\xi_i + \lambda c^T \mathbf{K} c \\ \text{subject to :} \quad & \xi_i \geq 1 - y_i\left(\sum_{j=1}^{n} c_j K(x_i, x_j) + b\right) \quad i = 1, \ldots, n \\ & \xi_i \geq 0 \quad i = 1, \ldots, n \end{aligned}$$

Standard Notation

In most of the SVM literature, instead of λ, a parameter C is used to control regularization:

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

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

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

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

$$\underset{c\in\mathbb{R}^n, b\in\mathbb{R}, \xi\in\mathbb{R}^n}{\text{argmin}} \quad C\sum_{i=1}^{n}\xi_i + \frac{1}{2}c^T\mathbf{K}c$$

$$\text{subject to :} \quad \xi_i \geq 1 - y_i\left(\sum_{j=1}^{n} c_j K(x_i, x_j) + b\right) \quad i = 1, \ldots, n$$

$$\xi_i \geq 0 \quad i = 1, \ldots, n$$

How to Solve?

$$\underset{c\in\mathbb{R}^n, b\in\mathbb{R}, \xi\in\mathbb{R}^n}{\text{argmin}} \quad C\sum_{i=1}^{n}\xi_i + \frac{1}{2}c^T\mathbf{K}c$$

$$\text{subject to :} \quad \xi_i \geq 1 - y_i\left(\sum_{j=1}^{n} c_j K(x_i, x_j) + b\right) \quad i = 1, \ldots, n$$

$$\xi_i \geq 0 \quad i = 1, \ldots, n$$

- 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.

Lagrangian

We derive the dual from the primal using the Lagrangian:

$$
\begin{aligned}
L(c, \xi, b, \alpha, \zeta) &= C \sum_{i=1}^{n} \xi_i + \frac{1}{2} c^T \mathbf{K} c \\
&\quad - \sum_{i=1}^{n} \alpha_i (y_i \{ \sum_{j=1}^{n} c_j K(x_i, x_j) + b \} - 1 + \xi_i) \\
&\quad - \sum_{i=1}^{n} \zeta_i \xi_i
\end{aligned}
$$

Dual I

Dual problem is:

$$\underset{\alpha,\zeta\geq 0}{\operatorname{argmax}} \inf_{c,\xi,b} L(c,\xi,b,\alpha,\zeta)$$

First, minimize L w.r.t. (c,ξ,b):

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

Dual II

Dual:

$$\underset{\alpha,\zeta\geq 0}{\operatorname{argmax}} \inf_{c,\xi,b} L(c,\xi,b,\alpha,\zeta)$$

Optimality conditions:

$$\begin{aligned} (1) \quad & c_i = \alpha_i y_i \\ (2) \quad & \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0 \\ (3) \quad & \alpha_i \in [0, C] \end{aligned}$$

Plug in (2) and (3):

$$\underset{\alpha\geq 0}{\operatorname{argmax}} \inf_{c} L(c,\alpha) = \frac{1}{2} c^T \mathbf{K} c + \sum_{i=1}^{n} \alpha_i \left(1 - y_i \sum_{j=1}^{n} K(x_i, x_j) c_j \right)$$

Dual II

Dual:

$$\underset{\alpha,\zeta \geq 0}{\operatorname{argmax}} \inf_{c,\xi,b} L(c, \xi, b, \alpha, \zeta)$$

Optimality conditions:

$$\begin{aligned} (1) \quad & c_i = \alpha_i y_i \\ (2) \quad & \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0 \\ (3) \quad & \alpha_i \in [0, C] \end{aligned}$$

Plug in (1):

$$\begin{aligned} \underset{\alpha \geq 0}{\operatorname{argmax}} L(\alpha) &= \textstyle\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i y_i K(x_i, x_j) \alpha_j y_j \\ &= \textstyle\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \alpha^T (\mathrm{diag}\mathbf{Y}) \mathbf{K} (\mathrm{diag}\mathbf{Y}) \alpha \end{aligned}$$

The Primal and Dual Problems Again

$$\underset{c\in\mathbb{R}^n, b\in\mathbb{R}, \xi\in\mathbb{R}^n}{\text{argmin}} \quad C\sum_{i=1}^{n}\xi_i + \frac{1}{2}c^T\mathbf{K}c$$

$$\text{subject to :} \quad \xi_i \geq 1 - y_i\left(\sum_{j=1}^{n} c_j K(x_i, x_j) + b\right) \quad i = 1, \ldots, n$$

$$\xi_i \geq 0 \quad i = 1, \ldots, n$$

$$\max_{\alpha\in\mathbb{R}^n} \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\alpha^T\mathbf{Q}\alpha$$

$$\text{subject to :} \quad \sum_{i=1}^{n} y_i\alpha_i = 0$$

$$0 \leq \alpha_i \leq C \quad i = 1, \ldots, n$$

SVM Training

- Basic idea: solve the dual problem to find the optimal α's, and use them to find b and c.
- The dual problem is easier to solve the primal problem. It has simple box constraints and a single equality constraint, and the problem can be decomposed into a sequence of smaller problems (see appendix).

Interpreting the solution

α tells us:

- c and b.
- The identities of the misclassified points.

How to analyze? Use the *optimality conditions*.

- Already used: derivative of L w.r.t. (c, ξ, b) is zero at optimality.
- Haven't used: complementary slackness, primal/dual constraints.

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

These optimality conditions are both necessary and sufficient for optimality: $(c, \xi, b, \alpha, \zeta)$ satisfy all of the conditions if and only if they are optimal for both the primal and the dual. (Also known as the Karush-Kuhn-Tucker (KKT) conditons.)

Interpreting the solution — c

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

Interpreting the solution — b

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

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

Interpreting the solution — sparsity

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

$$
\begin{aligned}
y_i f(x_i) > 1 \quad &\Rightarrow \quad (1 - y_i f(x_i)) < 0 \\
&\Rightarrow \quad \xi_i \neq (1 - y_i f(x_i)) \\
&\Rightarrow \quad \alpha_i = 0
\end{aligned}
$$

Interpreting the solution —- support vectors

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

Interpreting the solution — support vectors

So

$$y_i f(x_i) < 1 \Rightarrow \alpha_i = C.$$

Conversely, suppose $\alpha_i = C$:

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

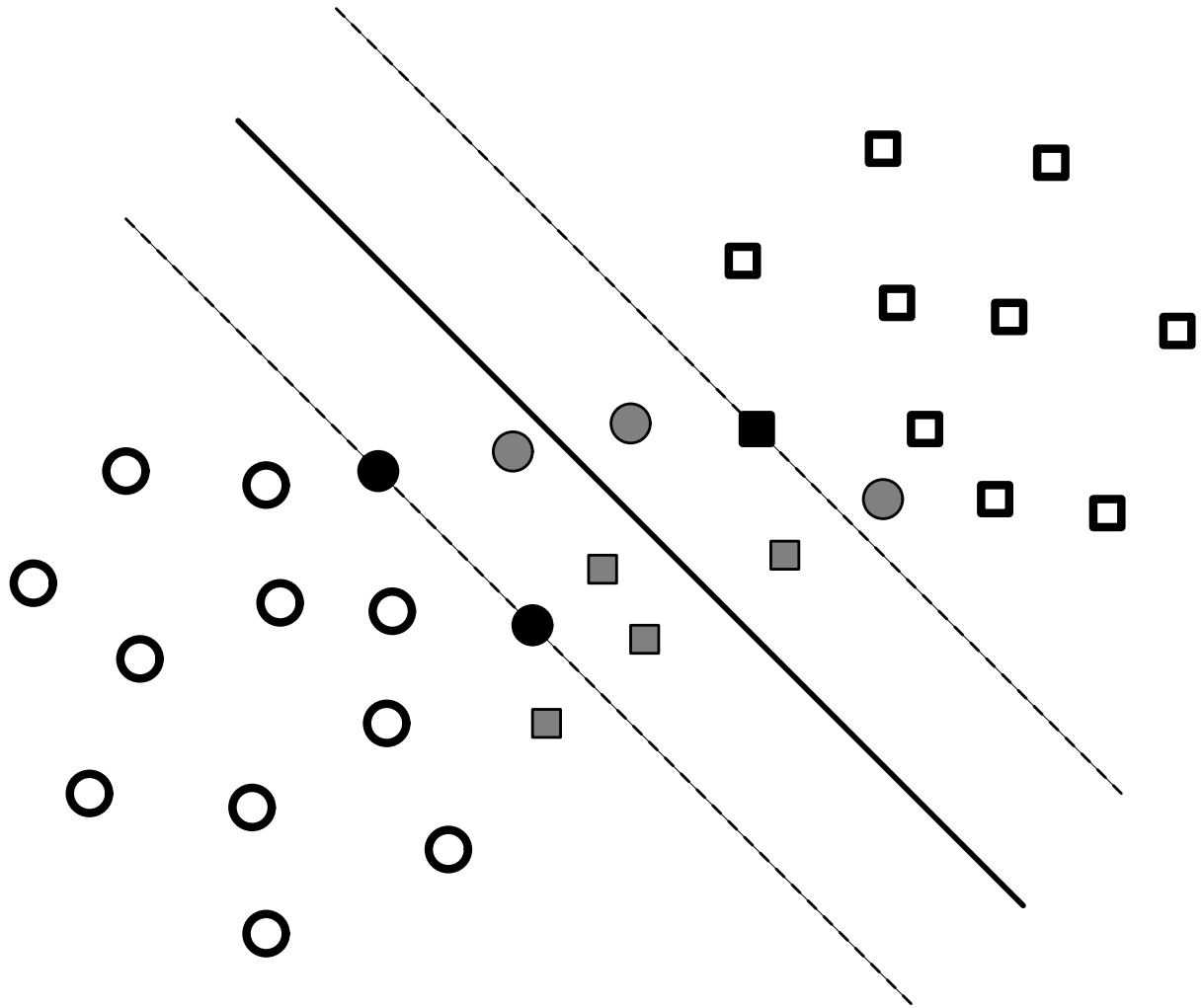
Interpreting the solution

Here are all of the derived conditions:

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

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

Geometric Interpretation of Reduced Optimality Conditions



Summary so far

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

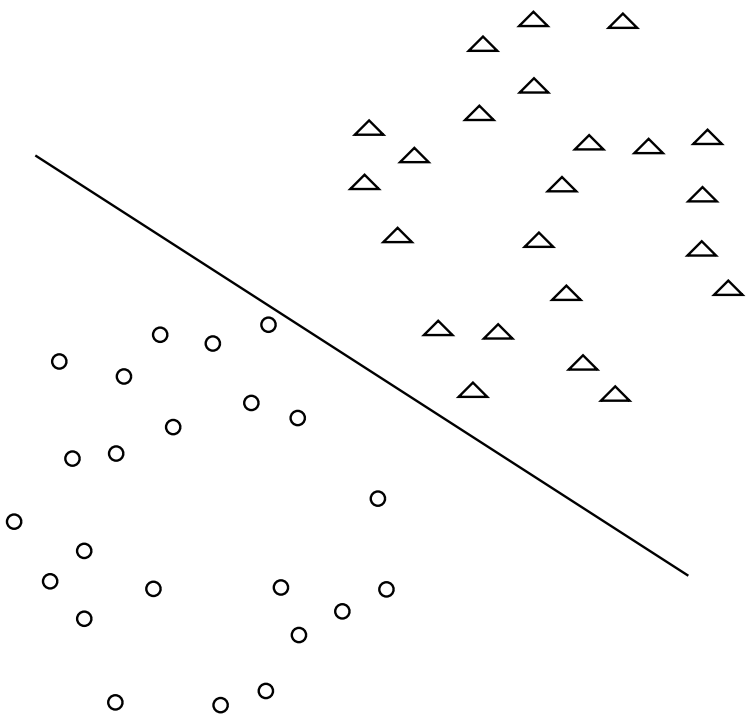
$$\underset{f \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (1 - y_i f(x_i))_+ + \lambda ||f||_{\mathcal{H}}^2.$$

- Solving the SVM means solving a constrained quadratic program.
- Solutions can be *sparse* – some coefficients are zero.
- The nonzero coefficients correspond to points that aren't classified correctly enough – this is where the "support vector" in SVM comes from.

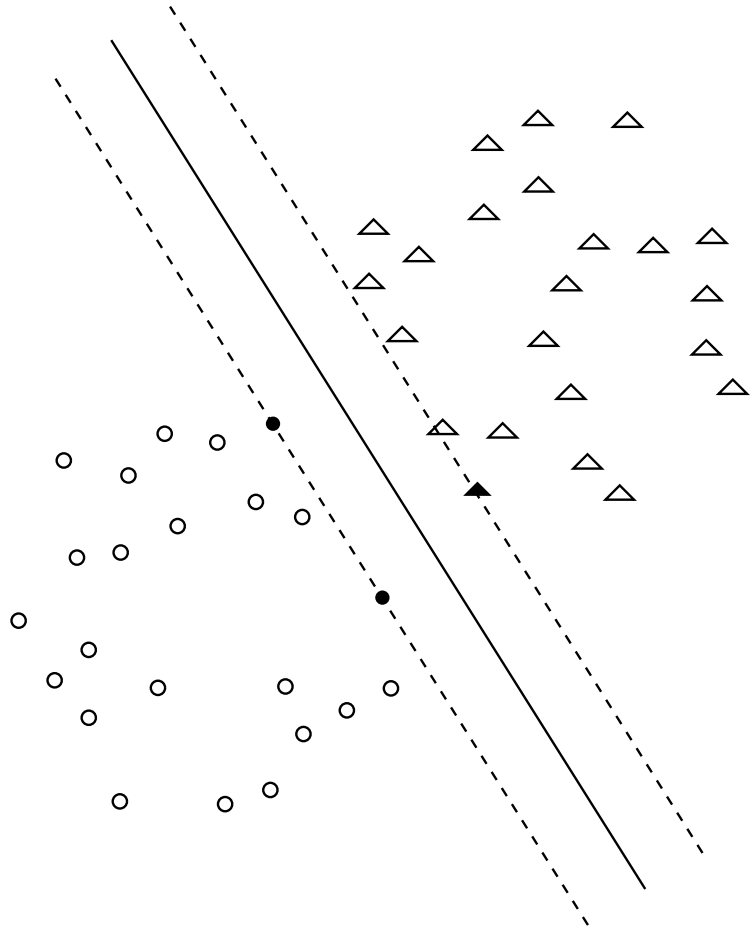
The Geometric Approach

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)

Maximal Margin Classification

Classification function:

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

w is a normal vector to the hyperplane separating the classes. We define the boundaries of the margin by $\langle w, x \rangle = \pm 1$.

What happens as we change $\|w\|$?

We push the margin in/out by rescaling w – the margin moves out with $\frac{1}{\|w\|}$. So maximizing the margin corresponds to minimizing $\|w\|$.

Maximal Margin Classification

Classification function:

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

w is a normal vector to the hyperplane separating the classes. We define the boundaries of the margin by $\langle w, x \rangle = \pm 1$.

What happens as we change $\|w\|$?

We push the margin in/out by rescaling w – the margin moves out with $\frac{1}{\|w\|}$. So maximizing the margin corresponds to minimizing $\|w\|$.

Maximal Margin Classification, Separable case

Separable means $\exists w$ s.t. all points are beyond the margin, i.e.

$$y_i \langle w, x_i \rangle \geq 1 \ , \ \forall i.$$

So we solve:

$$\begin{aligned} \underset{w}{\text{argmin}} \quad & \|w\|^2 \\ \text{s.t.} \quad & y_i \langle w, x_i \rangle \geq 1 \ , \ \forall i \end{aligned}$$

Maximal Margin Classification, Non-separable case

Non-separable means there are points on the wrong side of the margin, i.e.

$$\exists i \text{ s.t. } y_i \langle w, x_i \rangle < 1 \ .$$

We add slack variables to account for the wrongness:

$$\begin{aligned} \underset{\xi_i, w}{\text{argmin}} \quad & \sum_{i=1}^{n} \xi_i + \|w\|^2 \\ \text{s.t.} \quad & y_i \langle w, x_i \rangle \geq 1 - \xi_i \ , \ \forall i \end{aligned}$$

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:

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

- 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.
- The non-zero coefficients correspond to points not classified correctly enough – a.k.a. "support vectors."
- 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:

$$\max_{\alpha_W \in \mathbb{R}^{|W|},\ \alpha_R \in \mathbb{R}^{|R|}} \quad \sum_{\substack{i=1 \\ i \in W}}^{n} \alpha_i + \sum_{\substack{i=1 \\ i \in R}} \alpha_i$$

$$-\frac{1}{2}[\alpha_W\ \alpha_R] \begin{bmatrix} Q_{WW} & Q_{WR} \\ Q_{RW} & Q_{RR} \end{bmatrix} \begin{bmatrix} \alpha_W \\ \alpha_R \end{bmatrix}$$

$$\text{subject to :} \quad \sum_{i \in W} y_i \alpha_i + \sum_{i \in R} y_i \alpha_i = 0$$

$$0 \leq \alpha_i \leq C,\ \forall i$$

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:

$$\begin{aligned} \max_{\alpha_W \in \mathbb{R}^{|W|}} \quad & (\mathbf{1} - Q_{WR}\alpha_R)\alpha_W - \tfrac{1}{2}\alpha_W Q_{WW}\alpha_W \\ \text{subject to :} \quad & \textstyle\sum_{i \in W} y_i \alpha_i = -\sum_{i \in R} y_i \alpha_i \\ & 0 \leq \alpha_i \leq C, \ \forall i \in W \end{aligned}$$

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.