SVM Algebra (Duality)

David Helmbold

University of California, Santa Cruz
dph@soe.ucsc.edu

Fall '12, revised F'15. Note: still uses y's instead of t's

Problem note: using y's instead of t's

Given a set of labeled examples, $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)$ where each $\mathbf{x}_i \in \mathbb{R}^d$ and each $y_i \in \{+1, -1\}$, find a weight vector $\mathbf{w}$ and intercept b such that $\mathrm{sign}(\mathbf{w} \bullet \mathbf{x}_i + b) = y_i$ for all i. (assume linearly separable)

Want to maximize the minimum *margin*, but

$$\max_{\mathbf{w}, b} \min_i y_i(\mathbf{w} \bullet \mathbf{x}_i + b)$$

is not well defined (consider doubling $\mathbf{w}$ and b).

functional margin $= y(\mathbf{w} \bullet \mathbf{x} + b)$ depends on scaling

geometric margin $=$ distance between point and hyperplane

$$= \frac{y(\mathbf{w} \bullet \mathbf{x} + b)}{||\mathbf{w}||_2}$$

Want to maximize geometric margin: $\min_i \frac{y_i(\mathbf{w} \bullet \mathbf{x}_i + b)}{\|\mathbf{w}\|_2}$

Equivalent to:

$$\min_{\mathbf{w},b} \|\mathbf{w}\|_2 \qquad \text{subject to} \qquad y_i(\mathbf{w} \bullet \mathbf{x}_i + b) \geqslant 1 \text{ for all } i,$$

and to:

$$\min_{\mathbf{w},b} \frac{1}{2}(\mathbf{w} \bullet \mathbf{w}) \qquad \text{subject to} \qquad 0 \geqslant 1 - y_i(\mathbf{w} \bullet \mathbf{x}_i + b) \text{ for all } i.$$

Original:

$$\min_{\mathbf{w},b} \|\mathbf{w}\|_2 \qquad \text{subject to} \qquad y_i(\mathbf{w} \bullet \mathbf{x}_i + b) \geqslant 1 \text{ for all } i,$$

Primal problem:

$$\min_{\mathbf{w},b} \max_{\boldsymbol{\alpha} \succeq \mathbf{0}} \left[\frac{1}{2}(\mathbf{w} \bullet \mathbf{w}) + \sum_i \alpha_i \left(1 - y_i(\mathbf{w} \bullet \mathbf{x}_i + b)\right) \right]$$

Dual problem:

$$\max_{\boldsymbol{\alpha} \succeq \mathbf{0}} \min_{\mathbf{w},b} \left[\frac{1}{2}(\mathbf{w} \bullet \mathbf{w}) + \sum_i \alpha_i \left(1 - y_i(\mathbf{w} \bullet \mathbf{x}_i + b)\right) \right]$$

Lagrangian: $L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}(\mathbf{w} \bullet \mathbf{w}) + \sum_i \alpha_i \left(1 - y_i(\mathbf{w} \bullet \mathbf{x}_i + b)\right)$

$$\max_{\boldsymbol{\alpha} \succeq \mathbf{0}} \min_{\mathbf{w}, b} \underbrace{\frac{1}{2}(\mathbf{w} \bullet \mathbf{w}) + \sum_i \alpha_i \left(1 - y_i(\mathbf{w} \bullet \mathbf{x}_i + b)\right)}_{L(\mathbf{w}, b, \boldsymbol{\alpha})}$$

To solve inner min, differentiate $L(\mathbf{w}, b, \boldsymbol{\alpha})$ with respect to $\mathbf{w}$ and b:

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial w_k} = w_k - \sum_i \alpha_i y_i x_{i,k}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_i \alpha_i y_i \mathbf{x}_i \qquad \Rightarrow \quad \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = -\sum_i \alpha_i y_i \qquad \Rightarrow \quad \sum_i \alpha_i y_i = 0$$

Interesting!

$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$ means $\mathbf{w}$ is a weighted sum of examples (like perceptron)

$\sum_i \alpha_i y_i = 0$ means positive and negative examples have same total weight

Karush-Kuhn-Tucker conditions imply that for each constraint term

$$\alpha_i \left(1 - y_i(\mathbf{w} \bullet \mathbf{x}_i + b)\right)$$

<u>if</u> $\alpha_i \neq 0$ <u>then</u> the constraint is tight (i.e. $y_i(\mathbf{w} \bullet \mathbf{x}_i + b) = 1$), so ...

$\alpha_i > 0$ only when $\mathbf{x}_i$ is a support vector, and

$\mathbf{w}$ is a weighted sum of (signed) *support vectors*.

Get ready to plug into $L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}(\mathbf{w} \bullet \mathbf{w}) + \sum_i \alpha_i \left(1 - y_i(\mathbf{w} \bullet \mathbf{x}_i + b)\right)$

$$\mathbf{w} \bullet \mathbf{w} = \underbrace{\left(\sum_i \alpha_i y_i \mathbf{x}_i\right)}_{\mathbf{w}} \bullet \left(\sum_j \alpha_j y_j \mathbf{x}_j\right) = \sum_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \bullet \mathbf{x}_j)$$

$$\begin{aligned}
\sum_i \alpha_i \left(y_i(\mathbf{w} \bullet \mathbf{x}_i + b)\right) &= \sum_i \alpha_i y_i \left(\sum_j \alpha_j y_j \mathbf{x}_j\right) \bullet \mathbf{x}_i + \sum_i \alpha_i y_i b \\
&= \sum_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \bullet \mathbf{x}_j) + b \underbrace{\sum_i \alpha_i y_i}_{0}
\end{aligned}$$

So,

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}(\mathbf{w} \bullet \mathbf{w}) + \sum_i \alpha_i \left(1 - y_i(\mathbf{w} \bullet \mathbf{x}_i + b)\right)$$

$$L_{\text{subs}}(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \bullet \mathbf{x}_j) + \sum_i \alpha_i - \sum_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \bullet \mathbf{x}_j)$$

$$L_{\text{subs}}(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \bullet \mathbf{x}_j)$$

and we want to maximize $L_{\text{subs}}(\boldsymbol{\alpha})$ over $\boldsymbol{\alpha}$ (where each $\alpha_i \geqslant 0$).
This is a quadratic programming problem - can be done numerically.
From $\boldsymbol{\alpha}^*$, compute $\mathbf{w}^* = \sum_i \alpha_i^* y_i \mathbf{x}_i$, set b to "split the difference"

$$b^* = \frac{-1}{2}\left(\min_{i:y_i=+1}(\mathbf{w}^* \bullet \mathbf{x}_i) + \max_{j:y_i=-1}(\mathbf{w}^* \bullet \mathbf{x}_j)\right)$$

Sparseness

- Only for support vectors are α_i non-zero – usually few support vectors.
- Removing labeled examples only changes hypothesis if a support vector removed.
- If ℓ of m examples are support vectors, then m-fold cross validation (leave-one-out) error estimate is $\leqslant \ell/m$.
- Gives an expected error bound of ℓ/m.

Uses instances *only* through dot-product

- Optimization of $\boldsymbol{\alpha}$:

$$\max_{\boldsymbol{\alpha} \succeq \mathbf{0}} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \bullet \mathbf{x}_j)$$

- Prediction on new (or old) instance $\mathbf{x}$:

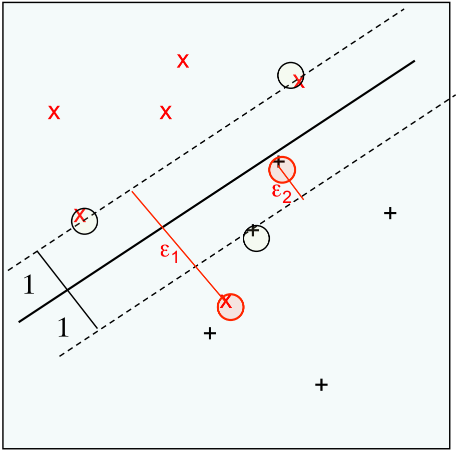
$$\mathbf{w}^* \bullet \mathbf{x} + b = \left(\sum_i \alpha_i^* y_i \mathbf{x}_i \right) \bullet \mathbf{x} + b = \sum_i \alpha_i^* y_i (\mathbf{x}_i \bullet \mathbf{x}) + b$$

- Even finding b:

$$b^* = \frac{-1}{2} \left(\min_{i: y_i = +1} (\mathbf{w}^* \bullet \mathbf{x}_i) + \max_{j: y_j = -1} (\mathbf{w}^* \bullet \mathbf{x}_j) \right)$$

Softmargin Idea

- Data doesn't always have good margin
- Allow Margin errors (imperfect classification)
- Let $\xi_i \geqslant 0$ be error on $\mathbf{x}_i$
- *Hinge loss* is 0 when margin $= 1$, increases linearly as margin drops
- trade off accuracy and sum of "errors"



Optimization problem (with trade-off C):

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}(\mathbf{w} \bullet \mathbf{w}) + C \sum_i \xi_i$$

subject to $y_i(\mathbf{w} \bullet \mathbf{x}_i + b) + \xi_i \geqslant 1$ and $\xi_i \geqslant 0$ for all i.

Lagrangian:

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}(\mathbf{w} \bullet \mathbf{w}) + C \sum_i \xi_i + \sum_i \alpha_i \left(1 - y_i(\mathbf{w} \bullet \mathbf{x}_i + b) - \xi_i\right) - \sum_i \mu_i \xi_i$$

After solving for $\mathbf{w}$, b, $\boldsymbol{\xi}$ we get $\mu_i = C - \alpha_i$ and dual problem

$$\max_{\boldsymbol{\alpha} \succeq \mathbf{0}} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \bullet \mathbf{x}_j)$$

subject to $0 \leqslant \alpha_i \leqslant C$ for all i.