Support Vector Machines

David Rosenberg

New York University

February 21, 2017

The SVM as a Quadratic Program

The Margin

Definition

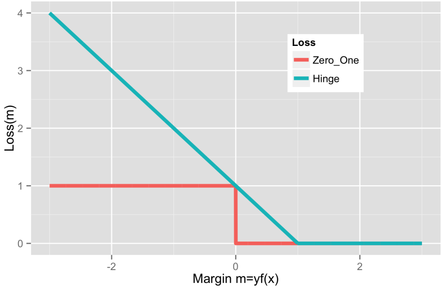
The **margin** (or **functional margin**) for predicted score $\hat{y}$ and true class $y \in \{-1, 1\}$ is $y\hat{y}$.

- The margin often looks like $yf(x)$, where $f(x)$ is our score function.
- The margin is a measure of how **correct** we are.
- We want to **maximize the margin**.
- Most classification losses depend only on the margin.

(This is distinct from but related to **geometric margin** from lab.)

Hinge Loss

- SVM/Hinge loss: $\ell_{\text{Hinge}} = \max\{1-m, 0\} = (1-m)_+$
- Margin $m = yf(x)$; "Positive part" $(x)_+ = x\mathbb{1}(x \geqslant 0)$.



Hinge is a **convex**, **upper bound** on $0-1$ loss. Not differentiable at $m = 1$.
We have a **"margin error"** when $m < 1$.

Support Vector Machine

- Hypothesis space $\mathcal{F} = \left\{ f(x) = w^T x + b \mid w \in \mathbf{R}^d, b \in \mathbf{R} \right\}$.
- ℓ_2 regularization (Tikhonov style)
- Loss $\ell(m) = \max\{1 - m, 0\} = (1 - m)_+$
- The SVM prediction function is the solution to

$$\min_{w \in \mathbf{R}^d, b \in \mathbf{R}} \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i\left[w^T x_i + b\right]\right).$$

SVM Optimization Problem (Tikhonov Version)

The SVM prediction function is the solution to

$$\min_{w \in \mathbf{R}^d, b \in \mathbf{R}} \frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i\left[w^T x_i + b\right]\right).$$

- unconstrained optimization
- not differentiable because of the max (right at the border of a margin error)
- Can we reformulate into a differentiable problem?

SVM Optimization Problem

- The SVM optimization problem is equivalent to

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i \\
\text{subject to} & \xi_i \geqslant \max\left(0, 1 - y_i\left[w^T x_i + b\right]\right).
\end{array}
$$

- Which is equivalent to

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i \\
\text{subject to} & \xi_i \geqslant \left(1 - y_i\left[w^T x_i + b\right]\right) \text{ for } i = 1, \ldots, n \\
 & \xi_i \geqslant 0 \text{ for } i = 1, \ldots, n
\end{array}
$$

SVM as a Quadratic Program

- The SVM optimization problem is equivalent to

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i \\
\text{subject to} & -\xi_i \leqslant 0 \text{ for } i = 1,\ldots,n \\
 & \left(1 - y_i\left[w^T x_i + b\right]\right) - \xi_i \leqslant 0 \text{ for } i = 1,\ldots,n
\end{array}
$$

- Differentiable objective function
- $n+d+1$ unknowns and $2n$ affine constraints.
- A quadratic program that can be solved by any off-the-shelf QP solver.
- Let's learn more by examining the dual.

The SVM Dual Problem

SVM Lagrange Multipliers

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^n \xi_i \\ \text{subject to} \quad & -\xi_i \leqslant 0 \text{ for } i = 1, \ldots, n \\ & \left(1 - y_i\left[w^T x_i + b\right]\right) - \xi_i \leqslant 0 \text{ for } i = 1, \ldots, n \end{aligned}$$

Lagrange Multiplier	Constraint
λ_i	$-\xi_i \leqslant 0$
α_i	$\left(1 - y_i\left[w^T x_i + b\right]\right) - \xi_i \leqslant 0$

$$L(w, b, \xi, \alpha, \lambda) = \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i \left(1 - y_i\left[w^T x_i + b\right] - \xi_i\right) + \sum_{i=1}^n \lambda_i \left(-\xi_i\right)$$

SVM Lagrangian

- The Lagrangian for this formulation is

$$
\begin{aligned}
& L(w,b,\xi,\alpha,\lambda) \\
= \quad & \frac{1}{2}\|w\|^2+\frac{c}{n}\sum_{i=1}^n \xi_i+\sum_{i=1}^n \alpha_i\left(1-y_i\left[w^Tx_i+b\right]-\xi_i\right)-\sum_i \lambda_i\xi_i \\
= \quad & \frac{1}{2}w^Tw+\sum_{i=1}^n \xi_i\left(\frac{c}{n}-\alpha_i-\lambda_i\right)+\sum_{i=1}^n \alpha_i\left(1-y_i\left[w^Tx_i+b\right]\right).
\end{aligned}
$$

- Primal and dual:

$$
\begin{aligned}
p^* &= \inf_{w,\xi,b}\sup_{\alpha,\lambda\succeq 0} L(w,b,\xi,\alpha,\lambda) \\
&\geqslant \sup_{\alpha,\lambda\succeq 0}\inf_{w,b,\xi} L(w,b,\xi,\alpha,\lambda)=d^*
\end{aligned}
$$

- Do we have $p^*=d^*$?

Strong Duality by Slater's constraint qualification

- The SVM optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & -\xi_i \leqslant 0 \text{ for } i = 1, \ldots, n \\ & \left(1 - y_i \left[w^T x_i + b\right]\right) - \xi_i \leqslant 0 \text{ for } i = 1, \ldots, n \end{aligned}$$

- Convex problem + affine constraints $\implies$ strong duality iff problem is feasible
- Constraints are satisfied by $w = b = 0$ and $\xi_i = 1$ for $i = 1, \ldots, n$,
    - so **we have strong duality** $\implies$

$$\begin{aligned} p^* &= \inf_{w,\xi,b} \sup_{\alpha,\lambda \succeq 0} L(w, b, \xi, \alpha, \lambda) \\ &= \sup_{\alpha,\lambda \succeq 0} \inf_{w,b,\xi} L(w, b, \xi, \alpha, \lambda) = d^* \end{aligned}$$

SVM Dual Function

- Lagrange dual is the inf over primal variables of the Lagrangian:

$$
\begin{aligned}
g(\alpha,\lambda) &= \inf_{w,b,\xi} L(w,b,\xi,\alpha,\lambda) \\
&= \inf_{w,b,\xi} \left[\frac{1}{2} w^T w + \sum_{i=1}^n \xi_i \left(\frac{c}{n} - \alpha_i - \lambda_i \right) + \sum_{i=1}^n \alpha_i \left(1 - y_i \left[w^T x_i + b \right] \right) \right]
\end{aligned}
$$

- Taking inf of convex and differentiable function of w, b, ξ.
  - Quadratic in w and linear in ξ and b.
- Thus optimal point iff $\partial_w L = 0$ $\partial_b L = 0$ $\partial_\xi L = 0$
- Note: $g(\alpha,\lambda) = -\infty$ when $\frac{c}{n} - \alpha_i - \lambda_i \neq 0$. (send $\xi_i \to \pm\infty$). This inf is NOT an optimum because it is never attained.

SVM Dual Function: First Order Conditions

Lagrange dual function is the inf over primal variables of L:

$$
\begin{aligned}
g(\alpha,\lambda) &= \inf_{w,b,\xi} L(w,b,\xi,\alpha,\lambda) \\
&= \inf_{w,b,\xi}\left[\frac{1}{2}w^Tw+\sum_{i=1}^n \xi_i\left(\frac{c}{n}-\alpha_i-\lambda_i\right)+\sum_{i=1}^n \alpha_i\left(1-y_i\left[w^Tx_i+b\right]\right)\right]
\end{aligned}
$$

$$\partial_w L = 0 \iff w-\sum_{i=1}^n \alpha_i y_i x_i = 0 \iff \boxed{w=\sum_{i=1}^n \alpha_i y_i x_i}$$

$$\partial_b L = 0 \iff -\sum_{i=1}^n \alpha_i y_i = 0 \iff \boxed{\sum_{i=1}^n \alpha_i y_i = 0}$$

$$\partial_{\xi_i} L = 0 \iff \frac{c}{n}-\alpha_i-\lambda_i = 0 \iff \boxed{\alpha_i+\lambda_i=\frac{c}{n}}$$

SVM Dual Function

- Substituting these conditions back into L, the second term disappears.
- First and third terms become

$$\frac{1}{2}w^Tw = \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j$$

$$\sum_{i=1}^{n}\alpha_i(1-y_i\left[w^Tx_i+b\right]) = \sum_{i=1}^{n}\alpha_i-\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_j^T x_i - b\underbrace{\sum_{i=1}^{n}\alpha_i y_i}_{=0}.$$

- Putting it together, the dual function is

$$g(\alpha,\lambda)=\begin{cases}\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_j^T x_i & \begin{array}{c}\sum_{i=1}^{n}\alpha_i y_i=0\\ \alpha_i+\lambda_i=\frac{c}{n},\text{ all } i\end{array}\\ -\infty & \text{otherwise.}\end{cases}$$

SVM Dual Problem

- The **dual function** is

$$g(\alpha,\lambda)=\begin{cases}\sum_{i=1}^n\alpha_i-\frac{1}{2}\sum_{i,j=1}^n\alpha_i\alpha_jy_iy_jx_j^Tx_i & \begin{matrix}\sum_{i=1}^n\alpha_iy_i=0\\ \alpha_i+\lambda_i=\frac{c}{n},\text{ all } i\end{matrix}\\ -\infty & \text{otherwise.}\end{cases}$$

- The **dual problem** is $\sup_{\alpha,\lambda\succeq 0}g(\alpha,\lambda)$:

$$\begin{aligned}\sup_{\alpha,\lambda}\quad & \sum_{i=1}^n\alpha_i-\frac{1}{2}\sum_{i,j=1}^n\alpha_i\alpha_jy_iy_jx_j^Tx_i\\ \text{s.t.}\quad & \sum_{i=1}^n\alpha_iy_i=0\\ & \alpha_i+\lambda_i=\frac{c}{n}\quad \alpha_i,\lambda_i\geqslant 0,\ i=1,\ldots,n\end{aligned}$$

SVM Dual Problem: Eliminating a Variable

- Can eliminate the λ variables:

$$
\begin{aligned}
\sup_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_j^T x_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\
& \alpha_i \in \left[0, \frac{c}{n}\right] \; i = 1, \dots, n.
\end{aligned}
$$

- Quadratic objective in n unknowns and $n+1$ constraints
- Efficient minimization algorithm: SMO (sequential minimal optimization)
- Next week we'll see what insights we get from the dual formulation...