Lecture 11
Support Vector Machines III

24 February 2016

Taylor B. Arnold
Yale Statistics
STAT 365/665

Notes:

- Problem 3 is due this Friday
- Problem 4 will be available on the course website later today, due next Friday

Today

- Further exploration of the SVM optimization problem
- Visualizing the effect of kernels and costs
- A more complex example

Review

We have the following specification of a support vector machine:

$$\begin{aligned} \min \quad & \frac{1}{2}||\beta||_2^2 \\ \text{s.t.} \quad & y_i(x_i^t\beta + \beta_0) > 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i > 0, \sum_i \xi_i \leq \text{Constant}. \end{aligned}$$

This defines a margin around the linear decision plane of width $\frac{1}{||\beta||}$, and tries to minimize the number of errors ξ of points that are on the wrong side of the margin.

Review, Dual

Computing the Lagrangian and calculating the dual function, we were able to re-write this as:

$$\max_{\alpha} \sum_{i} \alpha_i - \frac{1}{2} \cdot \sum_{i'} \sum_{i} \alpha_i \alpha_{i'} y_i y_{i'} x_i^t x_{i'}$$
$$\text{s.t.} \quad 0 \leq \alpha_i \leq C, \quad \sum_i \alpha_i y_i = 0.$$

Which defines a quadratic program with box constraints that can be solved by general purpose optimization engines.

Review, Kernel Trick

I also noted that the dual form of the problem only requires that we know the inner product between pairs of samples of the predictor matrix. In this way, if we want to do basis expansion, we only need to define the inner product rather than actually doing projection into a higher space. This is called the kernel trick.

The Kernel Trick, cont.

The projected inner product $< h(x_i), h(x_{i'}) >$ is usually written directly as $K(x_i, x_{i'})$ for a function K called the kernel. Popular choices include:

1. **Linear:** $K(x, x') =< x, x' >$
2. **Polynomial:** $K(x, x') = (1+ < x, x' >)^d$
3. **Radial:** $K(x, x') = exp(-\gamma||x - x'||^2)$
4. **Sigmoid:** $K(x, x') = tanh(\kappa_1 < x, x' > +\kappa_2)$

Notice that these all require approximately the same effort to calculate as the linear kernel.

Review, Representation

One side result of the dual calculation also showed us that the vector β can be written as a weighted sum of the inputs x_i:

$$\beta = \sum_i \alpha_i y_i x_i$$

This is particularly useful when used in conjunction with the kernel trick, where we instead have that:

$$\beta = \sum_i \alpha_i y_i h(x_i)$$

Review, Representation

Now if we want to estimate $h(x_k)^t\beta$ in order to do prediction, we again do not need to project into a higher dimensional space by can just use:

$$\begin{aligned} h(x_k)^t\beta &= \sum_i \alpha_i y_i h(x_k)^t h(x_i) \\ &= \sum_i \alpha_i y_i K(x_k, x_i) \end{aligned}$$

For this reason most support vector machine implementation usual store α or $\alpha \cdot y$ rather than β itself as this generalizes better to the kernel case.

Rethinking the primal problem

Let's return for a moment to the original primal problem we constructed:

$$\begin{aligned} \min \quad & \frac{1}{2}||\beta||_2^2 + C \cdot \sum_i \xi_i \\ \text{s.t.} \quad & y_i(x_i^t \beta + \beta_0) > 1 - \xi_i, \quad \xi_i > 0, \quad i = 1, \ldots, n. \end{aligned}$$

Notice that we will have either

$$\xi_i = 1 - y_i(x_i^t \beta + \beta_0) \quad \text{or} \quad \xi_i = 0$$

As otherwise, we could decrease ξ_i further and improve the objective function without breaking the constraints. A commonly used notation to specify this is called the positive part, denoted by a plus sign $(+)$ as a subscript:

$$\xi_i = \left[1 - y_i(x_i^t \beta + \beta_0)\right]_+ .$$

Rethinking the primal problem, cont.

We can actually substitute this directly into the primal problem to obtain an unconstrained form of the primal problem:

$$\min_{\beta} \quad \frac{1}{2}||\beta||_2^2 + C \cdot \sum_i \left[1 - y_i(x_i^t\beta + \beta_0)\right]_+$$

Making the substitution $\lambda = \frac{1}{2C}$, this becomes:

$$\min_{\beta} \quad \sum_i \left[1 - y_i(x_i^t\beta + \beta_0)\right]_+ + \lambda||\beta||_2^2$$

Which looks strikingly similar to ridge regression, but with the sum of squares replaced with a different measurement of goodness of fit.

Rethinking the primal problem

The value $[1 - y_i(x_i^t\beta + \beta_0)]_+$ is called the hinge loss. It behaves similarly in spirit to, still importantly different from, squared error or binomial deviance.

Penalized estimators

Using $f(x)$ to represent $X\beta + \beta_0$, we can write linear regression as either:

$$[1 - yf(x)]^2 + \lambda \cdot ||\beta||_2^2$$
$$[f(x) - y]^2 + \lambda \cdot ||\beta||_2^2$$

Logistic regression as:

$$\log[1 + e^{-yf(x)}] + \lambda \cdot ||\beta||_2^2$$

And support vector machines as:

$$[1 - yf(x)]_+ + \lambda \cdot ||\beta||_2^2$$

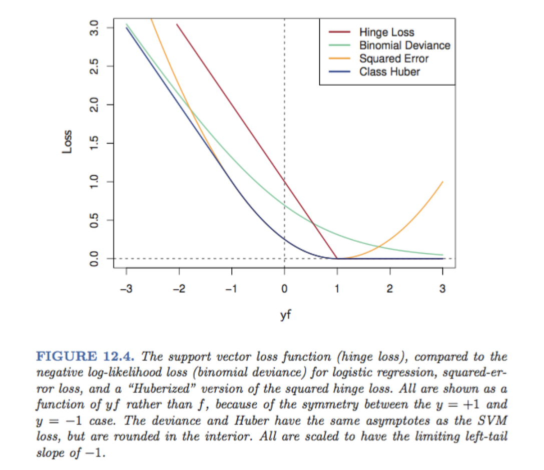


FIGURE 12.4. *The support vector loss function (hinge loss), compared to the negative log-likelihood loss (binomial deviance) for logistic regression, squared-error loss, and a "Huberized" version of the squared hinge loss. All are shown as a function of yf rather than f, because of the symmetry between the $y = +1$ and $y = -1$ case. The deviance and Huber have the same asymptotes as the SVM loss, but are rounded in the interior. All are scaled to have the limiting left-tail slope of -1.*

Kernels and the primal problem

The details go beyond the technical background I have assumed for this course, but it is possible to rewrite this unconstrained primal problem using the kernel trick as well:

$$\min_f \quad \sum_i [1 - y_i f(x_i)]_+ + \lambda ||f||_H^2$$

For a suitably chosen norm $|| \cdot ||_H$.

The actual optimization

We now have two formulations for solving support vector machines: the unconstrained primal problem given as a penalized estimator or the box-constrained dual problem. These can than be solved by applying a number of standard optimization techniques.

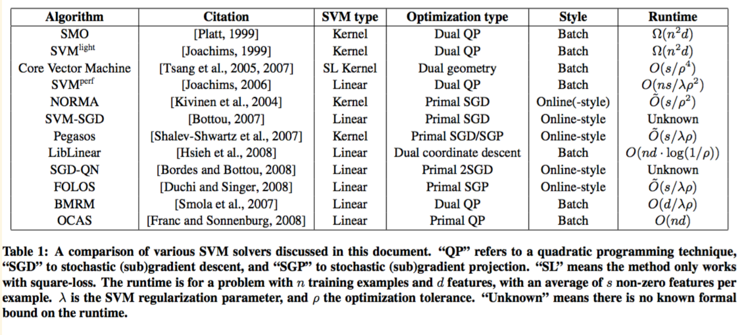
Historically, the dual formulation had been more popular because it was obvious how to deal with kernels and the box-constraints were easier to deal with than the constraints in the unmodified primal problem.

However, with the reformulation of the primal problem as a penalized unconstrained optimization objective, most new work is done on solving the primal problem directly.

The actual optimization, cont.

A well-written summary of recent advances that does not require extensive background knowledge is the following unpublished manuscript:

> *A. K. Menon. Large-scale support vector machines: algorithms and theory. Research Exam, University of California, San Diego, 2009.*
> `https://cseweb.ucsd.edu/~akmenon/ResearchExam.pdf`



Algorithm	Citation	SVM type	Optimization type	Style	Runtime
SMO	[Platt, 1999]	Kernel	Dual QP	Batch	$\Omega(n^2 d)$
SVM$^{\text{light}}$	[Joachims, 1999]	Kernel	Dual QP	Batch	$\Omega(n^2 d)$
Core Vector Machine	[Tsang et al., 2005, 2007]	SL Kernel	Dual geometry	Batch	$O(s/\rho^4)$
SVM$^{\text{perf}}$	[Joachims, 2006]	Linear	Dual QP	Batch	$O(ns/\lambda\rho^2)$
NORMA	[Kivinen et al., 2004]	Kernel	Primal SGD	Online(-style)	$\tilde{O}(s/\rho^2)$
SVM-SGD	[Bottou, 2007]	Linear	Primal SGD	Online-style	Unknown
Pegasos	[Shalev-Shwartz et al., 2007]	Kernel	Primal SGD/SGP	Online-style	$\tilde{O}(s/\lambda\rho)$
LibLinear	[Hsieh et al., 2008]	Linear	Dual coordinate descent	Batch	$O(nd \cdot \log(1/\rho))$
SGD-QN	[Bordes and Bottou, 2008]	Linear	Primal 2SGD	Online-style	Unknown
FOLOS	[Duchi and Singer, 2008]	Linear	Primal SGP	Online-style	$\tilde{O}(s/\lambda\rho)$
BMRM	[Smola et al., 2007]	Linear	Dual QP	Batch	$O(d/\lambda\rho)$
OCAS	[Franc and Sonnenburg, 2008]	Linear	Primal QP	Batch	$O(nd)$

Table 1: A comparison of various SVM solvers discussed in this document. "QP" refers to a quadratic programming technique, "SGD" to stochastic (sub)gradient descent, and "SGP" to stochastic (sub)gradient projection. "SL" means the method only works with square-loss. The runtime is for a problem with n training examples and d features, with an average of s non-zero features per example. λ is the SVM regularization parameter, and ρ the optimization tolerance. "Unknown" means there is no known formal bound on the runtime.

Primal and Dual: Final Thoughts

In terms of the understanding support vector machines, both formulations of the problem are quite important.

The dual brings to light the representation of the support vector machine as a weighted combination of a set of support vectors. It illustrates exactly what properties make a good support vector (dissimilar to other vectors with the same class; similar to those with different classes). It also shows one way of comparing and contrasting it with logistic regression.

The primal problem shows an entirely different way of comparing logistic regression with support vector machines. It also more clearly illustrates the role of the constant C in tuning the final model. The primal also serves as the motivation for important modifications such as replacing the hinge loss with a huberized variant.