Several Views of Support Vector Machines

Ryan M. Rifkin

Honda Research Institute USA, Inc.
Human Intention Understanding Group

2007

Tikhonov Regularization

- We are considering algorithms of the form

$$\min_{f\in\mathcal{H}} \sum_{i=1}^{n} v_i(Y_i) + \frac{\lambda}{2}||f||_K^2. \qquad (1)$$

- Different loss functions lead to different learning problems.
- Last class, we discussed *regularized least squares*, by choosing

$$v_i(y_i) = \frac{1}{2}(Y_i - y_i)^2.$$

- Support vector machines are another Tikhonov regularization algorithm ...

HONDA
The Power of Dreams

SVM Motivation: Problems with RLS

- RLS uses the square loss, which some might say does not "make sense" for classification. SVM uses the hinge loss (defined soon), which does "makes sense."
- Nonlinear RLS does not scale easily to large data sets. The SVM can have better scaling properties.
- The SVM has a (in my opinion weak) geometric motivation: the idea of *margin.*

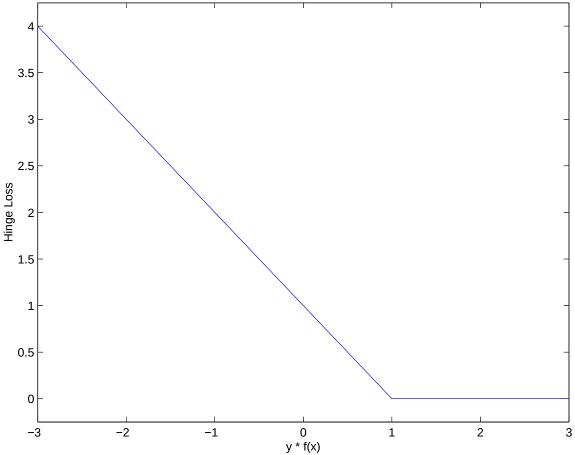
A loss function for classification

- The most natural loss for classification is probably the 0-1 loss: We pay zero if our prediction has the correct sign, and one otherwise (remember that functions in an RKHS make real-valued predictions).
- Unfortunately, the 0-1 loss is not convex. Therefore, we have little hope of being able to optimize this loss function in practice. (Note that the representer theorem *does* hold for the 0-1 loss.)
- A solution: the *hinge loss*, a convex loss that upper bounds the zero-one loss:

$$
\begin{aligned}
v(y) &= \max(1 - yY, 0) \\
&= (1 - yY)_+.
\end{aligned}
$$

HONDA The Power of Dreams

The hinge loss



Value-based SVM

- Substituting the loss function into the definition of Tikhonov regularization, we get an optimization problem

$$\min_{y \in \mathbb{R}^n} \sum_i (1 - y_i Y_i)_+ + \lambda y^t K^{-1} y.$$

- This is (basically) an SVM. So what?
- How will you solve this problem (find the minimizing y)? The hinge loss is not differentiable, so you cannot take the derivative and set it to zero.

Coefficient-based SVM

- Remember that the representer theorem says the answer has the form
$$f(\cdot) = \sum_i c_i k(X_i, \cdot).$$
- Using the transformation $y = Kc$ (or $c = K^{-1}y$), we can rewrite the SVM as
$$\min_{c \in \mathbb{R}^n} \sum_i (1 - (Kc)_i)_+ + \lambda c^t Kc.$$
- Again: so what?

The SVM: So What?

- The SVM has many interesting and desirable properties.
- These properties are not immediately apparent from the optimization problems we have just written.
- Optimization theory and geometry lead us to *algorithms* for solving the problem and *insights* in the nature of the solution.
- We will see that SVMs have a nice *sparsity* property: many (frequently most) of the c_i's turn out to be zero.

HONDA
The Power of Dreams

Nondifferentiable Functions and Constraints

- We can rewrite a piecewise differentiable convex linear function as a sum of differentiable functions over *constrained* variables.
- Case in point: instead of minimizing

$$(1 - yY)_+,$$

  I can minimize

$$\xi$$

  subject to the *constraints* that

$$\xi \geq 1 - yY \text{ and } \xi \geq 0.$$

- Two different ways of looking at the same thing.

HONDA
The Power of Dreams

The Hinge Loss, Constrained Form

- If I want to take a Lagrangian, I need to rewrite the loss function in terms of constraints. These constraints are also called *slack* variables.
- This rewriting is *orthogonal* to the issue of whether I think about y or c.
- In terms of y, we rewrite

$$\min_{y \in \mathbb{R}^n} \sum_i (1 - y_i Y_i)_+ + \lambda y^t K^{-1} y.$$

as

$$\begin{array}{cc} \min\limits_{y \in \mathbb{R}^n, \xi \in \mathbb{R}^n} & \sum_i \xi_i + \lambda y^t K^{-1} y \\ \text{subject to :} & \xi \geq (1 - yY) \\ & \xi \geq 0 \end{array}$$

HONDA The Power of Dreams

In terms of c

- In terms of the c, the constrained version of the problem is

$$\begin{aligned}\min_{c\in\mathbb{R}^n,\xi\in\mathbb{R}^n} \quad & \sum_i \xi_i + \lambda c^t K c \\ \text{subject to :} \quad & \xi \geq (1 - YKc) \\ & \xi \geq 0\end{aligned}$$

- Note how we get rid of the $(1 - Kc)_+$ by requiring that the ξ are nonnegative.

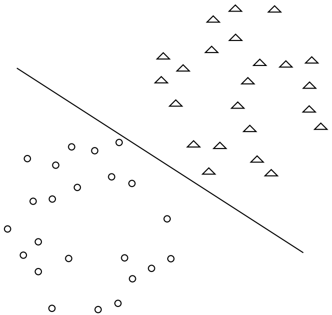
Solving an SVM, I

- Written in terms of c or y (and ξ), we have a problem where we're trying to minimize a convex quadratic function subject to linear constraints.
- In optimization theory, this is called a *convex quadratic program.*
- Algorithm I: Find or buy software that solves convex quadratic programs.
- This will work. However, this software generally needs to work with the matrix K. It will be slower than solving an RLS problem of the same size.
- As we will see, the SVM has special structure which leads to good algorithms.

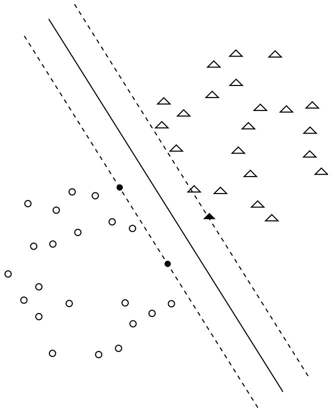
The geometric approach

- The “traditional” approach to explaining the SVM is via *separating hyperplanes* and *margin*.
- Imagine the positive and negative examples are separable by a linear function (i.e.a hyperplane).
- Define the margin as the distance from the hyperplane to the nearest example.
- Intuitively, larger margin will generalize better.

Large and Small Margin Hyperplanes



(a)



(b)

HONDA
The Power of Dreams

Classification With Hyperplanes

- Denote the hyperplane by w.
- $f(x) = w^t x$.
- A *separating* hyperplane satisfies $y_i(w^t x_i) > 0$ for the entire training set.
- We are considering homogeneous hyperplanes (i.e., hyperplanes that pass through the origin.)
- Geometrically, when we draw the hyperplane, we are drawing the set $\{x : w^t x = 0\}$, and the vector w is normal to this set.

Maximizing Margin, I

- Given a separating hyperplane w, let x^c be a training point closest to w, and define x^w to be the unique point in $\{x : w^t x = 0\}$ that is closest to x. (Both x^c and x^w depend on w.)
- Finding a maximum margin hyperplane is equivalent to finding a w that maximizes $||x^c - x^w||$.
- For some k (assume $k > 0$ for convenience),

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

HONDA
The Power of Dreams

Maximizing Margin, II

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

$$\begin{aligned} k = w^t(x^c - x^w) &= ||w||\ ||x^c - x^w|| \\ &\Longrightarrow ||x^c - x^w|| = \frac{k}{||w||} \end{aligned}$$

Maximizing Margin, III

- k is a "nuisance" parameter; WLOG, we fix it to 1. (Scaling a hyperparameter by a positive constant changes k and $||w||$, but not x^c or x^w.)
- With k fixed, maximizing $||x - x^w||$ is equivalent to maximizing $\frac{1}{||w||}$, or minimizing $||w||$, or minimizing $||w||^2$.
- The margin is now the distance between $\{x : w^t x = 0\}$ and $\{x : w^t x = 1.\}$
- Fixing k is fixing the scale of the function.

The linear homogeneous separable SVM

- Phrased as an optimization problem, we have

$$\begin{array}{ll} \min_{w \in \mathbb{R}^n} & ||w||^2 \\ \text{subject to}: & y_i w^t x_i - 1 \geq 0 \quad i = 1, \ldots, n \end{array}$$

- Note that $||w||^2$ is the RKHS norm of a linear function.
- We are minimizing the RKHS norm, subject to a “hard” loss.

HONDA
The Power of Dreams

From hard loss to hinge loss.

- We can introduce slacks ξ_i:

$$\begin{array}{ll} \min_{w \in \mathbb{R}^n, \xi \in \mathbb{R}^n} & ||w||^2 + \sum_i \xi_i \\ \text{subject to :} & \xi_i \geq 1 - y_i w^t x_i \quad i = 1, \ldots, n \\ & \xi_i \geq 0 \qquad\qquad\quad i = 1, \ldots, n \end{array}$$

- What happened to our beautiful geometric argument? What is the margin if we don't separate the data?
- Because we are nearly always interested classification problems that are *not* separable, I think it makes more sense to start with the RKHS and the hinge loss, rather than the concept of margin.

Fenchel Duality, Main Theorem (Reminder)

Theorem

Given convex functions f and g, under minor technical conditions,

$$\inf_{y,z} \{f(y) + g(y) + f^*(z) + g^*(-z)\} = 0,$$

at least one minimizer exists, and all minimizers y, z satisfy the complementarity equations:

$$\begin{aligned} f(y) - y^t z + f^*(z) &= 0 \\ g(y) + y^t z + g^*(-z) &= 0. \end{aligned}$$

Regularization Optimality Condition

- We are looking for y and z satisfying

$$R(y) - y^t z + R^*(z) = 0.$$

- For Tikhonov regularization,

$$\begin{aligned} R(y) &= \lambda y^t K^{-1} y. \\ R^*(z) &= \lambda^{-1} z^t K z. \end{aligned}$$

- The optimality condition for the regularizer is:

$$\begin{aligned} \frac{1}{2}\lambda y^t K^{-1} y - y^t z + \frac{1}{2}\lambda^{-1} z^t K z &= 0 \\ \frac{1}{2}(y - \lambda^{-1} K z)^t (\lambda K^{-1} y - z) &= 0 \\ y = \lambda^{-1} K z \Longleftrightarrow z = \lambda K^{-1} y. \end{aligned}$$

Regularization Optimality Condition

- For Tikhonov regularization, the optimal y and z satisfy

  $$y = \lambda^{-1} K z,$$

  independent of the loss function.
- Modified regularizers will lead to modified optimality conditions, again independent of the loss. Key future example: unregularized bias terms.
- The z's are closely related to the expansion coefficients via $c = \lambda^{-1} z$.

HONDA
The Power of Dreams

Loss Optimality Conditions

- For a pointwise loss function

$$V(y) = \sum_i v_i(y_i),$$

the conjugate of the sum is the sum of the conjugates:

$$\begin{aligned} V^*(z) &= \sup_y \left\{ y^t z - \sum_i v_i(y_i) \right\} \\ &= \sum_i \sup_{y_i} \{ y_i z_i - v_i(y_i) \} \\ &= \sum_i v_i^*(z_i). \end{aligned}$$

- Therefore, for each data point, we get a constraint

$$v_i(y_i) + y_i z_i + v_i^*(-z_i).$$

The exact form of the constraint is dictated by the loss.

The Hinge Loss Conjugate

- We need to derive $v^*(-z)$ for the hinge loss $v(y) = (1 - yY)_+$.
- We could use the graphical method (maybe on board).
- Note that $Y \in \{-1, 1\}$, so $yY = y/Y$.
- Alternate approach, a composition of functions...

The $\max(y, 0)$ nonlinearity.

- Suppose $f(y) = \max(y, 0) = (y)_+$
- $f^*(z) = \sup_y \{yz - (y)_+\}$
- Clearly, if $z < 0$ or $z > 1$, $f^*(z) = \infty$
- Clearly, if $z \in [0, 1]$, $f^*(z) = 0$
- Conclusion: $f^*(z) = \delta_{[0,1]}(z)$

The $1 - yY$ term.

- $g(y) = f(1 - yY)$
- $g^*(z) = \sup_y \{yz - f(1 - yY)\}$
- Substitute $\hat{y} = 1 - yY \iff y = Y - \hat{y}Y$
- $g^*(z) = \sup_{\hat{y}} \{(Y - \hat{y}Y)z - f(\hat{y})\} = Yz + f^*(-Yz)$

HONDA
The Power of Dreams

Putting it together

- $f(y) = (y)_+ \iff f^*(z) = \delta_{[0,1]}(z)$
- $g(y) = f(1 - yY) \iff g^*(z) = Yz + f^*(-Yz)$
- $v(y) = (1 - yY)_+$
- $v^*(z) = Yz + f^*(Yz) = Yz + \delta_{[0,1]}(-Yz)$
- $v^*(-z) = \delta_{[0,1]}\left(\frac{z}{Y}\right) - \frac{z}{Y}$

HONDA The Power of Dreams

The hinge loss optimality condition

$$\begin{aligned}
v(y) + yz + v^*(-z) &= 0 \\
(1 - yY)_+ + yz + \delta_{[0,1]}\left(\frac{z}{Y}\right) - \frac{z}{Y} &= 0 \\
(1 - yY)_+ - z\left(\frac{1}{Y} - y\right) + \delta_{[0,1]}\left(\frac{z}{Y}\right) &= 0 \\
(1 - yY)_+ - \frac{z}{Y}(1 - yY) + \delta_{[0,1]}\left(\frac{z}{Y}\right) &= 0
\end{aligned}$$

HONDA
The Power of Dreams

The complete SVM optimality conditions

Training an SVM means (conceptually) finding y, z satisfying

$$\begin{aligned} y &= \lambda^{-1} K z \\ (1 - yY)_+ &= \frac{z}{Y}(1 - yY) \\ \frac{z}{Y} &\in [0, 1]^n. \end{aligned}$$

HONDA The Power of Dreams

Analyzing the Loss Optimality Condition

- Remember, the loss function optimality condition is:

$$(1 - y_i Y_i)_+ - \frac{z_i}{Y_i}(1 - y_i Y_i) + \delta_{[0,1]}\left(\frac{z_i}{Y_i}\right) = 0$$

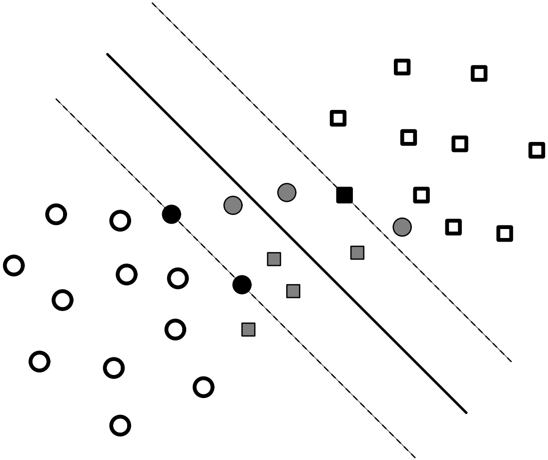
- Suppose that at optimality, $(1 - y_i Y_i) < 0$. We pay no loss at the *i*th point.
- Clearly, $\frac{z_i}{Y_i}(1 - y_i Y_i)$ must be zero as well.
- But that means that $z_i = 0$, and also that $c_i = 0$ in the functional expansion.
- Similarly, if $(1 - y_i Y_i) > 0$, then $\frac{z_i}{Y_i} = 1$.
- If $1 - y_i Y_i = 0$, we cannot say anything about z_i.

HONDA The Power of Dreams

What are support vectors?

- We see that points that are "well-classified" $(1 - y_i y_l < 0)$ have $z_i = c_i = 0$. *These points to do not contribute to the functional expansion.*
- The other points do contribute. They are called *support vectors.*
- If we are lucky, the number of support vectors will be small relative to the size of the training set.
- It is precisely this fact that makes the SVM architecture especially useful.
- Other key point: non-support vectors can be added, removed, or moved without changing the solution (assuming they always satisfy $(1 - y_i Y_i < 0)$).

Support Vectors: Graphical Interpretation



The primal and dual problems

$$\min_y R(y) + \sum_i v_i(y_i)$$

$$\min_y \frac{\lambda}{2} y^t K^{-1} y + \sum_i (1 - y_i Y_i)_+$$

$$\min_z R^*(z) + \sum_i v_i^*(-z_i)$$

$$\min_z \frac{\lambda^{-1}}{2} z^t K z + \sum_i \left(-\frac{z_i}{Y_i} + \delta_{[0,1]} \frac{z_i}{Y_i} \right)$$

HONDA
The Power of Dreams

A simple SVM algorithm

- We will develop a poor-man's but conceptually reasonable algorithm for solving

$$\min_z \frac{\lambda^{-1}}{2} z^t K z + \sum_i \left(-\frac{z_i}{Y_i} + \delta_{[0,1]} \frac{z_i}{Y_i} \right)$$

- We work with the z's rather than the y's because we don't want to deal with K^{-1}.
- Consider optimizing one of the z_i, and holding the others fixed.
- We are now trying to minimize

$$\lambda^{-1} \left(\frac{1}{2} K_{ii} z_i^2 + \sum_{j \neq i} (K_{ij} z_j) z_i \right) - \frac{1}{Y_i},$$

  subject to the constraint $\frac{z_i}{Y_i} \in [0, 1]$.
- This problem is easy to solve directly.
- Algorithm: Keep doing this until we're done.

A simple SVM algorithm, analyzed

- We start with the all-zero solution $z = 0$.
- Note that solving a subproblem for point i involves the kernel products between i and those j such that $z_j \neq 0$.
- If we have two points j and k such that neither z_j nor z_k ever become nonzero during the course of the algorithm, *we never need to compute* K_{jk}.
- Real SVM algorithms are basically (almost) this idea, combined with schemes for caching kernel products.

An unregularized bias

- The representer theorem says the answer has the form

$$f(\cdot) = \sum_i c_i k(X_i, \cdot).$$

- Suppose we decide to look for a function of the form

$$f(\cdot) = \sum_i c_i k(X_i, \cdot) + b,$$

and we do not regularize b.

- The modified problem is

$$\begin{array}{rl} \min_{c \in \mathbb{R}^n, \xi \in \mathbb{R}^n, b \in \mathbb{R}} & \sum_i \xi_i + \lambda c^t K c \\ & \xi \geq (1 - Kc + b) \\ & \xi \geq 0. \end{array}$$

- Why would we do such a thing?

An unregularized bias, thoughts

- "Why should my hyperplane have to go through the origin? I don't know that a priori."
- An unregularized bias says *constant functions are not penalized.*
- We are saying "Find me a function in the RKHS, plus some constant function."
- Alternate strategy: add a dimension of all 1's to the data, in feature space (e.g., $k(x_i, x_j) \leftarrow k(x_i, x_j) + 1$)
- The alternate strategy allows arbitrary hyperplanes, but penalizes the bias term.

Unregularized bias, pros and cons

- Pro: Some people think it feels better.
- Cons: The math gets more complicated.
- Suggestion: if you have a regularized bias, do it implicitly. Don't bother writing b^2 everywhere, that's a waste of ink.
- Suggestion: have a regularized bias.
- If you insist on an unregularized bias, Fenchel duality is a good way to talk about it ...

Fenchel Bias, I

- Instead of $y = Kc$, we have $y = Kc + b$.
- Suppose we have regularizer R (with conjugate $R^*(y)$).
- Adding an unregularized bias is really saying “I can shift all my values by some constant, and I consider that just as smooth.”
- The new regularizer is

$$R'(y) = \inf_b R(y - 1_n b)$$

HONDA The Power of Dreams

The conjugate of a biased regularizer

$$\begin{aligned}
R'(y) &= \inf_b R(y - 1_n b) \\
R'^*(z) &= \sup_y \{y^t z - \inf_b R(y - 1_n b)\} \\
&= \sup_{y,b} \{y^t z - R(y - 1_n b)\} \\
&= \sup_{\hat{y},b} \{(\hat{y} + 1_n b)^t z - R(\hat{y})\} \\
&= \sup_b \{(1_n^t z) b + \sup_{\hat{y}} \{\hat{y}^t z - R(\hat{y})\}\} \\
&= \delta_{\{0\}}(1_n^t z) + R^*(z).
\end{aligned}$$

The conjugate of a biased regularizer, thoughts

$$
\begin{aligned}
R'(y) &= \inf_b R(y - 1_n b) \\
R'^*(z) &= \delta_{\{0\}}(1_n^t z) + R^*(z).
\end{aligned}
$$

- In the primal, we say "allow a constant shift of the values."
- In the dual, we say $\sum_i z_i = 0$.
- **That's it!!!!!**

The conjugate of a biased regularizer, more thoughts

- We *don't* need to rederive the whole dual from the beginning.
- This result is general across regularizers and loss functions.
- This is an example of *infimal convolution,* see the Fenchel paper for details.
- For algorithms, the constraint $\sum_i z_i = 0$ means they modify two z's at a time rather than one.

HONDA
The Power of Dreams

Good Large-Scale SVM Solvers

- SVMLight: `http://svmlight.joachims.org`
- SVMTorch: `http://www.torch.ch`
- LIBSVM: `http://wws.csie.ntu.edu.tw/~cjlin/libsvm`

HONDA
The Power of Dreams

Musings on SVMs and RLS

- If we can solve one RLS problem, we can find a good λ (that minimizes LOO error.)
- There exists work on finding the "regularization path" of the SVM (Hastie et al. 04). The claim is they can find a good λ in the same time as it takes to solve one problem. The experiments do not convince me (and they do not do LOO error.)
- For large nonlinear problems, I cannot solve one RLS problem at all.
- The SVM is sparse. It is only a constant factor sparse, so it won't scale forever, but solving O(100,000) point nonlinear SVM problems is (somewhat) common.

HONDA The Power of Dreams

The elephant in the room.

- There are many good methods to help us choose λ.
- However, choosing k is usually the hard part.
- Note that λ is about choosing how much smoothness to insist on in an RKHS, but choosing k is about deciding which RKHS to use.
- If we only have a small number of parameters, we can grid search.
- But what about kernels like

$$k(x_i, x_j) = \exp\left(-\sum_d \gamma_d (x_{id} - x_{jd})^2\right),$$

  a generalization of the Gaussian where we have a lengthscale for each dimension?
- There are some recent attempts to deal with this, but nothing is too satisfactory in my opinion.