EECS 940 Presentation
Support Vector Machine

Yuanliang Meng
03-15-2012
Support Vector Machine

• We have discussed the idea of SVM, particularly in terms of quadratic programming, an instance of optimization problem.

• In this presentation we will discuss SVM in terms of regularization and regression, among other issues.
Regularization

- What is regularization?
  - It is a method to constrain the hypothesis space
Empirical error and penalization term

- A possible way to do regularization is to minimize the sum of empirical error (loss function) and a penalized term (stabilizer).

\[
\min_{f \in \mathcal{H}} \left[ \sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f) \right]
\]

- The empirical error is also known as a loss function. The most commonly used one is the sum of square error. The following is the example of ridge regression, we talked
• However, the loss function can have other forms than SSE. Recall the object function of SVM:

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^{N} \xi_i$$

subject to

$$\xi_i \geq 0, \quad y_i (x_i^T \beta + \beta_0) \geq 1 - \xi_i \quad \forall i,$$

• It can be rewritten as the following:

$$\min_{\beta_0, \beta} \sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \frac{\lambda}{2} \|\beta\|^2$$

$$f(x) = h(x)^T \beta + \beta_0$$

$$(k)_+ \equiv \max(k, 0).$$
Hinge loss of L1 SVM

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 $y_f$ 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 $-1$. 
More about the penalized term in regularization

• So far the penalization terms for various regularization problems are all square of 2-norms of some vector. They serve to “smooth” the model, and avoid over fitting.

• In some problems, they look different. For example, in a spline fitting problem, we want to control the curvature of a curve, and the penalization is corresponding to the integral of squared second order derivative. (Actually it is still a generalized “norm”.)

\[
RSS(f, \lambda) = \sum_{i=1}^{N} \{y_i - f(x_i)\}^2 + \lambda \int \{f''(t)\}^2 dt,
\]
Kernel method

• As we already know, we can map data from their feature space to some higher dimensional space, so they can be linearly separable in the new space.

\[ f(x) = \sum_{i=1}^{\infty} c_i \phi_i(x), \]

• What is even nicer is that we do not really need to calculate the mapped values. Instead, we just need to calculate a kernel function, which can be computationally economic. Note: we want the kernel to be general, not necessarily a matrix.

\[ f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i). \]
The relation between the basis and the kernel

- Therefore, the function $f$ can be represented in two ways. Technically, they are called the *primal* form and the *dual* form, respectively.
- A kernel function is positive definite and symmetric.
- Now we want to ask a question in general: Given a set of basis functions $\{\phi_i\}$, is it guaranteed to find a unique kernel?
- Under some conditions, the answer is “yes”.
Construct a kernel for chosen basis functions

- Suppose we have a sequence of positive numbers $\lambda_n$, and linearly independent functions $\phi_n$, we can construct a binary function $K(x, y)$ in the following: (Note: We do not know if it is a valid kernel yet.)

$$K(x, y) \equiv \sum_{n=0}^{\infty} \lambda_n \phi_n(x) \phi_n(y),$$

- The function $K(x, y)$ is Obviously positive definite and symmetric.
- Now the functions $\phi_n$ span a space.

$$f(x) = \sum_{n=0}^{\infty} a_n \phi_n(x)$$
• We want to make this space a **Hilbert Space**, i.e. a space with inner product defined. We choose the inner product as the following:

\[
< \sum_{n=0}^{\infty} a_n \phi_n(x), \sum_{n=0}^{\infty} d_n \phi_n(x) >_H \equiv \sum_{n=0}^{\infty} \frac{a_n d_n}{\lambda_n}
\]

• Then, it is easy to check the defined function \( K(x,y) \) has the following property:

\[
< f(y), K(y, x) >_H = \sum_{n=0}^{\infty} \frac{a_n \lambda_n \phi_n(x)}{\lambda_n} = \sum_{n=0}^{\infty} a_n \phi_n(x) = f(x)
\]

• Such a function \( K(x,y) \) is called the **reproducing kernel** of the Hilbert Space. It is unique. This kernel also defines a norm in this space:

\[
\| f \|_{K}^{2} = \sum_{n=0}^{\infty} \frac{a_n^2}{\lambda_n}
\]
• Generally speaking, the three things are equivalent:
  – Choosing a set of $\lambda_n, \phi_n$
  – Choosing reproducing kernel $K(x,y)$
  – Choosing a specific **reproducing kernel Hilbert Space (RKHS)**

• When the cardinality of $X$ is finite, $K$ is equivalent to a matrix $K_{ij} = K(x_i, x_j)$. This matrix is positive definite. The space consists of all the vectors $f$ with a finite norm $f^T K f$. Note: the dimensionality of the space in which $K$ is defined can still be infinite, such as the case of a radial basis kernel.
Generalized form of regularization

• A general class of regularization problems has the form:

\[
\min_{f \in \mathcal{H}} \left[ \sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f) \right]
\]

• Under some additional assumptions, the solution has the general form:

\[
f(X) = \sum_{k=1}^{K} \alpha_k \phi_k(X) + \sum_{i=1}^{N} \theta_i G(X - x_i)
\]

• A important subclass of the problems, which we are interested in, are those in which \( f \) belongs to a RKHS, and \( J(f) \) is the norm defined in the RKHS (or equivalently, defined by the reproducing kernel):

\[
\min_{f \in \mathcal{H}_K} \left[ \sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda \|f\|_{\mathcal{H}_K}^2 \right] \\
\min_{\{c_j\}} \left[ \sum_{i=1}^{N} L(y_i, \sum_{j=1}^{\infty} c_j \phi_j(x_i)) + \lambda \sum_{j=1}^{\infty} c_j^2 / \gamma_j \right]
\]
• The solution has the (dual) form we have seen before:

\[ f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i). \]

• And the penalty function is a square of norm of \( f \) (self inner product of \( f \)) defined by the kernel. Again, \( K \) is p.d.

\[ J(f) = \sum_{i=1}^{N} \sum_{j=1}^{N} K(x_i, x_j) \alpha_i \alpha_j \]

• Matrix form of \( J(f) \): \( \alpha^T K \alpha \)
Regression with SVM flavor

• Linear regression is to find a linear model:

\[ f(x) = x^T \beta + \beta_0 \]

• To estimate the parameters, we could consider minimizing

\[ H(\beta, \beta_0) = \sum_{i=1}^{N} V(y_i - f(x_i)) + \frac{\lambda}{2} ||\beta||^2 \]

Where

\[ V_\varepsilon(r) = \begin{cases} 0 & \text{if } |r| < \varepsilon, \\ |r| - \varepsilon & \text{otherwise.} \end{cases} \]
• This idea is somehow similar to the hinge loss in SVM.
  – In SVM, correctly classified points have no loss.
  – In the $\epsilon$-insensitive error measure just shown, small errors are totally ignored, and big errors are linearly calculated.

• Huber developed another loss function, commonly used.

\[
V_H(r) = \begin{cases} 
  \frac{r^2}{2} & \text{if } |r| \leq c, \\
  c|r| - \frac{c^2}{2} & \text{if } |r| > c,
\end{cases}
\]

**Figure 12.8.** The left panel shows the $\epsilon$-insensitive error function used by the support vector regression machine. The right panel shows the error function used in Huber’s robust regression (blue curve). Beyond $|c|$, the function changes from quadratic to linear.
Regression and kernels

• Certainly, we can use kernels to do regression, the same as classification.
• There is generally no need to compute the basis matrix, because we only need to know their inner product.
• The penalty function is based on squared norm.