9.520: Statistical Learning Theory and Applications February 27th, 2012

Regularized Least Squares and Support Vector Machines

Lecturer: Lorenzo Rosasco *Scribe: Charalampos Mavroforakis*

1 RLS and the Representer Theorem

1.1 Solving for a single λ

Find the function $f \in \mathcal{H}$ that minimizes the weighted sum of the square loss and the RKHS norm, i.e.

$$\arg\min_{f \in \mathcal{H}} \left\{ \frac{1}{2} \sum_{i=1}^{n} \left(f\left(x_i\right) - y_i \right)^2 + \frac{\lambda}{2} \left\| f \right\|_{\mathcal{H}}^2 \right\} \tag{1}$$

Using the representer theorem, we can write the solution to (1) as

$$f\left(\cdot\right) = \sum_{i=1}^{n} c_i k_{x_i} \tag{2}$$

for some $c \in \mathbb{R}^n$. We can also write the square norm of f as

$$\begin{aligned} \left\| f \right\|_{\mathcal{H}}^2 &= \langle f, f \rangle_{\mathcal{H}} = \langle \sum c_i k_{x_i}, \sum c_j k_{x_j} \rangle \\ &= \sum_i \sum_j c_i c_j \langle k_{x_i}, k_{x_j} \rangle \\ &= \sum_i \sum_j c_i c_j k\left(x_i, x_j\right) \\ &= c^T \mathbf{K} c \end{aligned} \tag{3}$$

Using (2) and (3), the RLS problem can be rewritten as follows:

$$\arg\min_{c \in \mathbb{R}^n} \left\{ \frac{1}{2} \left\| \mathbf{Y} - \mathbf{K}c \right\|_2^2 + \frac{\lambda}{2} c^T \mathbf{K} c \right\} \tag{4}$$

In order to obtain a solution, we need to set the partial derivative of (2) with respect to c to zero

$$\begin{aligned} \frac{\partial}{\partial c} \left(\frac{1}{2} \left\| \mathbf{Y} - \mathbf{K}c \right\|_2^2 + \frac{\lambda}{2} c^T \mathbf{K} c \right) = 0 \quad &\Rightarrow \\ -\mathbf{K}\left(\mathbf{Y} - \mathbf{K}c\right) + \lambda \mathbf{K} c = 0 \quad &\Rightarrow \\ \left(\mathbf{K} + \lambda \mathbf{I}\right) c = \mathbf{Y} \quad &\overset{\mathbf{K} \succcurlyeq 0}{\Rightarrow} \quad (5) \\ \left(\mathbf{K} + \lambda \mathbf{I}\right)^{-1} \mathbf{Y} = c \quad &\quad (6) \end{aligned}$$

Instead of solving (6) and in order to avoid the inversion of the matrix, we can solve (5), which is a linear system with n equations and n variables (c_i). Since $\mathbf{K} + \lambda \mathbf{I}$ is positive definite, we can use Cholesky factorization, an algorithm which has complexity $O\left(n^3\right)$. Assuming c_* is the solution of (5), given a new point x_*, we can predict $f\left(x_*\right)$ in the following way :

$$f(x_*) = \sum_{j=1}^{n} c_j \mathbf{K}_{x_j}(x_*) = \mathbf{K}_{x_j} c \tag{7}$$

which costs $O\left(n^2\right)$.

1.2 Solving for different λ's

The kernel matrix can be written as

$$\mathbf{K} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \tag{8}$$

where $\mathbf{Q}\mathbf{Q}^T = 0$ and $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{K}$, so $\mathbf{\Lambda}_{ii} \geq 0$. Then,

$$\begin{aligned} \mathbf{G}(\lambda) &= \mathbf{K} + \lambda\mathbf{I} = \\ &= \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T + \lambda\mathbf{I} \\ &= \mathbf{Q}(\mathbf{\Lambda} + \lambda\mathbf{I})\mathbf{Q}^T \end{aligned} \tag{9}$$

which implies that $\mathbf{G}^{-1}(\lambda) = \mathbf{Q}(\mathbf{\Lambda} + \lambda\mathbf{I})^{-1}\mathbf{Q}^T$. So, by paying $O\left(n^3\right)$ to calculate the eigendecomposition of $\mathbf{K}$ once, we can plug different values for λ to (9) and with $O\left(n^2\right)$ computations find the corresponding $\mathbf{G}^{-1}$. Thus, by using (6), we can easily calculate $c(\lambda)$.

1.3 The linear case

In the case of the linear kernel, i.e. $K(x_i, x_j) = x_i^T x_j$, the Kernel matrix can be written as $\mathbf{K} = \mathbf{X}\mathbf{X}^T$, and (4) becomes

$$\begin{aligned} &\arg\min_{c\in\mathbb{R}^n} \left\{ \frac{1}{2} \|\mathbf{Y} - \mathbf{K}c\|_2^2 + \frac{\lambda}{2} c^T \mathbf{K} c \right\} \\ =&\arg\min_{c\in\mathbb{R}^n} \left\{ \frac{1}{2} \left\|\mathbf{Y} - \mathbf{X}\mathbf{X}^T c\right\|_2^2 + \frac{\lambda}{2} c^T \mathbf{X}\mathbf{X}^T c \right\} \end{aligned} \tag{10}$$

Consider that for the prediction at a point x_* we use the following

$$\begin{aligned} f(x_*) &= \mathbf{K}_{x_*} c \\ &= x_*^T \mathbf{X}^T c \\ &= x_*^T w \end{aligned} \tag{11}$$

where $w = \mathbf{X}^T c$. If we substitute this in (10) and take the gradient with respect to w equal to 0, we have that

$$w = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1} \mathbf{X}^T\mathbf{Y} \tag{12}$$

Notice now that for a dataset $\mathbf{X}$ with n points and d features, the above calculation costs $O\left(d^2 n\right)$, as $\mathbf{X}^T\mathbf{X}$ is a $d \times d$ matrix. This means that if we have too many data compared to features (i.e. $n >> d$) it is better to use this linear system than relying to the application of SVD on the kernel matrix, which is the method that we saw previously.

2 Support Vector Machines

2.1 Loss function

Let us now consider a different loss function than the square loss. We define the hinge loss as

$$V\left(f\left(x,y\right)\right) \equiv \left(1-yf\left(x\right)\right)_+, \text{ where } \left(k\right)_+ \equiv \max\left(k,0\right) \tag{13}$$

This function is convex, as required in order to make the minimization problem tractable, but it is not differentiable (in the point where $y_i f\left(x_i\right)=1$). We will later introduce slack variables in order to fix this. Using the hinge loss function, the Tikhonov regularization problem becomes:

$$\arg\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}\left(1-y_i f\left(x_i\right)\right)_+ + \lambda\left\|f\right\|_{\mathcal{H}}^2 \tag{14}$$

By setting $C=\frac{1}{2\lambda n}$, we arrive at the form that is most common in the literature, i.e.

$$\arg\min_{f\in\mathcal{H}} C\sum_{i=1}^{n}\left(1-y_i f\left(x_i\right)\right)_+ + \frac{1}{2}\left\|f\right\|_{\mathcal{H}}^2 \tag{15}$$

Actually, there is also a bias term b in the formula, but we choose to omit it, i.e. the minimizer should be $f\left(x\right)+b$. By discarding this term, we force the linear kernel to pass from the origin, loosing the ability to slide the hyperplane wherever we want. In the case of a high dimensional kernel though, (e.g. the Gaussian) b doesn't make any difference. Alternatively, in the linear case, we can consider $wx+b$ as being $w'x'$, where $x'=\begin{pmatrix} x \\ 1 \end{pmatrix}$ and $w'=\begin{pmatrix} w \\ b \end{pmatrix}$, and choose whether to minimize $\|w\|$ or $\|w'\|$.

2.2 Arriving at the primal problem

In order to overcome the problem we mentioned before, namely the loss function not being differentiable in a point, we introduce slack variables and we replace $\left(1-y_i f\left(x_i\right)\right)_+$ with ξ_i. For each ξ_i, we require that $\xi_i \geq \left(1-y_i f\left(x_i\right)\right)_+$, making the new problem statement :

$$\begin{aligned} \arg\min_{f\in\mathcal{H}} & C\sum_{i=1}^{n}\xi_i + \frac{1}{2}\left\|f\right\|_{\mathcal{H}}^2 \\ \text{s.t. } & \xi_i \geq 1-y_i f\left(x_i\right), \quad i=1,\ldots,n \\ & \xi_i \geq 0 \end{aligned} \tag{16}$$

and by applying the representer theorem, we reach the following constraint quadratic programming problem:

$$\begin{aligned} \arg\min_{c\in\mathbb{R}^n,\xi\in\mathbb{R}^n} & C\sum_{i=1}^{n}\xi_i + \frac{1}{2}c^T\mathbf{K}c \\ \text{s.t. } & \xi_i \geq 1-y_i\left(\sum_{j=1}^{n}c_j K\left(x_i,x_j\right)\right), \quad i=1,\ldots,n \\ & \xi_i \geq 0 \end{aligned} \tag{17}$$

which defines the primal problem.

2.3 Formulating the dual problem

To cope with the primal problem, we need to formulate the Lagrangian and the solve the dual problem. Starting with the Lagrangian, we multiply each constraint with a variable (called Lagrange multiplier) and add them to the optimization function. The Lagrangian thus becomes:

$$\begin{aligned} L\left(c,\xi,\alpha,\zeta\right) =& C\sum_{i=1}^{n}\xi_i + \frac{1}{2}c^T\mathbf{K}c \\ & - \sum_{i=1}^{n}\alpha_i\left(y_i\sum_{j=1}^{n}c_jK\left(x_i,x_j\right) - 1 + \xi_i\right) \\ & - \sum_{i=1}^{n}\zeta_i\xi_i \end{aligned} \tag{18}$$

Now, in the dual problem, we try to maximize the infimum of the Lagrangian function, i.e.

$$\arg\max_{\alpha,\zeta}\inf_{c,\xi} L\left(c,\xi,\alpha,\zeta\right) \tag{19}$$

To construct the dual problem, we need to determine the optimal c and ξ in terms of the dual variables. We achieve this by differentiating the constraints with respect to the primal variables:

$$\begin{aligned} \frac{\partial}{\partial\xi_i}L = 0 \Rightarrow & C - \alpha_i - \zeta_i = 0 \\ & \Rightarrow 0 \le \alpha_i \le C \end{aligned} \tag{20}$$

$$\frac{\partial}{\partial c}L = 0 \Rightarrow c_i = \alpha_i y_i \tag{21}$$

so by plugging these back into (18) and substituting $c_i = \alpha_i y_i$, the dual problem becomes:

$$\begin{aligned} \max_{\alpha\in\mathbb{R}^n} & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}a^T\mathbf{Q}a \\ \text{s.t. } & \sum_{i=1}^{n}y_i\alpha_i = 0, \\ & 0 \le \alpha_i \le C, \quad i = 1,\ldots,n \end{aligned} \tag{22}$$

where $\mathbf{Q} = y^T\mathbf{K}y$.

The dual is much easier to solve than the primal, with only one box constraint and one inequality per point.

2.4 Optimality Conditions - Support Vectors

We have formulated the SVM as a quadratic programming problem. We can therefore use results from the field of optimization to derive certain properties of an optimal SVM solution. Specifically, the Karush-Kuhn-Tucker (KKT) conditions are necessary conditions for an optimal solution to any non-linear programming problem. They are sufficient conditions when the primal objective and inequality constraints are convex and continuously differentiable, and each equality constraint is an

affine function. This holds for the SVM problem, so the KKT conditions are both necessary and sufficient conditions for any optimal SVM solution.

The KKT conditions can be grouped into four categories: stationarity, primal feasibility, dual feasibility and complementary slackness. Stationarity requires that the gradient of the Lagrangian be zero. Primal and dual feasibility say that the solution must satisfy both the primal and dual constraints (i.e. any optimal solution must be a feasible one). Complementary slackness relates the value of each Lagrangian multiplier with its corresponding constraint. Recall that the variables α_i and ζ_i were introduced to incorporate the primal constraints into the dual objective.

$$\begin{aligned} \alpha_i &\rightarrow y_i \sum_{j=1}^{n} c_j K(x_i, x_j) - 1 + \xi_i \geq 0, \\ \zeta_i &\rightarrow \xi_i \geq 0. \end{aligned}$$

The complementary slackness condition states that either the primal constraint is satisfied with equality, or its corresponding Lagrangian multiplier is zero. More formally, if c, ξ, α and ζ are optimal solutions to the primal and dual, then

$$\begin{aligned} \alpha_i(y_i \sum_{j=1}^{n} c_j K(x_i, x_j) - 1 + \xi_i) &= 0, \\ \zeta_i \xi_i &= 0. \end{aligned}$$

Therefore, for any training point x_i, either $\alpha_i = 0$ or $y_i \sum_{j=1}^{n} c_j K(x_i, x_j) - 1 + \xi_i = 0$. Since $\xi_i \geq 0$, the latter case occurs when $y_i \sum_{j=1}^{n} c_j K(x_i, x_j) \leq 1$ (so the point is not classified 'as correctly' as we would like) and $\alpha_i = 0$ when $y_i \sum_{j=1}^{n} c_j K(x_i, x_j) > 1$ (by complementary slackness). We showed above that, at the optimum, $c_i = y_i \alpha_i$, so when $\alpha_i = 0$, the coefficient in the solution that corresponds to the ith training point will be zero – the solution is said to be "sparse." The points x_i for which $\alpha_i > 0$ are called the "support vectors," which gives the SVM its name.