english

CSCI 5525 Machine Learning Fall 2019

Lecture 7: Support Vector Machine (Part 2)

Feb 17th 2020

Lecturer: Steven Wu *Scribe: Steven Wu*

In the last lecture, we consider a general form of constrained optimization problem:

$$\min_{\mathbf{w}} F(\mathbf{w}) \quad \text{s.t.} \quad h_j(\mathbf{w}) \leq 0 \quad \forall j \in [m]$$

For each constraint, we introduce a Lagrangian multiplier (or dual variable) $\lambda_j \geq 0$, and write down the following Lagrangian function:

$$L(\mathbf{w}, \lambda) = F(\mathbf{w}) + \sum_{j=1}^{m} \lambda_j h_j(\mathbf{w})$$

Under "mild" condition (e.g. SVM problem, the so-called Slater's condition), *strong duality* holds

$$\max_{\lambda} \min_{\mathbf{w}} L(\mathbf{w}, \lambda) = \min_{\mathbf{w}} \max_{\lambda} L(\mathbf{w}, \lambda)$$

Let $\mathbf{w}^* = \arg\min_{\mathbf{w}}(\max_\lambda L(\mathbf{w}, \lambda))$ and $\lambda^* = \arg\max_\lambda(\min_{\mathbf{w}} L(\mathbf{w}, \lambda))$ denote the optimal primal and dual solutions respectively. When strong duality holds, we have the following KKT conditions:

- (Complementary slackness): last equality implies that $\lambda_j^* h_j(\mathbf{w}^*) = 0$ for all j.
- (Stationarity): $\mathbf{w}^*$ is the minimizer of $L(\mathbf{w}, \lambda^*)$ and thus has gradient zero

$$\nabla_{\mathbf{w}} L(w^*, \lambda^*) = \nabla F(w^*) + \sum_j \lambda_j^* \nabla h_j(\mathbf{w}^*) = \mathbf{0}$$

- (Feasibility): $\lambda_j \geq 0$ and $h_j(\mathbf{w}^*) \leq 0$ for all j.

The KKT conditions are necessary conditions for the optimal solutions. However, they are also sufficient when F is convex and the set of h_j are continuously differentiable convex functions.

Dual Formulation of SVM

Now we apply the tools Lagrange duality to the soft-margin SVM problem.

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n} \xi_i \quad \text{such that} \tag{1}$$

$$\forall i, \qquad y_i(\mathbf{w}^\intercal x_i) \geq 1 - \xi_i \tag{2}$$

$$\forall i, \qquad \xi_i \geq 0 \tag{3}$$

To derive the Lagrangian, we rewrite each constraint in (2) as

$$1 - \xi_i - y_i \mathbf{w}^\intercal x_i \leq 0$$

and introduce a dual variable $\lambda_i \geq 0$. For each constraint $\xi_i \geq 0$, we introduce a dual variable $\alpha_i \geq 0$. The set of variables $\mathbf{w}$ and ξ that are called the primal variables. This allows us to write down the *Lagrangian* objective:

$$L(\mathbf{w}, \xi, \lambda, \alpha) = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \lambda_i(1 - \xi_i - y_i \mathbf{w}^\intercal x_i) - \sum_{i=1}^{n} \alpha_i \xi_i$$

Now we can apply the KKT conditions to obtain some characterizations of the SVM solution. First, applying the staionarity condition $\nabla_{\mathbf{w},\xi} L(\mathbf{w}^*, \xi^*, \lambda^*, \alpha^*) = \mathbf{0}$:

$$\mathbf{w} = \sum_i y_i \lambda_i^* x_i \qquad (\tfrac{\partial L}{\partial \mathbf{w}} = 0)$$

$$C - \lambda_i^* - \alpha_i^* = 0 \quad \forall i \qquad (\tfrac{\partial L}{\partial \xi_i} = 0)$$

Let us plug these back into L:

$$\begin{aligned}
L(\mathbf{w}, \xi, \lambda, \alpha) &= C\sum_{i=1}^{n} \xi_i + \frac{1}{2}\left\|\sum_{i=1}^{n} y_i \lambda_i x_i\right\|_2^2 - \sum_{i=1}^{n} \alpha_i \xi_i + \sum_{i=1}^{n} \lambda_i (1 - \xi_i - y_i \mathbf{w}^\intercal x_i) && (4) \\
&= \frac{1}{2}\left\|\sum_{i=1}^{n} y_i \lambda_i x_i\right\|_2^2 + \sum_i \lambda_i - \sum_i \lambda_i \left(y_i \left(\sum_j y_j \lambda_j x_j \right)^\intercal x_i \right) \\
& && \text{(Plug in } C = \alpha_i + \lambda_i\text{)} \\
&= \frac{1}{2}\left\|\sum_{i=1}^{n} y_i \lambda_i x_i\right\|_2^2 + \sum_i \lambda_i - \sum_{i,j\in[n]} \lambda_i \lambda_j y_i y_j x_i^\intercal x_j && (5) \\
&= \sum_i \lambda_i - \frac{1}{2}\sum_{i,j\in[n]} \lambda_i \lambda_j y_i y_j x_i^\intercal x_j && (6)
\end{aligned}$$

The optimization problem then becomes:

$$\begin{aligned}
& \max_{\alpha,\lambda} \sum_i \lambda_i - \frac{1}{2}\sum_{i,j\in[n]} \lambda_i \lambda_j y_i y_j x_i^\intercal x_j \\
\text{such that for all } i: \quad & C = \lambda_i + \alpha_i \\
& \lambda_i, \alpha_i \geq 0
\end{aligned}$$

Observe that we could also replace the constraints by the following so that we only have one set of decision variables to optimize:

$$\text{for all } i: \qquad 0 \leq \lambda_i \leq C$$

This is a quadratic program with a quadratic objective function and a set of linear constraints. Suppose we are given the optimal solution λ^*. What is the linear predictor we get from this dual solution? We know from the KKT conditions that

$$\mathbf{w}^* = \sum_{i=1}^{n} y_i \lambda_i^* x_i = \sum_{i:\, \lambda_i^* > 0} y_i \lambda^* x_i$$

Any point i with $\lambda_i^* > 0$ is called a *support vector*, hence the name SVM.

Now let us apply complementary slackness from the KKT conditions:

$$\text{for all } i, \quad \alpha_i^* \xi_i^* = 0, \quad \lambda_i^* \left(1 - \xi_i^* - y_i \langle \mathbf{w}^*, x_i \rangle\right) = 0$$

For any support vector with $\lambda_i^* > 0$, we then also have

$$(1 - \xi_i^* - y_i \langle \mathbf{w}^*, x_i \rangle) = 0 \Leftrightarrow 1 - \xi_i^* = y_i \langle \mathbf{w}^*, x_i \rangle$$

We can break it down into the following cases:

- If $\xi_i^* = 0$, then $y_i \langle \mathbf{w}^*, x_i \rangle = 1$, which means the point is exactly $1/\|\mathbf{w}\|$ away from the decision boundary.
- If $\xi_i^* < 1$, then $y_i \langle \mathbf{w}^*, x_i \rangle \in (0, 1)$, then this point is classified correctly but pretty close to the decision boundary with distance less than $1/\|\mathbf{w}\|$.
- If $\xi_i^* > 1$, then $y_i \langle \mathbf{w}^*, x_i \rangle < 0$, then this point is classified incorrectly.

SVM can also be viewed as a form of compression, since we only need the support vectors to define the final solution.

Multiclass Extensions

SVM is inherently a classfication method for binary class $\mathcal{Y}$. There are many ways to take binary classificaton methods like SVM to solve multiclass classification problems. We discuss two standard approaches here. Let $\mathcal{Y} = \{1, \ldots, k\}$.

One-against-all. This involves solving k binary classification problems, each of which requires us to classify the current class j against all other classes. Given a dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, we can construct k datasets $D_1, \ldots, D_k$ such that

$$D_j = \{(x_i, \mathbf{1}[y_i = j])\}_{i=1}^{n}$$

Then run SVM k times: on each dataset D_j to obtain a weight vector $\mathbf{w}_j$. Finally, on any example x, we will predict

$$\hat{y} = \arg\max_{j \in \mathcal{Y}} \langle \mathbf{w}_j, x \rangle$$

One-against-one. Run SVM $k(k-1)/2$ times: for every pair $j, j' \in \mathcal{Y}$ such that $j < j'$, learn a weight vector $\mathbf{w}_{j,j'}$ that distinguishes between the two classes using the subset of data with labels j and j'. For each example x, the weight vector $\mathbf{w}_{jj'}$ "votes" for either label j or label j'. Finally, we predict the class with the highest votes given by the weight vectors $\mathbf{w}_{jj'}$.

We can also modify binary SVM directly to construct a multiclass SVM method.

Multiclass SVM Another idea similar to one-against-all is to train $\mathbf{w}_1, \ldots, \mathbf{w}_k$ simultaneously by asking the predictor to predict the right label on each example:

$$
\begin{aligned}
\min_{\mathbf{w}_1,\ldots,\mathbf{w}_k} \frac{1}{2}\sum_{j=1}^{k} \|\mathbf{w}_j\|_2^2 + C\sum_{i=1}^{n} \xi_i \qquad & \text{such that} \\
\forall i, \forall j \neq y_i \qquad \mathbf{w}_{y_i}^\intercal x_i \geq \mathbf{w}_j^\intercal x_i + 1 - \xi_i & \\
\forall i, \qquad \xi_i \geq 0 &
\end{aligned}
$$