CSCI 5525 Machine Learning Fall 2019

Lecture 7: Support Vector Machine (Part 2)

Sep 24 2019

Lecturer: Steven Wu *Scribe: Steven Wu*

Dual Formulation of SVM

Let us first recall the soft-margin SVM problem formulation.

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

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

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

Now we can introduce the tools of Lagrange duality and utilitze KKT conditions. First. we can 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 $\nabla_{\mathbf{w},\xi} L(\mathbf{w}^*, \xi^*, \lambda^*, \alpha^*) = \mathbf{0}$:

$$\mathbf{w} = \sum_i y_i \lambda_i^* x_i \tag{$\frac{\partial L}{\partial \mathbf{w}} = 0$}$$

$$C - \lambda_i^* - \alpha_i^* = 0 \quad \forall i \tag{$\frac{\partial L}{\partial \xi_i} = 0$}$$

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 a couple 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.

Kernels: Feature Expansion

Note that all of our derivation holds if we replace each feature vector x_i by some feature expansion $\phi(x_i)$. The optimization problem then becomes:

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

In linear regression, we have already seen several examples for such feature expansion. Some of these mappings lift the feature vector to much higher dimensional space.

Quadratic expansion For example, for $x \in \mathbb{R}^d$, consider the following variant of the quadratic (or polynomial of order 2) expansion:

$$\phi(x) = (1, \sqrt{2}x_1, \ldots, \sqrt{2}x_d, x_1^2, \ldots, x_d^2, \sqrt{2}x_1 x_2, \ldots, \sqrt{2}x_{d-1}x_d)$$

Under this expansion, the product

$$\phi(x)^\intercal \phi(x') = (1 + x^\intercal x')^2$$

which can be computed in $O(d)$ time, as opposed to $O(d^2)$.

Products of all subsets. Let's blow up the dimension even more. Consider the following feature expansion mapping

$$\phi(x) = \left(\prod_{i \in S} x_i \right)_{S \subseteq [d]}$$

Then we still compute the product in time $O(d)$ (instead of 2^d):

$$\phi(x)^\intercal \phi(x) = \prod_{i=1}^{d} (1 + x_i x_i')$$

Gaussian kernel. Next, let's be more ambitious and push the dimension to infinity. For any parameter $\sigma > 0$, consider a feature expansion such that

$$\phi(x)^\intercal \phi(x') = \exp\left(-\frac{\|x - x'\|_2^2}{2\sigma^2}\right)$$

This product can be computed in $O(d)$ time. So what is ϕ? Let us try the simple case of $d = 1$. Then

$$\begin{aligned}
\phi(x)\phi(y) &= \exp(-(x-y)^2/(2\sigma^2)) \\
&= \exp(-x^2/(2\sigma^2))\exp(-y^2/(2\sigma^2))\exp(xy/\sigma^2) \\
&= \exp(-x^2/(2\sigma^2))\exp(-y^2/(2\sigma^2))\sum_{j=0}^{\infty}\frac{1}{j!}\left(xy/\sigma^2\right)^j
\end{aligned}$$

This gives

$$\phi(x) = \exp(-x^2/(2\sigma^2))\left(1, \frac{x}{\sigma}, \frac{1}{2!}\left(\frac{x}{\sigma}\right)^2, \frac{1}{3!}\left(\frac{x}{\sigma}\right)^3 \ldots\right)$$

The product above is called RBF kernel or Gaussian kernel. We will revisit in the next lecture.