CS 181 Spring 2016 Section 5 Notes (Support Vector Machines)

1 Motivation

The idea for Support Vector Machines is that, for all the linear hyperplanes that exist, we want one that will create the largest space, or "margin", with the data. This gives us more insurance about our classifications being correct. To define the margin, we consider a hyperplane of the form

$$\boldsymbol{w}^T\boldsymbol{x} + b = 0$$

What is the perpendicular distance from a data point $\mathbf{x_n}$ to the decision boundary $y_w(x)$?

For two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ on the hyperplane, consider the projection with $\boldsymbol{w}$:

$$\boldsymbol{w}(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \boldsymbol{w}\boldsymbol{x}_1 - \boldsymbol{w}\boldsymbol{x}_2 = -b - (-b) = 0$$

Therefore, $\boldsymbol{w}$ is orthogonal to the hyperplane. So to get the distance between an arbitrary data point $\boldsymbol{x}$, we just need the length of the vector in the direction of $\boldsymbol{w}$ between the point and the vector. We let r signify the distance between a point and the hyperplane. Then $\boldsymbol{x}_\perp$ is defined as the vector on the hyperplane to satisfy the equation

$$\boldsymbol{x}_\perp + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2} = \boldsymbol{x}_n$$

Left multiply by $\mathbf{w}^T$:

$$\boldsymbol{w}^T\boldsymbol{x}_\perp + r\frac{\boldsymbol{w}^T\boldsymbol{w}}{\|\boldsymbol{w}\|_2} = \boldsymbol{w}^T\boldsymbol{x}_n \Rightarrow r = \frac{\boldsymbol{w}^T\boldsymbol{x}_n + b}{\|\boldsymbol{w}\|_2}$$

r then gives the distance between a point and the hyperplane. To make sure it's positive, we multiply by the label. Therefore, the margin of the dataset is the minimum such margin over all data:

$$\min_n \frac{t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b)}{\|\boldsymbol{w}\|_2}$$

What is the maximization problem for support vector machines?

We want the $\boldsymbol{w}$ and b that maximize the margin:

$$\arg\max_{\boldsymbol{w},b} \frac{1}{\|\boldsymbol{w}\|_2} \min_n t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b)$$

We can observe that $\boldsymbol{w}$ and b are invariant to changes of scale; therefore, without loss of generality, we can set

$$\min_n t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) = 1$$

So our optimization becomes

$$\arg\max_{w,b} \frac{1}{\|w\|_2} \text{ s.t. } \forall n \; t_n(w^T x_n + b) \geq 1$$

What is the corresponding minimization problem for support vector machines? Explain (at a high level) why the minimization problem for support vector machines is equivalent to the max-margin problem

We can invert w to change the max to a min:

$$\arg\min_{w,b} \frac{1}{2}\|w\|_2^2 \text{ s.t. } \forall n \; t_n(w^T x_n + b) \geq 1$$

Minimizing $\frac{1}{2}||w||^2$ is equivalent to maximizing $\frac{1}{||w||}$ because $||w|| \geq 0$. Furthermore, we note that the normalized orthogonal distance from a point to the decision boundary is invariant under scalar multiplication. To see this, we have

$$\frac{t_n(\mathbf{w^T}\mathbf{\Phi}(\mathbf{x_n}) + b)}{||w||^2} = \frac{\beta}{\beta} \cdot \frac{t_n(\mathbf{w^T}\mathbf{\Phi}(\mathbf{x_n}) + b)}{||w||^2} = \frac{t_n(\beta\mathbf{w})^\mathbf{T}\mathbf{\Phi}(\mathbf{x_n}) + (\beta b))}{||\beta w||^2}$$

Thus, since the data is linearly separable, so there exists a decision boundary with non-zero, positive margin for each example, we do not lose any generality in imposing the restraint $t_n(\mathbf{w^T}\mathbf{\Phi}(\mathbf{x_n}) + b) \geq 1$ (because we can just scale w until the minimal value is ≥ 1).

2 Dual Form

Use Lagrange multipliers to convert the new minimization problem into dual form.

Our original optimization problem, which looks for the parameters that maximize the margin, is

$$w^\star, b^\star = \arg\min_{w,b} \left\{ \frac{1}{2}||w||_2^2 \right\} \quad \text{s.t.} \quad t_n \cdot (w^\mathsf{T}\boldsymbol{\phi}(x_n) + b) \geq 1, \quad \forall n \in \{1, \ldots, N\}. \tag{1}$$

We introduce *Lagrange multipliers*, $\alpha_1, \ldots, \alpha_N \geq 0$, one for each inequality in Equation 1, i.e., one per datum, to obtain the Lagrangian function (Bishop appendix E does a good job of explaining Lagrange multipliers):

$$L(w, b, \boldsymbol{\alpha}) = \frac{1}{2}||w||_2^2 - \sum_{n=1}^{N} \alpha_n (t_n \cdot (w^\mathsf{T}\boldsymbol{\phi}(x_n) + b) - 1) \tag{2}$$

Our ideal values are

$$\boldsymbol{w}^*, b^* = \arg\min_{\boldsymbol{w},b} \max_{\alpha \geq 0} L(\boldsymbol{w}, b, \alpha)$$

Since our objective is quadratic, and we have linear constraints, we can (by Slater's condition) change the order to derive:

$$\max_{\boldsymbol{\alpha} \geq 0} \min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \alpha)$$

Taking derivatives, we see

$$\frac{\partial L}{\partial w_j} = w_j - \sum_{n=1}^{N} \alpha_n t_n \boldsymbol{\phi}(\boldsymbol{x}_n) = 0 \Rightarrow \boldsymbol{w}^* = \sum_{n=1}^{N} \alpha_n t_n \boldsymbol{\phi}(\boldsymbol{x}_n)$$

$$\frac{\partial L}{\partial b} = -\sum_{n=1}^{N} \alpha_n t_n = 0 \Rightarrow \sum_{n=1}^{N} \alpha_n t_n = 0$$

Plugging these optimal values into our Lagrange multiplier, we get:

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}(\sum_{i=1}^{N} \alpha_n t_n \phi(x_n))^2 - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{\phi}(x_i)^\mathsf{T} \boldsymbol{\phi}(x_j) - b \sum_{i=1}\sum_{j=1} \alpha_n t_n + \sum_{i=1}^{N} \alpha_n \\ &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{\phi}(\boldsymbol{x}_i)^T \boldsymbol{\phi}(\boldsymbol{x}_j) \end{aligned}$$

We then solve for $\boldsymbol{\alpha}$, maximizing L, conditioned on $\sum_{n=1}^{N} \alpha_n t_n = 0$. If desired, we can then plug in the optimal $\boldsymbol{\alpha}$ to obtain w^* and b^*.

3 Why bother with the dual form?

Solving the primal form, we obtain the optimal $\boldsymbol{w}$, but know nothing about the α_n. To classify a point, we must compute $\boldsymbol{w}^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x})$, which is expensive if the J feature dimension, produced by $\boldsymbol{\phi}$, is large.

Solving the dual problem, we obtain α_n (where $\alpha_n = 0$ for all but a few points, i.e. the support vectors). To classify a point, we compute $\sum_{n=1}^{N} \alpha_n^\star \, t_n \, K(\boldsymbol{x}_n, \boldsymbol{x}) + b^\star$ which is efficient if there are few support vectors. Further, now that we have a scalar product involving only the data vectors, we can apply the **kernel trick**.

4 Kernel trick

Given features $\boldsymbol{\phi}(\boldsymbol{x})$ for each data point $\boldsymbol{x}$, computing the features $\boldsymbol{\phi}$ may be expensive if the dimensionality is large. It may be more efficient to directly compute the dot product $\boldsymbol{\phi}(\boldsymbol{x})^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x}')$. This is known as the kernel trick:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x}') \tag{3}$$

As long as the **kernel** (or **similarity measure**) K is a valid inner product, the dual SVM problem can thus be solved without actually computing $\boldsymbol{\phi}$.

5 Practice Problems

1. **(Berkeley, Fall '11)**

   Suppose that we train two SVM's, the first containing all of the sample data points and the second trained on a data set constructed by removing some of the support vectors from the first set. How does the size of the optimal margin change between the first and second SVM?

2. **Weights of Hard Margin SVM (MIT 6.867, Fall '12)**

   Consider a hard-margin SVM classification scenario where we have linearly separable data $\{\boldsymbol{\phi}_n, t_n\}_{n=1}^N$, where $t_n \in \{-1, 1\}$. As usual, for a new data-point $\boldsymbol{\phi}_{new}$, we predict 1 if $\boldsymbol{w}^\mathsf{T} \boldsymbol{\phi}_{new} + b \geq 0$, and -1 otherwise. Now, assume $N = 4$, and that our data is $\boldsymbol{\phi}_1 = \langle 1, 1 \rangle$, $\boldsymbol{\phi}_2 = \langle 2, 2 \rangle$, $\boldsymbol{\phi}_3 = \langle -1.5, -1.5 \rangle$, $\boldsymbol{\phi}_4 = \langle 4, 4 \rangle$. Show that for any labeling $t_1, \ldots, t_4$ of our 4 data-points, the $\boldsymbol{w}^*$ we learn by optimizing the hard-margin SVM criterion will have $w_1^* = w_2^*$.

3. **SVM with Only Positive Training Examples (MIT 6.867, Fall '12)**

Suppose we have data $D = \{\boldsymbol{\phi}_n, t_n\}_{n=1}^N$, where $t_n \in \{-1, 1\}$, and that we would like to learn a linear classification boundary by optimizing an SVM-like criterion that completely ignores the negative training examples. In particular, letting $D^+ = \{\boldsymbol{\phi}_n \in D \mid t_n = 1\}$ (i.e., the set of positive training examples), we will look for

$$\arg\min_{\boldsymbol{w}} \frac{1}{2}||\boldsymbol{w}||^2 \quad \text{s.t.} \quad \boldsymbol{w}^\mathsf{T}\boldsymbol{\phi}_i \geq 1, \forall\ \boldsymbol{\phi}_i \in D^+. \tag{4}$$

Note that the above optimization problem does not include an offset parameter b!

(a) If instead our optimization problem above *did* include an offset parameter b (i.e., we minimized $||\boldsymbol{w}||^2$ subject to $\boldsymbol{w}^\mathsf{T}\boldsymbol{\phi}_i + b \geq 1, \forall\ \boldsymbol{\phi}_i \in D^+$), what would the minimizing weight-vector $\boldsymbol{w}^*$ be?

(b) If we've found a $\boldsymbol{w}^*$ that minimizes Equation (4), what is the value of $\min_{\boldsymbol{\phi}_i \in D+} \boldsymbol{w}^{*\mathsf{T}}\boldsymbol{\phi}_i$?

(c) Again, assuming we've found a $\boldsymbol{w}^*$ that minimizes Equation (4), suppose that for a new data-point $\boldsymbol{\phi}_{new}$ we predict 1 if $\boldsymbol{w}^{*\mathsf{T}}\boldsymbol{\phi}_{new} \geq (\min_{\boldsymbol{\phi}_i \in D+} \boldsymbol{w}^{*\mathsf{T}}\boldsymbol{\phi}_i) - \epsilon$ for some small $\epsilon > 0$, and -1 otherwise. Will this decision rule guarantee that all the training examples in D (both positive and negative) are classified correctly?

4. **SVM with Three Points (MIT 6.867, Fall '12)**

Consider the following dataset consisting of three points on the real line

$$(x_1 = -1, t_1 = 1), (x_2 = 0, t_2 = -1), (x_3 = 1, t_3 = 1),$$

which we will attempt to separate with a linear hyperplane through feature-space *that goes through the origin*. That is, our discriminant function will be $\boldsymbol{w}^\mathsf{T}\phi(x) \geq 0$, with no offset term b. The primal form of this problem is:

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{n=1}^{3}\xi_n \\ & \text{s.t.} \quad t_n(\boldsymbol{w}^\mathsf{T}\phi(x_n)) \geq 1 - \xi_n, \quad \xi_n \geq 0, \quad i = 1,2,3 \end{aligned}$$

The dual form is:

$$\begin{aligned} \text{Maximize} \quad & \sum_{n=1}^{3} a_n - \frac{1}{2}\sum_{n=1}^{3}\sum_{m=1}^{3} a_n a_m t_n t_m k(x_n, x_m) \\ & \text{s.t.} \quad 0 \leq a_n \leq C, \quad i = 1,2,3 \end{aligned}$$

(a) Suppose our kernel function is $k(x, x') = 1 + |xx'|$, where $|\cdot|$ is the absolute value. What feature mapping ϕ results in this kernel function?

(b) Using the feature mapping ϕ from your answer to the previous question, are the three training examples linearly separable in feature space?

(c) Consider any pair of kernels $k_1(x, x')$ and $k_2(x, x')$ such that the training points are linearly separable with k_1 but not with k_2. Are the data linearly separable if we use the kernel $k(x, x') = k_1(x, x') + k_2(x, x')$? Justify your answer.

(d) Using the kernel $k(x, x') = 1 + |xx'|$ that we had in part (a), express the value of $\boldsymbol{w}^\mathsf{T}\phi(x_2)$ in terms of the a_n (i.e., in dual form).

(e) If in our optimization we set $C < 1$, will we necessarily have $\xi_2 > 0$ (i.e., will the slack variable for the second training example exceed 0)? (Hint: you will probably want to use the expression you derived in the previous question for $\boldsymbol{w}^\mathsf{T}\phi(x_2)$ in answering this).

5. **(MIT 6.867, Fall '12)**

Suppose that K_1 and K_2 are valid kernel. Recall that a valid kernel has the matrix $(K)_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a positive semi-definite matrix for any finite set of examples $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$. Show that $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K_1(\boldsymbol{x}_i, \boldsymbol{x}_j) + K_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a valid kernel if K_1, K_2 are both valid kernels.

6. **(Berkeley, Fall '11)**

Recall that the equation of an ellipse in the 2-dimensional plane is $c(x_1 - a)^2 + d(x_2 - b)^2 - 1 = 0$. Show that an SVM using the polynomial kernel of degree 2, $K(\boldsymbol{x}, \boldsymbol{z}) = (1 + \boldsymbol{x} \cdot \boldsymbol{z})^2$ is equivalent to a linear SVM in the feature space $(x_1^2, x_2^2, x_1, x_2, x_1 x_2, 1)$. This shows that SVM's with this kernel can separate any elliptic region from the rest of the plane.