Introduction to Machine Learning **Fall Semester**

Recitation 6

Lecturer: Regev Schweiger *Scribe: Regev Schweiger*

6.1 Leave One Out

We used the notion of leave-one-out error to derive in class a learning guarantee for SVMs based on the fraction of support vectors in the training set.

Definition 6.1 *Let h_S denote the hypothesis returned by a learning algorithm A, when trained on a fixed sample S. Then, the leave-one-out error of A on a sample S of size m is defined by*

$$\hat{R}_{LOO}(A) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(h_{S-\{\mathbf{x}_i\}}(\mathbf{x}_i) \neq y_i)$$

Thus, for each $i \in \{1, \ldots, m\}$, A is trained on all the points in S except for $\mathbf{x}_i$, i.e., $S - \{\mathbf{x}_i\}$, and its error is then computed using $\mathbf{x}_i$. The leave-one-out error is the average of these errors. In the bound presented in class, we used an important property of the leave-one-out error stated in the following lemma.

Theorem 6.2 *The average leave-one-out error for samples of size $m \geq 2$ is an unbiased estimate of the average generalization error for samples of size $m-1$:*

$$\mathrm{E}_{S\sim\mathcal{D}^m}[\hat{R}_{LOO}(A)] = \mathrm{E}_{S'\sim\mathcal{D}^{m-1}}[error(h_{S'})]$$

where $\mathcal{D}$ denotes the distribution according to which points are drawn.

Proof: By the linearity of expectation, we can write

$$\begin{aligned}
\mathrm{E}_{S\sim\mathcal{D}^m}[\hat{R}_{LOO}(A)] &= \frac{1}{m}\sum_{i=1}^{m} \mathrm{E}_{S\sim\mathcal{D}^m}[\mathbb{I}(h_{S-\{\mathbf{x}_i\}}(\mathbf{x}_i) \neq y_i)] \\
&= \mathrm{E}_{S\sim\mathcal{D}^m}[\mathbb{I}(h_{S-\{\mathbf{x}_1\}}(\mathbf{x}_1) \neq y_1)] \\
&= \mathrm{E}_{S'\sim\mathcal{D}^{m-1},\mathbf{x}_1\sim\mathcal{D}}[\mathbb{I}(h_{S'}(\mathbf{x}_1) \neq y_1)] \\
&= \mathrm{E}_{S'\sim\mathcal{D}^{m-1}}[error(h_{S'})]
\end{aligned}$$

□

6.2 Kernelizing algorithms

6.2.1 Review of kernel SVM

Let us review why SVM is suitable for using the kernel trick. SVM finds a linear separator - but if we embed $\mathbf{x}$ in a larger space with $\phi : \mathbf{x} \to \phi(\mathbf{x})$, then we would have to run SVM on a larger (possibly infinite-dimensional) space. However, we have seen from the dual problem, that $\mathbf{x}_i$-s are involved in the training stage only through dot products such as $\mathbf{x}_i \cdot \mathbf{x}_j$. Also, we have seen that the optimal $\mathbf{w}$ is a linear combination of the data points. In the kernel space, this translates into

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \phi(\mathbf{x}_i)$$

When applying an inner product of $\mathbf{w}$ with a data point, we get:

$$\mathbf{w} \cdot \phi(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i y_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x})$$

This shows us that for SVM, even though it didn't seem like it at the beginning, *the data points* $\mathbf{x}$ *are involved in the computation only through dot products with other data points.* This allows us to replace expressions like $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x})$ with $K(\mathbf{x}_i, \mathbf{x})$.

This observation allows us to apply the kernel trick on a variety of algorithms. Whenever we can show (or convert) that an algorithm uses only dot products between data points, we can replace these dot products with kernel functions.

6.2.2 Kernel k-NN

As an example, let us convert the k-NN algorithm to use kernels. The k-NN algorithm, for a given $\mathbf{x}$, sorts all the data points by the distance $\|\mathbf{x} - \mathbf{x}_i\|^2$, and uses the labels of the closest k points to make a prediction.

At first glance, we don't see dot products here, but rather the norm of a distance. But, on closer inspection, we see that

$$\|\mathbf{x} - \mathbf{x}_i\|^2 = \langle \mathbf{x} - \mathbf{x}_i, \mathbf{x} - \mathbf{x}_i \rangle = \langle \mathbf{x}, \mathbf{x} \rangle - 2 \langle \mathbf{x}, \mathbf{x}_i \rangle + \langle \mathbf{x}_i, \mathbf{x}_i \rangle$$

So, if we were to apply ϕ to each data point (and to the query point $\mathbf{x}$), we would get:

$$\begin{aligned} \|\phi(\mathbf{x}) - \phi(\mathbf{x}_i)\|^2 &= \langle \phi(\mathbf{x}) - \phi(\mathbf{x}_i), \phi(\mathbf{x}) - \phi(\mathbf{x}_i) \rangle \\ &= \langle \phi(\mathbf{x}), \phi(\mathbf{x}) \rangle - 2 \langle \phi(\mathbf{x}), \phi(\mathbf{x}_i) \rangle + \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_i) \rangle \\ &= K(\mathbf{x}, \mathbf{x}) - 2K(\mathbf{x}, \mathbf{x}_i) + K(\mathbf{x}_i, \mathbf{x}_i) \end{aligned}$$

So this is the measure by which we sort the data points.

6.3 Separating line for quadratic kernel in SVM

We will visualize the results of using a quadratic kernel for SVM, where $X = \mathbb{R}^2$. Let $\mathbf{x}_i \in \mathbb{R}^2$ be samples and $y_i \in \{1, -1\}$ their labels, and assume we use SVM with a quadratic kernel. We have seen that in $\mathbb{R}^2$, the quadratic kernel $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x} \cdot \mathbf{x}')^2$ corresponds to the function:

$$\phi(x_1, x_2) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

In the standard linear kernel, the separating line (i.e., the line separating between the regions of positive labels and negative lables) in $\mathbb{R}^2$ is a straight infinite line: Exactly those points $\mathbf{x}$ for which $\mathbf{w} \cdot \mathbf{x} + b = 0 \Leftrightarrow w_1x_1 + w_2x_2 + b = 0$.

For the quadratic case, we can think of the SVM as equivalently embedding each point (x_1, x_2) in $\mathbb{R}^6$ using ϕ, and then solving a standard linear SVM. Denote the result of the SVM by $\mathbf{w}, b$. Note that $\mathbf{w} \in \mathbb{R}^6$, and therefore, the set of points on the separating region are exactly the points $\mathbf{x} = (x_1, x_2)$ for which

$$\phi(\mathbf{x}) \cdot \mathbf{w} + b = 0 \Rightarrow$$
$$(1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2) \cdot (w_1, w_2, \ldots, w_6) + b = 0$$

What are the possible shapes for the set of points that satisfy this equation? This equation is in fact the general equation for a quadratic curves. So, depending on $\mathbf{w}$, we can get any possible quadratic curve. For example, the separating line can be a circle around the origin if $w_1 = w_2 = w_3 = w_6 = 0, b = -1$ and $w_4 = w_5 = 1/R^2$. Then, we get the equation:

$$R^2 = x_1^2 + x_2^2$$

With other choices for $\mathbf{w}$, we could get all other quadratic curves, such as an ellipse, hyperbola, parabola; we can also get a straight line, as before, if $w_4 = w_5 = w_6 = 0$.

6.4 Separating line for Gaussian kernel in SVM

Now we will try to see how the separating line for a Gaussian kernel looks like when $X = \mathbb{R}^2$. Recall that the Gaussian kernel is:

$$K(\mathbf{x}, \mathbf{x}') = e^{-\frac{1}{\sigma^2}\|\mathbf{x}-\mathbf{x}'\|^2}$$

Now, we cannot use the same line of reasoning we used in the quadratic case, because the kernel space is infinite-dimensional. However, we know that

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \phi(\mathbf{x}_i)$$

Therefore, the set of points on the separating region is those for which

$$\begin{aligned} 0 &= \phi(\mathbf{x}) \cdot \mathbf{w} + b \\ &= \phi(\mathbf{x}) \cdot \sum_{i=1}^{m} \alpha_i y_i \phi(\mathbf{x}_i) + b \\ &= \sum_{i=1}^{m} \alpha_i y_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) + b \\ &= \sum_{i=1}^{m} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \\ &= \sum_{i=1}^{m} \alpha_i y_i e^{-\frac{1}{\sigma^2}\|\mathbf{x}-\mathbf{x}_i\|^2} + b \end{aligned}$$

To make sense of this equation, Let's look at one summand only:

$$e^{-\frac{1}{\sigma^2}\|\mathbf{x}-\mathbf{x}_i\|^2} = k$$

where $k = -b/\alpha_i y_i$. Therefore, this is a "latitude line" of a Gaussian centered around $\mathbf{x}_i$. From this, it is easy to see the general line is a the latitude line of a linear combination of Gaussians around each one of the support vectors (for which we know $\alpha_i > 0$). See Figures in lesson slides.