Kernels I

Kernels allow us to make complex, non-linear classifiers using Support Vector Machines.

Given x, compute new feature depending on proximity to landmarks l^1, l^2, l^3.

To do this, we find the "similarity" of x and some landmark l^i

$$f_i = similarity(x,\ l^i) = \exp\left(-\frac{\left||x - l^i|\right|^2}{2\sigma^2}\right)$$

This "similarity" function is called a **Gaussian Kernel**. It is a specific example of a kernel.

The similarity function can also be written as follows:

$$f_i = similarity(x,\ l^i) = \exp\left(-\frac{\sum_{j=1}^{n}\left(x_j - l_j^i\right)^2}{2\sigma^2}\right)$$

There are a couple properties of the similarity function:

$$if\ x \approx l^i,\ then\ f_i = \exp\left(-\frac{\approx 0^2}{2\sigma^2}\right) \approx 1$$

$$if\ x\ is\ far\ from\ l^i,\ then\ f_i = \exp\left(-\frac{(large\ number)^2}{2\sigma^2}\right) \approx 0$$

In other words, if x and the landmark are close, then the similarity will be close to 1, and if x and the landmark are far away from each other, the similarity will be close to 0.

Each landmark gives us the features in our hypothesis:

$$
\begin{gathered}
l^1 \rightarrow f_1 \\
l^2 \rightarrow f_2 \\
l^3 \rightarrow f_3 \\
\ldots \\
\mathrm{h}_\Theta(x) = \Theta_1 f_1 + \Theta_2 f_2 + \cdots
\end{gathered}
$$

σ^2 is a parameter of the Gaussian Kernel, and it can be modified to increase or decrease the **drop-off** of our feature f_i. Combined with looking at the values inside Θ, we can choose these landmarks to get the general shape of the decision boundary.

Kernels II

One way to get the landmarks is to put them in the **exact same** locations as all the training examples. This gives us m landmarks, with one landmark per training example.

Given example x:

$f_1 = similarity(x, l^1)$, $f_2 = similarity(x, l^2)$, $f_3 = similarity(x, l^3)$, and so on.

This gives us a "feature vector," f_i of all our features for example x_i. We may also set $f_0 = 1$ to correspond with Θ_0. Thus given training example x_i:

$$x^i \rightarrow \left[f_1^i = similarity\left(x^i, l^1\right),\ f_2 = similarity\left(x^i, l^2\right),\ f_3 = similarity\left(x^i, l^3\right)\right]$$

Now to get the parameters Θ we can use the SVM minimization algorithm but with f_i substituted in for x_i:

$$J(\theta) = C\sum_{i=1}^{m}\left[y^i cost_1(\theta^T f^i) + \left(1 - y^i\right)(cost_0(\theta^T f^i))\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

Using kernels to generate f(i) is not exclusive to SVMs and may also be applied to logistic regression. However, because of computational optimizations on SVMs, kernels combined with SVMs is much faster than with other algorithms, so kernels are almost always found combined only with SVMs.