07.01 - Support Vector Machines

1 Optimization objective

In support vector machine, we don't directly have λ. Instead, C is used, which penalizes the training set instead of the θ. In reality, it's almost the same, as we can set a small value for C and then we're giving more importance to the regularization term. We could think of $C = \frac{1}{\lambda}$.

A part from that, the objective function (minimizing cost function) is almost the same as the logistic regression one:

$$min_\theta C \sum_{i=1}^{m}[y^{(i)}cost_1(\theta^T x^{(i)}) + (1 - y^{(i)})cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2 \tag{1}$$

Where $cost_0$ and $cost_1$ are functions that are applied when $y = 0$ and $y = 1$, respectively.

SVMs don't output probabilities, they output directly the class they classificate into, using the following hyphotesis:

$$h_\theta(x) = \begin{cases} 1, & \text{if}\, \theta^T x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

2 Large Margin Intuition

We can see the SVM as a large margin classifier, meaning that if we have linearly separable data, it will separate it optimizing the boundary distance between the data (fig 1).

3 Kernels I

In logistic regression, when we had a non-linear problem, we had to use polynomial features. Kernels are another way of getting more features out of the training sample.

Kernels are similarity functions. In the video, he selects 3 landmarks and uses a gaussian kernel. He then uses those 3 landmarks to compute 3 features, using the gaussian to know the similarity between x and each of the landmark.

Doing that, we can create a decision boundary around the landmarks, where everything near them gets classified as 1 and everything that is far from them gets classified as 0 (following a gaussian shape).

4 Kernels II

4.1 Choosing Landmarks

For every training example, get a landmark exactly at that location. Then, features will be based around the similarity of a point in our data to other

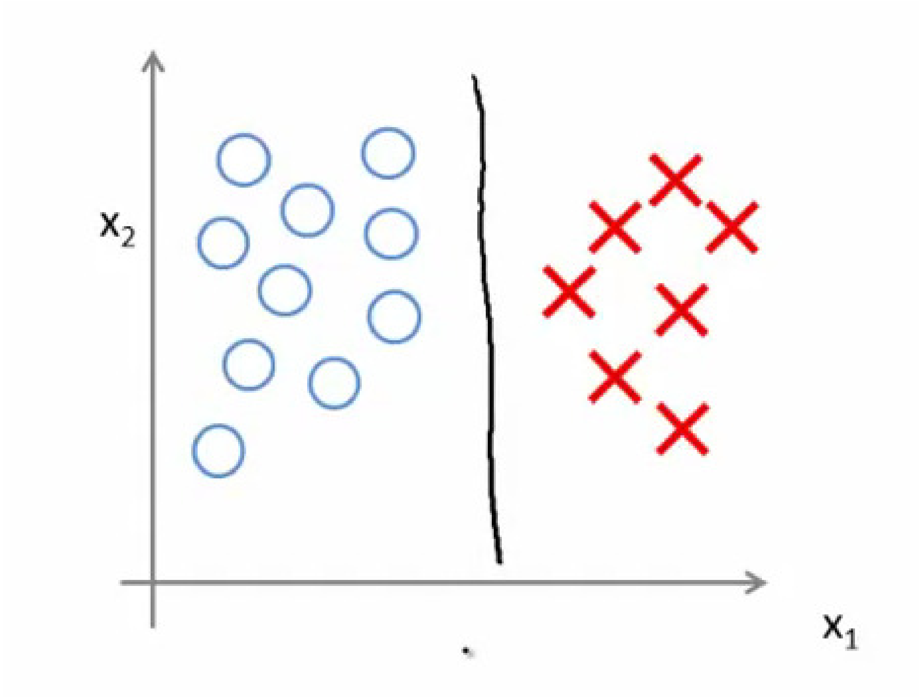

Figure 1: Example of linearly separable data correctly classified by an SVM.

points. This way we have one feature per datapoint, so we have m features.

Basically, we just substitute the features we had and we use this ones.

Mathematical implemenation note:

$$\sum_{j=1}^{n} \theta_j^2 = \theta^t M \theta \tag{3}$$

Where M is a matrix that depends on the kernel. This is done to scale to bigger training sets.

4.2 SVM Parameters

- Large C: Lower bias, high variance (small λ).
- Small C: Higher bias, low variance (large λ).

5 Using an SVM

- Use SVM software package to solve for paraemters θ.
- You still need to specify the choice of parameter C and the kernel.
- No kernel is sometimes called “linear kernel”.
- Do perfrom feature scaling before using the kernel.

Not all similarity functions make valid kernels. They need to satisfy a technical condition called “Mercer’s Theorem”.

5.1 Multiclass classification

Many SVM packages already have built-in multi-class classification functionality. Otherwise, use one-vs-all.

5.2 Logistic regression vs SVMs

Let n = number of features and m = number of training examples.

- If n is large relative to m -> Use logistic regression or SVM without kernel.
- If n is small, m is intermediate -> Use SVM with Gaussian kernel.
- If n is small, m is large -> Create/Add features and then use logistic regression or SVM without kernel.

Neural networks is likely to work well for mos of these settings, but they’re way slower to train.

SVMs don’t have to worry about local optima, because they solve a convex problem. NN may have problems with that, though.