Alin Tomescu
6.867 Machine learning | Week 2, Tuesday, September 10th, 2013| Lecture 2

Lecture 2: Supervised learning continued

Send questions to 6.867-staff@lists.csail.mit.edu for faster responses!

All homework-related questions should be posted by Friday!

Supervised learning: Given a few examples find a predictor function that works for new examples.

We are given a training set. We'll primarily talk about batch / offline supervised classification where training data is given upfront.

Training set: $S_n = \{(x^i, y^i), i = 1, \dots, n\}, (x^i, y^i) \sim p^*$

We select $\hat{h}: X = R^d \to Y = \{-1,1\}$

Goal is to minimize **generalization error** (risk):

$$R(\hat{h}) = E(x,y) \sim p^* \left\{Loss\left(y, \hat{h}(x)\right)\right\}$$

(expected value of loss on pairs sampled over p^*)

$$Loss\left(y, \hat{h}(x)\right) = \begin{cases} 1, if\ y \neq y' \\ 0, o.w. \end{cases}$$

Generative approach to solve the supervised learning problem

If I knew p^* I could perform optimal. I could evaluate what the generalization error is and find an $\hat{h}$ that would minimize it.

Estimate $\hat{p}(x, y)$ based on the training set S_n. We need constraints for estimating.

Once we have a guess on p^* we'll use a predictor $\hat{h}(\cdot) = \underset{h}{\text{argmin}}\ E(x,y) \sim \hat{p}\ \{Loss(y, h(x))\}$

Discriminative approach

$$\hat{R}_n(h) = \frac{1}{n} \sum_{i=1}^{n} Loss\left(y^i, h(x^i)\right)$$

$$h \in H - set\ of\ classifiers$$

We want to find an h minimize $\hat{R}_n(h)$

Use $\hat{h}(.) = \underset{h \in H}{argmin} \hat{R}_n(h)$

Alin Tomescu
6.867 Machine learning | Week 2, Tuesday, September 10th, 2013| Lecture 2

Linear classifiers

$$X = R^d$$

$$Y = \{-1,1\}$$

$$h(x;\theta,\theta_0) = sign(\theta x + \theta_0) = \begin{cases} +1, if\ \theta x + \theta_0 > 0 \\ -1, otherwise \end{cases}$$

You can get arbitrarily complex classifiers if you know how to handle linear classifiers.

$$\theta\vec{x} + \theta_0 = 0 \Leftrightarrow \theta(\vec{x} - \overrightarrow{x_0}) = 0 \Rightarrow \theta\overrightarrow{x_0} = -\theta_0 \Rightarrow \theta_0 = -\theta\overrightarrow{x_0}$$

Distance to boundary $\frac{y(\theta x)}{\|\theta\|}$, where $\|\theta\| = norm\ of\ theta$

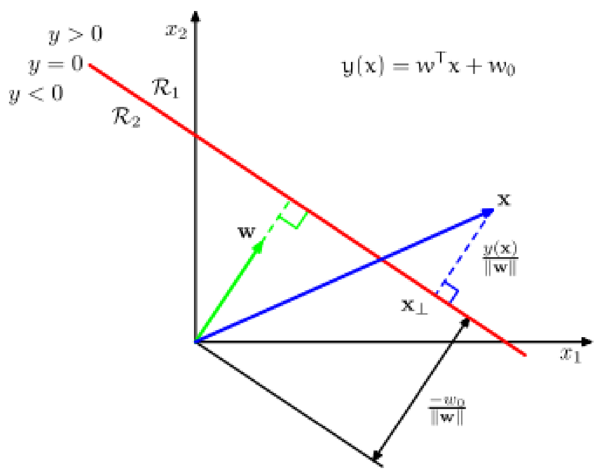

Figure 1: Geometry of linear discriminant functions (Bishop Figure 4.1).

TODO: Put graph in

Training without errors

Assumption 1: Training examples are linearly separable with margin γ:

$$\exists\theta^* s.t. \forall i, \frac{y^i\theta^* x^i}{\|\theta\|} > \gamma, \gamma > 0, i = 1, \dots, n$$

This says that all training examples are at least distance γ from the boundary.

Put another way, this means the examples are **linearly separable.**

Assumption 2: Training examples are bounded by a sphere/circle of radius r: $\|x^{(i)}\| \leq r, i = 1, \dots, n$

Perceptron algorithm

- Start at step 0: $\theta^{(0)} = 0$ (vector)
- Cycle through training samples correcting errors

Alin Tomescu

- If $y^{(i)}\left(\theta^{(k)}x^{(i)}\right) \leq 0$ (mistake), then $\theta^{(k+1)} = \theta^{(k)} + y^{(i)}x^{(i)}$

Thus, if our two assumptions hold, then our perception algorithm makes at most $\frac{r^2}{\gamma^2}$ mistakes.

The number of mistakes does not depend on the number of training examples or on the dimension of X, $\dim(X)$

Training without errors online

Assumption 1: Training examples are linearly separable with margin γ:

$$\exists \theta^* s.t. \forall \frac{y^i \theta^* x^i}{\|\theta\|} > \gamma, \gamma > 0, i = 1, \dots, \infty$$

Assumption 2: Training examples are bounded by a sphere/circle or radius r: $\left\|x^{(i)}\right\| \leq r, i = 1, \dots, \infty$

Perceptron algorithm online

- Start at step 0: $\theta^{(0)} = 0$ (vector)
- Cycle through training samples correcting errors
- If $y^{(i)}\left(\theta^{(k)}x^{(i)}\right) \leq 0$ (mistake), then $\theta^{(k+1)} = \theta^{(k)} + y^{(i)}x^{(i)}$

Once again, if our two assumptions hold, then our perception algorithm makes at most $\frac{r^2}{\gamma^2}$ mistakes.

Why the r^2/γ^2 bound?

$\cos\left(\theta^k, \theta^*\right) = \frac{\theta^k \theta^*}{\|\theta^k\| \times \|\theta^*\|}$, where θ^k is theta after k updates and theta star is the theta we assume exists

Step 1: Show that as we keep updating $\frac{\theta^k \theta^*}{||\theta^*||} \geq k\gamma$

$$\frac{\theta^k \theta^*}{||\theta^*||} = \frac{\left(\theta^{k-1} + y^i x^i\right) \times \theta^*}{||\theta^*||} = \frac{\theta^{k-1} \times \theta^*}{||\theta^*||} + \frac{y^i x^i \times \theta^*}{||\theta^*||}$$

$$\frac{y^i x^i \times \theta^*}{||\theta^*||} \geq \gamma$$

$$\Rightarrow \frac{\theta^k \theta^*}{\|\theta^*\|} \geq k\gamma$$

Step 2: Norm of our parameter vector does not increase too high: $\left\|\theta^k\right\|^2 \leq kr^2$

- We only update based on a mistake. We are correcting mistakes, which keeps the norm in check.

$$\cos\left(\theta^k, \theta^*\right) = \frac{\theta^k \theta^*}{\|\theta^k\| \times \|\theta^k\|} \geq \frac{k\gamma}{\|\theta^k\|} \geq \frac{k\gamma}{\sqrt{k} \times r} = \sqrt{k}\frac{\gamma}{r}$$

Why use the perception algorithm when we can find the maximum margin linear separator directly? We know γ.

Alin Tomescu

Maximum margin linear separator

$$minimizing \frac{1}{2}\|\theta\|^2$$

$$y^i \theta x^i \geq 1, \forall i$$

Unique answer