36-700: Probability and Mathematical Statistics I **Fall 2016**

Lecture 33: November 30

Lecturer: Siva Balakrishnan

33.1 Review and Outline

In the last class we began discussing classification:

1. The basic setup of classification
2. Error rate of a classifier
3. The Bayes classifier
4. Linear discriminant analysis
5. Logistic regression

33.2 Fitting logistic regression

In the last lecture we discussed logistic regression models where we assume that the probability:

$$P(Y = 1|X) = \frac{\exp(\beta_0 + \beta^T X)}{1 + \exp(\beta_0 + \beta^T X)}.$$

In order to fit a logistic regression model, we maximize the conditional likelihood, i.e. we observe samples of the form $(X_1, Y_1), \ldots, (X_n, Y_n)$ and we maximize,

$$\begin{aligned}\mathcal{L}(\beta_0, \beta) &= \prod_{i=1}^{n} P(Y_i = 1|X_i) \\ &= \prod_{i=1}^{n} \left(\frac{\exp(\beta_0 + \beta^T X_i)}{1 + \exp(\beta_0 + \beta^T X_i)}\right)^{Y_i} \left(\frac{\exp(\beta_0 + \beta^T X_i)}{1 + \exp(\beta_0 + \beta^T X_i)}\right)^{1-Y_i}.\end{aligned}$$

Unlike in linear regression, we cannot maximize the likelihood in closed form, so instead we use a method like gradient ascent. It turns out that the log-likelihood is a concave function so we can find the $(\widehat{\beta}_0, \widehat{\beta})$ that maximize the log-likelihood.

33.3 Connection between logistic regression and LDA

We have seen in the last class that in both LDA and logistic regression our decision boundary is linear, i.e. we declare that $Y = 1$ if

$$\alpha_0 + \alpha^T X \geq 0,$$

for some scalar α_0 and vector $\alpha \in \mathbb{R}^d$.

There is however, an important difference between LDA and logistic regression. In the two cases, we specify different likelihoods and fit the models differently.

In the LDA case, we used the joint likelihood:

$$\begin{aligned}\mathcal{L}(\beta_0, \beta) &= \prod_{i=1}^{n} P(X_i, Y_i) \\ &= \prod_{i=1}^{n} P(Y_i) P(X_i|Y_i),\end{aligned}$$

where we assumed that the first term is Bernoulli, while the second term is Gaussian.

On the other hand for logistic regression, we use the conditional likelihood:

$$\mathcal{L}(\beta_0, \beta) = \prod_{i=1}^{n} P(Y_i|X_i)$$

which we assumed was a logistic function. In this case, we do not even model the distribution of X.

In machine learning parlance, classifiers like LDA are called generative classifiers while classifiers like logistic regression are called discriminative classifiers.

Once you fit a generative classifier, you can generate new data, i.e. since you have estimated $P(Y = 1)$, $P(X|Y = 1)$ and $P(X|Y = 0)$, you can generate data by sampling Y and then sampling from the appropriate conditional distribution.

On the other hand, once you fit a logistic regressor you do not have any way to generate new Xs since you did not model them. You can however make predictions, i.e. predict Y given X which really is the goal of classification.

Often logistic regression will give superior predictions to LDA, because LDA is making a strong parametric assumption about the conditionals $P(X|Y)$ which are often not true. More broadly, we often prefer discriminative classifiers to generative ones when the goal is prediction.

33.4 Naive Bayes

Another popular generative method is called Naive Bayes. Here rather than assuming that the conditional $P(X|Y)$ is Gaussian we assume that the features are *independent*, i.e.:

$$P(X|Y) = \prod_{i=1}^{d} P(X^i|Y),$$

and then to use non-parametric density estimation to estimate the densities $P(X^i|Y)$. We estimate the probability $P(Y = 1)$ as before and then classify a new point as belonging to class 1 if:

$$\widehat{P}(Y = 1) \prod_{i=1}^{n} \widehat{P}(X^i|Y = 1) \geq (1 - \widehat{P}(Y = 1)) \prod_{i=1}^{n} \widehat{P}(X^i|Y = 0).$$

33.5 Empirical risk minimization

The other broad strategy is based on empirical risk minimization. In empirical risk minimization we select a set of classifiers $\mathcal{H}$ and then simply choose the one that minimizes the empirical risk, i.e. the classifier that is most accurate on the training data.

We pick the classifier:

$$\widehat{h} = \arg\min_{h \in \mathcal{H}} \widehat{L}(h).$$

A natural question is then: how good is $\widehat{h}$? This is very similar to our analysis of cross-validation.

We will first focus on the simple case, when the number of classifiers is small/finite. The best classifier is then:

$$h^* = \arg\min_{h \in \mathcal{H}} L(h),$$

and we would like to understand the difference $\Delta = L(\widehat{h}) - L(h^*)$, i.e. how much worse is the classifier we chose compared to the best classifier in $\mathcal{H}$.

In order to do this, we can use the Hoeffding bound, i.e. we know that for a fixed classifier h,

$$P(|\widehat{L}(h) - L(h)| \geq \epsilon) \leq 2 \exp(-2n\epsilon^2).$$

Why is this true?

Using the union bound we then have that:

$$P(\sup_{h\in\mathcal{H}} |\widehat{L}(h) - L(h)| \geq \epsilon) \leq 2|\mathcal{H}| \exp(-2n\epsilon^2).$$

Now, suppose we want to argue that often Δ is small. We can invert the above inequality to conclude that with probability at least $1-\delta$,

$$\sup_{h\in\mathcal{H}} |\widehat{L}(h) - L(h)| \leq \sqrt{\frac{2}{n}\log\left(\frac{2|\mathcal{H}|}{\delta}\right)} := \alpha.$$

Then we have that:

$$\begin{aligned}
\Delta &= L(\widehat{h}) - L(h^*) = L(\widehat{h}) - \widehat{L}(\widehat{h}) + \widehat{L}(\widehat{h}) - L(h^*) + \widehat{L}(h^*) - \widehat{L}(h^*) \\
&= L(\widehat{h}) - \widehat{L}(\widehat{h}) - (L(h^*) - \widehat{L}(h^*)) + \widehat{L}(\widehat{h}) - \widehat{L}(h^*) \\
&\leq L(\widehat{h}) - \widehat{L}(\widehat{h}) - (L(h^*) - \widehat{L}(h^*)) \\
&\leq |L(\widehat{h}) - \widehat{L}(\widehat{h})| + |L(h^*) - \widehat{L}(h^*)| \\
&\leq 2\alpha.
\end{aligned}$$

What this means is that for sufficiently large sample sizes, empirical risk minimization will pick close to the best possible classifier.

ERM is actually a very active area of research. There are two broad themes to the work in ERM.

On the statistical front, we often have $\mathcal{H}$ that are infinite, i.e. for instance we want to know how well does the best linear classifier do – there are infinitely many linear classifiers. Of course, many of the linear classifiers are very similar so our union bound is quite loose. There are many elegant ways (see for instance VC dimension) to deal with this.

On the computational front, if we have an infinite $\mathcal{H}$ (or even a very large one) it is most often not possible to compute the risk minimizer, i.e. finding the best linear classifier for example is a computationally intractable problem in general. In this case, we often replace the training error with something that is easier to minimize (like the logistic loss or hinge loss) and then try to argue that minimizing this easier loss is not much worse.