Predictions mathematics of LDA

LDA makes predictions by estimating the probability that a new set of inputs belongs to each class. The class that gets the highest probability is the output class and a prediction is made.

The model uses Bayes Theorem to estimate the probabilities. Bayes Theorem can be used to estimate the probability of the output class (k) given the input (x) using the probability of each class and the probability of the data belonging to each class:

P(Y=x|X=x) = (PIk * fk(x)) / sum(PIl * fl(x))

Where PIk refers to the base probability of each class (k) observed in your training data (e.g. 0.5 for a 50-50 split in a two class problem). In Bayes' Theorem this is called the prior probability.

PIk = nk/n

The f(x) above is the estimated probability of x belonging to the class. A Gaussian distribution function is used for f(x). Plugging the Gaussian into the above equation and simplifying we end up with the equation below. This is called a discriminate function and the class is calculated as having the largest value will be the output classification (y):

Dk(x) = x * (muk/siga^2) – (muk^2/(2*sigma^2)) + ln(PIk)

Dk(x) is the discriminate function for class k given input x, the muk, sigma^2 and PIk are all estimated from your data.

To understand some of topic and term of mentioned description-

LDA makes some simplifying assumptions about your data:

That your data is Gaussian, that each variable is is shaped like a bell curve when plotted.

That each attribute has the same variance, that values of each variable vary around the mean by the same amount on average.

With these assumptions, the LDA model estimates the mean and variance from your data for each class. It is easy to think about this in the univariate (single input variable) case with two classes.

The mean (mu) value of each input (x) for each class (k) can be estimated in the normal way by dividing the sum of values by the total number of values.

muk = 1/nk * sum(x)

Where muk is the mean value of x for the class k, nk is the number of instances with class k. The variance is calculated across all classes as the average squared difference of each value from the mean.

sigma^2 = 1 / (n-K) * sum((x – mu)^2)

Where sigma^2 is the variance across all inputs (x), n is the number of instances, K is the number of classes and mu is the mean for input x.

Here is an example:

Latent Dirichlet Allocation imagines that each document is a distribution over topics in your dataset, like {Topic 1: 0.8, Topic 2: 0.0, Topic 3: 0.1, Topic 4: 0.1}. Each of those topics, as we know, is a distribution over words, like {President: 0.5, united: 0.25, states: 0.25}. Now, to generate a document from that topic distribution, we first choose a random topic according to those probabilities (so we’d be very likely to choose topic 1, unlikely to choose topics 3 or 4, and would never choose topic 2). Once we’ve chosen our topic, we choose a word from that topic (so we’d choose “President” with high probability, and “united” or “states” with lower probability). That gives us a single word in our document. Then we repeat the process over and over again until we have however long of a document we’d like. Et Voilà, we’ve generated a document from a topic distribution!