LDA vs QDA vs FDA, and PCoA/MDS

David G. Khachatrian

August 15, 2019

1 LDA vs QDA vs FDA

We distinguish between

1. Fisher's Discriminant Analysis (FDA), which is a method of *feature extraction/dimensionality reduction*
2. (Linear/Quadratic) Discriminant Analysis (L/QDA), which is a classifier.

1.1 FDA

In FDA, you have n data points in $\mathbb{R}^d$ labeled into k categories. You'd like to reduce the number of dimensions to $m < d$ while

1. maximizing distance between clusters
2. minimizing variance within each cluster

In particular, we want to choose $w \in \mathbb{R}^{m \times d}$ that maximizes

$$\frac{w^T s_B w}{w^T s_W w} \tag{1}$$

where s_B represents the variance between clusters and s_W represents the variances within each cluster. You can solve this using e.g. Lagrange multipliers (we're only interested in the direction of w, so an equivalent optimization problem is to maximize $w^T s_B w$ while restricting $w^T s_W w := 1$) and you'll find that the optimal w satisfies $s_W^{-1} s_B w = \lambda w$. So w are the eigenvectors of $s_W^{-1} s_B$, and in fact you are limited to having $m := k-1$.

1.2 LDA/QDA

In LDA and QDA, we still have n data points labeled into k categories, but now we want to make a classifier using this dataset.

We make one key modeling assumption: *We assume the data for each label comes from a multivariate Normal (Gaussian) distribution.* From there you can give each distribution its μ and Σ using their usual maximum-likelihood-estimation (MLE) estimators.

Our decision boundaries will be where the probability of generating the datapoint is equal between two clusters. We classify a point x into the category whose associated Gaussian distribution has the highest probability density at that point.

Going through the math of

$$P(x \mid x \text{ came from category j}) = P(x \mid x \text{ came from category k})$$

you'll find while simplifying that there will be quadratic forms $x^T \Sigma_j x + \cdots - x^T \Sigma_k x = 0$. This means the final decision boundary between two classes will be **quadratic** (hence *quadratic* discriminant analysis). You

can piece together these pairwise decision boundaries to form your decision boundary for all the different classes.

If we make another simplifying assumption, namely that the covariance matrices for each category's Normal distribution are the *same*, then the quadratic forms cancel each other out and you'll have *linear* decision boundaries. Including this second assumption makes the classifier a *linear discriminant analysis classifier*.

Good links: https://www.youtube.com/playlist?list=PLehuLRPyt1Hy-4ObWBK4AbOxk97s6imfC (Lectures 2,3,4)

-8/15/19

2 PCoA/Classical MDS

In Principal Component Analysis (PCA), we have a design matrix X $(n \times p)$ and we want to find *principal components* which best describe the covariance of the data: $\Sigma_d = X_c^T X_c \in \mathbb{R}^{d \times d}$. What this amounts to is obtaining the eigenvectors v_i for Σ, whose percentage of variance is determined by the proportion $\lambda_i / \sum_i \lambda_i$.

Consider instead $K(L^2) := X_c X_c^T \in \mathbb{R}^{n \times n}$. This contains the same information as Σ, just packaged in a different way. From considering the SVD of X_c, we'd see that $K(L^2) = X_c X_c^T$ and $\Sigma = X_c^T X_c$ have the same eigenvalues. So if we were to perform PCA on $K(L^2)$ instead of Σ, we'd get principal directions that, axis-per-axis, describe the same proportion of the covariance! The only difference is that in the description for Σ, the principal components are described as a linear combination of the d dimensions, whereas when using $K(L^2)$ we'd end up describing them as a linear combination of the n samples. (Might be an odd thought to have "eigensamples", but it's worth pondering.)

I've laboriously kept writing the L^2 in $K(L^2)$ to denote that $X_c X_c^T$ gives the squared L^2 distances between the samples. That is, we choose $d(x, y) = \|x - y\|_2^2$.

In *Principal Coordinate Analysis (PCoA)*, also known as *classical/Torgerson's Multidimensional Scaling (Classical/Torgerson's MDS)*, **we choose whatever distance function d we'd like** in order to construct $K(d)$. We then perform the same machinery (e.g., SVD) to find its eigenvalues. An example of this where this could be useful is in gene expression analysis, where one may be more interested in average absolute log-fold differences between genes, i.e., for samples $x \in \mathbb{R}^p$,

$$d(x, y) = \frac{\sum_{k=1}^{p} |\log(x_k / y_k)|}{p}$$

You could load K with entries according to the above distance metric, and your visualization may more closely reflect "proximity" according to one studying differences in gene expression.

Good links:

- https://stats.stackexchange.com/a/132731/216799
- https://www.youtube.com/watch?v=GEn-_dAyYME

-8/15/19