UNIVERSITY OF WISCONSIN
DATA SCIENCE

*DS 740*

# Data Mining

## LDA Extended and QDA

### Multiple Predictors and Nonconstant Variance

**Important note**: Transcripts are **not** substitutes for textbook assignments.

# Learning Objectives

By the end of this lesson, you will be able to:

- State (and check) underlying assumptions for LDA classification with multiple predictors.
- Understand basic reason for QDA classification and state (and check) underlying assumptions.
- Understand how estimators are differently developed.
- Recognize situations appropriate for multiple-predictor classification.
- Apply multiple-predictor LDA and QDA via `lda` and `qda` functions.
- Compare models via cross-validation.

## Multiple Predictors → Multivariate

Consider using $p$ predictors, so we use a **vector** of $x_i$ of length $p$, containing predictor values, to predict response $y_i$

Results in assumptions for **multivariate distributions** of $X$'s; for each class $k = 1, 2, \ldots, K$

1. density $f_k(x)$ is multivariate normal with mean **vector** $\mu_k$
2. **covariance matrices** are the same: $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_K = \Sigma$

Derivation of estimator is very similar

Referring back to the goal of discriminant analysis, in which we want to predict a qualitative response y into one of k classes using one or more predictors x, we focus on the more than one predictor of x. So we introduce how multiple predictors can be implemented into our process. If we have p predictors, each data point now has p predictor values, which we arrange in a vector x sub i. Each predictor has a mean value within the class k and we arrange these p means in a vector mu sub k, one for each class k.

So we have k vectors of length p describing the means. The covariance is measured by a matrix with variances on the diagonal and covariances off diagonal. While we allow the mu of k's to be different for each class, we begin by assuming the covariance matrices are the same across all groups. In this case, the estimators we will derive are very similar to what we saw in the one predictor case. And the forms look very similar.

## Result: Bayes Classifier for LDA

**Recall goal:** pick $k$ to $\max_k p_k(x) = \Pr(Y = k|X = x)$

Mathematically, it can be shown $k$ is chosen to:

$$\max_k \left\{ x^T (\Sigma^{-1}\mu_k) + \left( -\frac{1}{2}\mu_k^T \Sigma^{-1}\mu_k + \log(\pi_k) \right) \right\}$$

constants for each $k$ → linear function

Same basic form as before, just using mean **vectors** and covariance **matrices** now.

As part of the similarity, the goal function which we are trying to maximize, is essentially the same simply with involving vectors rather than individual values now. So when we come up with a simplified goal function, it looks extremely similar to what we had derived in the previous lecture, only now instead of a single x, we have a vector x. And instead of a single mu sub k, it is a mu sub k as a vector. And instead of a single sigma squared, we have a covariance matrix, p by p in dimension.

This resulting goal function is, again, a linear combination of elements of the X factor, and thus this is still accurately described as a linear discriminant analysis.

## Estimates for LDA (p predictors)

Values for constants $\pi_k$, $\mu_k$, and $\Sigma$ are also similarly estimated from the data (note: each $x_i$ is now a vector):

- $\pi_k$ known, or use $\hat{\pi}_k = \frac{n_k}{n}$ for class $k$ = 1, 2, ..., $K$
- Use $\hat{\mu}_k = \frac{1}{n_k} \sum_{\{i:y_i=k\}} x_i$ for class $k$ = 1, 2, ..., $K$
- Use $\hat{\Sigma} = \frac{1}{n-K} \sum_{k=1}^{K} \left[ \sum_{\{i:y_i=k\}} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T \right]$

Estimating total of $K + Kp + p(p+1)/2 = (K + p/2)(p+1)$ parameters from data (potentially quite a bit more)

Estimating the various parameters for the goal function is still necessary, but again takes a very similar form. The x sub i's and mu of k's are now simply vectors of length p. However, this does mean we have quite a few more parameters to estimate. We still wind up with k estimators, pi hat sub k. But since each mu hat sub k is estimating p parameters, we have k times p total parameters for the mean functions that need to be estimated.

And estimating the parameters along the diagonal entries and above on the covariance matrix, results in p times p plus 1 divided by 2 parameters to be estimated. In total this gives us, k plus p divided by 2, the quantity times p plus 1, parameters to be estimated. This is decidedly more than we had previously.

## Quadratic Discriminant Analysis (QDA)

Results from removing one of the assumptions:

1. density $f_x(x)$ is multivariate normal with mean **vector** $\mu_k$
2. ~~covariance matrices are the same: $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_K = \Sigma$~~

Mathematically, it can be shown $k$ is chosen to:

$$\max_k \left\{ x^T \left( -\frac{1}{2} {\Sigma_k}^{-1} \right) x + x^T \left( {\Sigma_k}^{-1} \mu_k \right) + \left( -\frac{1}{2} \mu_k^T {\Sigma_k}^{-1} \mu_k - \frac{1}{2} log\,|\Sigma_k| + log(\pi_k) \right) \right\}$$

Quadratic Function (circled values are constants for each $k$)

As often is the case, it is unrealistic to assume that the covariance matrices are the same, although we can run a test check. What would be the consequence of dropping that assumption? The goal, which is to maximize a goal function, remains the same, but solving it is more arduous. Luckily, we can simply look at the form of the final goal function. We note it is more involved now within x transpose times a matrix times x vector involved in the first term. And this results in a linear combination of x squared values.

As well as the next term involving a linear combination of the x values. So this is now a quadratic function of the predictor values.

# Estimates for QDA

Estimators for parameters $\pi_k$ and $\mu_k$ are the same.

Allow separate covariance matrix $\Sigma_k$ for each class $k$ = 1, 2, ..., $K$

- Use $\hat{\Sigma}_k = \frac{1}{n_k - 1} \sum_{\{i:y_i=k\}} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$
- Estimating $p(p+1)/2$ parameters for each matrix!

Estimating total of $K + Kp + Kp(p+1)/2 = K(p+1)(p/2+1)$ parameters from data (even more)

Maximizing the goal function for quadratic discriminant analysis has the same estimators for parameters pi sub k and vectors mu sub k. It additionally allows separate covariance matrices sigma sub k for each class k, where the formula is computed using x vectors only from within class k. In each of those k matrices, we are estimating p times p plus 1 over 2 parameters, and so the total number of parameters that must be estimated are k plus k times p plus k times the number of parameters for each covariance matrix.

## Comparing LDA and QDA

**Practically, there is a trade-off:**

- QDA is more flexible
- However, QDA requires estimating more parameters (needing more data) and thus adds complexity

**Mathematically, also a tradeoff:**

- QDA can have lower bias
- but QDA has higher variance

If you have enough data to estimate all parameters for QDA, test for evidence of difference between covariance matrices.

When comparing LDA and QDA as possible models, computationally there is realistically no difference, but practically there is a trade off. As a model, QDA is more flexible in that it accommodates different covariance structures for the different classes. But this in turn requires estimating more parameters, which means that we would need more data and we add complexity to our model. The mathematical consequence of this is also a trade off- the well-known bias variance trade-off.

QDA can have a lower bias, but it does have a higher variance because of the added complexity. If you have enough data to estimate all parameters of a QDA, test to see if there is significant evidence of a difference between covariance matrices. If there is, you can proceed with the QDA. Otherwise, you can go ahead and use LDA.

# Computation

**Process:**

Use `lda` and `qda` functions in MASS package

**Inputs:** `formula` and `data` (and `prior`)

**Values:** `means` and `scaling`

**Testing assumptions:**

**Normality:** `hzTest` function from MVN package

**Constant covariance:** `boxM` function from biotools package

As you've already been introduced to the lda function, the use of the qda function in the MASS package should feel very comfortable. Inputs and values of the function work in a very similar manner. We will need to be able to test assumptions for the two tests that we are working with.

For the normality assumption, we will be using the Henze-Zirkler's multivariate normality test, which was previously introduced in a 705 class. This is applied using the mhz function from the mvnormalTest package. To test the constant covariance assumption, we'll be using the BoxM function from the MVTests package. We'll take a look at this in an example in today's video.

**Notes:**

R Manual Pages: lda function, qda function, hzTest function, boxM function.

# Iris example

Using **iris** data set, already loaded in R, with variables.

- Four size measurements (all in cm) of iris plant: `Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width`
- `Species`, one of "setosa", "versicolor", and "virginica"

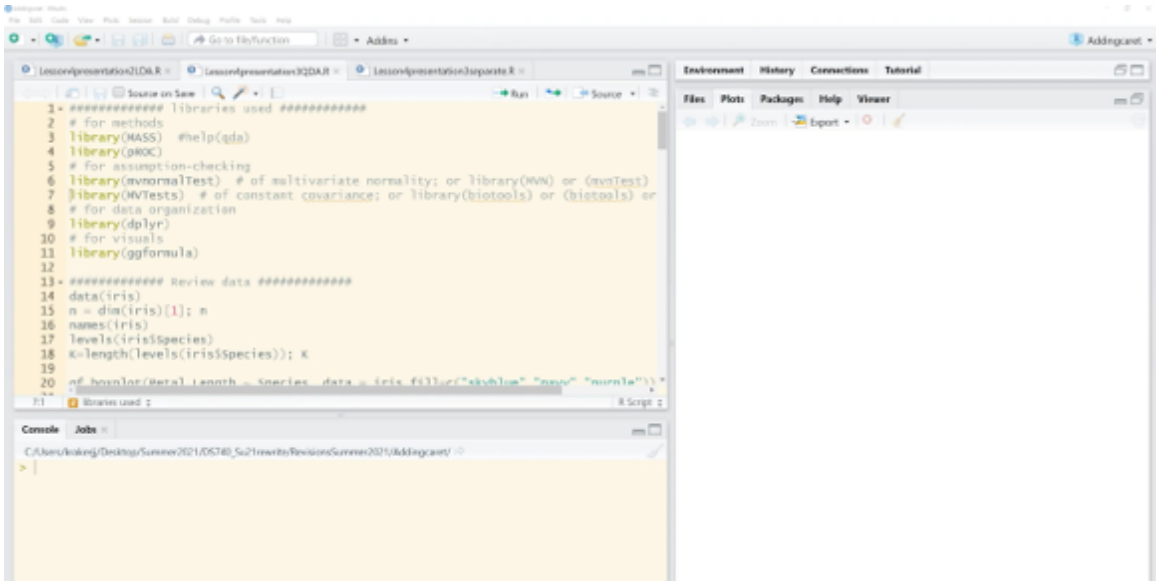
**Purposes:**

1. Fit both LDA and QDA.
2. Predict classification.
3. Compare models via cross-validation.

We'll be revisiting the iris data set. This time working with all four of the predictors to predict the species. And will be fitting both LDA and QDA models using those models to predict classifications. And finally compare the models via cross validation methods.

**Notes:**

Link: Iris data set




This slide represents a video/screencast in the lecture. The transcript does not substitute video content.

We next apply QDA, Quadratic Discriminant Analysis function, found in the MASS library. We're also going to be using ROC curves for visualization. And we'll need to open a couple libraries to conduct tests of our assumptions. Additionally, we'll be doing some data organization, as well as some visualization.

Our data is set up as previously. And we again observe that we've got three levels of our iris species, so of our response categorical variable. Taking a look at the response values, or the petal length split up by those response values, it is very clear that there are sizable differences in petal length split up by species.

Quadratic discriminant analysis is advisable, since we do not have constant variance within these groups. And so we're going to proceed in that direction. As we've done previously, we could specify our CV groups. In fact, this is the exact same specification of those cross-validation groups. And we could apply LDA on one predictor.

We actually did this precisely as will be done in this example in the previous recording. The difference here is we're going to generalize this so that we can apply both LDA and QDA, as well as include different potential models-- that is, using different subsets of predictors.

So I'm going to set my response and store that in y and then check. And there's various ways you can do this. I'm doing an if-else check. I'm actually

doing a double if check to make sure that my methodapplied is either LDA or QDA so that the modelfit is the appropriate model.

Now, as I specified up here, methodapplied is LDA. If methodapplied is LDA, the output of this will store in modelfit the LDA fit on the specified model, which is just species, split or fit on petal length for the iris data.

So running this will run this line, since I specified methodapplied to be LDA. Taking a look at modelfit, I should see precisely the same output as I saw in the previous presentation, as well as getting the same output and repredicted error, as well as the same cross-validated predictions.

The adjustment has to be made for the cross-validation predictions as well. So if we take a look back at the code from the previous presentation, inside our cross-validation, we had simply specified LDA of the particular model we wanted.

To make this a little bit more general purpose, general usage, we now check which method is applied and then apply the corresponding method to the model that we have chosen. So we are taking this generalization and putting it inside our cross-validation.

Importantly, we're only going to fit this to the train set within the cross-validation loop. Everything else will be the same as before. So it's just this change from one line to these four lines to specify the appropriate model. So I run this. I should, again, get the same output as we observed in the previous presentation.

Now I can store this CV error for model L1. Of course, it seems like a little bit of a level of complexity if that's all we did. But we're actually trying to make this into a-- easier to apply for both QDA and different model specifications. So instead of using LDA, I can now use, and in my coding, check for QDA, as well as apply a particular model.

So here, I'm going to apply QDA to the same one-predictor model that we talked about previously, run the same lines of code, with an emphasis on getting to the cross-validation error. And it looks very similar in terms of-- we have the same error rate.

But importantly, we do have different classifications here-- that is, different incorrect classifications here. And that's because we are fitting or cross-validating a different model. And this was our QDA with one predictor. So I'm going to store that CV error. I'm going to rename it as CVErrorQ1.

The next step would be, instead of just using one predictor, now I'm going to use all four predictors. So I specify, going back to linear discriminant analysis. And I'm going to fit species on all the remaining predictors. And if you weren't sure what those were, let's double check-- names of iris. We were going to be fitting species on the remaining variables. So that is the method and model applied.

So taking iris species, running these same steps as before, again, because we've made this generalizable to different models, we can actually just rerun the same set of code. And it will go through and make sure that we're using the correct model, either LDA or QDA-- I'm sorry, the correct method, either LDA or QDA, and the correct model on the appropriate data set.

Once again, store that, in this case as CDErrorL4, and then run through our final method and model, which is going to be QDA with the response fit on all four predictors. Run the same set of code. And make sure we store the result.

And we now have CV error for each of four models-- LDA with one predictor, QDA with one predictor, LDA with four, and QDA with four. It appears that, because we're looking at the cross-validated error, or, that is, the honest error, it appears that four predictors is generally doing a better job-- not dramatically better, but better job at classifying.

And we might say, well, LDA seems to do slightly better. But it's important to recognize that equal covariances are not reasonable. And so we're going to double check assumptions down here to verify that. So I'm going to define all my xvars by piping iris to select just the four x variables.

And I'm going to pipe iris through filtering-- that is, selecting only the rows that are of the species setosa-- and then selecting just the columns that we want-- the columns that are predictors. So we'll define those four matrices, check for multivariate normality, which, as we said on the previous slide, is reasonable or very close to reasonable. And that multivariate normality is needed, or is an assumption, for either LDA or QDA.

And then we're going to do a check for equal covariance matrices. It is extremely clear, when we take a look at this output with a tiny, tiny p-value, that there is very strong evidence of a difference in covariance matrices. So QDA is the better option, because we do not have equal covariance matrices. And QDA allows for unequal covariance matrices.

So that's the application of QDA and multiple predictors. If you'd like, we are now going to run a ROC with QDA predictions for a two-level response. And that two-level response is going to be an indicator of virginica species. We're

going to run both LDA and QDA on everything except species, of course-- that is, on all the numeric predictors except species.

We can keep track of the probabilities, the posterior probabilities, from each of those methods, and then fit a ROC curve for the LDA fit on all four predictors. And that looks similar, but a little bit higher, a little bit area underneath the curve, than on just using one predictor.

And QDA is fit in the same general way, where our response is the two-outcome response, virginica. And predictor is the posterior probability values. Now, when we plot this one, keep an eye on the plots. You'll see that there is an even bigger area under the curve. And so QDA with all four predictors is the method we would choose to go with in this example.

Notes:

See the online course for a downloadable R file containing the set of commands used in this demonstration.

## Additional Notes

- Violations of assumptions mean that the model may be poor at classifying the data; this can be observed in assessment.
- Bayes classifier simply uses the *k* for which the posterior is most likely; however, the threshold probability can be adjusted to gain in either sensitivity or specificity for the classification decision rule.

# Summary One

We have seen that including more than one predictor still results in maximization of a linear goal function, which in turn means that we still have a linear discriminant analysis as long as we retain the assumption of constant variance across classes.

Once we remove the constant variance assumption, the resulting goal function is quadratic in $x$-values, an thus we are conducting a quadratic discriminant analysis.

# Summary Two

The forms of the estimators of the parameters are similar, even when we have multiple predictors, but the **number** of parameters to be estimated increases dramatically.

When choosing between LDA and QDA models, first verify that you have enough data to estimate all parameters for QDA; then a test for evidence of difference between covariance matrices can be run. If the test is significant, use QDA; otherwise it is reasonable to use LDA.