Statistical Methods in Natural Language Processing (NLP)

Class 11: Machine Learning: LDA, QDA, FDA, C5.0, Model Comparison

Charalambos (Haris) Themistocleous

Department of Philosophy, Linguistics and Theory of Science, Centre for Linguistic Theory and Studies in Probability

Machine Learning

- Machine Learning Approaches
- Linear Discriminant Analysis
- Functional Discriminant Analysis
- C5.0

Applications

- speech to text
- text to speech
- Spoken dialect identification
- spoken document retrieval
- spoken language translation
- dialogue systems.

Types of Machine Learning

- Supervised: Learning from Labelled Data.
- Unsupervised: Unsupervised Learning.
- Reinforcement: Learning by doing with delayed reward.

Supervised

- Classification
- Regression

Unsupervised

- Clustering
- Compression

Process

1. Collect data
2. Exploring and preparing the data (dummy coding missing values, missing values)
3. training a model on the data
4. evaluating model performance
5. improving model performance

Linear Discriminant Analysis

- Classification
- Dimensionality Reduction

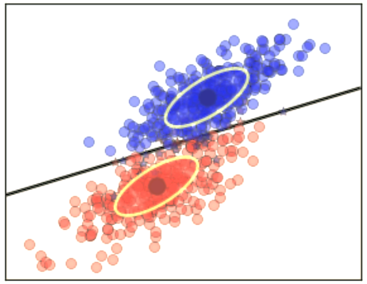


Figure: Linear Discriminant Analysis

Linear Discriminant Analysis

- LDA can be derived from simple probabilistic models which model the class conditional distribution of the data P(X—y=k) for each class k.

Predictions can then be obtained by using Bayes rule:

$$P(y=k|X) = \frac{P(X|y=k)P(y=k)}{P(X)} = \frac{P(X|y=k)P(y=k)}{\sum_l P(X|y=l)\cdot P(y=l)}$$

and we select the class k which maximizes this conditional probability.

LDA

P(X—y) is modelled as a multivariate Gaussian distribution with density:

$$p(X|y=k) = \frac{1}{(2\pi)^n|\Sigma_k|^{1/2}}\exp\left(-\frac{1}{2}(X-\mu_k)^t\Sigma_k^{-1}(X-\mu_k)\right)$$

We estimate from the training data the

- **class priors** $P(y=k)$ from the proportion of instances of class k.
- **the class means** μ_k from the empirical sample class means and
- **the covariance matrices** from the empirical sample class covariance matrices

LDA

The Gaussians for each class are assumed to share the same covariance matrix: $\Sigma_k = \Sigma$ for all k. This leads to linear decision surfaces between, as can be seen by comparing the log-probability ratios $\log[P(y=k|X)/P(y=l|X)]$:

$$\log\left(\frac{P(y=k|X)}{P(y=l|X)}\right) = 0 \Leftrightarrow (\mu_k - \mu_l)\Sigma^{-1}X = \frac{1}{2}(\mu_k^t\Sigma^{-1}\mu_k - \mu_l^t\Sigma^{-1}\mu_l)$$

LDA: A simple example

```
import numpy as np
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
y = np.array([1, 1, 1, 2, 2, 2])
clf = LinearDiscriminantAnalysis()
clf.fit(X, y)

print(clf.predict([[-0.8, -1]]))
```

Example from **scikit-learn.org**

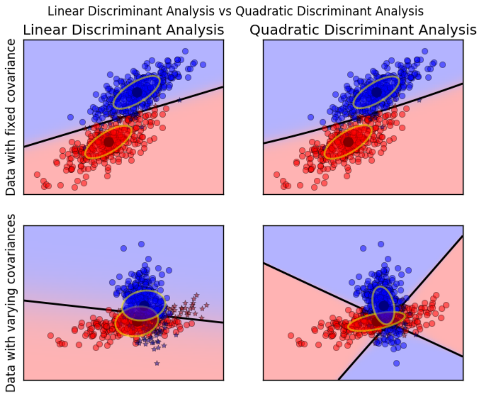
What if assumptions are not met?

- Quadratic Discriminant Analysis (QDA)
- Functional Discriminant Analysis (FDA)

Example

Vowels Database

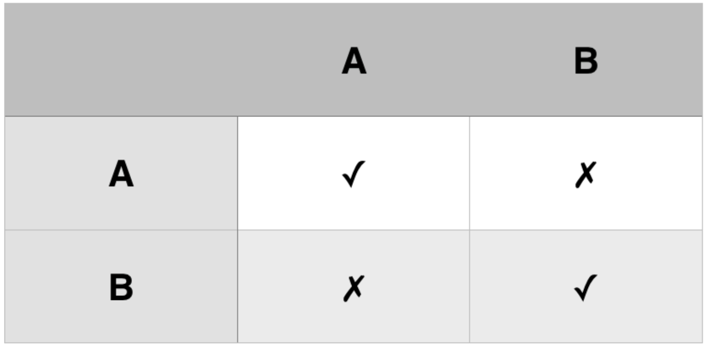
Quadratic Discriminant Analysis (QDA)



Flexible discriminant analysis (FDA)

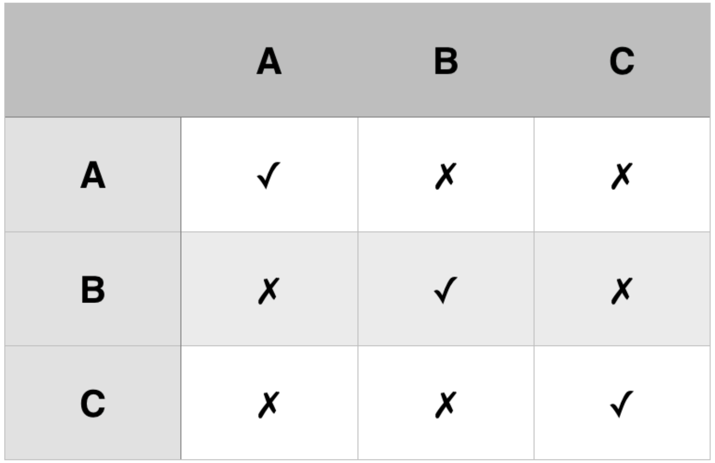
The flexible discriminant analysis (FDA) employs non-parametric techniques for the classification of the categorical variables (Trevor et al., 1994). Therefore, it does not requireunlike the LDAthat the data should be normally distributed. Because not all the predictors of this study are normally distributed, 95 the FDA is expected to offer a better classification accuracy than the LDA (from Themistocleous, submitted).

Confusion Matrix: Two Classes



	A	B
A	✓	✗
B	✗	✓

Confusion Matrix: Three Classes



	A	B	C
A	✓	✗	✗
B	✗	✓	✗
C	✗	✗	✓

Confusion Matrix

- True Positive: Correctly classified as the class of interest (Yes – Yes)
- True Negative: Correctly classified as not the class of interest (No–No)
- False Positive: Incorrectly classified as the class of interest (Yes – No)
- False Negative: Incorrectly classified as not the class of interest (No – Yes)

Accuracy

```
               | predicted default
actual default |        CG |       SMG | Row Total |
---------------|-----------|-----------|-----------|
            CG |       328 |       134 |       462 |
               |     0.374 |     0.153 |           |
---------------|-----------|-----------|-----------|
           SMG |       201 |       214 |       415 |
               |     0.229 |     0.244 |           |
---------------|-----------|-----------|-----------|
  Column Total |       529 |       348 |       877 |
---------------|-----------|-----------|-----------|
```

Beyond Accuracy

```
Confusion Matrix and Statistics

           Reference
Prediction  CG SMG
       CG  328 201
       SMG 134 214

                Accuracy : 0.618
                  95% CI : (0.5849, 0.6503)
     No Information Rate : 0.5268
     P-Value [Acc > NIR] : 3.178e-08

                   Kappa : 0.2275
  Mcnemar's Test P-Value : 0.000311

             Sensitivity : 0.7100
             Specificity : 0.5157
          Pos Pred Value : 0.6200
          Neg Pred Value : 0.6149
              Prevalence : 0.5268
          Detection Rate : 0.3740
    Detection Prevalence : 0.6032
       Balanced Accuracy : 0.6128

        'Positive' Class : CG
```

Evaluating Model Performance

- Kappa statistic
- Sensitivity
- Specificity
- ROC

Kappa

The kappa statistic adjusts accuracy by accounting for the possibility of a correct prediction by chance alone. This is especially important for datasets with a severe class imbalance, because a classifier can obtain high accuracy simply by always guessing the most frequent class. The kappa statistic will only reward the classifier if it is correct more often than this simplistic strategy.

- Poor agreement = less than 0.20
- Fair agreement = 0.20 to 0.40
- Moderate agreement = 0.40 to 0.60
- Good agreement = 0.60 to 0.80
- Very good agreement = 0.80 to 1.00

Kappa Statistic

The equation for is:

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e},$$

Pr_o refers to the proportion of the actual agreement and Pr_e refers to the expected agreement between the classifier and the true values, under the assumption that they were chosen at random.

Sensitivity and specificity

The **sensitivity** (a.k.a. recall) of a model (also called the true positive rate) measures the proportion of positive examples that were correctly classified (see also recall).

$$sensitivity = \frac{TruePositive}{TruePositive + FalseNegative}$$

Sensitivity and specificity

The **specificity** of a model (also called the true negative rate) measures the proportion of negative examples that were correctly classified.

$$sensitivity = \frac{TrueNegative}{TrueNegative + FalsePositive}$$

Precision and recall

The **precision** (also known as the positive predictive value) is the proportion of positive examples that are truly positive; in other words, when a model predicts the positive class, how often is it correct? A precise model will only predict the positive class in cases that are very likely to be positive. It will be very trustworthy.

$$precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

Precision and recall

Recall (a.k.a., Sensitivity) is a measure of how complete the results are. As shown in the following formula, this is defined as the number of true positives over the total number of positives. You may have already recognized this as the same as sensitivity. However, in this case, the interpretation differs slightly. A model with a high recall captures a large portion of the positive examples, meaning that it has wide breadth.

$$recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

F-measure

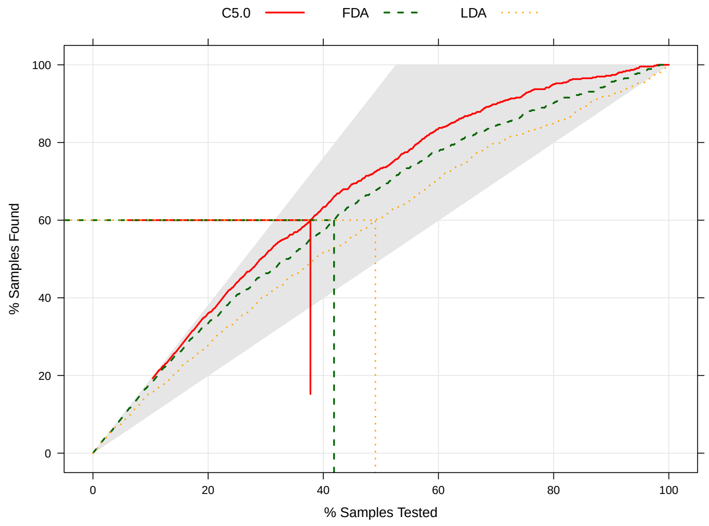
A measure of model performance that combines precision and recall using harmonic mean into a single number is known as the F-measure (also sometimes called the F1 score or F-score). The harmonic mean is used rather than the common arithmetic mean since both precision and recall are expressed as proportions between zero and one, which can be interpreted as rates.

$$Fmeasure = \frac{2 \times precision \times recall}{recall + precision} = \frac{2 \times TruePositive}{2 \times TruePositive + FalsePositive + FalseNegative}$$

Receiver Operating Characteristic (ROC) curve

The **Receiver Operating Characteristic (ROC)** curve is commonly used to examine the trade-off between the detection of true positives, while avoiding the false positives.

ROC



C5.0

1. The C5.0 is a classification algorithm developed by Ross Quinlan (Quinlan, 1993).
2. It assesses class factors, such as the dialect, based on a defined set of predictors.
3. It evaluates recursively the data and employs the predictors that can provide the best splitting of the data into more refined categories.
4. The splitting criterion is the difference in information entropy (a.k.a., the normalized information gain).

C5.0

Task Classify whether a fricative consonant is stressed or unstressed.

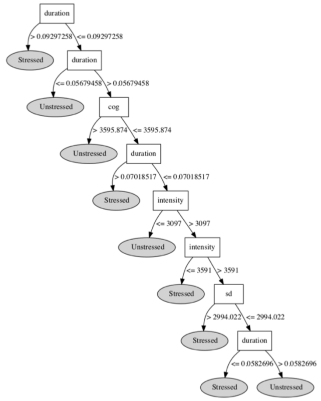


Figure 7: *Decision Tree produced by the machine learning and classification algorithm C5.0.*

C5.0

1. The predictor that provides the highest normalized information gain is the one selected for the decision (see also Woehrling et al., 2009, who provide classification a regional French varieties, using a different decision tree method).
2. Typically, each split is also an interpretation of the variation or impurity in the data.
3. The algorithm will stop when a criterion is met, such as when there are not enough data left to split.
4. Finally, C5.0 provides both tree and rule models (for an application of C4.5, which is an earlier iteration of C5.0, on accent classification, see Vieru et al. (2011) and for the classification of stressed and unstressed fricatives using C5.0, see Themistocleous et al. (2016)).

Next Class

- Evaluating the Naive Bayes algorithm with Mehdi.
- Find the Accuracy