MAA704, Classification and evaluation

Christopher Engström

December 10, 2014

Todays lecture

- Classification vs clustering
- Soft Independent Modelling of Class Analogy (SIMCA)
- Linear discriminant analysis (LDA)
- Evaluation

Classification

Classification and clustering are both used to group data (observations) into multiple groups where those in the same group are in some way "similar" and those in different groups are not.
However the goal of classification and clustering is a bit different.

- In clustering we first cluster (divide into groups) the data and then try to interpret the result.
- Clustering aims to answer questions such as: Out of these patients with the same disease, are there different types of patients which need different treatment?
- In classification we already have the interpretation (some classes) and we try to find the class belonging of every observation.
- Classificationaims to answer questions such as: Using some test, can we find out if a patient have some disease or not?

Classification problems

- Typically when solving a classification problem you need some data to trin your method on (observations where you know which class they belong to).
- After training the method on this data we hope to be able to use it on new data with unknown class.
- Usually finding the class itself is very time consuming and/or costly or simply not possible at the time we collect data, while the data we got for the observations is much cheaper.
- Another point to keep in mind is that the data classification used to train on is not always complete and of good quality, if it is we usually call it a gold standard.
- If the training data is incomplete or the classification on the trainingdata itself is not fully reliable it is usually called a silver standard.

Classification problems

We will take a look at two different classification methods

- SIMCA- which is a method based on PCA.
- LDA- which tries to find a line which when projected upon well seperates two classes.

Repetition Principal component analysis

Overview of the Covariance method.

- 1) Remove the mean from every individual set of measurements, order the measurements as the rows of matrix X.
- 2) Calculate the Covariance matrix $M = X^\top X$, (or use the correlation matrix).
- 3) Calculate the eigenvalues and eigenvectors of M.
- 4) Choose amount of eigenvectors to include, for example enough to at least keep 0.9 of the "variance".
- 5) Calculate $T_L = XW_L$ which is our new dataset.

SIMCA

Soft Independent Modelling of Class Analogy (SIMCA) is a classification method based on PCA, used a lot in for example chemometrics.

- SIMCA is a soft classification method i.e. it allows a single object to belong to more than one class.
- Some objects can end up with no class, either indicating a potential new class or more commonly an outlier.
- SIMCA does not imply that clusters have any specific shape or distribution of the clusters for each class.
- However where PCA fails to describe a class well (such as for a S-shaped cluster) it similarly SIMCA will have problems.

SIMCA

Basic outline:

1. Do any preprocessing needed on your training data (can be different for observations belonging to different classes).
2. Calculate principal components for each class and pick a suitable number of components to keep for each class (can be different for each class).
3. Choose a confidence level (usually 95%) and calculate statistical measures needed to classify new observations.
4. Assign new observations to a classes using a F-test by looking at the "distance" from each class.

Step 1: preprocessing

In this step data any preprocessing of data is done, such as for example:

- Missing values: For example by replacing them by the mean (or using any other imputation method).
- Outliers: Possibly identify and remove outliers if deemed appropriate.
- Do NOT standardize the variables (removing mean and/or standardize such that the variance is 1).
- Other preprocessing of the data as needed.

Step 2: PCA step

For each set of n observations $x_1^l, \ldots x_n^l$ for class l we:

- Calculate the covariance matrix and find the principal components.
- Choose a suitable number of principal components to represent the class. ($T_k = XW_k$).

Step 3: Calculate what we need to classify new observations

For each class l we have p-original variables, k-principal components, n observations and the loadings matrix W_k (from the previous step).

- For each observation x_i^l (in class) calculate the projection of the observation onto the mean:

$$\hat{x}_i^l = \bar{x}^l + W_k^l (W_k^l)^\top (x_i^l - \bar{x}^l)$$

  where $\bar{x}^l$ is the mean over all observations of the class (in the original p variables.
- Calculate the distance D_i^l for each observation x_i^l (in class) using the euclidean norm:

$$D_i^l = ||x_i^l - \hat{x}_i^l||$$

- Calculate within class variance $(s_l)^2$:

$$(s_l)^2 = \frac{\sum_{i=1}^{n_l} (D_i^l)^2}{(p - k_l)(n_l - k_l - 1)}$$

Step 4: Classify new observations

For each new observation y_i we look at each class l and decide if if should belong to that class:

- Calculate the distance D_i^l to the class mean:

$$\hat{y}_i^l = \bar{x}^l + W_k^l (W_k^l)^\top (y_i - \bar{x}^l)$$

$$D_i^l = ||y_i - \hat{y}_i||$$

- We then calculate the variance $(s^l)^2$:

$$(s_i^l)^2 = \frac{(D_i^l)^2}{p - k_l}$$

- To decide if the observation should belong to class l we perform an F-test by calculating $(s_i^l / s_l)^2$ and compare this value with the quantile of the F-distribution with $(p - k_l, (p - k_l)(n_l - k_l - 1))$ degrees of freedom and choosen confidence level (found in a table). If the calculated value is smaller than this then we assign the observation to this class.

Comments

Some things to note:

- The confidence level can be seen as the probability that an observation belonging to the class is classified as that class, hence a higher confidence level means that it is easier for an observation to be classified as that class.
- For a more indepth understanding of the statistics involved we referer to other courses in statistics since it would be to much to go through here.

Example: Fisher's Iris data

Fisher's Iris data:

- Standard dataset containing sepal and petal width and length for three types of Iris flowers, Setosa, Versicolor and Virginica.
- Aim is to classify the flowers based on these measurements.
- One species (Setosa) is relatively easy to classify, while the other two are much harder to separate.
- The data contains 150 observations, 50 from each species.

Example: Fisher's Iris data

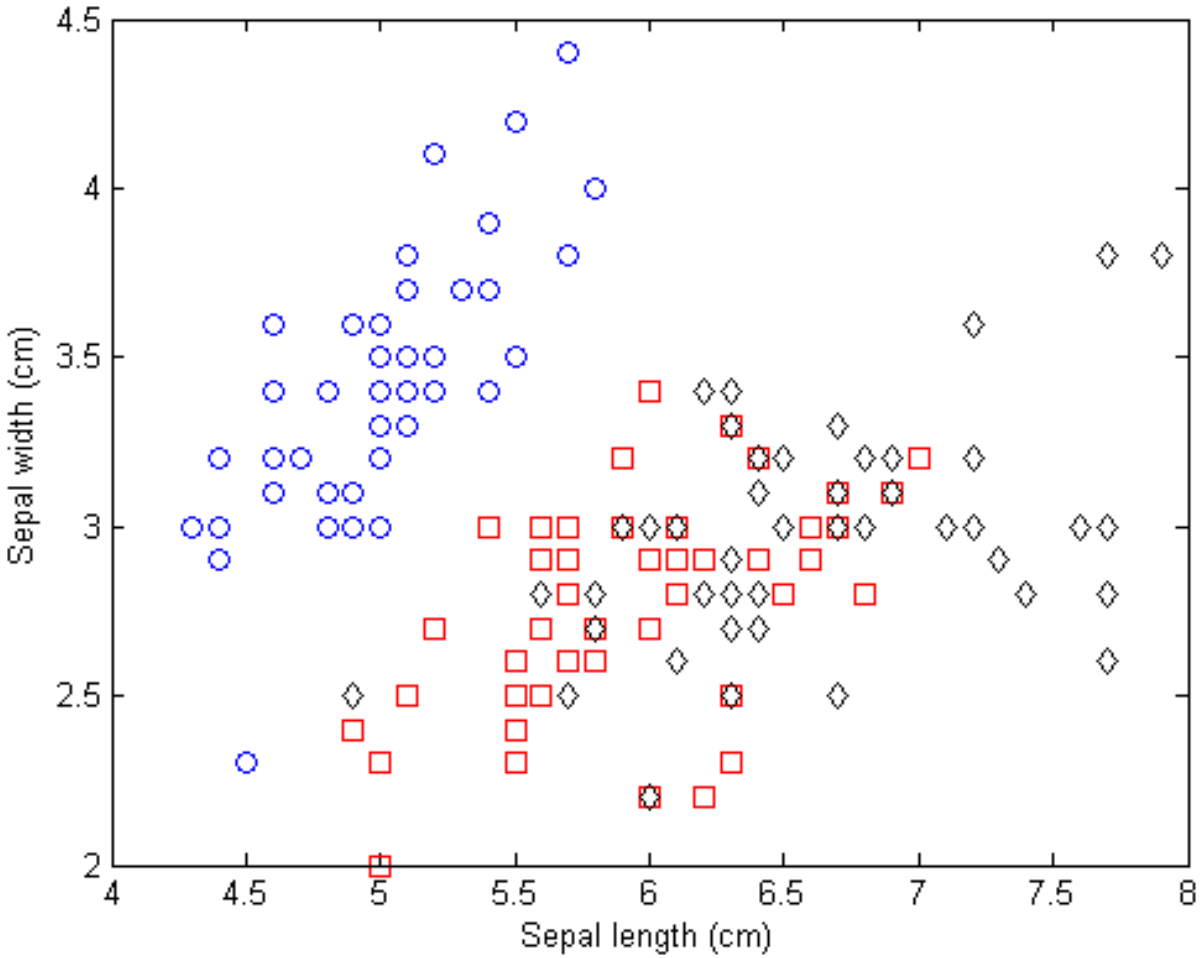

Figure: Plot of the first two features: sepal Length and sepal width for the three flowers.

Example: Fisher's Iris data

In order to have something to test on we will only train the method on the first 40 of each type (leaving 30 for testing).

- We will show the procedure for one of the classes (setosa) in detail (the others can be done in the same way).
- We start by extracting the 40 observations belonging to the class and calculate the covariance matrix In order to calculate principal components.

$$C_{\text{setosa}} = \begin{bmatrix} 0.131 & 0.097 & 0.013 & 0.013 \\ 0.097 & 0.130 & 0.002 & 0.012 \\ 0.013 & 0.002 & 0.030 & 0.005 \\ 0.013 & 0.012 & 0.005 & 0.010 \end{bmatrix}$$

Example: Fisher's Iris data

- Next we calculate the eigenvalues of the covariance matrix which the eigenvectors as the columns of (V) and the eigenvalues as the elements of e.

$$\mathrm{V} = \begin{bmatrix} -0.031 & 0.477 & 0.521 & 0.707 \\ -0.067 & -0.401 & -0.587 & 0.700 \\ -0.195 & -0.765 & 0.612 & 0.057 \\ 0.978 & -0.165 & 0.098 & 0.082 \end{bmatrix}, \quad e = \begin{bmatrix} 0.0073 \\ 0.0235 \\ 0.0397 \\ 0.2300 \end{bmatrix}$$

- Sorting the eigenvalues and calculating the proportion ot total variance we get: $(0.765,\ 0.898,\ 0.976,\ 1)$.
- Of these we choose to keep the first 2 (explaining nearly 90% of the variance.

Example: Fisher's Iris data

- Picking the first two principal components gives loading vector W_2^{set}:

$$W_2^{set} = \begin{bmatrix} 0.707 & 0.521 \\ 0.700 & -0.587 \\ 0.057 & 0.612 \\ 0.082 & 0.098 \end{bmatrix}$$

- Next we want to calculate the within class variance s^2_{set}.

Example: Fisher's Iris data

- For each observation x_i^{set} (in class) calculate the projection of the observation onto the mean:

$$\hat{x}_i^{set} = \bar{x}^{set} + W_2^{set}(W_2^{set})^\top(x_i^{set} - \bar{x}^{set})$$

where $\bar{x}^{set}$ is the mean over all observations of the class (in the original p variables.

- For the first observation we get:

$$\hat{x}_1^{set} = \begin{bmatrix} 5.04 \\ 3.45 \\ 1.46 \\ 0.24 \end{bmatrix} + W_2^{set}(W_2^{set})^\top \left(\begin{bmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{bmatrix} - \begin{bmatrix} 5.04 \\ 3.45 \\ 1.46 \\ 0.24 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 5.07 & 3.52 & 1.44 & 0.24 \end{bmatrix}^\top$$

Example: Fisher's Iris data

- Next we calculate the distance D_i^{set} for each observation (in class)

$$D_i^{set} = ||x_i^{set} - \hat{x}_i^{set}||$$

- For the first observation we get

$$D_1^{set} = \left|\left| \begin{bmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{bmatrix} - \begin{bmatrix} 5.04 \\ 3.45 \\ 1.46 \\ 0.24 \end{bmatrix} \right|\right| = \left|\left| \begin{bmatrix} 0.06 \\ 0.05 \\ -0.06 \\ -0.04 \end{bmatrix} \right|\right|$$

$$= \sqrt{0.06^2 + 0.05^2 + 0.06^2 + 0.04^2} = 0.106$$

Example: Fisher's Iris data

- Calculating this for all observations (in class) we get within class variance as

$$(s_{set})^2 = \frac{\sum_{i=1}^{n_{set}} (D_i^{set})^2}{(p - k_{set})(n_{set} - k_{set} - 1)} = 0.1422$$

where we have the number of observations of class $n_{set} = 40$, number of features $p = 4$ and number of principal components $k_{set} = 2$.

Example: Fisher's Iris data

We are now ready to try to classify new observations.

- We pick two observations $y_1 = (5, 3.5, 1.3, 0.3)$ and $y_2 = (5.9, 3, 5.1, 1.8)$ of which the first is of Setosa and the second is of Virginica from the test set.
- First we calculate the distance to the class mean:

$$\hat{y}_i^{set} = \bar{x}^{set} + W_2^{set}(W_2^{set})^\top(y_i - \bar{x}^{set})$$

$$D_i^{set} = ||y_i - \hat{y}_i||$$

- Calculating this for the two observations gives

$$D_1^{set} = 0.119, \;\; D_2^{set} = 2.583$$

- We then calculate the variance $(s^{set})^2$:

$$(s_i^{set})^2 = \frac{(D_i^{set})^2}{p - k_l}$$

- For the two observations we get

$$(s_1^{set})^2 = 0.0071, \;\; (s_2^{set})^2 = 3.3365$$

Example: Fisher's Iris data

- Last we perform the F-test for both by calculating $F_i = (s_i^{set}/s_{set})^2 = (s_1^{set})^2/s_{set}^2$

$$t_1 = 0.0071/0.1422 = 0.0499$$

$$t_2 = 3.3365/0.1422 = 23.4710$$

- We choose 95% as our confidence level which means that we need

$$F(p - k_{set}, (p - k_{set})(n_{set} - k_{set} - 1))_{(0.95)}$$

$$= F(2, 74)_{(0.95)} = 3.125$$

- The test value for the first observation is smaller than $F(2, 74)_{(0.95)}$ hence we classify it as "Setosa" while the second is larger and therefor is not classified as that.

Example: Fisher's Iris data

If we do this for the whole testset we would see that in fact this classifies the dataset into Setosa / not Setosa correctly for all observations of the test set.

- However the same is not true for Versicolor and Virginia, in both cases they get those that are Setosa right (never classified as either),
- However all Versicolor and Virginia sample are also classified as the other.
- To handle this we need to lower the confidence level, for example if we lower it to 20% instead we no longer make any wrong classifications. But 2 of the observations are not classified as anything at all.

Evaluation

To visualise the result a confusionmatrix can be very useful.

This is a matrix where every element C_{ij} is the amount of observations of class i classified as class j (using our classifier).

All diagonal elements represent correctly classified elements, while the non diagonal elements represent misclassifications.

Evaluation

Since our classifier allows for an observation to be part of more than one class we might also neet to include the pairs $(set, ver), (set, vir), (ver, vir)$ the tripple (set, ver, vir) and those no classified in any class.

Using our "best" classifier with lower confidence level for the Versicolor and Virginia classes we get (where the 4th row/column represent no class):

$$\begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 9 & 0 & 1 \\ 0 & 0 & 9 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

.

Evaluation

If we classified the whole dataset (including the part we trained on) we would instead have (with states set,ver,vir,(ver+vir),none)

$$\begin{bmatrix} 50 & 0 & 0 & 0 & 0 \\ 0 & 42 & 2 & 2 & 4 \\ 0 & 1 & 43 & 3 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Of course when there are some that do not exist (no observation is 2 species or none) then those rows could be omitted, resulting in a 3×5 matrix in this case instead.

Linear discriminant analysis

Our object is to reduce the dimension while preserving as much of the information discriminating the classes from each other as possible.

- Also called Fisher's linear disciminant.
- Assuming we have N D-dimensional samples.
- N_1 of these samples belonging to class c_1 and N_2 belonging to class c_2.
- Our aim is to project the samples $\mathbf{x}$ on a line $y = \mathbf{w}^\top \mathbf{x}$. Resulting in a scalar value for every sample.
- We want to find the line which when projected upon, best seperates the two classes.

Linear discriminant analysis

To evaluate the class separability we will use something called the Fisher linear discriminant:

- Maximizing this measure will give us the "best" line.
- But first we need to look at some components we will need.

Linear discriminant analysis

We start by looking at the distance between the projected means of the classes.

- In the original space the mean $\boldsymbol{\mu}_i$ of a class is easily found:

$$\boldsymbol{\mu}_i = \frac{1}{N_i} \sum_{x \in c_i} \mathbf{x}$$

- And for the projected means we get:

$$\tilde{\mu}_i = \frac{1}{N_i} \sum_{y \in c_i} \mathbf{y} = \frac{1}{N_i} \sum_{x \in c_i} \mathbf{w}^\top \mathbf{x} = \mathbf{w}^\top \boldsymbol{\mu}_i$$

- We could now use the distance between the projected means:

$$|\tilde{\mu}_1 - \tilde{\mu}_2| = |\mathbf{w}^\top (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)|$$

Linear discriminant analysis

Although the distance between the projected means might seperate the classes well, it does not take into consideration the variance of the data.

- If there is a high variance in the same direction as the one we would get when maximizing the direction of the means, we could get bad seperability anyway.
- To solve this we will look at the variance within a class (also called the scatter) on the projected line:

$$\tilde{S}_i^2 = \sum_{y \in c_i} (y - \tilde{\mu}_i)^2$$

- Adding the scatter of both classes and we get the "within-class scatter":

$$(\tilde{S}_1^2 + \tilde{S}_2^2)$$

Linear discriminant analysis

Fishers linear discriminant is defined as the function $\mathbf{w}^\top \mathbf{x}$ which maximizes $J(\mathbf{w})$:

$$J(\mathbf{w}) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{S}_1^2 + \tilde{S}_2^2}$$

We are now looking for a projection where elements from the same class are projected close to eachother (low within-class scatter) and the distance between the projected class means are far apart.

Linear discriminant analysis

To maximize $J(\mathbf{w})$ we start by writing it using $\mathbf{w}$.

- We start by looking at the "within-class scatter":

$$\tilde{S}_i^2 = \sum_{y \in c_i} (y - \tilde{\mu}_i)^2 = \sum_{y \in c_i} (\mathbf{w}^\top \mathbf{x} - \mathbf{w}^\top \boldsymbol{\mu}_i)^2$$

-

$$= \sum_{y \in c_i} \mathbf{w}^\top (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^\top \mathbf{w}$$

- We call: $\mathrm{S}_i = \sum_{x \in c_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^\top$ and $\mathrm{S}_W = \mathrm{S}_1 + \mathrm{S}_2$

  and get:

$$\tilde{S}_1^2 + \tilde{S}_2^2 = \mathbf{w}^\top (\mathrm{S}_1 + \mathrm{S}_2)\mathbf{w} = \mathbf{w}^\top \mathrm{S}_W \mathbf{w}$$

Linear discriminant analysis

If we instead look at the distance between projected means we get:

$$|\tilde{\mu}_1 - \tilde{\mu}_2|^2 = (\mathbf{w}^\top \boldsymbol{\mu}_1 - \mathbf{w}^\top \boldsymbol{\mu}_2)^2$$

$$\mathbf{w}^\top (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \mathbf{w}$$

We call $\mathsf{S}_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top$ and we get:

$$|\tilde{\mu}_1 - \tilde{\mu}_2|^2 = \mathbf{w}^\top \mathsf{S}_B \mathbf{w}$$

Linear discriminant analysis

Combining the two gives the following function in $\mathbf{w}$ which we want to maximize:

$$J(\mathbf{w}) = \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}$$

For which we can find the maximum by setting the gradient to zero:

$$\frac{d}{d\mathbf{w}} J(\mathbf{w}) = 0$$

Linear discriminant analysis

With the help of some matrix calculus this is equivalent to the eigenvalue problem:

$$S_W^{-1} S_B \mathbf{w} = J\mathbf{w}$$

With solution:

$$\mathbf{w}^* = \arg\max \left[\frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}} \right] = S_W^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

Linear discriminant analysis

Overview

- 1) Calculate the mean of each class $\boldsymbol{\mu}_i = \frac{1}{N_i} \sum_{x \in c_i} \mathbf{x}$.
- 2) Calculate the within class variance $S_i = \sum_{x \in c_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^\top$
- 3) Add the two $S_W = S_1 + S_2$.
- 4) Calculate $\mathbf{w}^* = S_W^{-1}(\mu_1 - \mu_2)$ to get one point on the line, ((0, 0) is another).
- 5) Now we can classify a new observation $\mathbf{x}$ by projecting it on this line $y = \mathbf{w}^\top \mathbf{x}$.
- 6) Choose a threshold T, all observations which gives a $y > T$ we classify into class 1 and all else into class 2.
- If the two classes have approximately the same distribution on the projected line the mean of the projected means are often a good choice $T = (1/2)(\tilde{\mu}_1 + \tilde{\mu}_2)$

Linear discriminant analysis

A short note on generalizations to more than two classes.

- Fisher's linear discriminant itself can be generalized for more classes, will result in finding a subspace $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{C-1}$ where C is the number of classes instead.
- Another alternative is to classify every class itself with respect to all the other classes, giving C classifiers that can then be combined.
- It is also possible to use the classifiers of every pair of classes.

Linear discriminant analysis

While LDA works well for Gaussian distributions, some other shapes or types of distributions can give problems.

- Some shapes of the classes can lead to problems, for example two interwined "C" shapes.
- If both clusters have the same mean we cannot classify them.
- Or if the information lies not in the mean of the classes but in the variance it will also fail.
- There is another similar method with the same name (Linear disciminant analysis), which assumes a gaussian distribution and classifies using a probabilistic perspective.

Example: Retun to Fisher's Iris data

To illustrate the method we use LDA on the same data as before, however we look at only two of the flowers, namely Setosa and Versicolor.

- In order to be able to plot the line seperating the two classes we will only consider the first two features, sepal length and sepal width.
- Thus we now have 100 observations out of two classes with two features each.

Example: Fisher's Iris data

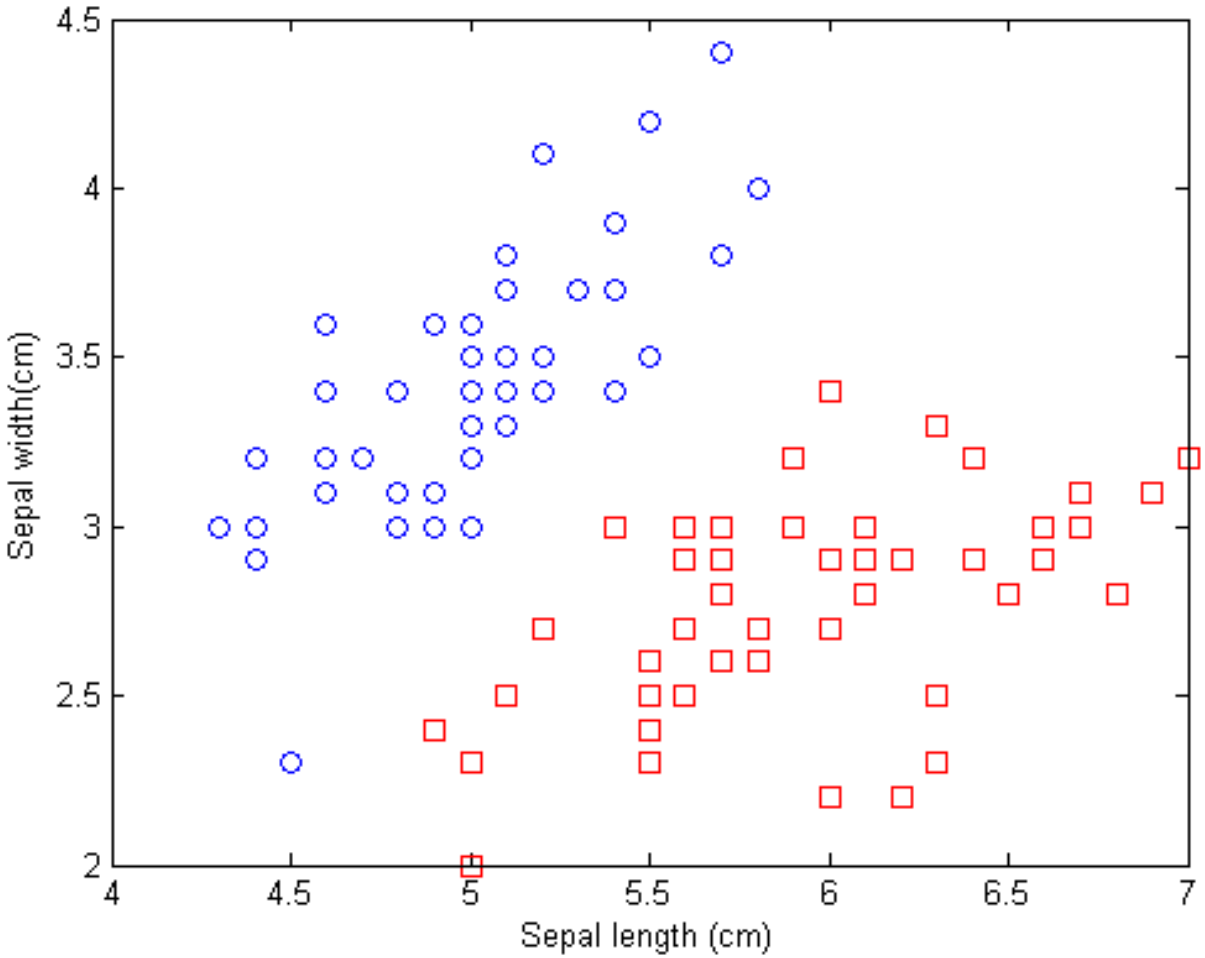

Figure: Plot of the first two features: sepal Length and sepal width for Setosa and Versicolor.

Example: Retun to Fisher's Iris data

- We start by calculating the mean and variance for each class.
-

$$\mu_{set} = \frac{1}{50} \sum_{x \in set} \mathbf{x} = \begin{bmatrix} 5.006 \\ 3.428 \end{bmatrix}$$

$$\mu_{ver} = \frac{1}{50} \sum_{x \in ver} \mathbf{x} = \begin{bmatrix} 5.936 \\ 2.770 \end{bmatrix}$$

-

$$S_{set} = \sum_{x \in set} (\mathbf{x} - \boldsymbol{\mu}_{set})(\mathbf{x} - \boldsymbol{\mu}_{set})^\top = \begin{bmatrix} 0.124 & 0.099 \\ 0.099 & 0.144 \end{bmatrix}$$

$$S_{ver} = \sum_{x \in c_{ver}} (\mathbf{x} - \boldsymbol{\mu}_{ver})(\mathbf{x} - \boldsymbol{\mu}_{ver})^\top = \begin{bmatrix} 0.266 & 0.085 \\ 0.085 & 0.099 \end{bmatrix}$$

Example: Fisher's Iris data

- We get

$$S_W = S_{set} + S_{ver} = \begin{bmatrix} 0.391 & 0.184 \\ 0.184 & 0.242 \end{bmatrix}$$

- With inverse

$$S_W^{-1} = \begin{bmatrix} 0.391 & 0.184 \\ 0.184 & 0.242 \end{bmatrix}^{-1} = \begin{bmatrix} 4.00 & -3.043 \\ -3.043 & 3.447 \end{bmatrix}$$

- This gives

$$\mathbf{w}^* = S_W^{-1}(\mu_{set} - \mu_{ver})$$

$$= \begin{bmatrix} 4.00 & -3.043 \\ -3.043 & 3.447 \end{bmatrix} \left(\begin{bmatrix} 5.006 \\ 3.428 \end{bmatrix} - \begin{bmatrix} 5.936 \\ 2.770 \end{bmatrix} \right) = \begin{bmatrix} -5.718 \\ 7.072 \end{bmatrix}$$

Example: Fisher's Iris data

- The slope of the line we project onto is then $k = w(1)/w(2) = -5.718/7.072 = -0.8086$ and the orthogonal line seperating the two classes $k_2 = -1/k$.
- To classify the observations we project them on the line $y = \mathbf{w}^*\mathbf{x}$ and then choose a threshold T.
- As initial value of T we choose the mean of the projected means on the line: $T = (1/2)(\mu_{set} + \mu_{ver}) = 9.37$

Example: Fisher's Iris data

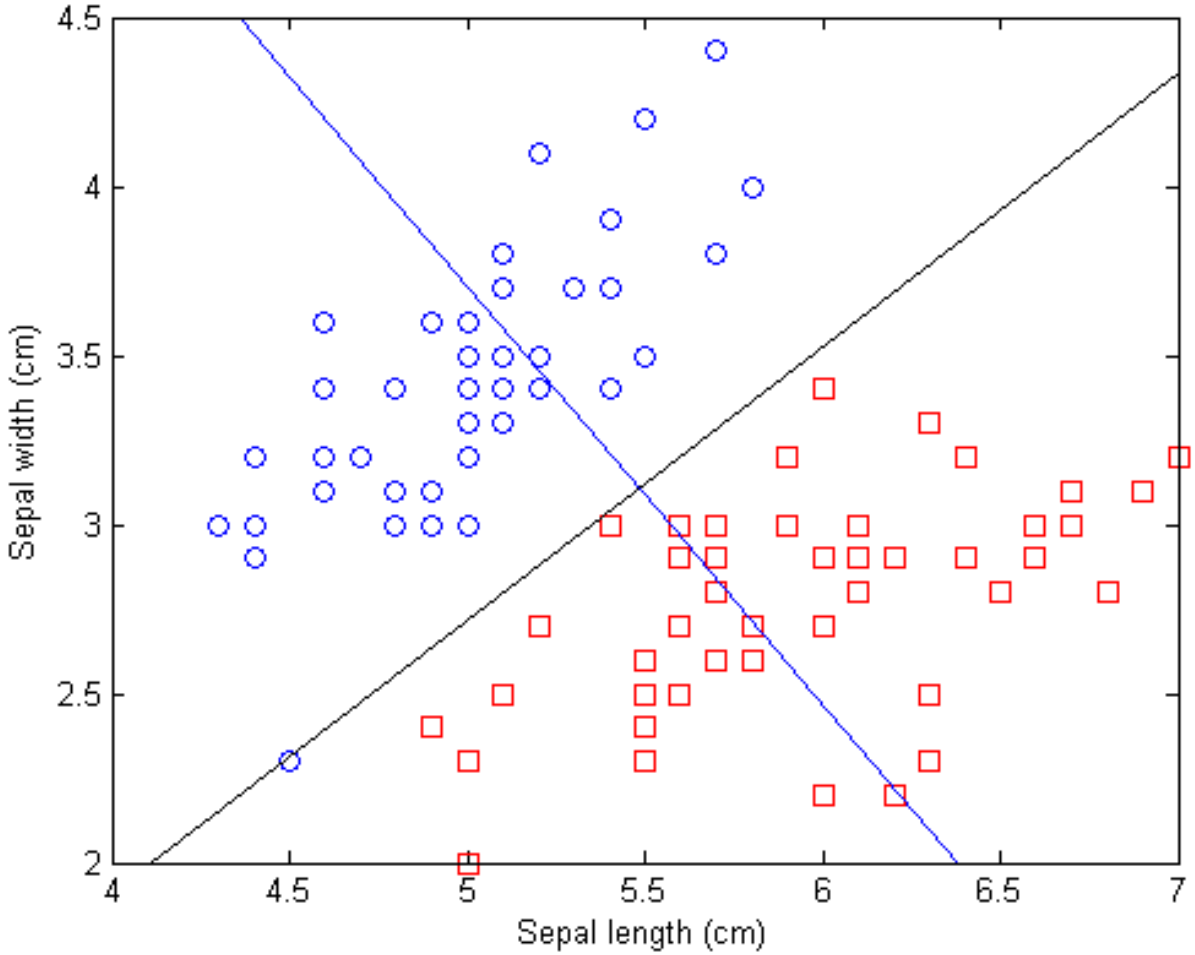

Figure: Plot of the first two features: sepal Length and sepal width for Setosa and Versicolor. and line data is projected upon. (A line parallel to the one calculated earlier is drawn instead)

If we classify all observations we can also set up the confusion matrix which in this case is:

$$\begin{bmatrix} 49 & 1 \\ 0 & 50 \end{bmatrix}$$

This could also be seen graphically in the previous slide where one of the blue circles (Setosa) was on the wrong side of the black line.

Evaluation

While the confusion matrix is useful, it does not account for overfitting, which is essentially when the method is trained "to much" on training data.

- If you have a lot of data you might be able to simply train on one part of the data and test on another, however that might not be an option if the data is small.
- Instead you can do something called n-fold crossvalidation.
- This is also commonly called leave-one-out crossvalidation.

Crossvalidation

Doing a n-fold crossvalolidation is simple, but can be quite time consuming depending on model used.

- 1) Divide your data into n parts randomly.
- 2) Train remove one of the parts and train your classifier on the rest.
- 3) Test your classifier on the removed part of the data (not used for trainiing).
- 4) Do this for each of the n parts and take an average of the result (usually percentage correctly classified observations).
- 5) It is also very useful to calculate the sample standard deviation of the same things we calculated in 4).

Recall, precision and F-measure

Some classifiers return a value rather than class. In order to classify an observation we then check how high/low this value is and classify it accordingly.

- One example of this is SIMCA if we leave the choice of confidence level as something we want to optimize as well.
- Very common when the "cost" of a misclassification is not equal, for example in medical test where you rather have some extra false positives in order to avoid false negatives.
- In this case the precision, recal and F-measure are very useful.

Recall, precision and F-measure

Assuming we are interested in how accurately we classify observations into class 1.

- Precision is a measure on how sure we can be that someone classified as "class 1" actually belong to class 1.

$$P = \frac{\text{number correctly classified as class1}}{\text{total number classified as class1}}$$

- Recal is a measure on how large proportion of those in class 1, are actually classified as class 1.

$$R = \frac{\text{number correctly classified as class1}}{\text{total number of observations of class 1}}$$

- F-measure is the harmonic mean of the two, giving equal weight to accurate prediction and recal.

$$F = 2\frac{P \cdot R}{P + R}$$

- Different weight can also be put on precision or recal depending on what is deemed most important.

Recall, precision and F-measure

Different weight can also be put on precision or recal depending on what is deemed most important.

This is denoted by F_β where β is a non-negative value.

$$F_\beta = (1 + \beta^2)\frac{P \cdot R}{\beta^2 P + R}$$

For example F_2 gives twice the wieght on Precision and $F_{0.5}$ gives halv the weight on precision.