Week 6

Lecture

- SVM
  - Hard margin (support vectors, margin, decision boundary)
  - Soft margin
  - Nonlinear with kernel trick

- Dimensionality reduction
  - PCA (steps, how to determine the number of principle components?)
  - SVD (the decomposition steps)

Tutorial

Task 1: Applying SVM to classify breast cancer data

Step 1: Load data, split into train/test set and normalize it.

Step 2: Create SVM classifiers with RBF kernel.

Step 3: Evaluation.

Task 2: Tuning SVM parameters

Step 1: load the moons dataset.

Step 2: Create SVM with RBF kernel

Step 3: Try different values for gamma and C.

Task 3: Dimensionality reduction using PCA

Step 1: Using PCA reduce the dimensionality of breast cancer dataset.

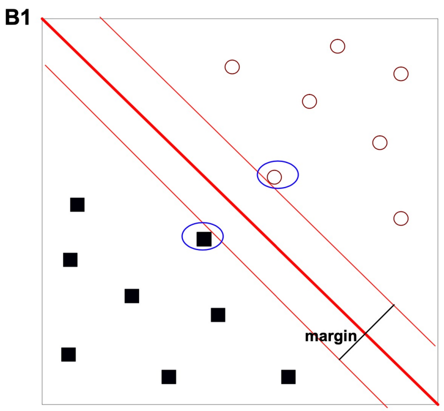
Step 2: Applying KNN for the reduced data.

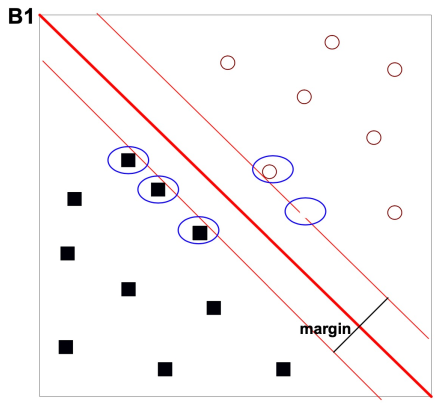
Step 3: choose the number of principal components.

TODO:

1. Create linear SVM and RBF SVM.
2. Grid search for RBF SVM.
3. Apply PCA to the MNIST data (preserve 95% var)
4. Decompress the reduced dataset back.

SVM – hard margin





The support vectors just touch the margin of the decision boundary

It is possible to have more than 1 support vector for each class

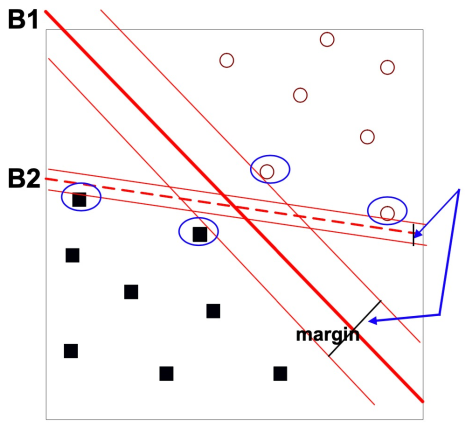
For our example: 5 support vectors, 3 for class square and 2 for class circle

Support vectors are the examples (data points) that lie closest to the decision boundary; they are circled

Margin – the separation between the boundary and the closest examples

The boundary is in the middle of the margin

Which hyperplane should we select - B1 or B2?
Which one is likely to classify more accurately new data**?**



- The hyperplane with the bigger margin, B1

SVM selects the maximum margin hyperplane

SVM – soft margin

We can modify our method to allow some misclassifications, i.e. by considering the trade-off between the margin width and the number of misclassifications

The optimisation problem formulation is similar but there is an additional parameter *C* in the definition of the optimization function

C is a hyper-parameter that allows for a trade-off between maximizing the margin and minimizing the training error

- Large *C*: more emphasis on minimizing the training error than maximizing the margin

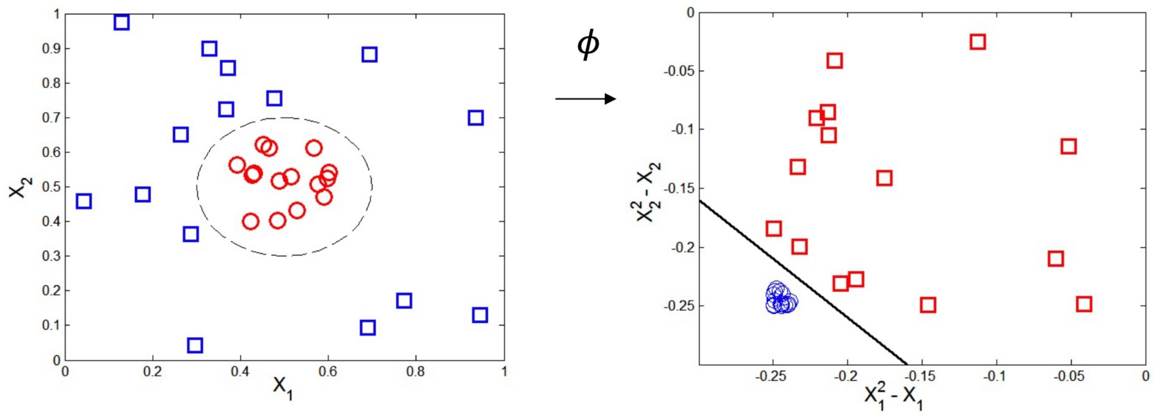
SVM – nonlinear

Transform the data from its original feature space to a new space where a linear boundary can be used to separate the data

If the transformation is non-linear and to a higher dimensional space, it is more likely than a linear decision boundary can be found in it

The learned linear decision boundary in the new feature space is mapped back to the original feature space, resulting in a non-linear decision boundary in the original space

- Non-linearly separable data in the original space
- Becomes linearly separable in the new space



transformation from old to new space: $\phi = (x_1, x_2) \rightarrow (x_1{}^2 - x_1, x_2{}^2 - x_2)$

SVM – kernel trick

$$\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$$

$$\cancel{1)\,\mathbf{x}_i \xrightarrow{\Phi} \Phi(\mathbf{x}_i),\mathbf{x}_j \xrightarrow{\Phi} \Phi(\mathbf{x}_j)}$$
$$\cancel{2)\,\Phi(\mathbf{x}_i).\Phi(\mathbf{x}_j)}$$

$$\Phi:(x_1,x_2)=(x_1^2,\sqrt{2}x_1x_2,x_2^2)$$

$$\mathbf{u}\xrightarrow{\Phi}\Phi(\mathbf{u}),\mathbf{v}\xrightarrow{\Phi}\Phi(\mathbf{v})$$

$$\Phi(\mathbf{u})\cdot\Phi(\mathbf{v})=(u_1^2,\sqrt{2}u_1u_2,u_2^2)\cdot(v_1^2,\sqrt{2}v_1v_2,v_2^2)=$$
$$=u_1^2v_1^2+2u_1u_2v_1v_2+u_2^2v_2^2=(u_1v_1)^2+(u_2v_2)^2+2u_1u_2v_1v_2=$$
$$=(u_1v_1+u_2v_2)^2=(\mathbf{u}\cdot\mathbf{v})^2$$

$$\longrightarrow \Phi(\mathbf{u})\cdot\Phi(\mathbf{v})=(\mathbf{u}\cdot\mathbf{v})^2$$

$$\downarrow$$

$$K(\mathbf{u},\mathbf{v})=\Phi(\mathbf{u})\cdot\Phi(\mathbf{v})=(\mathbf{u}\cdot\mathbf{v})^2$$

$$K(\mathbf{x},\mathbf{y})=(\mathbf{x}\cdot\mathbf{y}+1)^p - \textit{polynomial kernel}$$
$$K(\mathbf{x},\mathbf{y})=e^{-\frac{\|x-y\|^2}{2\sigma^2}} - RBF$$
$$K(\mathbf{x},\mathbf{y})=\tanh(k\mathbf{x}\cdot\mathbf{y}-\theta) - \textit{tanget hyperbolic}$$
$$(\textit{satisfies Mercer's Th. only for some } k \textit{ and } \theta)$$

Given: *N* examples with dimensionality *m* (i.e. *m* features)

Find: *m* new axes $Z_1, \ldots, Z_m$ orthogonal to each other such that
$Var(Z_1) > Var(Z_2) \ldots > Var(Z_m)$

Z1,…, Zm are called principal components

The principal components are vectors that define a new coordinate system

They are ordered based on how much variance they capture

- The first axis goes in the direction of the highest variance in the data
- The second axis is orthogonal to the first one and goes in the direction of the second highest variance
- The third one is orthogonal to both the first and second and goes in the direction of the third highest variance, and so on

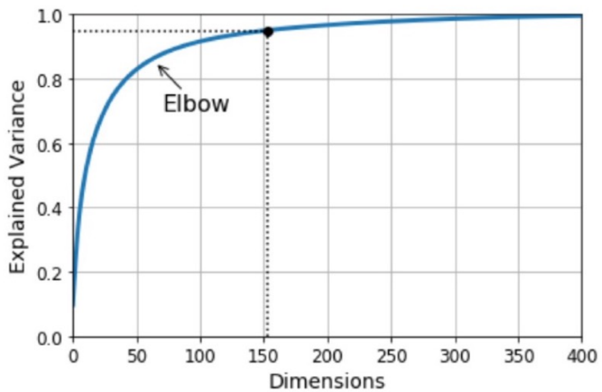
PCA

Method 1: Set min % of variance that should be preserved, e.g. 95%

- Choose *k* such that Z_1, Z_2,…Z_k capture 95% of the variance

Method 2: (Elbow method)

- Plot number of dimensions as a function of variance
- There is usually an elbow in the curve where the variance stops growing fast



- 95% variance is at 153 dimensions
- Elbow (subjective) - e.g. 100 dimensions