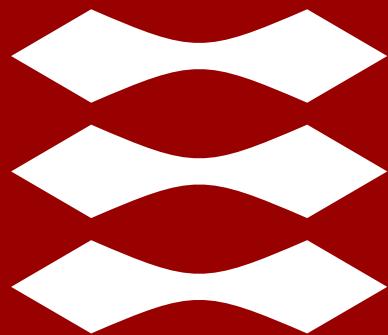


**DTU**



Lazaros Nalpantidis

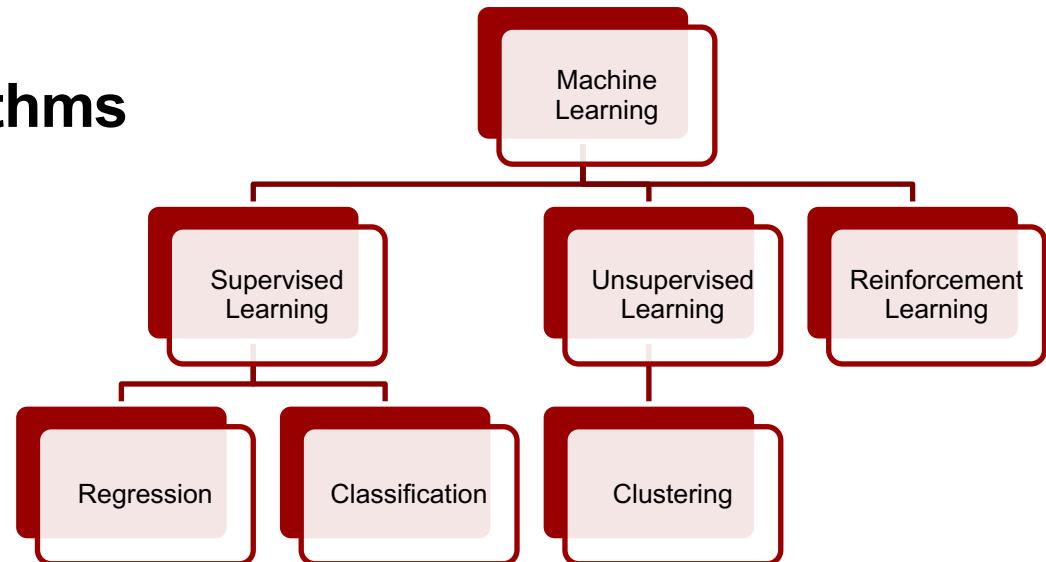
# Classification

- What is Classification?
- Dimensionality Reduction
  - PCA
  - Face detection
- Classifiers
  - k-Nearest Neighbors
  - Support Vector Machines
- Class/Category Recognition
- Summary

- What is Classification?
- Dimensionality Reduction
  - PCA
  - Face detection
- Classifiers
  - k-Nearest Neighbors
  - Support Vector Machines
- Class/Category Recognition
- Summary

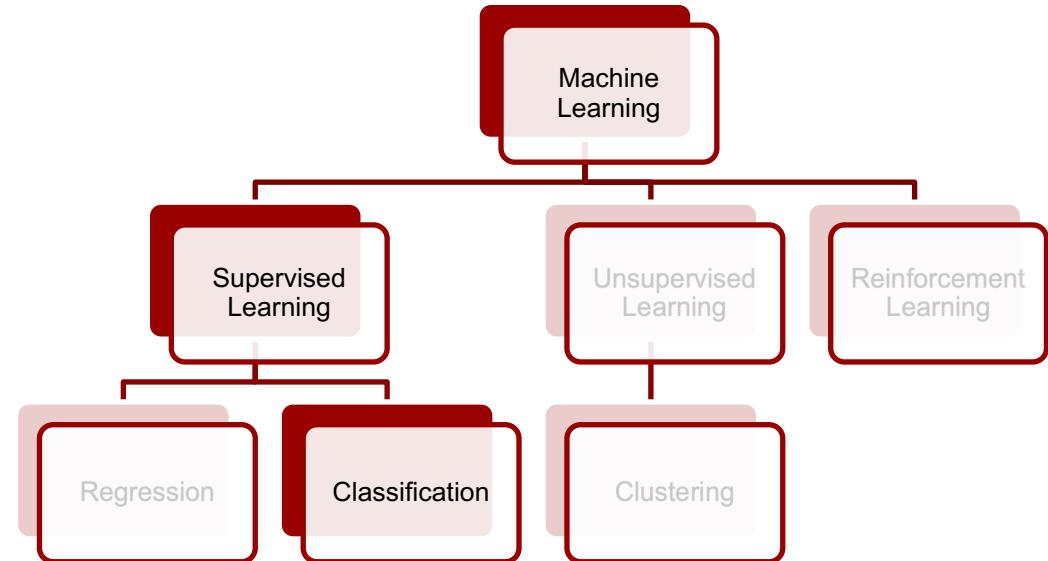
# What is Classification?

# A taxonomy of ML algorithms



- **Supervised Learning** – The target values are known
  - Regression – The target value is numeric
  - Classification – The target value is nominal
- **Unsupervised Learning** – The target values are unknown
  - Clustering – Group together similar instances
- **Reinforcement Learning** – Interacting with a dynamic environment the system must perform a certain goal.

# Classification



- **Supervised Learning** – The target values are known
  - Classification – The target value is nominal

# Classification

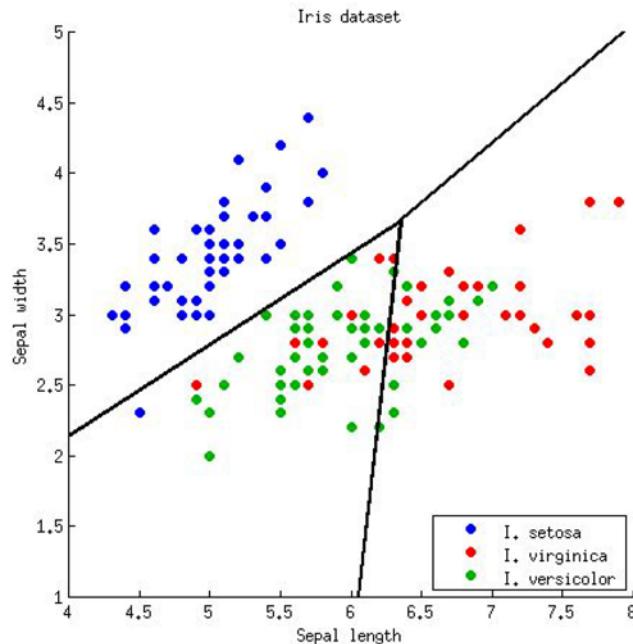
- Some terminology:
  - **Concept:** The description of the dataset. What can be learned by the examined dataset (e.g. the labels)
  - **Instances:** Individual and independent examples of the concept (e.g. an image)
  - **Attributes:** The features/dimensions that describe each instance.

	Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)	Type
1	5.1	3.5	1.4	0.2	Iris setosa
2	4.9	3.0	1.4	0.2	Iris setosa
3	4.7	3.2	1.3	0.2	Iris setosa
4	4.6	3.1	1.5	0.2	Iris setosa
5	5.0	3.6	1.4	0.2	Iris setosa
...					
51	7.0	3.2	4.7	1.4	Iris versicolor
52	6.4	3.2	4.5	1.5	Iris versicolor
53	6.9	3.1	4.9	1.5	Iris versicolor
54	5.5	2.3	4.0	1.3	Iris versicolor
55	6.5	2.8	4.6	1.5	Iris versicolor
...					
101	6.3	3.3	6.0	2.5	Iris virginica
102	5.8	2.7	5.1	1.9	Iris virginica
103	7.1	3.0	5.9	2.1	Iris virginica
104	6.3	2.9	5.6	1.8	Iris virginica
105	6.5	3.0	5.8	2.2	Iris virginica
...					



# What is Classification?

- The final output of a classification algorithm is usually decision boundaries that separate the classes.



- So, the examined concept is classified according to which boundary side it is found on.

# Image classification pipeline

- Our input is a training dataset that consists of  $N$  images, each labeled with one of  $K$  different classes.
- Then, we use this training set to train a classifier to learn what every one of the classes looks like.
- In the end, we evaluate the quality of the classifier by asking it to predict labels for a new set of images that it's never seen before. We'll then compare the true labels of these images to the ones predicted by the classifier.

- What is Classification?
- Dimensionality Reduction
  - PCA
  - Face detection
- Classifiers
  - k-Nearest Neighbors
  - Support Vector Machines
- Class/Category Recognition
- Summary

# Dimensionality Reduction

- When each concept is described by a large number of attributes, we say that the concept is described by a vector with large dimensionality.
- Generally, large dimensional vectors are well suited for machine learning.
- However,
  - some of the attributes may be redundant, i.e, attributes with small variance.
  - having many dimensions, might lead to the curse of dimensionality (data sparsity).
- Solution :
  - Reduce the dimensionality of the problem to boost speed and performance.

# Dimensionality Reduction

- Principal Component Analysis (PCA)

# Dimensionality Reduction

- Principal Component Analysis (PCA)
  - A solution for dealing with/reducing high dimensionality.
- Characteristics
  - Creates **new features** that are linear combinations of the original features
  - New features are orthogonal to each other
  - Keep the new features that account for a large amount of the variance in the original dataset
  - Re-base the dataset's coordinate system in a new space defined by its lines of greatest variance

# Dimensionality Reduction

- Principal Component Analysis (PCA)

1. Center the input data

2. Calculate Covariance Matrix

$$C = \frac{1}{N-1} X^T X$$

3. Compute eigenvectors & eigenvalues of the Covariance Matrix

- The first principal component is the eigenvector of the covariance matrix that has the largest eigenvalue

- » This vector points towards the direction of the largest variance of the data

- » The corresponding eigenvalue defines the magnitude of this vector

- The second largest eigenvector is orthogonal to the largest eigenvector, and points into the direction of the second largest spread of the data.

4. Sort the eigenvectors according to their eigenvalues

5. Calculate the variance score (significance).

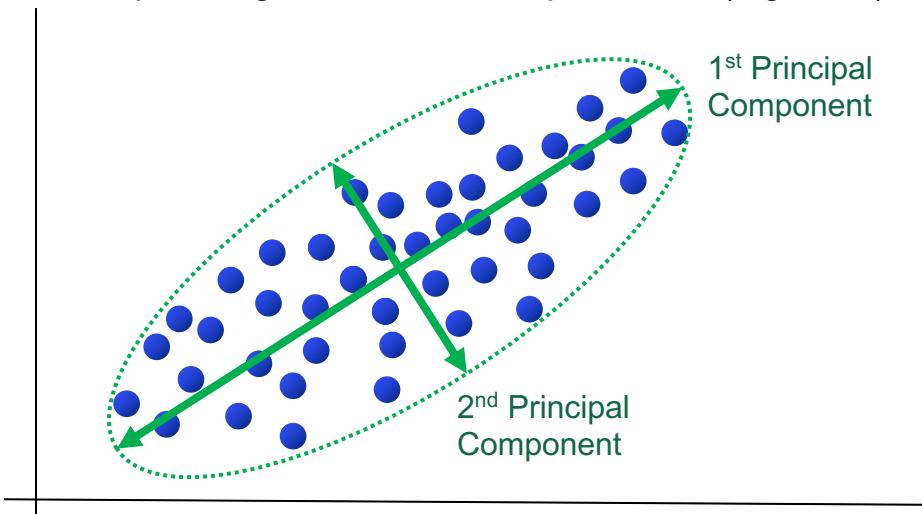
6. Keep eigenvectors that explain most (e.g. 95%) variance / remove the rest! ( $\leftarrow$  dimensionality reduction)

7. Project instances to eigenvectors y.

$$y = W^t x$$

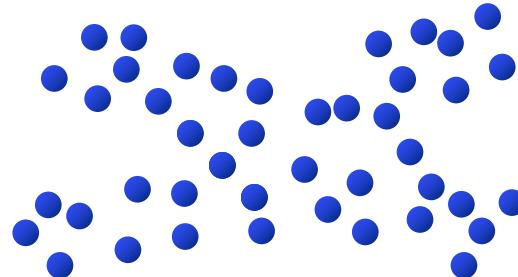
# Dimensionality Reduction

- Principal Component Analysis (PCA)
  - A solution for dealing with/reducing high dimensionality.
- **Eigenvector:** points to the direction of variance.
- **Eigenvalues:** shows how much of the total variance is explained by the corresponding eigenvector.
- Keep the eigenvectors that explain most (e.g. 95%) of the variance.



# Dimensionality Reduction

- Principal Component Analysis (PCA)
  - A solution for dealing with/reducing high dimensionality.
- **Eigenvector:** points to the direction of variance.
- **Eigenvalues:** shows how much of the total variance is explained by the corresponding eigenvector.
- Keep the eigenvectors that explain most (e.g. 95%) of the variance.



← What are the Principal Components here?

- What is Classification?
- Dimensionality Reduction
  - PCA
  - Face detection
- Classifiers
  - k-Nearest Neighbors
  - Support Vector Machines
- Class/Category Recognition
- Summary

# Eigenfaces

- Face Recognition
- Consider many images of faces (each image with the same dimensions).
- Consider each image, not as a 2D table, but just as a very long vector!
  - How many dimensions does such a vector have?
- We want to construct a low-dimensional linear subspace that best explains the variation in our set of face images.
- Most face images lie on a low-dimensional subspace determined by the first some(!) directions of maximum variance.
- Use PCA to find the vectors(eigenfaces) that determine this subspace
- Then, all face images can be expressed as linear combinations of the eigenfaces

*M. Turk and A. Pentland, 1991. Eigenfaces for Recognition, Journal of Cognitive Neuroscience, 3(1), 71-86.*

Large dataset of face images

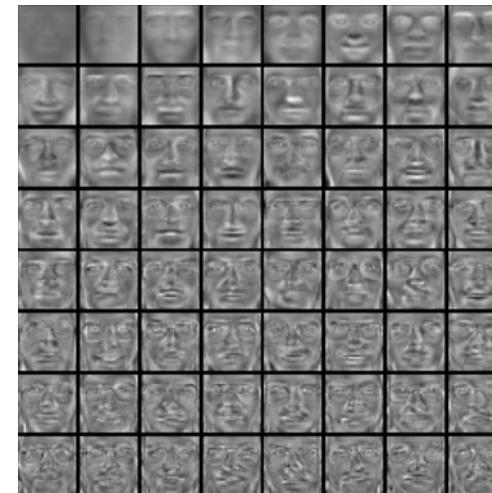


Mean Image



PCA

The first 64 eigenvectors (eigenfaces)



- Then an image of our dataset becomes:

$$\text{Image} = \text{Mean Image} + c_1 * (\text{1st Eigenface}) + c_2 * (\text{2nd Eigenface}) + \dots$$

- So, these coefficients  $c_1, c_2, \dots, c_{64}$  represent the face image!!
- We can perform Face Recognition
  - For a new face image
    - We can calculate its 64 coefficients
    - Find the closest training face (most similar coefficients) in the 64-dimensional space (the most similar face of my training dataset)
    - **Classify** the new face as this most similar training face.

- What is Classification?
- Dimensionality Reduction
  - PCA
  - Face detection
- Classifiers
  - k-Nearest Neighbors
  - Support Vector Machines
- Class/Category Recognition
- Summary

# k-Nearest Neighbors (k-NN)

k-NN is a geometric method for classification.

- It is based on the assumption that neighbor instances belong to the same class.

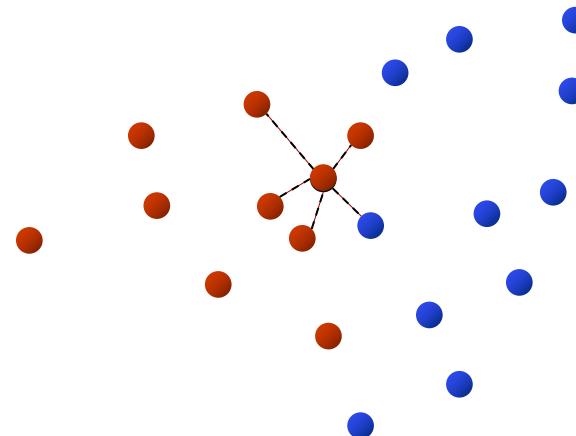
For classifying a new instance  $x_*$  :

- we calculate its distance from all the other instances.
- Then we select the k-nearest instances.
- The probability of the class is given by:

$$p(C_j|x_*) = \frac{k_j}{K}$$

where  $k_j$  is the number of nearest neighbors that belong to class  $j$ , and  $K$  is the total number of nearest neighbors.

# k-Nearest Neighbors (k-NN)



e.g.  $k=5$

What is the class of the black dot?

The black dot belongs to the “red” class with probability 4/5

# k-Nearest Neighbors (k-NN)

The parameters of k-NN are:

- the number of nearest neighbors  $k$
- the distance metric that would be used.
  - Most common distance metric is Euclidean -- other possibilities are: Mahalanobis, Manhattan etc.

# k-Nearest Neighbors (k-NN)

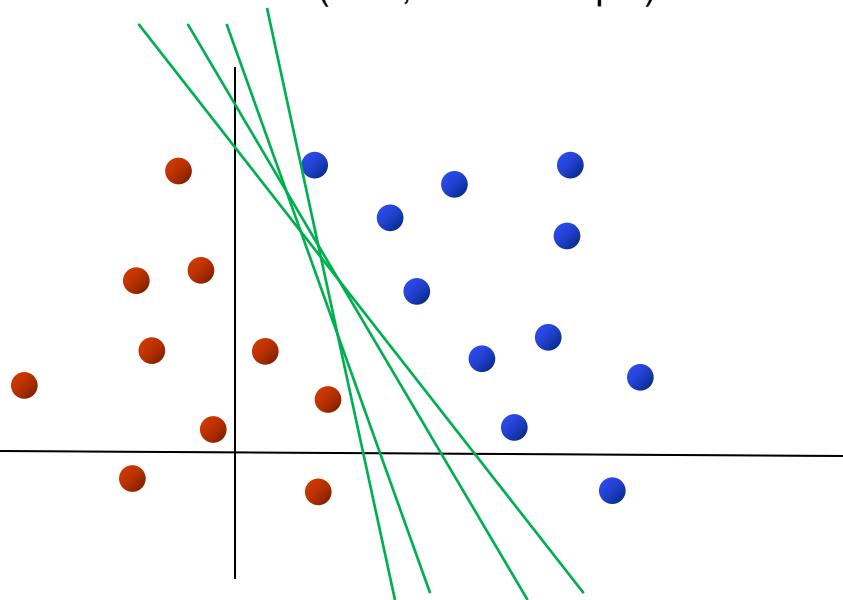
Characteristics of k-NN:

- Very small numbers of k do not perform well on noisy data.
- k-NN is a non-linear classifier.
- Despite its simplicity is powerful.
- Can be very slow for large datasets.

- What is Classification?
- Dimensionality Reduction
  - PCA
  - Face detection
- Classifiers
  - k-Nearest Neighbors
  - Support Vector Machines
- Class/Category Recognition
- Summary

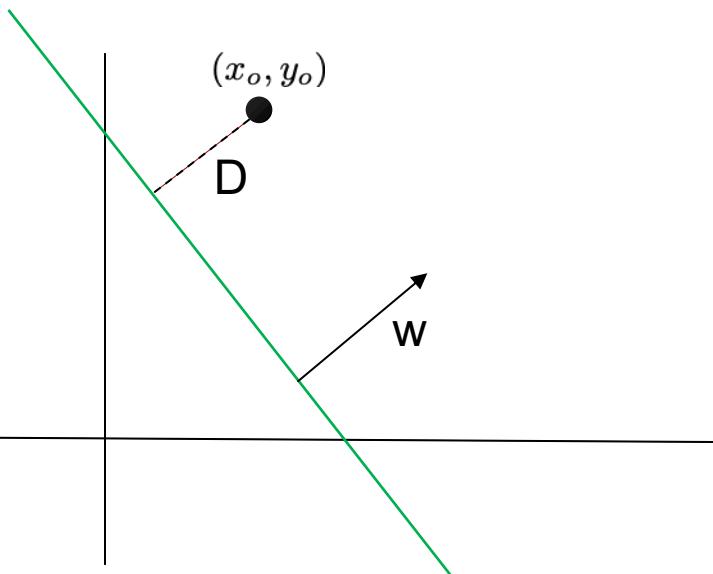
# Support Vector Machines (SVMs)

- SVMs are geometric methods for classification
  - Find a line/surface in feature space that separates the classes
- Linear Classifier: (here, a 2D example)



*Which line is best?*

- Lines in 2D



$$\mathbf{w} = \begin{bmatrix} p \\ q \end{bmatrix}$$
$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$px + qy + b = 0$$

$$\mathbf{w}^T \cdot \mathbf{x} + b = 0$$

$$D = \frac{|px_0 + qy_0 + b|}{\sqrt{p^2 + q^2}} = \frac{\mathbf{w}^T \cdot \mathbf{x} + b}{|\mathbf{w}|}$$

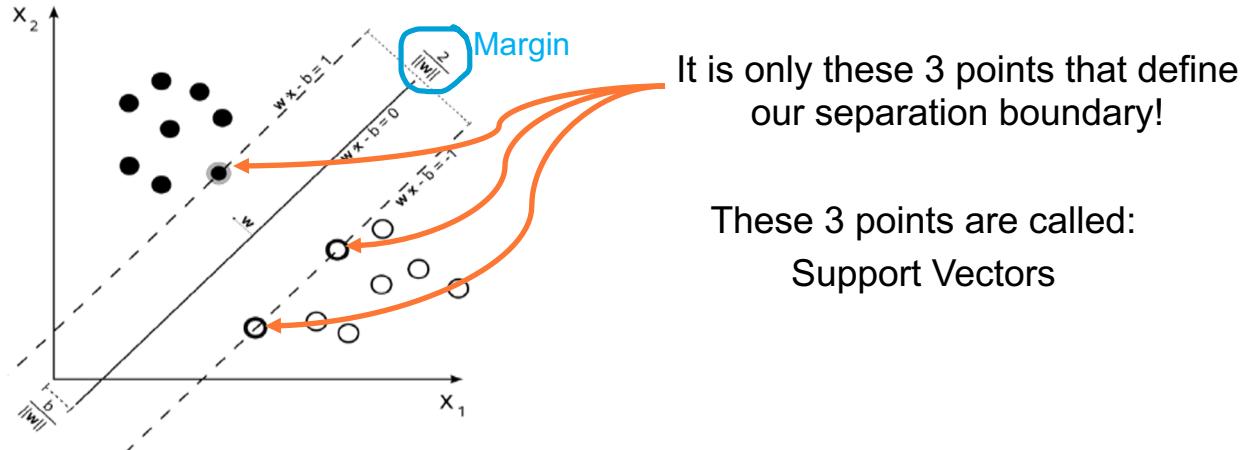
# Support Vector Machines (SVMs)

- SVMs are geometric methods for classification
- They derive a linear decision boundary which satisfies two criteria:
  1. The distance between the instances and the boundary is as large as possible (max margin).
  2. The instances are classified as good as possible.
- Thus, the decision boundary is a hyperplane  $h(x)$  defined as:

$$h(x) = w^T x + b$$

...the goal is to find the  $w$  that satisfy the above criteria.

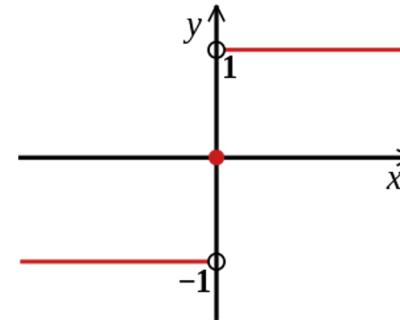
- SVMs are geometric methods for classification
- They derive a linear decision boundary which satisfies two criteria:
  1. The distance between the instances and the boundary is as large as possible (max margin).
  2. The instances are classified as good as possible.



# Support Vector Machines (SVMs)

- Maximize the margin  $\frac{2}{\|\mathbf{w}\|}$  while correctly classifying all training data points
- The goal is to find the  $\mathbf{w}$  that satisfies the criteria of:  $h(x) = \mathbf{w}^T x + b$
- The classification rule is expressed as:  $f(x_*) = \text{sgn}(h(x_*))$
- The sign function is defined as:

$$\text{sgn}(x) := \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases}$$



- Thus, a new instance is labeled with 1 if it is located above the boundary and with -1 otherwise.

# Support Vector Machines (SVMs)

- Extending SVMs!
  - Moving beyond 2D
  - There might be more than 2 classes
  - Data might not be linearly separable

# Support Vector Machines (SVMs)

- Extending SVMs!
  - **Moving beyond 2D**
  - There might be more than 2 classes
  - Data might not be linearly separable
- Our math work just fine for more than 2 dimensions!
  - Instead of lines, we get planes/hyperplanes...

# Support Vector Machines (SVMs)

- Extending SVMs!
  - Moving beyond 2D
  - **There might be more than 2 classes**
  - Data might not be linearly separable
  - Multi-class SVMs
    - Possible solutions:
      - 1-vs-all
        - » Combine multiple binary classifiers.
        - » Choose the one with the largest decision value
      - 1-vs-1
        - » Train one classifier for each pair of classes
        - » Choose the class that most classifiers vote for

# Support Vector Machines (SVMs)

- Extending SVMs!
  - Moving beyond 2D
  - There might be more than 2 classes
  - **Data might not be linearly separable**
  - The kernel trick
    - We can use kernels (functions)
      - We are moving our problem in a higher- (or even infinite-) dimensional space.
      - In this new space our problem is linearly separable!
    - Typical kernels:
      - Linear
      - Polynomial
      - Gaussian (Radial Basis Function – RBF)

- What is Classification?
- Dimensionality Reduction
  - PCA
  - Face detection
- Classifiers
  - k-Nearest Neighbors
  - Support Vector Machines
- Class/Category Recognition
- Summary

# Class/Category Recognition

- Bag of Words



# Class/Category Recognition

- Bag of Words

## Analogy to documents

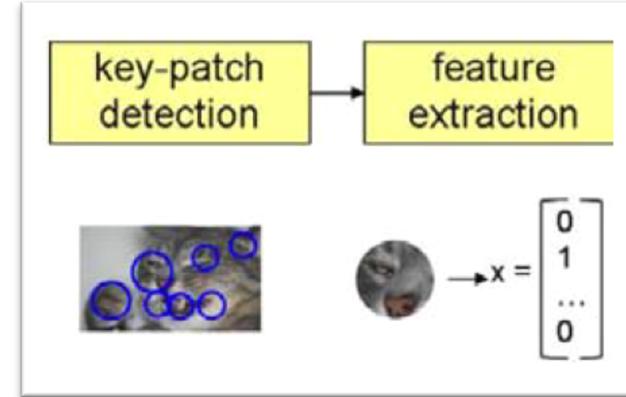
Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that reach our brain via our eyes. For a long time it was believed that the retinal image was processed by the visual centers in the cerebral cortex. This was a movie screen theory. In 1960, two researchers discovered that the eye, cell, optical nerve, image and Hubel and Wiesel.

Hubel and Wiesel have demonstrated that the message about the image falling on the retina undergoes a top-down analysis in a system of nerve cells stored in columns. In this system each column has its specific function and is responsible for a specific detail in the pattern of the retinal image.

China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry said the surplus would be created by a predicted 30% increase in exports to \$750bn, compared with \$660bn. The Chinese government, annoyed that China's trade surplus, commerce, exports, imports, US, yuan, bank, domestic, foreign, increase, trade, value

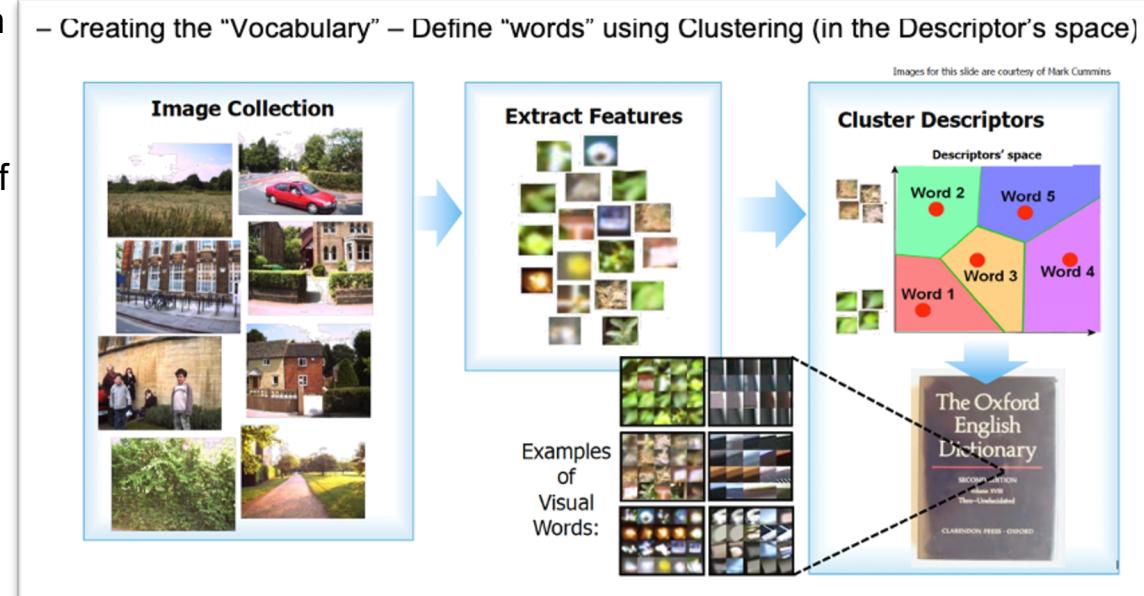
# Class/Category Recognition

- **Bag of Words**
  - Feature Detection
  - Feature Extraction/Description (e.g. SIFT, ...)
  - Codebook/Vocabulary Generation
  - Image description:
    - Histogram Computation
    - a bag of visual words is a vector of occurrence counts of a vocabulary of local image features.
  - Classification of new image in the descriptor's space



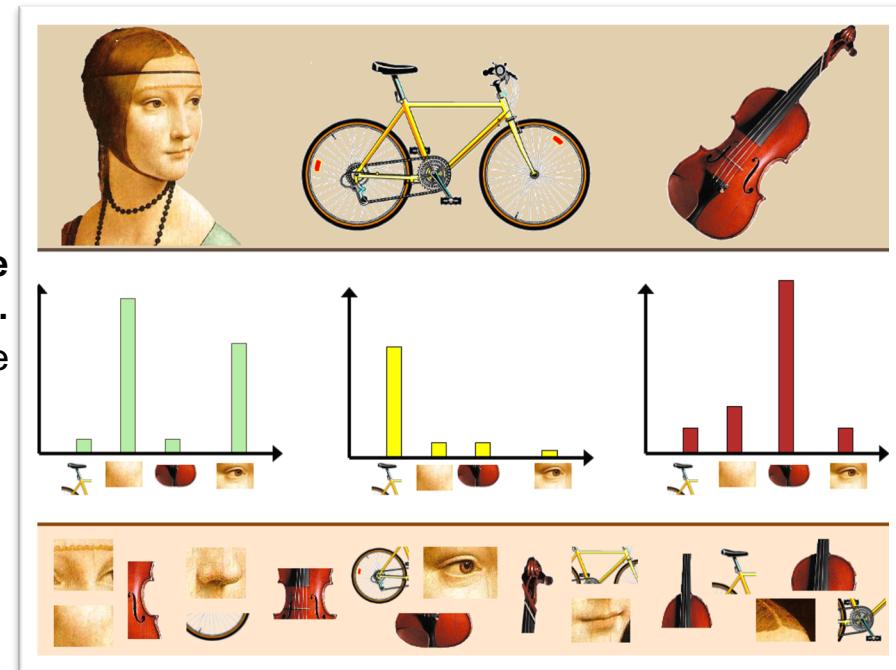
# Class/Category Recognition

- **Bag of Words**
  - Feature Detection
  - Feature Extraction/Description (e.g. SIFT, ...)
- **Codebook/Vocabulary Generation**
- Image description:
  - Histogram Computation
  - a bag of visual words is a vector of



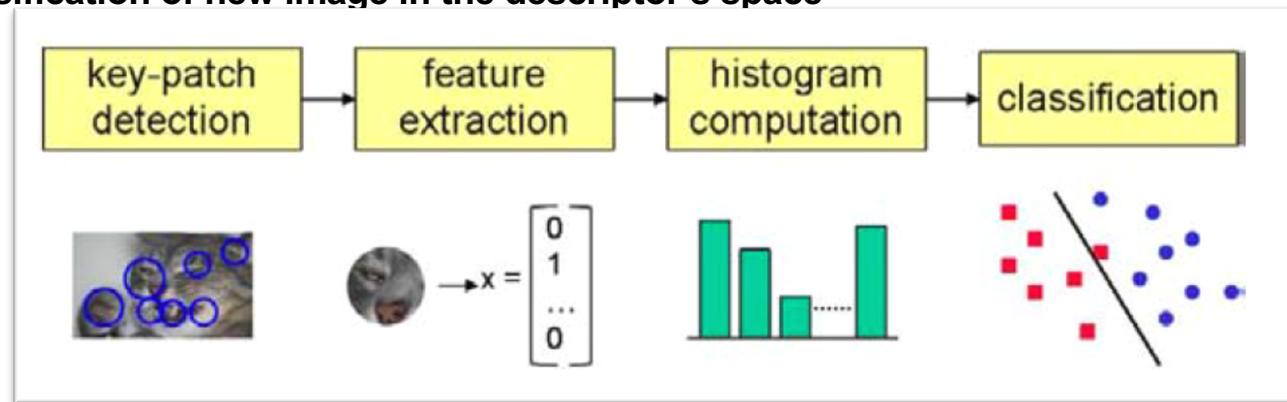
# Class/Category Recognition

- **Bag of Words**
  - Feature Detection
  - Feature Extraction/Description (e.g. SIFT, ...)
  - Codebook/Vocabulary Generation
- **Image description:**
  - Histogram Computation
  - **a bag of visual words is a vector of occurrence counts of a vocabulary of local image features.**
- Classification of new image in the descriptor's space



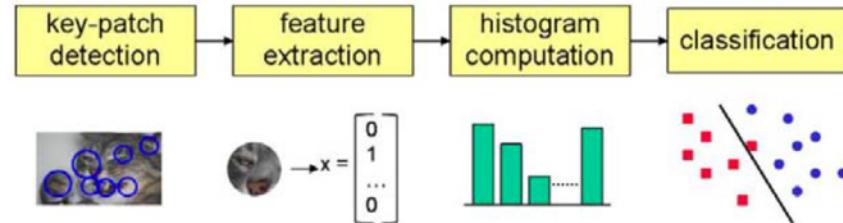
# Class/Category Recognition

- **Bag of Words**
  - Feature Detection
  - Feature Extraction/Description (e.g. SIFT, ...)
  - Codebook/Vocabulary Generation
  - Image description:
    - Histogram Computation
    - a bag of visual words is a vector of occurrence counts of a vocabulary of local image features.
- **Classification of new image in the descriptor's space**



- What is Classification?
- Dimensionality Reduction
  - PCA
  - Face detection
- Classifiers
  - k-Nearest Neighbors
  - Support Vector Machines
- Class/Category Recognition
- Summary

- Machine Learning techniques can find the best label for a new instance, based on known labeled training instances.
  - We might
    - know what we are looking for → **object detection**
    - have a specific rigid object we are trying to recognize → **instance recognition**
    - want to recognize instances of extremely varied classes (e.g. animals or furniture) → **category/class recognition**



*"Woven into all of these techniques is the topic of learning, since hand-crafting specific object recognizers seems like a futile approach given the complexity of the problem."*

R. Szelinski, "Computer Vision: Algorithms and Applications", 2010.

- Latest methods rely on deep neural networks.

- **Object Detection in images:**

- YOLO

- » J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 779-788.
    - » <https://arxiv.org/abs/1506.02640>

- Faster R-CNN

- » S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.
    - » <https://arxiv.org/abs/1506.01497>

- **Classification in 3D point clouds**

- PointNet

- » R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 77-85.
    - » <https://arxiv.org/abs/1612.00593>

Lazaros Nalpantidis

# Classification