# Introduction to Pattern Recognition

UNIT 5TH

PRASHANT TOMAR

The term pattern recognition refers to the task of placing some object to a correct class based on the measurements about the object. Usually this task is to be performed automatically with the help of computer. Objects to be recognized, measurements about the objects, and possible classes can be almost anything in the world. For this reason, there are very different pattern recognition tasks. A system that makes measurements about certain objects and thereafter classifies these objects is called a **pattern recognition system. Example-**

A spam (junk-mail) filter is another example of pattern recognition systems. A spam filter recognizes automatically junk e-mails and places them in a different folder (e.g. /dev/null) than the user's inbox.

Some pattern recognition tasks are everyday tasks (e.g. speech recognition) and some pattern recognition tasks are not so everyday tasks.
For example, it is very difficult to 'teach' a computer to read hand-written text. A part of the challenge follow because a letter 'A' written by a person B can look highly different than a letter 'A' written by another person. For this reason, it is worthwhile to model the variation within a class of objects (e.g. hand-written 'A's).

# Machine Perception

- Build a machine that can recognize patterns:
  - Speech recognition
  - Fingerprint identification
  - OCR (Optical Character Recognition)
  - DNA sequence identification

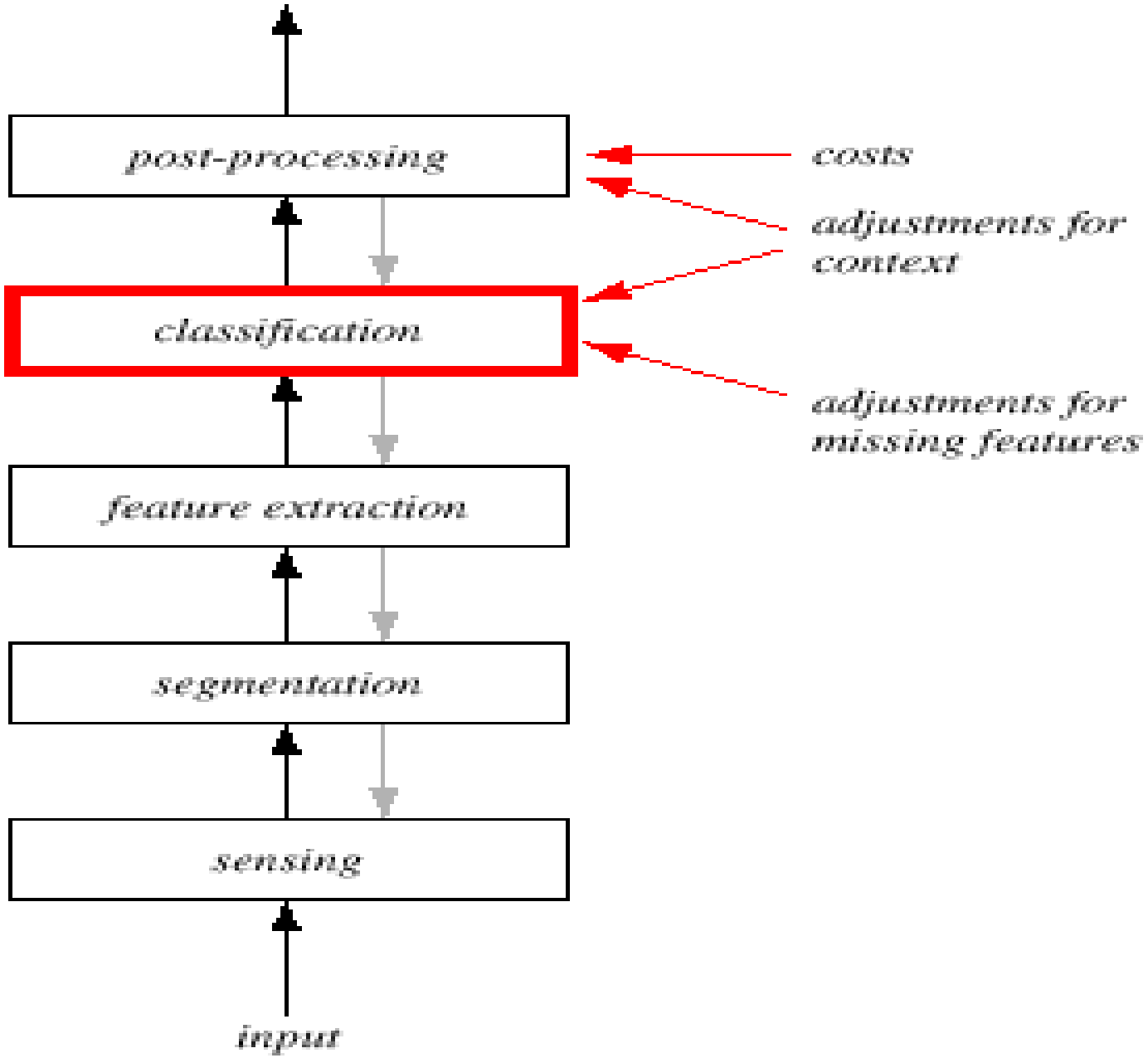
Basic Pattern Recognition Systems:

Many pattern recognition systems can be thought to consist of five stages:

1. Sensing (measurement);
2. Pre-processing and segmentation;
3. Feature extraction;
4. Classification;
5. Post-processing;

A majority of these stages are very application specific.

**Sensing** refers to some measurement or observation about the object to be classified. For example, the data can consist of sounds or images and sensing equipment can be a microphone array or a camera.


post-processing
classification
feature extraction
segmentation
sensing
input
costs
adjustments for context
adjustments for missing features

**Pre-processing** refers to filtering the raw data for noise suppression and other operations performed on the raw data to improve its quality. In segmentation, the measurement data is partitioned so that each part represents exactly one object to be classified. For example in address recognition, an image of the whole address needs to be divided to images representing just one character. The result of the segmentation can be represented as a vector that is called a pattern vector.

**Feature extraction**. Especially when dealing with pictorial information the amount of data per one object can be huge. A high resolution facial photograph (for face recognition) can contain 1024*1024 pixels. The pattern vectors have then over a million components. The most part of this data is useless for classification.

In **feature extraction**, we are searching for the features that best characterize the data for classification The result of the feature extraction stage is called a feature vector. The space of all possible feature vectors is called the feature space. In face recognition, a widely used technique to reduce the number features is principal component analysis (PCA). PCA is a statistical technique to reduce the dimensionality of a data vector while retaining most of the information that the data vector contains.

**The classifier:** takes as an input the feature vector extracted from the object to be classified. It places then the feature vector (i.e. the object) to class that is the most appropriate one. In address recognition, the classifier receives the features extracted from the sub-image containing just one character and places it to one of the following classes: ’A’,’B’,’C’..., ’0’,’1’,...,’9’. The classifier can be thought as a mapping from the feature space to the set of possible classes. Note that the classifier cannot distinguish between two objects with the same feature vector.

**Post-processing:** A pattern recognition system rarely exists in a vacuum. The final task of the pattern recognition system is to decide upon an action based on the classification result(s). A simple example is a bottle recycling machine, which places bottles and cans to correct boxes for further processing

**Statistical pattern recognition:** Statistical pattern recognition relates to the use of statistical techniques for analysing data measurements in order to extract information and make justified decisions. Applications such as data mining, web searching, multimedia data retrieval, face recognition, and cursive handwriting recognition, all require robust and efficient pattern recognition techniques.

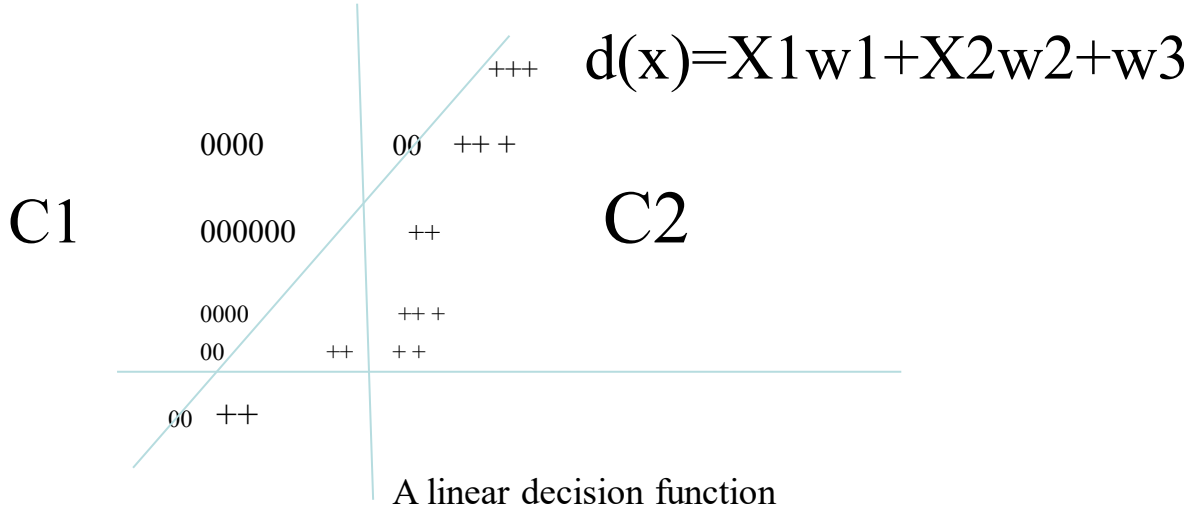
It is based on the use of decision function to classify object. A decision function map vector X into decision regions of D . i.e.

f:X→D where f is decision function.

Example: let O={O1,O2,…On}is set of universe object and each object Oi have K-observable attribute V={v1,v2…vk}.

Suppose X={x1,x2….xm}is a subset of V, whose value represent unique character of the object Oi therefore the set of attributes X subset of V provide the partition of the set V at least two part say C>=2 grouping or classification of the object Oi, hence the region are used to classify each Oi as belonging to at most one of the C classes.

When there are two classes C1 and C2,the values of the objects pattern vector may tend to cluster into two disjoint groups. In case ,a linear decision function d(X) can be used to determine on object's class.




Wi are the parameters or weights that are adjusted to find a separating line for the classes.

If d(X)<0 then object is classified as belonging to C1.

If d(X)>0 then object is classified as belonging to C2.

If d(X)=0 then object is classified as indeterminate.

## Principal Component Analysis (PCA) :

In face recognition, a widely used technique to reduce the number features is principal component analysis (PCA). PCA is a statistical technique to reduce the dimensionality of a data vector while retaining most of the information that the data vector contains.

- One approach to deal with high dimensional data is by reducing their dimensionality.
- Project high dimensional data onto a lower dimensional sub-space using linear or non-linear transformations.

$$x = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_N \end{bmatrix} \text{--->} \textit{reduce dimensionality} \text{--->} y = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_K \end{bmatrix} \quad (K \ll N)$$

- Each dimensionality reduction technique finds an appropriate transformation by satisfying certain criteria (e.g., information loss, data discrimination, etc.)

- The goal of PCA is to reduce the dimensionality of the data while retaining as much as possible of the variation present in the dataset.
- Find a basis in a low dimensional sub-space:
  - Approximate vectors by projecting them in a low dimensional sub-space:

(1) Original space representation

$$x = a_1 v_1 + a_2 v_2 + ... + a_N v_N$$

where $v_1, v_2, ..., v_n$ is a base in the original N-dimensional space

$$\begin{bmatrix} a_1 \\ a_2 \\ ... \\ a_N \end{bmatrix}$$

# 2) Lower-dimensional sub-space representation:

$$\hat{x} = b_1 u_1 + b_2 u_2 + ... + b_K u_K$$

where $u_1, u_2, ..., u_K$ is a base in the $K$-dimensional sub-space (K<N)

$$\begin{bmatrix} b_1 \\ b_2 \\ ... \\ b_K \end{bmatrix}$$

- *Note:* if K=N, then x^ =X

- Example (K=N):

$$v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, v_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{(standard basis)}$$

$$x_v = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = 3v_1 + 3v_2 + 3v_3$$

$$u_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, u_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, u_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{(some other basis)}$$

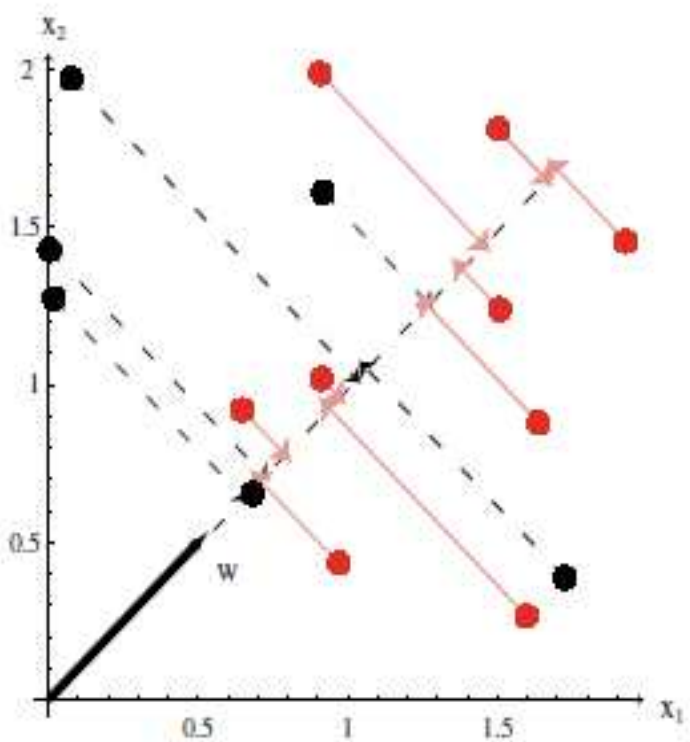
$$x_u = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = 0u_1 + 0u_2 + 3u_3$$

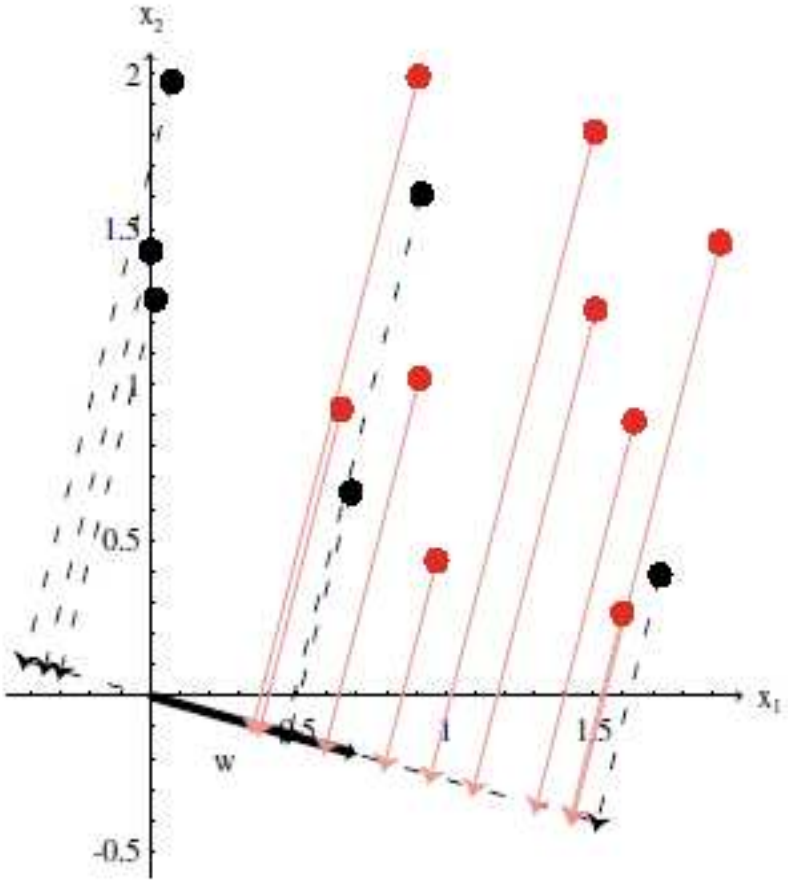
thus, $x_v = x_u$

# Linear Discriminant Analysis (LDA):

**Linear discriminant analysis** (**LDA**) is a generalization of **Fisher's linear discriminant**, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

- Perform dimensionality reduction "while preserving as much of the class discriminatory information as possible".
- Seeks to find directions along which the classes are best separated.
- Takes into consideration the scatter *within-classes* but also the scatter *between-classes*.
- More capable of distinguishing image variation due to identity from variation due to other sources such as illumination and expression
- LDA computes a transformation that maximizes the between-class scatter while minimizing the within-class scatter:


x2
2
1.5
1
0.5
w
0.5
1
1.5
x1


x2
2
1.5
1
0.5
-0.5
w
1
1.5
x1

- Notation:

- Suppose there are $C$ classes
- Let $\boldsymbol{\mu}_i$ be the mean vector of class $i$, $i = 1, 2, .., C$
- Let $M_i$ be the number of samples within class $i$, $i = 1, 2, .., C$,
- Let $M = \sum_{i=0}^{C} M_i$ be the total number of samples. and

Within-class scatter matrix:

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{M_i} (y_j - \boldsymbol{\mu}_i)(y_j - \boldsymbol{\mu}_i)^T$$

(each sub-matrix has rank 1 or less, i.e., outer product of two vectors)

Between-class scatter matrix:

($S_b$ has at most rank C-1)

$$S_b = \sum_{i=1}^{C} (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$

$$\boldsymbol{\mu} = 1/C \sum_{i=1}^{C} \boldsymbol{\mu}_i \quad \text{(mean of entire data set)}$$

For example, an educational researcher may want to investigate which variables discriminate between high school graduates who decide (1) to go to college, (2) to attend a trade or professional school, or (3) to seek no further training or education. For that purpose the researcher could collect data on numerous variables prior to students' graduation. After graduation, most students will naturally fall into one of the three categories. *Discriminant Analysis* could then be used to determine which variable(s) are the best predictors of students' subsequent educational choice.

A medical researcher may record different variables relating to patients' backgrounds in order to learn which variables best predict whether a patient is likely to recover completely (group 1), partially (group 2), or not at all (group 3). A biologist could record different characteristics of similar types (groups) of flowers, and then perform a discriminant function analysis to determine the set of characteristics that allows for the best discrimination between the types.

Classification Technique:

1. Nearest neighbor (NN).
2. Bayes Classifier
3. SVM
4. K-means clustering

**SVM(Support Vector machine):**

Single-layer networks have a simple and efficient learning algorithm, but have very limited expressive power—they can learn only linear decision boundaries in the input space. Multilayer networks, on the other hand, are much more expressive—they can represent general nonlinear functions—but are very hard to train because of the abundance of local minima and the high dimensionality of the weight space. So we will explore a relatively new family of learning methods called **support vector machines (SVMs)** or, more generally, **kernel machines**.

A support vector machine with 25,000 support vectors achieved an error rate of 1.1%. This is remarkable because the SVM technique, like the simple nearest- neighbor approach, required almost no thought or iterated experimentation on the part of the developer.

To some extent, kernel machines give us the best of both worlds. That is, these methods use an efficient training algorithm and can represent complex, nonlinear functions

# 1. Nearest neighbor (NN):

**K-nearest-neighbor (kNN)** classification is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. K-nearest-neighbor classification was developed from the need to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine.

Nearest neighbour **classifiers** are a class of non-parametric methods used in statistical classification(or pattern recognition). The method classifies objects based on closest training examples in the feature space.

Nearest-neighbor density estimation applied to predicting the class of an object. To classify measurement $x$, take the $k$ nearest training measurements and choose the most popular class among them. The quality of this method depends crucially on the distance metric. This method is very sensitive to irrelevant features, so it is usually combined with feature selection

The k-nearest-neighbor classifier is commonly based on the Euclidean distance between a test sample and the specified training samples. Let $\mathbf{x}i$ be an input sample with $p$ features $(xi1,xi2,\ldots,xip)$ , $n$ be the total number of input samples ($i$=1,2,…,$n$) and $p$ the total number of features ($j$=1,2,…,$p$) . The Euclidean distance between sample $\mathbf{x}i$ and $\mathbf{x}l$ ($l$=1,2,…,$n$) is defined as

$$d(\mathbf{x}i,\mathbf{x}l)=\sqrt{}\ (xi1-xl1)^2+(xi2-xl2)^2+\cdots+(xip-xlp)^2$$

# K-means Clustering:

In some classifiers, such as linear support-vector machines and linear neural networks, it's possible to do backward elimination more efficiently. You train the classifier once, and then remove the feature that has the smallest input weight.
These methods can be extended to non-linear SVMs and neural networks, but it gets somewhat more complicated there.
Slide
Another whole strategy for feature selection is to make new features. One very drastic method is to try to cluster all of the inputs in your data set into a relatively small number of groups, and then learn a mapping from each group into an output.
So, in this case, we might divide the input points into k-clusters. The stars indicate the cluster centers.

Then, for each cluster, we would assign the majority class. Now, to predict the value of a new point, we would see which region it would land in, and predict the associated class.

This is different from nearest neighbors in that we actually discard all the data except the cluster centers. This has the advantage of increasing interpretability, since the cluster centers represent "typical" inputs.

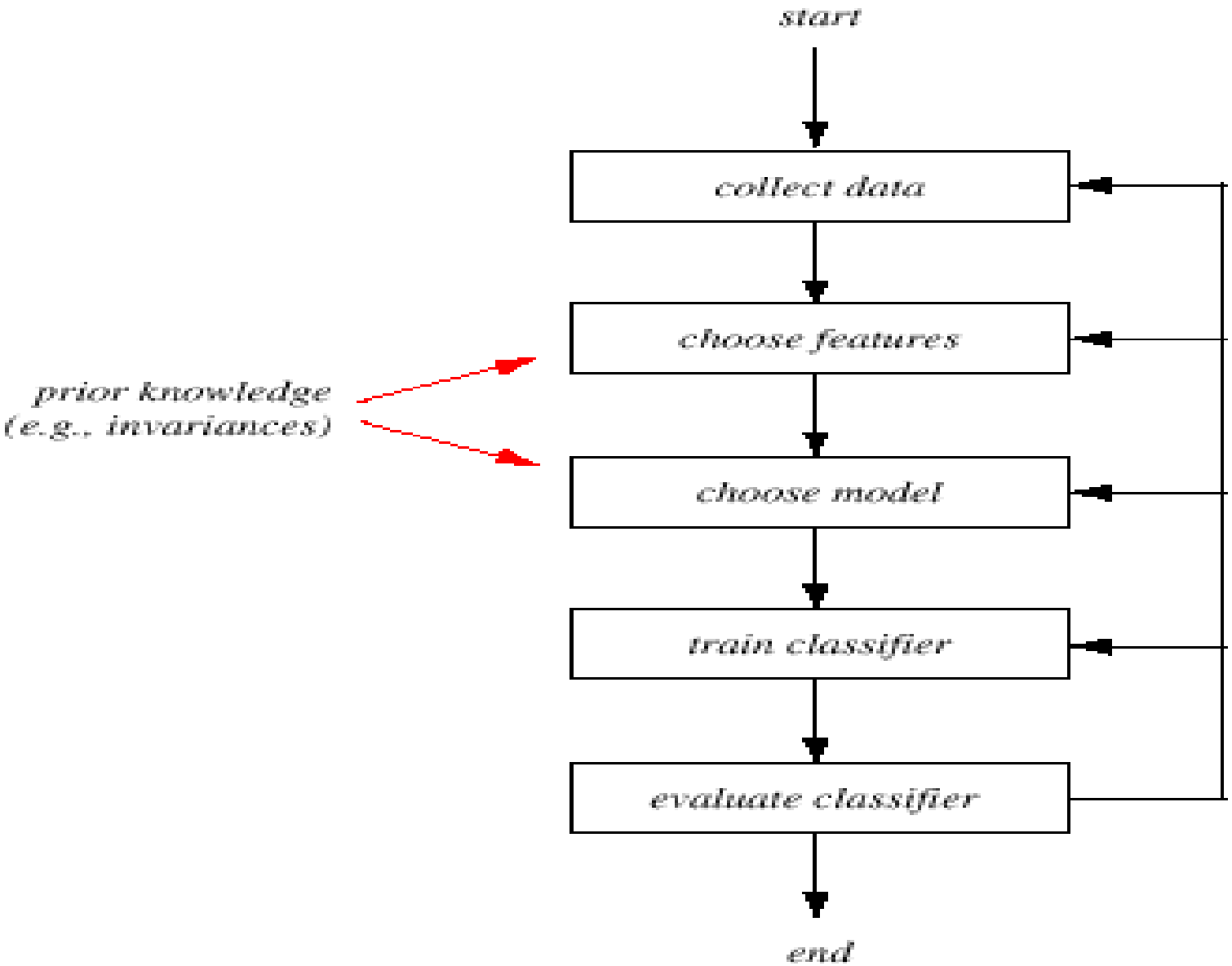
So, what makes a good clustering? There are lots and lots of different technical choices. The basic idea is usually that you want to have clusters in which the distance between points in the same group is small and the distance between points in different groups is large.
Clustering, like nearest neighbor, requires a distance metric, and the results you get are as scale-sensitive as they are in nearest-neighbor.
One of the simplest and most popular clustering methods is K-means clustering. It tries to minimize the sum, over all the clusters, of the variance of the points within the cluster (the distances of the points to the geometric center of the cluster).
Unfortunately, it only manages to get to a local optimum of this measure, but it's usually fairly reasonable.
Slide 8.4.23 Here is the code for the k-means clustering algorithm. You start by choosing k, your desired number of clusters. Then, you can randomly choose k of your data points to serve as the initial cluster centers.
Slide 8.4.24 Then, we enter a loop with two steps. The first step is to divide the data up into k classes, using the cluster centers to make a Voronoi partition of the data. That is, we assign each data point to the cluster center that it's closest to.
Now, for each new cluster, we compute a new cluster center by averaging the elements that were assigned to that cluster on the previous step.
We stop when the centers quit moving. This process is guaranteed to terminate.

**Properties of k-means clustering:**

The K-means clustering is 'an application' of the minimum distance classifier to clustering. Therefore, the k-means clustering suffers from the same drawbacks as the minimum distance classification. If the clusters are of different sizes (i.e. they contain a different number of feature vectors), the k-means clustering is not suc- cessful. If the clusters have very different covariances or scales of different features are different, then the clustering by k-means is not appropriate.

# The Design Cycle

- Data collection
- Feature Choice
- Model Choice
- Training
- Evaluation
- Computational Complexity


start
collect data
choose features
prior knowledge
(e.g., invariances)
choose model
train classifier
evaluate classifier
end

- Data Collection

  – How do we know when we have collected an adequately large and representative set of examples for training and testing the system?

- Feature Choice

  – Depends on the characteristics of the problem domain.

  – Simple to extract, invariant to irrelevant transformation, insensitive to noise.

- Model Choice

  – Unsatisfied with the performance of our fish classifier and want to jump to another class of model

- Training
  - Use data to determine the classifier.
  - Many different procedures for training classifiers and choosing models exists.

- Evaluation

  – Measure the error rate (or performance) and switch from one set of features to another one