Pattern recognition

"To understand is to perceive patterns" – Sir Isaiah Berlin, Russian philosopher

“The more relevant patterns at your disposal, the better your decisions will be. This is hopeful news to proponents of artificial intelligence, since computers can surely be taught to recognize patterns” – Herbert Simon, Nobel laureate, 1978
Outline

- Introduction of concept
- Syntactical pattern recognition
- Statistical pattern recognition
- Reading Sonka 9.2, 9.4 and Gonzalez and Woods 12 (selected subsections)
  - Building simple classifiers
    - Supervised classification
      - Minimum distance classifier
      - Bayesian classifiers
    - Unsupervised classification
      - K-means algorithm
What is Pattern Recognition?

- Visual pattern recognition tries to answer a basic question: What is in the image (video)? (where, when)
- Examples:
  - What object is this based on shape, position? (geometry-based object recognition)
  - What kind of pixel is this based on local image properties? (appearance-based pattern recognition)
Example: face detection
Finding faces

- Faces “look like” templates (at least when they’re frontal).
- General strategy:
  - search image windows at a range of scales
  - Correct for illumination
  - Present corrected window to classifier

- Issues
  - How corrected?
  - What features?
  - What classifier?
  - what about lateral views?
Challenge 1: viewpoint variation

Michelangelo 1475-1564
Challenge 2: illumination

slide credit: S. Ullman
Challenge 3: occlusion

Magritte, 1957
Challenges 4: scale
Challenge 5: deformation

Xu, Beihong

1943
Challenge 6: background clutter

Klimt, 1913
Approaches in pattern recognition

- **Statistical**
  - Operates in the feature space
  - Classifies objects based on their features
  - The features of an object provide a **quantitative** description of the object

- **Syntactical**
  - Based on a **qualitative** description of an object
Approaches in pattern recognition: structural (syntactic)

- **Section 9.4 (only pp. 410-411-412)**
- hierarchic perspective suitable for recognizing complex patterns
- The simplest sub-pattern to be recognized is called a primitive.
- The complex pattern is represented in terms of interrelationships between primitives
Approaches in pattern recognition: statistical

- Terminology: features, feature vectors, classes, classifiers
- Each pattern is represented in terms of d features and is viewed as a point in a d-dimensional space
- Issue: how do we choose the features?
- The effectiveness of the chosen representation (feature set) is determined by how well patterns from different classes are separated.
- Given a set of training samples for each class, the objective is to establish decision boundaries in the feature space, and thus to separate patterns belonging to different classes
Statistical pattern recognition: Building classifiers

- Each pattern is a point in feature space

Three types of flowers described by two measurements (features)
- Petal length
- Petal width
Features and patterns

- Can’t really classify things
- Have to classify their perceived pattern
- Terminology:
  Properties we can measure: **features**
  Collections of features: **feature vector**, which is a description of a **pattern**

- Features have to be quantitative measures (scalar or vector-valued)
- Examples: area, average curvature on the outer boundary, average intensity, etc.
- Questions:
  - How many features?
  - Which ones?
  - Relative importance?
Classes

- Pattern recognition doesn’t produce general, open-ended object descriptions
- “which of these possibilities is it?”
- Finite sets of possibilities are called **classes**
- Might be a specific “yes/no” (two-class problem)
- Might be multiple options (e.g., OCR)
General classification process

- Extract features
- Assemble features into a feature vector or pattern
- Assign to a class
  - Might be a single classification
  - Might rank possible classifications
  - Might produce (relative) probabilities
Building classifiers (cont’d)

- **Key ideas:**
  - Patterns from the same class should cluster together in feature space
  - **Supervised training:** learn the properties of the cluster (distribution) for each class
    - Example: minimum distance classifier
  - **Unsupervised training:** find the clusters from scratch; no information about the class structure is provided
    - Example: k-means classifier
Minimum-distance classifier

- Reading Gonzalez and Woods excerpt pp. 698-701
- Idea: Use a single prototype for each class $\omega_i$ (usually the class’s mean $m_i$)
- Training:
  - compute each class’s prototype (mean)
- Classification:
  - Assign unlabeled patterns to the nearest prototype in the feature space
Decision boundaries

- If we partition the feature space according to the nearest prototype, we create decision boundaries.

Decision boundary between two classes; means are represented by the dark dot and square.
Minimum distance classifier (cont’d)

- Simpler calculation: The distances to the prototypes don’t need to be computed
- What we need is finding the distance to the nearest prototype (minimal)
Linear decision boundaries

- **Linearity:**

\[ g(x) = w^T x + w_0 \]

where

\[ w = m_1 - m_2 \]
\[ w_0 = -\frac{1}{2}(m_1^T m_1 - m_2^T m_2) \]

- \( g(x) > 0 \) Assign \( x \) to \( \omega_1 \)
- \( g(x) < 0 \) Assign \( x \) to \( \omega_2 \)
- \( g(x) = 0 \) Undecided
Discriminants

- A function used to test the class membership is called a discriminant.
- The two-class result can be extended to multiple cases.
- Construct a single discriminant $g_i(x)$ for each class $\omega_i$, and assign $x$ to class $\omega_i$ if $g_i(x) > g_j(x)$ for all other classes $\omega_j$. 
Performance of the minimum distance classifier: discussion

Figure 4.5  Three classes with complex structure: Classification using nearest mean will yield poor results.
Unsupervised training: clustering

- The goal is to find natural groupings of patterns
- It is useful if the training set is not pre-labeled
- Usually some a priori information is available (for instance number of classes)
The k-means algorithm

- is an unsupervised version of minimum distance classification; requires number of classes k.

1. Place \( k \) points into the space represented by the feature vectors that are being clustered. These points represent initial group centroids (prototypes).

2. Assign each object to the group that has the closest centroid.
3. When all vectors have been assigned, recalculate the positions of the $K$ centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move.
The k-means algorithm (cont’d)

- Trajectory of the means $m_1$ and $m_2$
Outline

- Introduction of concept
- Syntactical pattern recognition
- Statistical pattern recognition
- Reading Sonka 9.2, 9.4 and Gonzalez and Woods 12 (selected subsections)
  - Building simple classifiers
    - Supervised classification
      - Minimum distance classifier
      - Bayesian classifiers
    - Unsupervised classification
      - K-means algorithm
Main concept

- Perform classification using a probabilistic framework
- When we classify, we measure the feature vector \( \mathbf{x} \)
  
  “Given that this pattern has features \( \mathbf{x} \), what is the probability that it belongs to class \( \omega_i \)?”
Notation: probabilities

○ The probability of discrete event $A$ occurring is $P(A)$
○ The continuous random variable $x$ has a probability density function (pdf) is $p(x)$
○ For vector-valued random variables $\mathbf{x}$, we write this as $p(\mathbf{x})$
Conditional probabilities

- We write the conditional probability of $A$ given $B$ as $P(A/B)$

- This means “the probability of $A$ given $B” or “given $B$, what is the probability of $A$?”

- Or for random variables, $p(x/A)$ and $p(x/A)$
Training of Bayesian Classifiers

- Suppose that we measure features for a large training set taken from class $\omega_i$.

- Each of these training patterns has a different value $\mathbf{x}$ for the features. This can be written as the class-conditional probability: $p(\mathbf{x}/\omega_i)$

- In other words, How often do things in class $\omega_i$ exhibit features $\mathbf{x}$?
Bayesian Classification

- When we classify, we measure the feature vector $\mathbf{x}$
  “Given that this pattern has features $\mathbf{x}$, what is the probability that it belongs to class $\omega_i$?”
- Mathematically, this is written as $P(\omega_i/ \mathbf{x})$
How are $P(\omega_i/x)$ and $p(x/\omega_i)$ related?

Through the Bayes Theorem...
General structure of a Bayesian classifier

- Measure for each class \( p(\mathbf{x}/\omega_i) \)

**Prior Knowledge:**
- Measure or estimate \( P(\omega_i) \) in the general population.
  (Can sometimes aggregate the training set if it is a reasonable sampling of the population)

**Classification:**
- 1. Measure feature (\( \mathbf{x} \)) for new pattern.
- 2. Calculate posterior probabilities \( P(\omega_i/\mathbf{x}) \) for each class with Bayes formula
- 3. Choose the one with the largest posterior \( P(\omega_i/\mathbf{x}) \)
Normally-distributed class-conditional probabilities

\[ p(x / \omega_i) = \frac{1}{\sqrt{2\pi \sigma_i}} e^{-\frac{1}{2} \left( \frac{x-\mu_i}{\sigma_i} \right)^2} \]

**FIGURE 12.10** Probability density functions for two 1-D pattern classes. The point \( x_0 \) shown is the decision boundary if the two classes are equally likely to occur.
From probabilities to discriminants

- We want to compute the discriminant for class $\omega_i$ assuming that the likelihood has a normal distribution...
Extending to multiple features

The key term in the discriminant for a 1-D normal distribution is:

\[(x - \mu_i)^2 / \sigma_i^2\]

Natural extension to multiple features:
- normalizing each feature’s “distance from the mean” by the respective standard deviation
- then use minimum distance classification
(by considering the use of prior class probabilities as well)

Normalizing each feature by its variance is called naïve Bayes because it does not take into account the relationships between features
The multivariate normal distribution

- In multiple dimensions, the normal distribution takes on the following form:

\[
p(x) = \left( \frac{1}{\sqrt{2\pi}} \right)^d \frac{1}{|C|^{1/2}} e^{-\frac{1}{2}(x-m)^TC^{-1}(x-m)}
\]

- The covariance matrix consists of the variances of the variables (features) along the main diagonal and the covariances between each pair of variables (features) in the other matrix positions.

\[
COV = \frac{\sum_{i=1}^{n} (X_i - \bar{x})(Y_i - \bar{y})}{n - 1}
\]
Multivariate normal Bayesian classification

- For multiple classes in a p-dimensional feature space, each class $\omega_i$ has its own mean vector $m_i$ and covariance matrix $C_i$.
- The class-conditional probabilities are:

$$p(x|\omega_i) = \frac{1}{(2\pi)^{d/2}|C_i|^{-1/2}} e^{-\frac{1}{2}(x-m_i)^T C_i^{-1} (x-m_i)}$$
Mahalonobis distance

- The expression \((x - m_i)^T C^{-1} (x - m_i)\)

- can be considered as a distance between feature vector \(x\) and class \(i\). \(C\) is the covariance matrix computed for class \(i\).

- It can be proven that the Mahalonobis distance satisfies all properties of a distance function.

- Mahalonobis distance is more useful than Euclidian distance when features have different scales; the importance of each feature in classification is lost with Euclidian distance.

\[(600^2 + 0.8^2) = 600.0007\]
Equivalence between classifiers

- Pattern recognition using multivariate normal distributions and equal priors is simply a minimum Mahalonobis distance classifier.