Recap

• (Ch 12) Clustering
  • The curse of dimensionality
  • Multivariate normal distribution
  • The clustering problem
  • $k$-means algorithm

Today

• (Ch 12) Clustering
  • $k$-means algorithm
  • Vector quantization
The clustering problem

• Given a dataset \( \{x\} \), separate the data items into clusters so that
  • Items within a cluster are close to each other
  • Items in different clusters are far from each other

• There are two problems to solve
  • Determine the number of clusters
  • Assign each item to a cluster

• Note that we are taking unlabeled data and assigning a class label to each item
$k$-means clustering

- Pick a value for $k$, which is the number of clusters
- Select $k$ random cluster centers
- Iterate the following two steps until convergence
  - Assign each data item to the nearest cluster center
  - Update each cluster center as the mean of the items assigned to its cluster
$k$-means clustering result: iris example

true labels

$k$-means with $k = 2$ clusters
$k$-means clustering result: iris example
Groceries in Portugal example

- The dataset consists of the annual grocery spending of 440 customers

- Each customer’s spending is recorded in 6 categories:
  fresh food, milk, grocery, frozen, detergents/paper, delicatessen

- Each customer is labeled by
  - Channel (Channel 1, Channel 2)
  - Region (Region 1, Region 2, Region 3)

for a total of 6 channel/region labels
Visualizing the data: groceries example

- The scatterplot matrix
  - along 6 spending dimensions
  - with 6 channel/region labels
does not reveal any structure

- At first glance, it does not look like clustering will help
$k$-means clustering: groceries example
Trying to find structure: groceries example

• It is reasonable to think that there are certain customer “types”
  • Customers who cook meals at home would spend more on fresh food
  • Customers with children would spend more on milk

• We don’t know what these customer types are, but we can let the cluster centers stand for them

• Even though each channel/region has many types of customers, perhaps each of them has a characteristic mix of customer types
Cluster center histograms: groceries example

• For each channel/region, we make a histogram of customers that map to each of 10 cluster centers ("customer types")

• There is more similarity in the mix within channels than regions

• We can now classify a group of customers (of arbitrary size) from an unknown channel/region
Classifying data of varying size

• The classifiers of Chapter 11 all assumed that each feature vector $\mathbf{x}$ had the same number of entries

• Many datasets have items of different size
  • Images usually have different numbers of pixels
  • Audio signals (and other time series) usually have different durations

• We will use vector quantization to map variable length data to fixed-length feature vectors using cluster center histograms
Pattern vocabulary: conceptual example

• Suppose we want to classify images as either beach or prairie

• We slice each training image into $10 \times 10$ pixel subimages and cluster all subimages to construct a pattern vocabulary of $k$ patterns.

key cluster centers

sand  water  dry grass  cloud/sky
Feature vectors: conceptual example

- To represent an image as a fixed-length feature vector
  - Slice the image into $10 \times 10$ pixel subimages
  - Assign each subimage to the nearest of the $k$ patterns (i.e. cluster centers)
  - The counts form a feature vector of dimension $k$

$$x_i = \begin{bmatrix} 5 \\ 5 \\ 0 \\ 5 \end{bmatrix}$$

- These feature vectors are the fixed-length inputs for a classifier
Classification with vector quantization

• Build a pattern vocabulary
  • Slice the training set of signals into pieces of fixed size $d$
  • Cluster all the pieces and find $k$ cluster centers (typically using $k$-means)

• Represent each signal as a feature vector
  • Slice each signal in the training and test sets into pieces of size $d$
  • Count the number of slices nearest each cluster center to obtain a $k$-dimensional feature vector

• Train a classifier using the training feature vectors and evaluate it using the test feature vectors
The project: activity from accelerometer data

• The dataset consists of Fitbit-like accelerometer signals, each of which
  • Can be of arbitrary length
  • Consists of 3 dimensions (x, y, z) of data sampled at 32 Hz
  • Is labeled with one of 14 activities, such as “brushing teeth”

https://archive.ics.uci.edu/ml/datasets/Dataset+for+ADL+Recognition+with+Wrist+ worn+Accelerometer

• Your task is to train a classifier to take an accelerometer signal and
  map it to an activity
The project: looking at the raw data
The project: building a pattern vocabulary

- Slice each signal into non-overlapping pieces of 1 second duration, which gives you pieces of size $d = 32 \times 3 = 96$

- Cluster the 96-dimensional vectors to $k$ cluster centers using scikit-learn’s $k$-means clustering algorithm ($k = 480$)
The project: representation and classification

• Represent each signal as a $k$-dimensional feature vector

• Train a multiclass classifier such as scikit-learn’s random forest on the training vectors

• Evaluate the classifier using the test vectors

• Improve the classifier by tuning parameters