Maximize the margin

\[ \max_{\gamma, \mathbf{w}, w_0} \gamma \]
\[ y_i (\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq \gamma, \forall i \in \{1, \ldots, N\} \]
A convention: Normalized margin

Canonical hyperplanes

\[ x_i = \bar{x}_i + \lambda \frac{w}{||w||} \]

\[ \begin{align*}
    w \cdot x^+ + w_0 &= +1 \\
    w \cdot x^- + w_0 &= -1 \\
    x^+ &= x^- + 2\gamma \frac{w}{||w||}
\end{align*} \]

What if data are still not linearly separable?

\[ \min_{w, w_0} \|w\|^2 \]

\[ y_i (w \cdot x_i + w_0) \geq 1 \quad \forall i \in \{1, \ldots, N\} \]

- If data are not linearly separable, some points don’t satisfy margin constraint:
  - in case: \( y_i (w \cdot x_i + w_0) < 1 \)
  - or \( 1 - y_i (w \cdot x_i + w_0) > 0 \)
  - How bad is the violation?
  - Tradeoff margin violation with \( \|w\| \):
    \[ \min_{w, w_0} \|w\|^2 + \sum_{i=1}^{N} \epsilon \left( 1 - y_i (w \cdot x_i + w_0) \right)^2 \]
Stochastic gradient descent for SVMs

- Perceptron minimization:
  \[
  \sum_{i=1}^{N} (-y_i (w \cdot x_i + w_0))_+
  \]
- SGD for Perceptron:
  \[
  w^{(t+1)} \leftarrow w^{(t)} + \eta \left[ y_t (w^{(t)} \cdot x_t) \leq 0 \right] y_t x_t
  \]
- SVMs minimization:
  \[
  \|w\|^2 + C \sum_{i=1}^{N} (1 - y_i (w \cdot x_i + w_0))_+
  \]
- SGD for SVMs:
  \[
  w^{(t+1)} \leftarrow w^{(t)} + \eta \left( C \mathbb{1} (1 - y_t (w \cdot x_t + w_0)) y_t x_t - 2 w^{(t)} \right)
  \]
  - pick \(C, \eta\) using CV (or other methods)

Side note: What’s the difference between SVMs and logistic regression?

- SVM:
  \[
  \min_{w, w_0} \|w\|^2 + C \sum_{i=1}^{N} (1 - y_i (w \cdot x_i + w_0))_+
  \]
- Logistic regression:
  \[
  p(y = 1 \mid w, x) = \frac{1}{1 + e^{-(w \cdot x + w_0)}}
  \]
  Log loss:
  \[
  - \sum \log p(y = 1 \mid w, x) = \sum \log \left( 1 + e^{-(w \cdot x + w_0)} \right)
  \]
  leads to sparse solutions
Visualizing the SVM decision boundary with and without kernels

Mixture model example

Training Error: 0.270
Test Error: 0.288
Bayes Error: 0.210

Training Error: 0.26
Test Error: 0.20
Bayes Error: 0.21

From Hastie, Tibshirani, Friedman book

©2017 Emily Fox

CS 446: Machine Learning
Mixture model example – kernels

SVM - Degree-4 Polynomial in Feature Space

SVM - Radial Kernel in Feature Space

From Hastie, Tibshirani, Friedman book

Summary for SVMs
What you need to know...

- Maximizing margin
- Derivation of SVM formulation
- Non-linearly separable case
  - Hinge loss
  - a.k.a. adding slack variables
- SVMs = Perceptron + $L_2$ regularization
- Can optimize SVMs with SGD
  - Many other approaches possible

Clustering:
Grouping
Related Docs
Motivating clustering approaches

Goal: Structure documents by topic

Discover groups (clusters) of related articles

SPORTS  WORLD NEWS
Why might clustering be useful?

I don’t just like sports!

Learn user preferences

Set of clustered documents read by user

Cluster 1

Cluster 2

Cluster 3

Cluster 4

Use feedback to learn user preferences over topics
Clustering: An unsupervised learning task

What if some of the labels are known?
Training set of labeled docs

- SPORTS
- WORLD NEWS
- ENTERTAINMENT
- SCIENCE
Multiclass classification problem

Example of supervised learning

Clustering

No labels provided
...uncover cluster structure from input alone

Input: docs as vectors $x_i$
Output: cluster labels $z_i$

An unsupervised learning task
What defines a cluster?

Cluster defined by center & shape/spread

Assign observation $x_i$ (doc) to cluster $k$ (topic label) if
- Score under cluster $k$ is higher than under others
- For simplicity, often define score as distance to cluster center (ignoring shape)

Hope for unsupervised learning

Easy

Impossible

In between
Other (challenging!) clusters to discover...

![Cluster visualizations](image-url)
**k-means: A clustering algorithm**

**k-means**

Assume

\[-\text{Score} = \text{distance to cluster center}\]

(smaller better)
k-means algorithm

0. **Initialize cluster centers**
   \( \mu_1, \mu_2, \ldots, \mu_k \)

1. **Assign observations to closest cluster center**

   \[ z_i \leftarrow \arg \min_j \| \mu_j - x_i \|_2^2 \]

   Inferred label for obs \( i \), whereas supervised learning has given label \( y_i \)

2. **Revise cluster centers as mean of assigned observations**

   \[ \mu_j \leftarrow \frac{1}{n_j} \sum_{i \in S_j} x_i \]

3. **Repeat 1.+2. until convergence**

   \( (k=3 \text{ clusters}) \)
k-means algorithm

0. Initialize cluster centers
1. Assign observations to closest cluster center
2. Revise cluster centers as mean of assigned observations

\( \mu_j = \frac{1}{n_j} \sum_{i: z_i = j} x_i \)
k-means as coordinate descent

A coordinate descent algorithm

1. Assign observations to closest cluster center
   \[ z_i \leftarrow \arg\min_j \|\mu_j - x_i\|_2^2 \]

2. Revise cluster centers as mean of assigned observations
   \[
   \mu_j = \frac{1}{n_j} \sum_{i: z_i = j} x_i
   \]
   equivalent to
   \[
   \mu_j \leftarrow \arg\min_{\mu} \sum_{i: z_i = j} \|\mu - x_i\|_2^2
   \]
A coordinate descent algorithm

1. Assign observations to closest cluster center
   
   \[ z_i \leftarrow \arg \min_j \| \mu_j - x_i \|_2^2 \]

2. Revise cluster centers as mean of assigned observations
   
   \[ \mu_j \leftarrow \arg \min_\mu \sum_{i: z_i = j} \| \mu - x_i \|_2^2 \]

Alternating minimization

1. (z given µ) and 2. (µ given z)

= coordinate descent
Convergence of k-means

Converges to:

- Global optimum
- Local optimum
- neither

Convergence of k-means to local mode
Convergence of k-means to local mode
Smart initialization with k-means++

k-means++ overview

Initialization of k-means algorithm is critical to quality of local optima found

**Smart initialization:**

1. Choose first cluster center uniformly at random from data points
2. For each obs $\mathbf{x}$, compute distance $d(\mathbf{x})$ to nearest cluster center
3. Choose new cluster center from amongst data points, with probability of $\mathbf{x}$ being chosen proportional to $d(\mathbf{x})^2$
4. Repeat Steps 2 and 3 until k centers have been chosen
k-means++ visualized

k-means++ visualized
k-means++ visualized

k-means++ visualized
k-means++ pros/cons

Computationally costly relative to random initialization, but the subsequent k-means often converges more rapidly

Tends to improve quality of local optimum and lower runtime

Assessing quality of the clustering and choosing the # of clusters
Which clustering do I prefer?

k-means objective

$k$-means is trying to minimize the sum of squared distances:

$$\sum_{j=1}^{k} \sum_{i: z_i = j} ||\mu_j - x_i||^2_2$$
Cluster heterogeneity

Measure of quality of given clustering:
\[
\sum_{j=1}^{k} \sum_{i:z_i=j} ||\mu_j - x_i||^2
\]

Lower is better!

What happens as \( k \) increases?

Can refine clusters more and more to the data \( \rightarrow \) overfitting!

Extreme case of \( k=N \):
- can set each cluster center equal to datapoint
- heterogeneity =

Lowest possible cluster heterogeneity decreases with increasing \( k \)
How to choose $k$?

Summary for k-means
What you can do now...

- Describe potential applications of clustering
- Describe the input (unlabeled observations) and output (labels) of a clustering algorithm
- Determine whether a task is supervised or unsupervised
- Cluster documents using k-means
- Interpret k-means as a coordinate descent algorithm