INTRODUCTION TO MACHINE LEARNING

SUPPORT VECTOR MACHINE

The slides are from Raymond J. Mooney (ML Research Group @ Univ. of Texas)

Mingon Kang, Ph.D.
Department of Computer Science @ UNLV
Linear Separators

- Binary classification can be viewed as the task of separating classes in feature space:

\[ w^T x + b = 0 \]
\[ w^T x + b > 0 \]
\[ w^T x + b < 0 \]

\[ f(x) = \text{sign}(w^T x + b) \]
Linear classifiers: Which Hyperplane?

- Lots of possible choices for \( a, b, c \).
- A Support Vector Machine (SVM) finds an optimal* solution.
  - Maximizes the distance between the hyperplane and the “difficult points” close to decision boundary
  - One intuition: if there are no points near the decision surface, then there are no very uncertain classification decisions

This line represents the decision boundary:
\[
ax + by - c = 0
\]
Support Vector Machine (SVM)

- SVMs maximize the margin around the separating hyperplane.
  - A.k.a. large margin classifiers
- The decision function is fully specified by a subset of training samples, the support vectors.
- Solving SVMs is a quadratic programming problem
Maximum Margin: Formalization

- \( \mathbf{w} \): decision hyperplane normal vector
- \( \mathbf{x}_i \): data point \( i \)
- \( y_i \): class of data point \( i \) (+1 or -1)
- Classifier is: \( f(\mathbf{x}_i) = \text{sign}(\mathbf{w}^\top \mathbf{x}_i + b) \)
- Functional margin of \( \mathbf{x}_i \) is: \( y_i (\mathbf{w}^\top \mathbf{x}_i + b) \)
- The functional margin of a dataset is twice the minimum functional margin for any point
  - The factor of 2 comes from measuring the whole width of the margin
- Problem: we can increase this margin simply by scaling \( \mathbf{w}, b \)....
Distance from example to the separator is \[ r = y \frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|} \]

Examples closest to the hyperplane are support vectors.

Margin \( \rho \) of the separator is the width of separation between support vectors of classes.

Derivation of finding \( r \):
Dotted line \( \mathbf{x}' - \mathbf{x} \) is perpendicular to decision boundary so parallel to \( \mathbf{w} \).
Unit vector is \( \mathbf{w}/|\mathbf{w}| \), so line is \( r\mathbf{w}/|\mathbf{w}| \).
\( \mathbf{x}' = \mathbf{x} - yr\mathbf{w}/|\mathbf{w}| \).
\( \mathbf{x}' \) satisfies \( \mathbf{w}^T \mathbf{x}' + b = 0 \).
So \( \mathbf{w}^T(\mathbf{x} - yr\mathbf{w}/|\mathbf{w}|) + b = 0 \).
Recall that \( |\mathbf{w}| = \sqrt{\mathbf{w}^T \mathbf{w}} \).
So \( \mathbf{w}^T \mathbf{x} - yr|\mathbf{w}| + b = 0 \).
So, solving for \( r \) gives:
\[ r = y(\mathbf{w}^T \mathbf{x} + b)/|\mathbf{w}| \]
Linear SVM Mathematically

The linearly separable case

- Assume that the functional margin of each data item is at least 1, then the following two constraints follow for a training set \( \{(x_i, y_i)\} \)

\[
\begin{align*}
    w^T x_i + b &\geq 1 \quad \text{if } y_i = 1 \\
    w^T x_i + b &\leq -1 \quad \text{if } y_i = -1
\end{align*}
\]

- For support vectors, the inequality becomes an equality

- Then, since each example’s distance from the hyperplane is

\[
r = y \frac{w^T x + b}{\|w\|}
\]

- The functional margin is:

\[
= \frac{2}{\|w\|}
\]
Linear Support Vector Machine (SVM)

- **Hyperplane**
  \[ w^T x + b = 0 \]

- **Extra scale constraint:**
  \[ \min_{i=1,...,n} |w^T x_i + b| = 1 \]

- This implies:
  \[ w^T (x_a - x_b) = 2 \]
  \[ \rho = \|x_a - x_b\|_2 = \frac{2}{\|w\|_2} \]
Worked example: Geometric margin

- Maximum margin weight vector is parallel to line from (1, 1) to (2, 3). So weight vector is (1, 2).
- Decision boundary is normal (“perpendicular”) to it halfway between.
- It passes through (1.5, 2)
- So \( y = x_1 + 2x_2 - 5.5 \)
- Geometric margin is \( \sqrt{5} \)
Worked example: Functional margin

- Let’s minimize $w$ given that $y_i(w^T x_i + b) \geq 1$
- Constraint has $=$ at SVs; $w = (a, 2a)$ for some $a$
- $a + 2a + b = -1$  \hspace{1cm}  $2a + 6a + b = 1$
- So, $a = 2/5$ and $b = -11/5$
- Optimal hyperplane is: $w = (2/5, 4/5)$ and $b = -11/5$
- Margin $\rho$ is $2/|w| = 2/\sqrt{(4/25+16/25)} = 2/(2\sqrt{5}/5) = \sqrt{5}$
Then we can formulate the quadratic optimization problem:

Find \( w \) and \( b \) such that

\[
\frac{2}{\|w\|} = \text{maximized; and for all } \{(x_i, y_i)\}
\]

\[
w^T x_i + b \geq 1 \text{ if } y_i = 1; \quad w^T x_i + b \leq -1 \text{ if } y_i = -1
\]

A better formulation (\( \min \|w\| = \max 1/\|w\| \)):

Find \( w \) and \( b \) such that

\[
\Phi(w) = \frac{1}{2} w^T w \text{ is minimized;}
\]

and for all \( \{(x_i, y_i)\} \):

\[
y_i (w^T x_i + b) \geq 1
\]
Solving the Optimization Problem

Find \( \mathbf{w} \) and \( b \) such that
\[
\Phi(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} \text{ is minimized; }
\]
and for all \( \{(\mathbf{x}_i, y_i)\} \): \( y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \)

- This is now optimizing a quadratic function subject to linear constraints
- Quadratic optimization problems are a well-known class of mathematical programming problem, and many (intricate) algorithms exist for solving them (with many special ones built for SVMs)
- The solution involves constructing a dual problem where a Lagrange multiplier \( \alpha_i \) is associated with every constraint in the primary problem:

Find \( \alpha_1...\alpha_N \) such that
\[
Q(\boldsymbol{\alpha}) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \text{ is maximized and }
\]
(1) \( \sum \alpha_i y_i = 0 \)
(2) \( \alpha_i \geq 0 \) for all \( \alpha_i \)
The Optimization Problem Solution

- The solution has the form:

\[
\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i \quad b = y_k - \mathbf{w}^T \mathbf{x}_k \text{ for any } \mathbf{x}_k \text{ such that } \alpha_k \neq 0
\]

- Each non-zero \( \alpha_i \) indicates that corresponding \( \mathbf{x}_i \) is a support vector.

- Then the classifying function will have the form:

\[
f(\mathbf{x}) = \sum \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b
\]

- Notice that it relies on an inner product between the test point \( \mathbf{x} \) and the support vectors \( \mathbf{x}_i \)
  - We will return to this later.

- Also keep in mind that solving the optimization problem involved computing the inner products \( \mathbf{x}_i^T \mathbf{x}_j \) between all pairs of training points.
Soft Margin Classification

- If the training data is not linearly separable, slack variables $\xi_i$ can be added to allow misclassification of difficult or noisy examples.
- Allow some errors
  - Let some points be moved to where they belong, at a cost
- Still, try to minimize training set errors, and to place hyperplane “far” from each class (large margin)
Soft Margin Classification
Mathematically

- The old formulation:

\[
\text{Find } w \text{ and } b \text{ such that }
\Phi(w) = \frac{1}{2} w^T w \text{ is minimized and for all } \{(x_i, y_i)\}
\]
\[
y_i (w^T x_i + b) \geq 1
\]

- The new formulation incorporating slack variables:

\[
\text{Find } w \text{ and } b \text{ such that }
\Phi(w) = \frac{1}{2} w^T w + C \sum \xi_i \text{ is minimized and for all } \{(x_i, y_i)\}
\]
\[
y_i (w^T x_i + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \text{ for all } i
\]

- Parameter C can be viewed as a way to control overfitting
  - A regularization term
Soft Margin Classification — Solution

- The dual problem for soft margin classification:

  \[ Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \alpha_i \alpha_j y_i y_j x_i^T x_j \]

  is maximized and

  \[ \sum \alpha_i y_i = 0 \]

  \[ 0 \leq \alpha_i \leq C \text{ for all } \alpha_i \]

- Neither slack variables \( \xi_i \) nor their Lagrange multipliers appear in the dual problem!

- Again, \( x_i \) with non-zero \( \alpha_i \) will be support vectors.

- Solution to the dual problem is:

  \[ w = \sum \alpha_i y_i x_i \]

  \[ b = y_k (1 - \xi_k) - w^T x_k \text{ where } k = \arg\max_k \alpha_k' \]

  \[ f(x) = \sum \alpha_i y_i x_i^T x + b \]

  \( w \) is not needed explicitly for classification!
Classification with SVMs

- Given a new point $x$, we can score its projection onto the hyperplane normal:
  - I.e., compute score: $w^T x + b = \Sigma \alpha_i y_i x_i^T x + b$
  - Decide class based on whether $<$ or $> 0$

- Can set confidence threshold $t$.
  - Score $> t$: yes
  - Score $< -t$: no
  - Else: don’t know
Linear SVMs: Summary

- The classifier is a separating hyperplane.

- The most “important” training points are the support vectors; they define the hyperplane.

- Quadratic optimization algorithms can identify which training points $x_i$ are support vectors with non-zero Lagrangian multipliers $\alpha_i$.

- Both in the dual formulation of the problem and in the solution, training points appear only inside inner products:

$$f(x) = \sum \alpha_i y_i x_i^T x + b$$

Find $\alpha_1...\alpha_N$ such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \alpha_i \alpha_j y_i y_j x_i^T x_j$ is maximized and

1. $\sum \alpha_i y_i = 0$
2. $0 \leq \alpha_i \leq C$ for all $\alpha_i$
Non-linear SVMs

Datasets that are linearly separable (with some noise) work out great:

But what are we going to do if the dataset is just too hard?

How about … mapping data to a higher-dimensional space:
Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:

\[ \Phi: \mathbf{x} \rightarrow \varphi(\mathbf{x}) \]
Non-linear SVMs: Feature spaces

Image by MIT OpenCourseWare.
The “Kernel Trick”

- The linear classifier relies on an inner product between vectors $K(x_i, x_j) = x_i^T x_j$
- If every datapoint is mapped into high-dimensional space via some transformation $\Phi: x \rightarrow \phi(x)$, the inner product becomes:
  $$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$
- A kernel function is some function that corresponds to an inner product in some expanded feature space.
- Example:
  2-dimensional vectors $x = [x_1, x_2]$; let $K(x_i, x_i) = (1 + x_i^T x_i)^2$.
  Need to show that $K(x_i, x_i) = \phi(x_i)^T \phi(x_i)$:
  $$K(x_i, x_i) = (1 + x_i^T x_i)^2 = 1 + x_i^2 x_i^2 + 2 x_i^T x_i x_i + x_i^2 x_i^2 + 2 x_i^T x_i x_i + 2 x_i x_i = [1, x_i^2 \sqrt{2} x_i x_i, x_i^2 \sqrt{2} x_i, \sqrt{2} x_i] [1, x_i^2 \sqrt{2} x_i x_i, x_i^2 \sqrt{2} x_i, \sqrt{2} x_i]$$
  $$= \phi(x_i)^T \phi(x_i)$$
  where $\phi(x) = [1, x_1^2 \sqrt{2} x_1 x_2, x_2^2 \sqrt{2} x_1, \sqrt{2} x_2]$
Kernels

- Why use kernels?
  - Make non-separable problem separable.
  - Map data into better representational space

- Common kernels
  - Linear
  - Polynomial $K(x,z) = (1 + x^T z)^d$
    - Gives feature conjunctions
  - Radial basis function (infinite dimensional space)

$$K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2}$$