Support Vector Machines (SVM)

Lecture-11

Slides Credits
Saad Ali, Andrew Zisserman, Guo-Jun QI, many more
Application

- Pattern recognition
- Object classification/detection
Usage

- The classifier must be trained using a set of negative and positive examples.

- The classifier “learns” the regularities in the data

- After training classifier is capable of classifying an unknown example with a high degree of accuracy.
Human Detection

Navneet Dalal and Bill Triggs “Histograms of Oriented Gradients for Human Detection” CVPR05
Blocks, Cells

- 16x16 blocks of 50% overlap.
  - 7x15=105 blocks in total

- Each block should consist of 2x2 cells with size 8x8.
Votes

- Each block consists of 2x2 cells with size 8x8
- Quantize the gradient orientation into 9 bins (0-180)
  - The vote is the gradient magnitude
  - Interpolate votes linearly between neighboring bin centers.
    - Example: if $\theta=85$ degrees.
    - Distance to the bin center Bin 70 and Bin 90 are 15 and 5 degrees, respectively.
    - Hence, ratios are $5/20=1/4$, $15/20=3/4$.
  - The vote can also be weighted with Gaussian to down weight the pixels near the edges of the block.
Final Feature Vector

- Concatenate histograms
  - Make it a 1D vector of length 3780.

Visualization
Application: Pedestrian detection in Computer Vision

Objective: detect (localize) standing humans in an image
• cf face detection with a sliding window classifier

Method: the HOG detector

Slides Credits
Andrew Zisserman
Training data and features
Feature: histogram of oriented gradients (HOG)
Averaged positive examples

Slides Credits
Andrew Zisserman
Algorithm

Training (Learning)

- Represent each example window by a HOG feature vector

\[ x_i \in \mathbb{R}^d, \]

- Train a SVM classifier

Testing (Detection)

- Sliding window classifier

\[ f(x) = w^T x + b \]
Linear Classifiers

\[ f(x, w, b) = \text{sign}(w \cdot x + b) \]

How would you classify this data?

Label \( y \):
- \textbf{denotes +1}
- \textbf{denotes -1}

Parameters:

Slide Credits: Guo-Jun QI
Linear Classifiers

\[ f(x, w, b) = \text{sign}(w x + b) \]

How would you classify this data?
Linear Classifiers

\[ f(x, w, b) = \text{sign}(w x + b) \]

- denotes +1
- denotes -1

How would you classify this data?

Parameters

Slide Credits: Guo-Jun QI
Linear Classifiers

\[ f(x, w, b) = \text{sign}(w x + b) \]

\( \cdot \) denotes +1
\( \circ \) denotes -1

Any of these would be fine..

..but which is the best?

Slide Credits: Guo-Jun QI
Linear Classifiers

\[ f(x, w, b) = \text{sign}(w \cdot x + b) \]

How would you classify this data?

Parameters

\[ x \rightarrow f \rightarrow y_{\text{est}} \]

- denotes +1
- denotes -1

Misclassified to +1 class

Slide Credits: Guo-Jun Qi
Define the **margin** of a linear classifier as the width that the boundary could be increased by before hitting a data point.

\[
f(x, w, b) = \text{sign}(w x + b)
\]
Maximum Margin

The maximum margin linear classifier is the linear classifier with the maximum margin.

1. Maximizing the margin makes sense according to intuition
2. Implies that only support vectors are important; other training examples can be discarded without affecting the training result.

Support Vectors are those datapoints that the margin pushes up against

• denotes +1
○ denotes -1

Parameters

Linear SVM

The maximum margin linear classifier is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Slide Credits: Guo-Jun QI
**Maximum Margin**

\[ f(x, w, b) = \text{sign}(w \cdot x + b) \]

- \( x \) denotes +1
- \( y \) denotes -1

Keeping only support vectors will not change the maximum margin classifier.

- Robust to the small changes (noise) in non-support vectors

**Linear SVM**

Slide Credits: Guo-Jun Qi
Linear Classifier

- Binary classifier $\rightarrow$ 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 Classifier

- Finding perpendicular to the plane (line)

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

\[ W^T X' + b = 0 \]
\[ W^T X'' + b = 0 \]
\[ (W^T X' + b) - (W^T X'' + b) = 0 \]
\[ W^T X' + b - W^T X'' - b = 0 \]
\[ W^T X' - W^T X'' = 0 \]
\[ W^T (X' - X'') = 0 \]

\( W \) is perpendicular to line (hyper plane)
Linear Classifier

- Binary classifier → Task of separating classes in feature space
- Distance from a point to a line (hyper plane)

\[
\mathbf{w}^T \mathbf{x} + b = 1
\]

\[
\mathbf{w}^T \mathbf{x} + b = 0
\]

Unit vector

\[
\hat{w} = \frac{\mathbf{w}}{||\mathbf{w}||}
\]

\[
|\hat{W}(X - X_n)| = \frac{\mathbf{w}}{||\mathbf{w}||}(X - X_n)
\]

\[
|\hat{W}(X - X_n)| = \frac{1}{||\mathbf{w}||}(W.X - W.X_n)
\]

\[
|\hat{W}(X - X_n)| = \frac{1}{||\mathbf{w}||}(W.X + b - W.X_n - b)
\]

\[
|\hat{W}(X - X_n)| = \frac{1}{||\mathbf{w}||}
\]
Margin

Margin $2\gamma$ of the separator is the width of separation between classes.

\[ w^T x + b = 1 \]

\[ w^T x + b = -1 \]

\[ \hat{w}(X_+ - X_-) = \frac{WX_+ - WX_-}{\|W\|} \]

\[ \hat{w}(X_+ - X_-) = \frac{WX_+ + b - WX_- - b}{\|W\|} \]

\[ \hat{w}(X_+ - X_-) = \frac{WX_+ + b - (WX_- + b)}{\|W\|} \]

\[ \hat{w}(X_+ - X_-) = \frac{2}{\|W\|} \]
Linear SVM Mathematically

- **Goal:** 1) Correctly classify all training data
  
  \[ \begin{align*}
  w x_i + b & \geq 1 \quad \text{if } y_i = +1 \\
  w x_i + b & \leq -1 \quad \text{if } y_i = -1 \\
  y_i (w x_i + b) & \geq 1 \quad \text{for all } i
  \end{align*} \]

  2) Maximize the Margin

  \[ M = \frac{2}{w^t w} \]

- We can formulate a Quadratic Optimization Problem and solve for \( w \) and \( b \)

- **Minimize**

  \[ \Phi(w) = \frac{1}{2} w^t w \]

  subject to

  \[ y_i (w x_i + b) \geq 1 \quad \forall i \]
Linear SVM

- Now we can formulate the quadratic optimization problem as

Given a linearly separable training sample $S = ((x_1,y_1),...,(x_l,y_l))$, the hyperplane $(\mathbf{w},b)$ that solves the optimization problem

- minimizes $\langle \mathbf{w}^* \mathbf{w} \rangle$
- subject to $y_i(\langle \mathbf{w} \cdot x_i \rangle + b) \geq 1 \quad i = 1, ..., l$
Linear SVM

minimizes $\langle \mathbf{w}^* \mathbf{w} \rangle$
subject to $y_i (\langle \mathbf{w}^* \mathbf{x}_i \rangle + b) \geq 1 \quad i = 1, \ldots, l$

Convert the problem from the primal form into the dual form

$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \langle \mathbf{w}^* \mathbf{w} \rangle - \sum_{i=1}^{l} \alpha_i [y_i (\langle \mathbf{w}^* \mathbf{x}_i \rangle + b) - 1]$

\[
\frac{\partial L(\mathbf{w}^*, \alpha^*, \beta^*)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i = 0
\]

\[
\frac{\partial L(\mathbf{w}^*, \alpha^*, \beta^*)}{\partial b} = \sum_{i=1}^{l} y_i \alpha_i = 0
\]

\[
\mathbf{w} = \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i
\]

\[
0 = \sum_{i=1}^{l} y_i \alpha_i
\]
Linear SVM

minimizes $\langle w^* w \rangle$
subject to $y_i(\langle w^* x_i \rangle + b) \geq 1$ $i = 1, ..., l$
Linear SVM

Now we plug the newly defined $w$ into the $L(w,b,\alpha)$

$L(w,b,\alpha) = \frac{1}{2} \langle w \rangle - \sum_{i=1}^{l} \alpha_i [y_i (\langle w \rangle x_i + b) - 1] =

w = \sum_{i=1}^{l} y_i \alpha_i x_i$
Linear SVM

- The dual form of the original problem is

\[
\text{maximize } W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \mathbf{x}_j \rangle \\
\text{subject to } \sum_{i=1}^{l} y_i \alpha_i = 0, \quad \alpha_i \geq 0 \quad i = 1, \ldots, l.
\]

The optimal weight vector given by:

\[
\mathbf{w}^* = \sum_{i=1}^{l} y_i \alpha_i^* \mathbf{x}_i
\]

realizes the maximal margin hyperplane with the geometric margin given by:

\[
\gamma = \frac{1}{\| \mathbf{w} \|_2}
\]
Solving the Optimization Problem

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

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

(1) $\sum a_i y_i = 0$

(2) $\alpha_i \geq 0$ for all $\alpha_i$
Finding $b$

\[ y_k (W^T X_k + b) = 1 \]
\[ (W^T X_k + b) = \frac{1}{y_k} \]
\[ (W^T X_k + b) = y_k \]
\[ b = y_k - W^T X_k \]
The Optimization Problem Solution

- The solution has the form:

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

- \( \alpha_i \) must satisfy Karush-Kuhn-Tucker conditions:
  \[ \alpha_i \left[ y_i (w^T x_i + b) - 1 \right] = 0, \text{ for any } i \]
  - If \( \alpha_i > 0, \quad y_i (w^T x_i + b) - 1 = 0, \) \( x_i \) is on the margin
  - If \( y_i (w^T x_i + b) > 1, \) \( \alpha_i = 0 \)
  - Each non-zero \( \alpha_i \) indicates that corresponding \( x_i \) is a support vector.
Maximum Margin

- denotes +1
- denotes -1

\[ y_i(w^T x_k + b) - 1 = 0 \]

\( w, b \) depends only on Support Vectors via active constraints

Slide Credits: Guo-Jun QI
The Optimization Problem Solution

To classify the new test point \( x \), we use

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

Find \( \alpha_1 \ldots \alpha_N \) 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. \( \alpha_i \geq 0 \) for all \( \alpha_i \)
Dataset with noise

- **Hard Margin**: So far, all data points are classified correctly
  - No training error
- **What if the training set is noisy?**
Slack variables $\xi_i$ to allow misclassification:

\[ y_i (w^T x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0 \]

What should our quadratic optimization criterion be?

We expect $\xi_i$ to be small.

\[ \Phi(w) = \frac{1}{2} w^T w + C \sum \xi_i \]
Hard Margin v.s. Soft Margin

- The old formulation:

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

- The new formulation incorporating slack variables:

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

- Similar solution can be obtained to that of hard margin
- Parameter C can be viewed as a way to control overfitting.
• data is linearly separable
• but only with a narrow margin
C = Infinity  hard margin
C = 10  soft margin

Slide Credit Andrew Zisserman
XOR Problem

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$Y = X_1 \oplus X_2$

Not linearly separable

XOR data are not linearly separable

Mapping \((x_1, x_2)\) to \((x_1, x_1 \cdot x_2)\)
Non-linear SVMs: Feature spaces

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

\[ \Phi: x \rightarrow \phi(x) \]
Non-linear SVMs

- If every data point is mapped into high-dimensional space via some transformation $\Phi$: $x \rightarrow \phi(x)$, optimization problem is similar:

  Find $\alpha_1...\alpha_N$ such that
  \[ Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \] is maximized
  \[ (1) \quad \sum \alpha_i y_i = 0 \]
  \[ (2) \quad \alpha_i \geq 0 \text{ for all } \alpha_i \]

- Classifying function is:

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

- But relies on inner product $\phi(x_i)^T \phi(x)$

Slide Credits: Guo-Jun QI
The “Kernel Trick”

- SVM relies on
  - Linear: $K(x_i,x_j) = x_i^T x_j$
  - Non-linear: $K(x_i,x_j) = \varphi(x_i)^T \varphi(x_j)$
- Feature mapping is time-consuming.
- Use a kernel function that directly obtains the value of inner product
- Feature mapping $\varphi$ is not necessary in this case.

- Example:
  2-dimensional vectors $x = [x_1 \ x_2]$; let $K(x_i,x_j) = (1 + x_i^T x_j)^2$
  It is inner product of $\varphi(x) = [1 \ x_i^2 \ \sqrt{2} x_i x_2 \ x_2^2 \ \sqrt{2} x_i \ \sqrt{2} x_2]$
  Verify: $K(x_i,x_j) = (1 + x_i^T x_j)^2$
    $= 1 + x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 + 2 x_{i1} x_{j1} + 2 x_{i2} x_{j2}$
    $= [1 \ x_{i1}^2 \ \sqrt{2} x_{i1} x_{i2} \ x_{i2}^2 \ \sqrt{2} x_{i1} \ \sqrt{2} x_{i2}]^T [1 \ x_{j1}^2 \ \sqrt{2} x_{j1} x_{j2} \ x_{j2}^2 \ \sqrt{2} x_{j1} \ \sqrt{2} x_{j2}]$
    $= \varphi(x_i)^T \varphi(x_j)$,
\[ K(x_i, x_j) = (1 + x_i \cdot x_j)^2 \]
\[ = (1 + (x_{i1}, x_{i2}) \cdot (x_{j1}, x_{j2}))^2 \]
\[ = (1 + x_{i1}x_{j1} + x_{i2}x_{j2})^2 \]
\[ = 1 + 2(x_{i1}x_{j1} + x_{i2}x_{j2}) + (x_{i1}x_{j1} + x_{i2}x_{j2})^2 \]
\[ = 1 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + x_{i1}^2x_{j1}^2 + x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} \]
Examples of Kernel Functions

- Linear: $K(x_i, x_j) = x_i^T x_j$

- Polynomial of power $p$: $K(x_i, x_j) = (1 + x_i^T x_j)^p$

- Gaussian (radial-basis function network): 
  $$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

- Sigmoid: $K(x_i, x_j) = \tanh(\beta_0 x_i^T x_j + \beta_1)$

Slide Credits: Guo-Jun QI
Non-linear SVMs Mathematically

- Dual problem formulation:

Find $\alpha_1 \ldots \alpha_N$ such that

$$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

is maximized and

1. $\sum \alpha_i y_i = 0$
2. $\alpha_i \geq 0$ for all $\alpha_i$

- The solution is:

$$f(x) = \sum \alpha_i y_i K(x_i, x_j) + b$$

- Optimization techniques for finding $\alpha_i$’s remain the same!

Slide Credits: Guo-Jun QI
The feature is mapped to a high dimensional space where
- training data are separable.
- Inner product is computed by kernel function.
- Optimization problem is similar to linear SVM
References


Christopher J. C. Burges: A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Bernhard Scholkopf and A. J. Smola: Learning with Kernels. 2002

A useful website: www.kernel-machines.org

Software:

LIBSVM: www.csie.ntu.edu.tw/~cjlin/libsvm/
SVMLight: svmlight.joachims.org/
Other Links

- http://www.robots.ox.ac.uk/~az/lectures/ml/
- Andrew Ng. Part V. Support Vector Machines (note)