Supper Vector Machines

Chen Yu
Indiana University

Adapted from the slides by Martin Law

Linear Classifier

• How to classify the data? Which is the best?

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

Maximum Margin

• Margin: the width that the boundary could be increased by before hitting a data point.

Support Vectors

• Maximum margin linear classifier
• Support vectors are those datapoints that the margin pushes up against.
Why Maximum Margin

• It feels safe.
• Empirically it works well.
• If the location of the boundary is not perfect due to noise, this gives us the least chance of misclassification.
• Not sensitive to removal of any non support vector datapoints.
• Some theory called VC dimension.

Specifying a line and margin

Estimate the Margin

• What is the distance expression for a point \( x \) to a line \( wx + b = 0 \)?

\[
d(x) = \frac{|x \cdot w + b|}{\sqrt{w^Tw}} = \frac{|x \cdot w + b|}{\sqrt{\sum_{i=1}^{d} w_i^2}}
\]

• Let \( \{x_1, ..., x_n\} \) be our data set and let \( y_i \in \{1,-1\} \) be the class label of \( x_i \).

• The decision boundary should classify all points correctly \( \Rightarrow y_i(w^T x_i + b) \geq 1, \quad \forall i \)

• The decision boundary can be found by solving the following constrained optimization problem

\[
\text{Minimize } \frac{1}{2}||w||^2 \\
\text{subject to } y_i(w^T x_i + b) \geq 1 \quad \forall i
\]

• This is a constrained optimization problem.
The Primal problem

Minimize \( \frac{1}{2} ||w||^2 \)
subject to \( 1 - y_i (w^T x_i + b) \leq 0 \) for \( i = 1, \ldots, n \)

- The Lagrange function is
  \[
  \mathcal{L} = \frac{1}{2} w^T w + \sum_{i=1}^{n} \alpha_i \left( 1 - y_i (w^T x_i + b) \right)
  \]
  - Note that \( ||w||^2 = w^T w \)
- Setting the gradient of \( \mathcal{L} \) w.r.t. \( w \) and \( b \) to zero, we have
  \[
  w + \sum_{i=1}^{n} \alpha_i (-y_i) x_i = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{n} \alpha_i y_i x_i
  \]
  \[
  \sum_{i=1}^{n} \alpha_i y_i = 0
  \]

- If we substitute \( w = \sum_{i=1}^{n} \alpha_i y_i x_i \) to \( \mathcal{L} \), we have
  \[
  \mathcal{L} = \frac{1}{2} \left( \sum_{i=1}^{n} \alpha_i y_i x_i + b \right)^2 + \sum_{i=1}^{n} \alpha_i \left( 1 - y_i \left( \sum_{j=1}^{n} \alpha_j y_j x_i + b \right) \right)
  \]
  \[
  = \frac{1}{2} \sum_{i=1}^{n} \alpha_i y_i x_i + b \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{n} \alpha_j y_j x_i - b \sum_{i=1}^{n} \alpha_i y_i
  \]
  - The constraint \( \sum_{i=1}^{n} \alpha_i y_i = 0 \)
  - This is a function of \( \alpha_i \) only

The Dual Problem

- The new objective function is in terms of \( \alpha_i \) only
- It is known as the dual problem: if the primal problem has an optimal solution, the dual problem also has an optimal solution, and the corresponding optimal values are the same.
- The objective function of the dual problem needs to be maximized!
  \[
  \max. \quad W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j
  \]
  - With the constraints
  subject to \( \alpha_i \geq 0 \), \( \sum_{i=1}^{n} \alpha_i y_i = 0 \)

\[
\max. \quad W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j
\]
subject to \( \alpha_i \geq 0 \), \( \sum_{i=1}^{n} \alpha_i y_i = 0 \)

- This is a quadratic programming (QP) problem
  - A global maximum of \( \alpha_i \) can always be found
- \( w \) can be recovered by
  \[
  w = \sum_{i=1}^{n} \alpha_i y_i x_i
  \]
- KTT condition:
  \[
  \alpha_i [y_i (w^T x_i + b) - 1] = 0 \text{ for all } i
  \]
  \[
  x_i \text{ with } \alpha_i > 0 \text{ are support vectors}
  \]
Solution

- Many of the $\alpha_i$ are zero
  - $w$ is a linear combination of a small number of data points
- $\mathbf{x}_i$ with non-zero $\alpha_i$ are called support vectors (SVs)
  - The decision boundary is determined only by the SVs
  - Let $t_j (j=1, ..., s)$ be the indices of the $s$ support vectors. We can write $w = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$
- For testing with a new data $\mathbf{z}$
  - Compute $w^T \mathbf{z} + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} (\mathbf{x}_{t_j}^T \mathbf{z}) + b$ and classify $\mathbf{z}$ as class 1 if the sum is positive, and class 2 otherwise
  - Note: $w$ need not be formed explicitly

Example

Non-Separable Case

- We allow “error” $\xi_i$ in classification; it is based on the output of the discriminant function $w^T \mathbf{x} + b$

A new optimization problem

- If we minimize $\sum_i \xi_i$, $\xi_i$ can be computed by

\[
\begin{cases}
    w^T \mathbf{x}_i + b \geq 1 - \xi_i & y_i = 1 \\
    w^T \mathbf{x}_i + b \leq -1 + \xi_i & y_i = -1 \\
    \xi_i \geq 0 & \forall i
\end{cases}
\]

- $\xi_i$ are “slack variables” in optimization
- Note that $\xi_i = 0$ if there is no error for $\mathbf{x}_i$
- We want to minimize $\frac{1}{2} ||w||^2 + C \sum_i \xi_i$
  - $C$: tradeoff parameter between error and margin
- The optimization problem becomes

Minimize $\frac{1}{2} ||w||^2 + C \sum_i \xi_i$

subject to $y_i (w^T \mathbf{x}_i + b) \geq 1 - \xi_i$, $\xi_i \geq 0$
The dual of this new constrained optimization problem is

\[
W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j
\]

subject to \( C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0 \)

Once again, a QP solver can be used to find \( \alpha_i \)
\( w \) is recovered as \( w = \sum_{j=1}^{n} \alpha_j y_j x_j \)
This is very similar to the optimization problem in the linear separable case, except that there is an upper bound \( C \) on \( \alpha_i \).

Non-Linear Classification

So far, we have only considered large-margin classifiers with a linear decision boundary

How to generalize it to become nonlinear?
Key idea: transform \( x \) to a higher dimensional space.

- Input space: the space the point \( x \) are located
- Higher-dimension space: the space of \( \phi(x) \) after transformation

Why transform?
- Linear operation in the higher-dimensional space is equivalent to non-linear operation in a lower-dimensional space
- Classification can become easier with a proper transformation.

Kernel Trick

The SVM optimization problem

\[
W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j
\]

subject to \( C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0 \)

The data points only appear as inner product
As long as we can calculate the inner product in the transformed space, we do not really need the mapping of data points.
Define the kernel function \( K \) by

\[
K(x_i, x_j) = \phi(x_i)^T \phi(x_j)
\]

• The SVM optimization problem

\[
W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j
\]

subject to \( C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0 \)

• The data points only appear as inner product

As long as we can calculate the inner product in the transformed space, we do not really need the mapping of data points.

• Define the kernel function \( K \) by

\[
K(x_i, x_j) = \phi(x_i)^T \phi(x_j)
\]
Suppose $\phi(.)$ is given as follows:

$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

An inner product in the feature space is

$$\langle \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right), \phi\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right)\rangle = (1 + x_1y_1 + x_2y_2)^2$$

So, if we define the kernel function as follows, there is no need to carry out $\phi(.)$ explicitly

$$K(x, y) = (1 + x_1y_1 + x_2y_2)^2$$

This use of kernel function to avoid carrying out $\phi(.)$ explicitly is known as the **kernel trick**.

**New Problem**
- Change all inner products to kernel functions
- For training,
  $$\max_{\alpha} \ W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$
  subject to $C \geq \alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i y_i = 0$

  With kernel function
  $$\max_{\alpha} \ W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$
  subject to $C \geq \alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i y_i = 0$

**Testing**
- For testing, the new data $z$ is classified as class 1 if $f \geq 0$, and as class 2 if $f < 0$

  Original
  $$w = \sum_{j=1}^{s} \alpha_j y_j x_j$$
  $$f = w^T z + b = \sum_{j=1}^{s} \alpha_j y_j x_j^T z + b$$

  With kernel function
  $$w = \sum_{j=1}^{s} \alpha_j y_j \phi(x_j)$$
  $$f = \langle w, \phi(z) \rangle + b = \sum_{j=1}^{s} \alpha_j y_j K(x_j, z) + b$$

- Since the training of SVM only requires the value of $K(x_i, x_j)$, there is no restriction of the form of $x_i$ and $x_j$.
- $K(x_i, x_j)$ is just a similarity measure comparing $x_i$ and $x_j$.
- For a test object $z$, the discriminant function essentially is a weighted sum of the similarity between $z$ and a pre-selected set of the support vectors.
  $$f(z) = \sum_{x_i \in \mathcal{S}} \alpha_i y_i K(z, x_i) + b$$
  $\mathcal{S}$ : the set of support vectors
Example

- Suppose we have 5 1D data points
  - \(x_1=1, x_2=2, x_3=4, x_4=5, x_5=6\), with 1, 2, 6 as class 1 and 4, 5 as class 2 \(\Rightarrow y_1=1, y_2=1, y_3=-1, y_4=-1, y_5=1\)
- We use the polynomial kernel of degree 2
  - \(K(x,y) = (xy+1)^2\)
  - \(C\) is set to 100
- We first find \(\alpha_i\) (\(i=1, \ldots, 5\)) by
  subject to \(100 \geq \alpha_i \geq 0, \sum_{i=1}^{5} \alpha_i y_i = 0\)

  By using a QP solver, we get
  - \(\alpha_1=0, \alpha_2=2.5, \alpha_3=0, \alpha_4=7.333, \alpha_5=4.833\)
  - Note that the constraints are indeed satisfied
  - The support vectors are \((x_2=2, x_4=5, x_5=6)\)
- The discriminant function is
  \[
  f(z) = 2.5(1)(2z+1)^2 + 7.333(-1)(5z+1)^2 + 4.833(1)(6z+1)^2 + b
  \]
  \[
  = 0.6667z^2 - 5.333z + 9
  \]

Summary

- SVM is useful.
- Two key concepts of SVM: maximization of the margin and the kernel trick
- A lesson learnt in SVM: a linear algorithm in a HD space is equivalent to a non-linear algorithm in a LD space
- Standard linear algorithms can be generalized to its non-linear version by going to a HD space. E.g. Kernel principal component analysis.
- Many SVM implementations are available on the web for you to try on your data set.