DATA MINING 2
Support Vector Machine

Riccardo Guidotti

a.a. 2022/2023

Slides edited from Tan, Steinbach, Kumar, Introduction to Data Mining
Maximum Margin Hyperplanes

• Find a linear hyperplane (decision boundary) that separates the data.
Maximum Margin Hyperplanes

- One possible solution.
Maximum Margin Hyperplanes

- Another possible solution.
Maximum Margin Hyperplanes

• Other possible solutions.
Maximum Margin Hyperplanes

• Let’s focus on $B_1$ and $B_2$.
• Which one is better?
• How do you define better?
Maximum Margin Hyperplanes

• The best solution is the hyperplane that \textbf{maximizes} the \textit{margin}.
• Thus, $B_1$ is better than $B_2$. 
Linear SVM: Separable Case

• A linear SVM is a classifier that searches for a hyperplane with the largest margin (a.k.a. maximal margin classifier).

• $w$ and $b$ have to be learned.

• Given $w$ and $b$ the classifiers work as

$$f(\vec{x}) = \begin{cases} 
1 & \text{if } \vec{w} \cdot \vec{x} + b \geq 1 \\
-1 & \text{if } \vec{w} \cdot \vec{x} + b \leq -1
\end{cases}$$

Example calculus dot product

$$w = [0.3 \ 0.2] \quad x = [1 \ 2] \quad b = -2$$

$$w \cdot x + b = 0.3*1 + 0.2*2 + (-2) = -1.3$$
Linear SVM: Separable Case

What is the distance expression for a point $x$ to a line $wx + b = 0$ (the decision boundary)?

$$d(x) = \frac{|x \cdot w + b|}{\sqrt{\|w\|^2}} = \frac{|x \cdot w + b|}{\sqrt{\sum_{i=1}^{d} w_i^2}}$$
Linear SVM: Separable Case

- The distance between $B_1$ and $b_{11}$ is $1/\|w\|$
- The distance between $b_{11}$ and $b_{12}$, i.e., the margin is
  \[
  \text{Margin} = \frac{2}{\|\vec{w}\|}
  \]
- In order to \textbf{maximize the margin} we need to minimize $\|w\|$
Learning a Linear SVM

- Learning the SVM model is equivalent to determining \( w \) and \( b \).
- How to find \( w \) and \( b \)?
- Objective is to **maximize the margin**.
- Which is equivalent to minimize
- Subject to to the following constraints
- This is a constrained optimization problem that can be solved using the *Lagrange* multiplier method.
- Introduce Lagrange multiplier \( \lambda \) (or \( \alpha \))

\[
\text{Margin} = \frac{2}{||\vec{w}||} \\
L(\vec{w}) = \frac{||\vec{w}||^2}{2} \\
y_i = \begin{cases} 
1 & \text{if } \vec{w} \cdot \vec{x}_i + b \geq 1 \\
-1 & \text{if } \vec{w} \cdot \vec{x}_i + b \leq -1 
\end{cases} \\
y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \ i = 1,2,...,N
\]
Constrained Optimization Problem

Minimize $\|\mathbf{w}\| = \langle \mathbf{w} \cdot \mathbf{w} \rangle$ subject to $y_i((\mathbf{x}_i \cdot \mathbf{w}) + b) \geq 1$ for all $i$

Lagrangian method: maximize $\inf_{\mathbf{w}} L(\mathbf{w}, b, \alpha)$, where

$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\| - \sum_i \alpha_i [(y_i(\mathbf{x}_i \cdot \mathbf{w}) + b) - 1]$

At the extremum, the partial derivative of $L$ with respect to both $\mathbf{w}$ and $b$ must be 0. Taking the derivatives, setting them to 0, substituting back into $L$, and simplifying yields:

Maximize $\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$

subject to $\sum_i y_i \alpha_i = 0$ and $\alpha_i \geq 0$
A Geometrical Interpretation

Implies that only support vectors matter; other training examples are ignorable.
Example of Linear SVM

![Graph showing the linear SVM decision boundary with support vectors highlighted.]

The equation of the decision boundary is:

\[-6.6431x_1 - 9.3232x_2 + 7.9327 = 0\]

The table below lists the support vectors and their corresponding values of \(\lambda\):

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>y</th>
<th>(\lambda)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3858</td>
<td>0.4687</td>
<td>1</td>
<td>65.5261</td>
</tr>
<tr>
<td>0.4871</td>
<td>0.611</td>
<td>-1</td>
<td>65.5261</td>
</tr>
<tr>
<td>0.9218</td>
<td>0.4103</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>0.7382</td>
<td>0.8936</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>0.1763</td>
<td>0.0579</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.4057</td>
<td>0.3529</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.9355</td>
<td>0.8132</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>0.2146</td>
<td>0.0099</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\(\lambda = \alpha\)
Linear SVM: Non-separable Case

- What if the problem is not linearly separable?
- We must allow for errors in our solution.
Slack Variables

- The inequality constraints must be relaxed to accommodate the nonlinearly separable data.
- This is done introducing slack variables $\xi (x_i)$ into the constrains of the optimization problem.
- $\xi$ provides an estimate of the error of the decision boundary on the misclassified training examples.
Learning a Non-separable Linear SVM

- Objective is to minimize
- Subject to to the constraints
- where $C$ and $k$ are user-specified parameters representing the penalty of misclassifying the training instances
- Lagrangian multipliers are constrained to $0 \leq \lambda \leq C$.

$$L(w) = \frac{||w||^2}{2} + C\left(\sum_{i=1}^{N} \xi_i\right)$$

$$y_i = \begin{cases} 1 & \text{if } \vec{w} \cdot \vec{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \vec{w} \cdot \vec{x}_i + b \leq -1 + \xi_i \end{cases}$$

Non-linear SVM

• What if the decision boundary is not linear?

• How about... mapping data to a higher-dimensional space:
Non-linear SVMs: Feature Spaces

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

\[ \Phi : \mathbb{R}^2 \to \mathbb{R}^3 \]
\[ (x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2} x_1 x_2, x_2^2) \]
Non-linear SVM

• The trick is to transform the data from its original space $x$ into a new space $\Phi(x)$ (phi) so that a linear decision boundary can be used.

\[
x_1^2 - x_1 + x_2^2 - x_2 = -0.46. \]
\[
\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1).
\]
\[
w_4x_1^2 + w_3x_2^2 + w_2\sqrt{2}x_1 + w_1\sqrt{2}x_2 + w_0 = 0.
\]

• Decision boundary $\vec{w} \cdot \Phi(\vec{x}) + b = 0$
Learning a Nonlinear SVM

• Optimization problem

\[
\begin{align*}
\min_w & \quad \frac{||w||^2}{2} \\
\text{subject to} & \quad y_i(w \cdot \Phi(x_i) + b) \geq 1, \forall \{(x_i, y_i)\}
\end{align*}
\]

• Which leads to the same set of equations but involve \(\Phi(x)\) instead of \(x\).

\[
f(z) = \text{sign}(w \cdot \Phi(z) + b) = \text{sign}\left(\sum_{i=1}^{n} \lambda_i y_i \Phi(x_i) \cdot \Phi(z) + b\right).
\]

Issues:

• What type of mapping function \(\Phi\) should be used?

• How to do the computation in high dimensional space?
  • Most computations involve dot product \(\Phi(x) \cdot \Phi(x)\)
  • Curse of dimensionality?
The Kernel Trick

- $\Phi(x) \cdot \Phi(x) = K(x_i, x_j)$
- $K(x_i, x_j)$ is a kernel function (expressed in terms of the coordinates in the original space)
- Examples:
  
  $K(x, y) = (x^T y + 1)^d$
  $K(x, y) = \exp(-\|x - y\|^2/(2\sigma^2))$
  $K(x, y) = \tanh(\kappa x^T y + \theta)$

Examples of Kernel Functions

• Polynomial kernel with degree \( d \)
  \[ K(x, y) = (x^T y + 1)^d \]

• Radial basis function kernel with width \( \sigma \)
  \[ K(x, y) = \exp(-\|x - y\|^2/(2\sigma^2)) \]
  • Closely related to radial basis function neural networks
  • The feature space is infinite-dimensional

• Sigmoid with parameter \( \kappa \) and \( \theta \)
  \[ K(x, y) = \tanh(\kappa x^T y + \theta) \]
  • It does not satisfy the Mercer condition on all \( \kappa \) and \( \theta \)

• Choosing the Kernel Function is probably the most tricky part of using SVM.
The Kernel Trick

• The linear classifier relies on 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 $\phi$: $x \to \phi(x)$, the inner product becomes:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

• A kernel function is a function that is equivalent to an inner product in some feature space.

• Example:

2-dimensional vectors $x = [x_1, x_2]$; let $K(x_i, x_j) = (1 + x_i^T x_j)^2$

Need to show that $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$:

$$K(x_i, x_j) = (1 + x_i^T x_j)^2 = 1 + x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 + 2 x_{i1} x_{j1} + 2 x_{i2} x_{j2} =$$

$$= [1 \ x_{i1}^2 \ \sqrt{2} x_{i1} x_{i2} \ \sqrt{2} x_{i1} x_{j1} \ x_{i2}^2 \ \sqrt{2} x_{i1} x_{j2} \ \sqrt{2} x_{i2} x_{j1} \ \sqrt{2} x_{i2} x_{j2}]^T [1 \ x_{j1}^2 \ \sqrt{2} x_{j1} x_{j2} \ x_{j2}^2 \ \sqrt{2} x_{j1} \ \sqrt{2} x_{j2}] =$$

$$= \phi(x_i)^T \phi(x_j), \text{ where } \phi(x) = [1, \ x_1^2, \ \sqrt{2} x_1 x_2, \ x_2^2, \ \sqrt{2} x_1, \ \sqrt{2} x_2]$$

• Thus, a kernel function implicitly maps data to a high-dimensional space (without the need to compute each $\phi(x)$ explicitly).
The Kernel Trick

Advantages of using kernel:

• Don’t have to know the mapping function $\Phi$.

• Computing dot product $\Phi(x) \cdot \Phi(y)$ in the original space avoids curse of dimensionality.

Not all functions can be kernels

• Must make sure there is a corresponding $\Phi$ in some high-dimensional space.

• Mercer’s theorem (see textbook) that ensures that the kernel functions can always be expressed as the dot product in some high dimensional space.

\[ f(z) = \text{sign}(w \cdot \Phi(z) + b) = \text{sign}(\sum_{i=1}^{n} \lambda_i y_i K(x_i, z) + b). \]
Constrained Optimization Problem with Kernel

Minimize $\|w\| = \langle w \cdot w \rangle$ subject to $y_i((\langle x_i \cdot w \rangle + b) \geq 1$ for all $i$

Lagrangian method: maximize $\inf_w L(w, b, \alpha)$, where

$$L(w, b, \alpha) = \frac{1}{2} \|w\| - \sum_i \alpha_i [y_i(x_i \cdot w) + b - 1]$$

At the extremum, the partial derivative of $L$ with respect both $w$ and $b$ must be 0. Taking the derivatives, setting them to 0, substituting back into $L$, and simplifying yields:

Maximize $\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j K(x_i, x_j)$

subject to $\sum_i y_i \alpha_i = 0$ and $\alpha_i \geq 0$

$\lambda = \alpha$
Example

class 1

\[
\begin{array}{ccc}
\times & \times & \times \\
1 & 2 & 3
\end{array}
\]

class 2

\[
\begin{array}{ccc}
\bigcirc & \bigcirc & \times \\
4 & 5 & 6
\end{array}
\]

class 1
Example

- Suppose we have 5 one-dimensional data points
  - \( x_1=1, x_2=2, x_3=4, x_4=5, x_5=6 \), with values 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
  - \( K(x,z) = (xz+1)^2 \)
  - \( C \) is set to 100

- We first find \( \alpha_i \) (\( i=1, \ldots, 5 \)) by

\[
\begin{align*}
\alpha_j y_i y_j (x_i x_j + 1)^2 \\
\max. \quad \sum_{i=1}^{5} \alpha_i - \frac{1}{2} \sum_{i=1}^{5} \sum_{i=1}^{5} \alpha_i \alpha_j y_i y_j (x_i x_j + 1)^2 \\
\text{subject to } 100 \geq \alpha_i \geq 0, \sum_{i=1}^{5} \alpha_i y_i = 0
\end{align*}
\]
Example

• 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
  \[
  f(z) = 2.5(1)(2z + 1)^2 + 7.333(-1)(5z + 1)^2 + 4.833(1)(6z + 1)^2 + b
  = 0.6667z^2 - 5.333z + b
  \]

• $b$ is recovered by solving $f(2)=1$ or by $f(5)=-1$ or by $f(6)=1$, as $x_2$ and $x_5$ lie on the line $\phi(w)^T\phi(x) + b = 1$ and $x_4$ lies on the line $\phi(w)^T\phi(x) + b = -1$

• All three give $b=9$  \[ f(z) = 0.6667z^2 - 5.333z + 9 \]
Example

Value of discriminant function

class 1

1 2

4 5

6

class 2
class 1
Support Vector Machine (SVM)

- SVM represents the decision boundary using a subset of the training examples, known as the **support vectors**.

- The basic idea behind SVM lies within the concept of **maximal margin hyperplane**.
Characteristics of SVM

• Since the learning problem is formulated as a convex optimization problem, efficient algorithms are available to find the global minima of the objective function (many of the other methods use greedy approaches and find locally optimal solutions).

• Overfitting is addressed by maximizing the margin of the decision boundary, but the user still needs to provide the type of kernel function and cost function.

• Difficult to handle missing values.

• Robust to noise.

• High computational complexity for building the model.
References

• Support Vector Machine (SVM). Chapter 5.5. Introduction to Data Mining.
• http://www.kernel-machines.org/
• http://www.support-vector.net/
• An Introduction to Support Vector Machines. N. Cristianini and J. Shawe-Taylor.