Support Vector Machines

Gautam Kunapuli
University of Wisconsin-Madison
Outline

• **Example of classification problems**
• Formulating Support Vector Machines
• SVM Properties
• Soft-Margin SVMs
• Kernels and Nonlinear Data
• Regularization
Example: Diabetes Diagnosis

<table>
<thead>
<tr>
<th>Patient #</th>
<th>Plasma gluc. (mm Hg)</th>
<th>Diastolic BP (mm)</th>
<th>Fold test (mm)</th>
<th>2-hr Insulin (kg/m²)</th>
<th>BMI (kg/m²)</th>
<th>pedigree function</th>
<th>Age (yrs)</th>
<th>diabetes diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>85</td>
<td>66</td>
<td>29</td>
<td>0</td>
<td>26.6</td>
<td>0.351</td>
<td>31</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>183</td>
<td>64</td>
<td>0</td>
<td>0</td>
<td>23.3</td>
<td>0.672</td>
<td>32</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>89</td>
<td>66</td>
<td>23</td>
<td>94</td>
<td>28.1</td>
<td>0.167</td>
<td>21</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>137</td>
<td>40</td>
<td>35</td>
<td>168</td>
<td>43.1</td>
<td>2.288</td>
<td>33</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>116</td>
<td>74</td>
<td>0</td>
<td>0</td>
<td>25.6</td>
<td>0.201</td>
<td>30</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>78</td>
<td>50</td>
<td>32</td>
<td>88</td>
<td>31</td>
<td>0.248</td>
<td>26</td>
<td>yes</td>
</tr>
<tr>
<td>7</td>
<td>197</td>
<td>70</td>
<td>45</td>
<td>543</td>
<td>30.5</td>
<td>0.158</td>
<td>53</td>
<td>yes</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>768</td>
<td>166</td>
<td>72</td>
<td>19</td>
<td>175</td>
<td>25.8</td>
<td>0.587</td>
<td>51</td>
<td>yes</td>
</tr>
</tbody>
</table>
Example: Diabetes Diagnosis

Do Not Have Diabetes

- Blood glucose = 30
- Body mass index = 120 kg/m²
- Diastolic bp = 79 mm Hg
- Age = 32 years

Have Diabetes

- Blood glucose = 22
- Body mass index = 160 kg/m²
- Diastolic bp = 80 mm Hg
- Age = 63 years

- Blood glucose = 40
- Body mass index = 180 kg/m²
- Diastolic bp = 95 mm Hg
- Age = 49 years

- Blood glucose = 21
- Body mass index = 140 kg/m²
- Diastolic bp = 99 mm Hg
- Age = 37 years
Example: Diabetes Diagnosis

Do Not Have Diabetes

- Blood glucose = 30
- Body mass index = 120 kg/m²
- Diastolic BP = 79 mm Hg
- Age = 32 years

Have Diabetes

- Blood glucose = 22
  - Body mass index = 160 kg/m²
  - Diastolic BP = 80 mm Hg
  - Age = 18 years

- Blood glucose = 22
  - Body mass index = 75 kg/m²
  - Diastolic BP = 73 mm Hg
  - Age = 27 years

- Blood glucose = 46
  - Body mass index = 110 kg/m²
  - Diastolic BP = 110 mm Hg
  - Age = 55 years

- Blood glucose = 40
  - Body mass index = 150 kg/m²
  - Diastolic BP = 80 mm Hg
  - Age = 63 years

- Blood glucose = 45
  - Body mass index = 180 kg/m²
  - Diastolic BP = 95 mm Hg
  - Age = 49 years

- Blood glucose = 21
  - Body mass index = 140 kg/m²
  - Diastolic BP = 99 mm Hg
  - Age = 37 years
Example: Diabetes Diagnosis

**Do Not Have Diabetes**
- Blood glucose = 30
- Body mass index = 120 kg/m²
- Diastolic BP = 79 mm Hg
- Age = 32 years

**Have Diabetes**
- Blood glucose = 46
- Body mass index = 110 kg/m²
- Diastolic BP = 110 mm Hg
- Age = 55 years

Training examples with features:
- Blood glucose = 22
- Body mass index = 160 kg/m²
- Diastolic BP = 80 mm Hg
- Age = 18 years

Training labels for the examples to identify their class:
- Blood glucose = 22
- Body mass index = 160 kg/m²
- Diastolic BP = 80 mm Hg
- Age = 63 years

- Blood glucose = 40
- Body mass index = 150 kg/m²
- Diastolic BP = 100 mm Hg
- Age = 63 years

- Blood glucose = 45
- Body mass index = 180 kg/m²
- Diastolic BP = 95 mm Hg
- Age = 49 years

- Blood glucose = 21
- Body mass index = 140 kg/m²
- Diastolic BP = 99 mm Hg
- Age = 37 years
Example: Diabetes Diagnosis

Learn a classification function that can discriminate between the two classes.

**Do Not Have Diabetes**
- Blood glucose = 30
- Body mass index = 120 kg/m²
- Diastolic bp = 79 mm Hg
- Age = 32 years

**Have Diabetes**
- Blood glucose = 22
- Body mass index = 160 kg/m²
- Diastolic bp = 80 mm Hg
- Age = 18 years

- Blood glucose = 22
- Body mass index = 160 kg/m²
- Diastolic bp = 80 mm Hg
- Age = 63 years

- Blood glucose = 40
- Body mass index = 150 kg/m²
- Diastolic bp = 110 mm Hg
- Age = 55 years

- Blood glucose = 45
- Body mass index = 180 kg/m²
- Diastolic bp = 95 mm Hg
- Age = 49 years

- Blood glucose = 21
- Body mass index = 140 kg/m²
- Diastolic bp = 99 mm Hg
- Age = 37 years
## Example: Credit Card Application

<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>Age (yrs)</th>
<th>Income ($)</th>
<th>Rent/Own? (binary)</th>
<th>Monthly Rent/Mort.</th>
<th>Manager’s Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Greg J.</td>
<td>38</td>
<td>65,000</td>
<td>rent</td>
<td>1,050</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>James T.</td>
<td>24</td>
<td>21,000</td>
<td>rent</td>
<td>350</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>Hannah M.</td>
<td>45</td>
<td>98,000</td>
<td>own</td>
<td>2,400</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>Rashard K.</td>
<td>19</td>
<td>19,500</td>
<td>own</td>
<td>400</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>Xavier N.</td>
<td>29</td>
<td>75,000</td>
<td>rent</td>
<td>1,570</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>Jillian A.</td>
<td>29</td>
<td>39,000</td>
<td>own</td>
<td>1,000</td>
<td>yes</td>
</tr>
<tr>
<td>7</td>
<td>Ramon H.</td>
<td>35</td>
<td>103,000</td>
<td>rent</td>
<td>3,000</td>
<td>yes</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2000</td>
<td>Mary C.</td>
<td>55</td>
<td>45,000</td>
<td>rent</td>
<td>1,200</td>
<td>no</td>
</tr>
</tbody>
</table>

*Learn a classification function* to determine who gets a credit card.
Example: Handwritten Digit Recognition

Learn a classification function that can discriminate between multiple classes.

In this example, the classification function discriminates between 4s and not 4s. So, still an example of a 2-class problem.
Goal of Classification: Generalization

Learn a classification function that can discriminate between the two classes. Classification function can also generalize to unseen examples, and classify them correctly.
Outline

- Example of classification problems
- **Formulating Support Vector Machines**
- SVM Properties
- Soft-Margin SVMs
- Kernels and Nonlinear Data
- Regularization
Setting Up The SVM Problem

Find a linear classification function (a hyperplane) \( f(x) = w^T x - b \) such that

\[
\text{sign}(f(x)) = +1, \quad \text{when diabetes}
\]

\[
\text{sign}(f(x)) = -1, \quad \text{when not diabetes}
\]
The SVM Problem

Find a linear classification function (a hyperplane) $f(x) = w'x - b$ such that

$\text{sign}(f(x)) = +1$, when diabetes

$\text{sign}(f(x)) = -1$, when not diabetes
The Notion of Margin

The data set is linearly separable. There exist many different classifiers! Which one is the best?

The hyperplane is shown along with its margin, or how much it separates the two classes.
The Notion of Margin

The data set is \textit{linearly separable}. There exist many different classifiers! \textbf{Which one is the best?}

Do we want a skinny margin? Or a fat margin?
The Notion of Margin

If we want to learn a classifier that generalizes best, need one that achieves the **maximum margin**.
A Theoretical Digression

• When we have infinitely many solutions that reduce the error (variance), which one should we pick?
• We introduce a bias! Our bias is toward simpler solutions (minimal complexity).
• How do we measure complexity of hyperplanes with respect to data? Vapnik-Chervonenkis dimension
A Theoretical Digression: VC Dimension

- Complexity for a **class of functions** $H$ is measured by **VC Dimension**
- A given set of $\ell$ points can be labeled in $2^\ell$ ways.
- If for each labeling, some function $f$ from $H$ can be found which correctly assigns those labels, we say that that set of points is **shattered by $H$**.
- The VC dimension for the set of functions $H$ is defined as the **maximum number of training points that can be shattered**
Vladimir Vapnik showed that there is a connection between VC dimension and margin (Vapnik, 1995)

\[ h \leq \frac{4R^2}{\gamma^2} \]

To minimize the VC dimension (and hence complexity), we have to maximize the margin!
Formulating the SVM

\[ \mathbf{w}' \mathbf{x} - \alpha = 0 \]

\[ \mathbf{w}' \mathbf{x} - \beta = 0 \]

Distance of a point in Class \( \times \) to the margin is

\[ \frac{|\mathbf{w}' \mathbf{x} - \alpha|}{||\mathbf{w}||} \]

Distance of a point in Class \( \bullet \) to the margin is

\[ \frac{|\mathbf{w}' \mathbf{x} - \beta|}{||\mathbf{w}||} \]
Formulating the SVM

Distance between the hyperplanes is the margin:

\[ \gamma = \frac{|\alpha - \beta|}{\|w\|} \]

Distance of a point in Class \( \times \) to the margin is:

\[ \frac{|w'x - \alpha|}{\|w\|} \]

Distance of a point in Class \( \bullet \) to the margin is:

\[ \frac{|w'x - \beta|}{\|w\|} \]
Formulating the SVM

Distance between the hyperplanes is the margin

\[ \gamma = \frac{|\alpha - \beta|}{\|w\|} \]

Set \( \alpha = b + 1 \) and \( \beta = b - 1 \), then the margin is

\[ \gamma = \frac{2}{\|w\|} \]

which should be maximized.
We are given labeled data points
\[(x_i, y_i)_{i=1}^{\ell}\]
We need to learn a hyperplane
\[w'x - b = 0\]
such that

1. all the points in class with labels \(y_i = +1\), lie above the margin, that is
\[w'x_i - b \geq 1\]
2. all the points in class with labels \(y_i = -1\), lie below the margin, that is
\[w'x_i - b \leq -1\]
3. the margin is maximized
\[
\gamma = \frac{2}{\|w\|}
\]
Formulating the SVM

Maximizing $\gamma = \frac{2}{\|w\|}$ is equivalent to minimizing

$$\frac{1}{2}\|w\|^2$$

with the constraints that

$\mathbf{w}'\mathbf{x}_i - b \geq 1$ when $y_i = +1$

$\mathbf{w}'\mathbf{x}_i - b \leq -1$ when $y_i = -1$

Optimization problem for a support vector machine:

$$\min \frac{1}{2}\|\mathbf{w}\|^2$$

s.t. $y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1 \quad \forall i = 1 \ldots \ell$
Optimization problem for a support vector machine:

\[
\min_{\mathbf{w}, b} \frac{1}{2} \| \mathbf{w} \|_2^2 \\
\text{s.t. } y_i (\mathbf{w}' \mathbf{x}_i - b) \geq 1 \quad \forall i = 1 \ldots \ell
\]

- **Convex, quadratic** minimization problem called the **primal problem**. Guaranteed to have a **global minimum**.
- Further properties of the formulation can be studied by deriving the **dual problem**
- Introduce **Lagrange multipliers**, \( \{\alpha_i\}_{i=1}^\ell \), one for each constraint (hence data point). These are the **dual variables**.
- Construct the Lagrangian function of primal and dual variables (note that by definition all \( \alpha_i \geq 0 \))

\[
L(\mathbf{w}, b, \alpha_i) = \frac{1}{2} \mathbf{w}' \mathbf{w} - \sum_{i=1}^\ell \alpha_i [y_i (\mathbf{w}' \mathbf{x}_i - b) - 1]
\]
Differentiate the Lagrangian \textbf{with respect to the primal variables}

\[
\nabla_w L(w, b, \alpha_i) = 0 : \quad w = \sum_{i=1}^{\ell} \alpha_i y_i x_i
\]

\[
\nabla_b L(w, b, \alpha_i) = 0 : \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0
\]

These are the \textbf{first order optimality conditions}. We can now \textbf{eliminate} the primal variables by \textbf{substituting the first order conditions} into the Lagrangian.
support vector machine **primal problem**

\[
\begin{align*}
\min & \quad \frac{1}{2} \|w\|^2_2 \\
\text{s.t.} & \quad y_i (w'x_i - b) \geq 1 \quad \forall i = 1 \ldots \ell
\end{align*}
\]

support vector machine **dual problem**

\[
\begin{align*}
\max & \quad -\frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j x_i'x_j + \sum_{i=1}^{\ell} \alpha_i \\
\text{s.t.} & \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0, \\
& \quad \alpha_i \geq 0, \quad \forall i = 1 \ldots \ell
\end{align*}
\]

Why bother with the dual, and solving for \( \alpha \)?

- convex optimization problem. **No duality gap**
- Dual has fewer constraints. **Easier to solve**
- Dual solution is sparse. **Easier to represent**
Outline

• Example of classification problems
• Formulating Support Vector Machines
• **SVM Properties**
  • Soft-Margin SVMs
• Kernels and Nonlinear Data
• Regularization
Characteristics of the Solution

Recall the first order condition:

$$w = \sum_{i=1}^{\ell} \alpha_i y_i x_i$$

The final solution is a linear combination of the training data!

Additional optimality condition: complementarity slackness

$$0 \leq \alpha_i \perp y_i(w'x - b) - 1 \geq 0$$

$$\alpha_i = 0 \quad \text{and} \quad y_i(w'x_i - b) > 1 \quad (\text{point not on hyperplane})$$

or

$$\alpha_i > 0 \quad \text{and} \quad y_i(w'x_i - b) = 1 \quad (\text{point on hyperplane})$$
Characteristics of the Solution

Recall the first order condition:

\[ w = \sum_{i=1}^{\ell} \alpha_i y_i x_i \]

The final solution is a **sparse** linear combination of the training data!

only **support vectors** have \( \alpha_i > 0 \) (non-zero). All other vectors have \( \alpha_i = 0 \), and this makes the solution sparse!

\[ \alpha_i = 0 \quad \text{and} \quad y_i(w'x_i - b) > 1 \quad \text{(point not on hyperplane)} \]

\[ \alpha_i > 0 \quad \text{and} \quad y_i(w'x_i - b) = 1 \quad \text{(point on hyperplane)} \]
Characteristics of the Solution

Recall the first order condition:

\[ w = \sum_{i=1}^{\ell} \alpha_i y_i x_i \]

The final solution is a **sparse** linear combination of the training data!

only **support vectors** have \( \alpha_i > 0 \) (non-zero). All other vectors have \( \alpha_i = 0 \), and this makes the solution **sparse**!

solution does not change if all other vectors with \( \alpha_i = 0 \) are **deleted**!
Outline

• Example of classification problems
• Formulating Support Vector Machines
• SVM Properties
• **Soft-Margin SVMs**
• Kernels and Nonlinear Data
• Regularization
Linearly Inseparable Data

So far, assumed that the data is **linearly separable**. However, this assumption is not valid in most **real-world applications**.

Need to extend **hard-margin support vector machines** to be able to handle noisy data.

This results in the **soft-margin support vector machine**.
Soft-margin Support Vector Machine

**hard-margin** support vector machine

\[
\begin{align*}
\text{min} & \quad \frac{1}{2} \| w \|_2^2 \\
\text{s.t.} & \quad y_i (w' x_i - b) \geq 1 \quad \forall i = 1 \ldots \ell
\end{align*}
\]

**soft-margin** support vector machine

\[
\begin{align*}
\text{min} & \quad \frac{1}{2} \| w \|_2^2 + C \sum_{i=1}^{\ell} \xi_i \\
\text{s.t.} & \quad y_i (w' x_i - b) \geq 1 - \xi_i \quad \forall i = 1 \ldots \ell \\
\xi_i & \geq 0
\end{align*}
\]
Soft-margin Support Vector Machine

**hard-margin** support vector machine

\[
\begin{align*}
\min & \quad \frac{1}{2} \| w \|^2 \\
\text{s.t.} & \quad y_i (w' x_i - b) \geq 1 \quad \forall i = 1 \ldots \ell
\end{align*}
\]

**soft-margin** support vector machine

\[
\begin{align*}
\min & \quad \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{\ell} \xi_i \\
\text{s.t.} & \quad y_i (w' x_i - b) \geq 1 - \xi_i \quad \forall i = 1 \ldots \ell \\
\xi_i & \geq 0
\end{align*}
\]

errors (slack variables) to measure loss of misclassified data points
Soft-margin Support Vector Machine

**hard-margin** support vector machine

\[
\min \quad \frac{1}{2} \| w \|_2^2 \\
\text{s.t.} \quad y_i (w' x_i - b) \geq 1 \quad \forall i = 1 \ldots \ell
\]

**soft-margin** support vector machine

\[
\min \quad \frac{1}{2} \| w \|_2^2 + C \sum_{i=1}^{\ell} \xi_i \\
\text{s.t.} \quad y_i (w' x_i - b) \geq 1 - \xi_i \quad \forall i = 1 \ldots \ell \\
\xi_i \geq 0
\]

Errors (slack variables) to measure loss of misclassified data points

Regularization constant that trades off between complexity and loss
Soft-margin Dual

**hard-margin svm dual**

\[
\text{max} \quad -\frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j x'_i x_j + \sum_{i=1}^{\ell} \alpha_i \\
\text{s.t.} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad \forall i = 1 \ldots \ell
\]

**soft-margin svm dual**

\[
\text{max} \quad -\frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j x'_i x_j + \sum_{i=1}^{\ell} \alpha_i \\
\text{s.t.} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0 \\
0 \leq \alpha_i \leq C, \quad \forall i = 1 \ldots \ell
\]
Some things to note about the Dual

\[
\begin{align*}
\text{max} & \quad -\frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{\ell} \alpha_i \\
\text{s.t.} & \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0 \\
& \quad 0 \leq \alpha_i \leq C, \quad \forall i = 1 \ldots \ell
\end{align*}
\]

Note that the regularization constant is set by the user.

This is an important parameter that can cause dramatically different behaviors on the same data set.

Note that the dual solution depends only on the inner products of the training data.

This is an important observation that allows us to extend linear SVMs to handle nonlinear data.
Outline

• Example of classification problems
• Formulating Support Vector Machines
• SVM Properties
• Soft-Margin SVMs
• Kernels and Nonlinear Data
• Regularization
Nonlinear Data Sets

No linear classifier can classify this nonlinear data.

Naïve solution: Transform the input data into a higher dimension using the following nonlinear transformation:

\[
\begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix}
\rightarrow
\begin{bmatrix}
  x_1^2 \\
  \sqrt{2}x_1x_2 \\
  x_2^2
\end{bmatrix}
\]
No linear classifier exists in 2-d space, but a linear classifier can classify this nonlinear data in 3-d space!

Naïve solution: Transform the input data into a higher dimension using the following nonlinear transformation:

\[
\begin{bmatrix}
  x_1 \\
  x_2 
\end{bmatrix} \rightarrow \begin{bmatrix}
  x_1^2 \\
  \sqrt{2} x_1 x_2 \\
  x_2^2
\end{bmatrix}
\]
Nonlinear Data Sets

Naïve solution: Transform the input data into a higher dimension using the following nonlinear transformation:

\[
\begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix}
\rightarrow
\begin{bmatrix}
  x_1^2 \\
  \sqrt{2}x_1x_2 \\
  x_2^2
\end{bmatrix}
\]
Naïve Approach: Explicit Transformation

transform data to a higher dimensional feature space

train a classifier in the high dimensional space using support vector machines

project the learned classifier back to the original space
Naïve Approach: Explicit Transformation

Recall that SVM training relies only on inner products of the training data.

In this case, it will rely on the inner-products of the transformed data.
Inner Products in Feature Space

Let two points in the original input space be
\[ x = (x_1, x_2) \]
\[ z = (z_1, z_2) \]

After transformation, in the **high-dimensional feature space**, they become
\[ \phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \]
\[ \phi(z) = (z_1^2, \sqrt{2}z_1z_2, z_2^2) \]

What do inner products in this **transformed feature space** look like?
\[ \langle \phi(x), \phi(w) \rangle = \langle (x_1^2, x_2^2, \sqrt{2}x_1 x_2), (w_1^2, w_2^2, \sqrt{2}w_1 w_2) \rangle \]
\[ = x_1^2 w_1^2 + x_2^2 w_2^2 + 2x_1 w_1 x_2 w_2 \]
\[ = (x_1 w_1 + x_2 w_2)^2 \]
\[ = \langle x, w \rangle^2 \]
The Kernel Trick

inner products in high-dimensional feature space

\[ \langle \phi(x), \phi(w) \rangle = \langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), (w_1^2, w_2^2, \sqrt{2}w_1w_2) \rangle \]
\[ = x_1^2w_1^2 + x_2^2w_2^2 + 2x_1w_1x_2w_2 \]
\[ = (x_1w_1 + x_2w_2)^2 \]
\[ = \langle x, w \rangle^2 \]

inner products in the original input space

the two inner products are related to each other through the kernel function
Better Approach: The Kernel Trick

Use a kernel function to directly learn a nonlinear classifier!

No need for explicit transformations

Can use existing approaches with slight modification!

transform data to a higher dimensional feature space

train a classifier in the high dimensional space using support vector machines

project the learned classifier back to the original space
The Kernel Trick

**linear support vector machine**

\[
\max \quad -\frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{\ell} \alpha_i \\
\text{s.t.} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0 \\
\quad 0 \leq \alpha_i \leq C \quad \forall i = 1 \ldots \ell
\]

**kernel support vector machine**

\[
\max \quad -\frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \kappa(x_i, x_j) + \sum_{i=1}^{\ell} \alpha_i \\
\text{s.t.} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0 \\
\quad 0 \leq \alpha_i \leq C \quad \forall i = 1 \ldots \ell
\]

Use a kernel function to directly learn a nonlinear classifier!

No need for explicit transformations

Can use existing approaches with slight modification!
Some Popular Kernels

Some popular kernels

• Linear kernel: $\kappa(x, z) = \langle x, z \rangle$

• Polynomial kernel: $\kappa(x, z) = (\langle x, z \rangle + c)^d, \ c, d \geq 0$

• Gaussian kernel: $\kappa(x, z) = e^{-\frac{\|x-z\|^2}{\sigma}}, \ \sigma > 0$

• Sigmoid kernel: $\kappa(x, z) = \tanh^{-1} \eta \langle x, z \rangle + \theta$

Kernels can also be constructed from other kernels:

• Conical (not linear) combinations, $\kappa(x, z) = a_1 \kappa_1(x, z) + a_2 \kappa_2(x, z)$

• Products of kernels, $\kappa(x, z) = \kappa_1(x, z) \kappa_2(x, z)$

• Products of functions, $\kappa(x, z) = f_1(x)f_2(z)$, $f_1, f_2$ are real valued functions.
Polynomial Kernels

$$\kappa(x, z) = \left(\langle x, z \rangle + 1 \right)^d$$
Gaussian Kernels

\[ \kappa(x, z) = \exp \left( -\frac{\|x-z\|^2}{\sigma} \right) \]

Gaussian Kernel, gamma =
Gaussian Kernels

\[ \kappa(x, z) = \exp - \frac{||x-z||^2}{\sigma} \]
Outline

• Example of classification problems
• Formulating Support Vector Machines
• SVM Properties
• Soft-Margin SVMs
• Kernels and Nonlinear Data
• Regularization
Regularization and Over-fitting

**soft-margin** support vector machine

\[
\begin{align*}
\text{min} & \quad \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{\ell} \xi \\
\text{s.t.} & \quad y_i (w' x_i - b) \geq 1 - \xi_i \quad \forall i = 1 \ldots \ell \\
& \quad \xi \geq 0
\end{align*}
\]

The regularization parameter, \( C \), is chosen a priori, defines the relative trade-off between norm (complexity / smoothness / capacity) and loss (error penalization)

We want to find classifiers that minimize \((\text{regularization} + C \cdot \text{loss})\)

Regularization
- introduces inductive bias over solutions
- controls the complexity of the solution
- imposes smoothness restriction on solutions

As \( C \) increases, the effect of the regularization decreases and the SVM tends to overfit the data

**kernel** support vector machine

\[
\begin{align*}
\text{max} & \quad -\frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \kappa(x_i, x_j) + \sum_{i=1}^{\ell} \alpha_i \\
\text{s.t.} & \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0 \\
& \quad 0 \leq \alpha_i \leq C \quad \forall i = 1 \ldots \ell
\end{align*}
\]
The Effect of C on Classification

$C = 0.001$
The Effect of C on Classification

$C = 0.01$
The Effect of C on Classification

\[ C = 0.1 \]
The Effect of $C$ on Classification

$C = 1$
The Effect of $C$ on Classification

$C = 10$
The Effect of C on Classification

\[ C = 100 \]
SVM Algorithms over the Years

• **Earliest solution approaches**: Quadratic Programming Solvers (CPLEX, LOQO, Matlab QP, SeDuMi)

• **Decomposition methods**: SVM chunking (Osuna et. al., 1997); implementation: SVMlight (Joachims, 1999)

• **Sequential Minimization Optimization** (Platt, 1999); implementation: LIBSVM (Chang et. al., 2000)

• **Interior Point Methods** (Munson and Ferris, 2006), **Successive Over-relaxation** (Mangasarian, 2004)

• **Co-ordinate Descent Algorithms** (Keerthi et. al., 2009), **Bundle Methods** (Teo et. al., 2010)