Support Vector Machines

Razvan C. Bunescu

Department of Computer Science @ CCI

razvan.bunescu@uncc.edu
Max-Margin Classifiers: Separable Case

- Linear model for binary classification:
  \[ y(x) = w^T \varphi(x) + b \]

- Training examples:
  \[(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N), \text{ where } t_n \in \{+1, -1\}\]

- Assume training data is linearly separable:
  \[ t_n y(x_n) > 0, \text{ for all } 1 \leq n \leq N \]

  \[ \Rightarrow \text{perceptron solution depends on:} \]
  - initial values of \(w\) and \(b\).
  - order of processing of data points.
Maximum Margin Classifiers

• Which hyperplane has the smallest generalization error?
  – The one that maximizes the margin \([\text{Computational Learning Theory}]\)
    • margin = the distance between the decision boundary and the closest sample.
Geometric Interpretation

\[ h(x) = w^T x + w_0 \]
Maximum Margin Classifiers

- The distance between a point $x_n$ and a hyperplane $y(x)=0$ is:

$$
\frac{|y(x_n)|}{||w||} = \frac{t_n y(x_n)}{||w||} = \frac{t_n (w^T \varphi(x_n) + b)}{||w||}
$$
Maximum Margin Classifiers

- Margin = the distance between hyperplane $y(x)=0$ and closest sample:

$$\min_n \left[ \frac{t_n (w^T \varphi(x_n) + b)}{||w||} \right]$$

- Find parameters $w$ and $b$ that maximize the margin:

$$\arg \max_{w,b} \left\{ \frac{1}{||w||} \min_n [t_n (w^T \varphi(x_n) + b)] \right\}$$

- Rescaling $w$ and $b$ does not change distances to the hyperplane:

$\Rightarrow$ for the closest point(s), set $t_n (w^T \varphi(x_n) + b) = 1$

$\Rightarrow$ this means $t_n (w^T \varphi(x_n) + b) \geq 1, \quad \forall n \in \{1, \ldots, N\}$
Max-Margin: Quadratic Optimization

- Constrained optimization problem:

  minimize:

  \[ J(w, b) = \frac{1}{2} \| w \|^2 \]

  subject to:

  \[ t_n (w^T \varphi(x_n) + b) \geq 1, \quad \forall n \in \{1, \ldots, N\} \]

- Solved using the technique of **Lagrange Multipliers**.
  
  - [derivation shown at the end of slides, mandatory for 8156].
Max-Margin: Quadratic Optimization

• Equivalent **dual representation**:

maximize:

\[ L_D(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m t_n t_m k(x_n, x_m) \]

subject to:

\[ \alpha_n \geq 0, \; n = 1, \ldots, N \]

\[ \sum_{n=1}^{N} \alpha_n t_n = 0 \]

- \( k(x_n, x_m) = \varphi(x_n)^T \varphi(x_n) \) is the *kernel* function.

- where \( w = \sum_{n=1}^{N} \alpha_n t_n \varphi(x_n) \) and \( \sum_{n=1}^{N} \alpha_n t_n = 0 \)

Exactly like in the Kernel Perceptron!
KKT conditions

1. **primal constraints**: \( t_n y(x_n) - 1 \geq 0 \)

1. **dual constraints**: \( \alpha_n \geq 0 \)

2. **complementary slackness**: 
   \[
   \alpha_n \left\{ t_n y(x_n) - 1 \right\} = 0
   \]

\[\Rightarrow \quad \text{for any data point, either } \alpha_n = 0 \text{ or } t_n y(x_n) = 1\]

\[S = \{n \mid t_n y(x_n) = 1\} \text{ is the set of support vectors}\]
Max-Margin Solution

- After solving the dual problem ⇒ know \( \alpha_n \), for \( n = 1 \ldots N \)

\[
\mathbf{w} = \sum_{n=1}^{N} \alpha_n t_n \varphi(x_n) = \sum_{m \in S} \alpha_m t_m \varphi(x_m)
\]

\[
b = \frac{1}{|S|} \sum_{n \in S} \left( t_n - \sum_{m \in S} \alpha_m t_m k(x_n, x_m) \right)
\]

- Linear discriminant function becomes:

\[
y(x) = \sum_{m \in S} \alpha_m t_m k(x, x_m) + b
\]

⇒ In both training and testing, examples are used only through the *kernel function*!
An SVM with Gaussian kernel
Max-Margin Classifiers: Non-Separable Case

- Allow data points to be on the wrong side of the margin boundary.
  - Penalty that increases with the distance from the boundary.
Max-Margin: Quadratic Optimization

- Optimization problem:

\[
\begin{align*}
\text{minimize:} & \quad J(w, b) = \frac{1}{2} \|w\|^2 + C \sum_{n=1}^{N} \xi_n \\
\text{subject to:} & \quad t_n (w^T \varphi(x_n) + b) \geq 1 - \xi_n, \quad \forall n \in \{1, \ldots, N\} \\
& \quad \xi_n \geq 0
\end{align*}
\]

- Solve it using the technique of Lagrange Multipliers.
Max-Margin: Quadratic Optimization

• Dual representation:

maximize:

\[
L_D(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m t_n t_m k(x_n, x_m)
\]

subject to:

\[
0 \leq \alpha_n \leq C, \quad n = 1, \ldots, N
\]

\[
\sum_{n=1}^{N} \alpha_n t_n = 0
\]

• \(k(x_n, x_m) = \varphi(x_n)^T \varphi(x_n)\) is the kernel function.
(Some of the) KKT conditions

1. primal constraints: \( t_n y(x_n) - 1 + \xi_n \geq 0 \)

1. dual constraints: \( 0 \leq \alpha_n \leq C \)

2. complementary slackness: \( \alpha_n \{ t_n y(x_n) - 1 + \xi_n \} = 0 \)

\[ \Rightarrow \text{for any data point, either } \alpha_n = 0 \text{ or } t_n y(x_n) = 1 - \xi_n \]

\( S = \{ n \mid t_n y(x_n) = 1 - \xi_n \} \) is the set of support vectors

\( M = \{ n \mid 0 < \alpha_n < C \} \) is the set of SVs that lie on the margin.
Max-Margin Solution

- After solving the dual problem ⇒ know $\alpha_n$, for $n = 1 \ldots N$

$$w = \sum_{n=1}^{N} \alpha_n t_n \varphi(x_n) = \sum_{m \in S} \alpha_m t_m \varphi(x_m)$$

$$b = \frac{1}{|M|} \sum_{n \in M} \left( t_n - \sum_{m \in S} \alpha_m t_m k(x_n, x_m) \right)$$

- Linear discriminant function becomes:

$$y(x) = \sum_{m \in S} \alpha_m t_m k(x, x_m) + b$$

⇒ In both training and testing, examples are used only through the kernel function!
Support Vector Machines

- Optimization problem:

\[
\begin{aligned}
\text{minimize:} & \quad J(w, b) = \frac{1}{2} \|w\|^2 + C \sum_{n=1}^{N} \xi_n \\
\text{subject to:} & \quad t_n (w^T \varphi(x_n) + b) \geq 1 - \xi_n, \quad \forall n \in \{1, \ldots, N\} \\
& \quad \xi_n \geq 0
\end{aligned}
\]

- Implemented in `sklearn`:
  

upper bound on the **misclassification error** on the training data.
SVMs for Regression

- Use an $\varepsilon$-insensitive error function ($\varepsilon > 0$) to obtain *sparse solutions*.
  - Penalty that increases with the distance from the $\varepsilon$-insensitive “tube”.

![Diagram showing SVMs for Regression](image-url)
SVMs for Regression: Quadratic Optimization

- Optimization problem:

\[
\begin{align*}
\text{minimize:} & \quad J(w, b) = \frac{1}{2} \| w \|^2 + C \sum_{n=1}^{N} (\xi_n + \hat{\xi}_n) \\
\text{subject to:} & \quad t_n \leq w^T \varphi(x_n) + b + \varepsilon + \xi_n \\
& \quad t_n \geq w^T \varphi(x_n) + b - \varepsilon - \hat{\xi}_n \\
& \quad \xi_n, \hat{\xi}_n \geq 0, \quad \forall n \in \{1, \ldots, N\}
\end{align*}
\]

- Solve it using the technique of Lagrange Multipliers.
SVMs for Regression: Sparse Solution

- After solving the dual problem $\Rightarrow$ know $\alpha_n, \hat{\alpha}_n$ for $n = 1 \ldots N$

$$w = \sum_{n=1}^{N} (\alpha_n - \hat{\alpha}_n) \varphi(x_n) = \sum_{m \in S} (\alpha_m - \hat{\alpha}_m) \varphi(x_m)$$

- $S$ is the set of support vectors:
  i.e. points for which either $\alpha_n \neq 0$ or $\hat{\alpha}_n \neq 0$
  $\Rightarrow$ points that lie on the boundary of the $\varepsilon$-insensitive tube or outside the tube

$$y(x) = w^T x + b = \sum_{m \in S} (\alpha_m - \hat{\alpha}_m) k(x, x_m) + b$$

$\Rightarrow$ In both training and testing, examples are used only through the kernel function!
SVMs for Regression: Sparse Solution
SVMs for Ranking

[Joachims, KDD’02]

• Problem:
  – For a query $q$, a search engine returns a set of documents $D$.
  – Want to rank $d_i$ higher than $d_j$ if $d_i$ is more relevant to $q$ than $d_j$.

• Solution:
  – Learn a ranking function $f(q,d) = w^T \phi(q,d)$
  – Rank $d_i$ higher than $d_j$ if $f(q,d_i) \geq f(q,d_j) \iff w^T \phi(q,d_i) \geq w^T \phi(q,d_j)$
  – Training data:
    • Set $\{(q_k, d_i, d_j) \mid d_i \text{ ranked higher than } d_j \text{ for query } q_k\}$.
    • Relative rankings obtained from clickthrough data.
SVMs for Ranking [Joachims, KDD’02]

- Optimization problem:

\[
\begin{align*}
\text{minimize:} & \quad J(w) = \frac{1}{2} \|w\|^2 + C \sum \xi_{k,i,j} \\
\text{subject to:} & \quad w^T \varphi(q_k, d_i) \geq w^T \varphi(q_k, d_i) + 1 - \xi_{k,i,j} \\
& \quad \xi_{k,i,j} \geq 0
\end{align*}
\]

\[
w^T(\varphi(q_k, d_i) - \varphi(q_k, d_i)) \geq 1 - \xi_{k,i,j}
\]

⇒ equivalent with a classification problem
SVMs for Ranking [Joachims, KDD’02]

- After solving the quadratic problem:

\[ w = \sum_{k,l} \alpha_{k,l} \varphi(q_k, d_l) \]

\[ \Rightarrow f(q, d) = w^T \varphi(q, d) = \sum_{k,l} \alpha_{k,l} \varphi(q_k, d_l) \varphi(q, d) = \sum_{k,l} \alpha_{k,l} K(q_k, d_l, q, d) \]

\[ \Rightarrow \text{In both training and testing, examples are used only through the kernel function!} \]
Learning Scenarios for SVMs

- Classification.
- Ranking.
- Regression.
- Ordinal Regression.
- One Class Learning.
- Learning with Positive and Unlabeled examples.
- Transductive Learning.
- Semi-Supervised Learning.
- Multiple Instance Learning.
- Structured Outputs.
**Practical Issues**

- **Data Scaling:**
  - Between $[-1,+1]$ or $[0, 1]$.
  - Use same scaling factors in training and testing!

- **Parameter Tuning:**
  - Most SVM packages specify reasonable default values.
    - Tuning helps, especially with kernels that tend to overfit.
  - Grid search is simple and effective:
    - For RBF kernels, need to tune $C$ and $\gamma$:
      - $C \in \{2^{-5}, 2^{-3}, ..., 2^{15}\}$, $\gamma \in \{2^{-15}, 2^{-13}, ..., 2^3\}$

- Read LibSVM’s “[A practical guide to SVM classification](#)”. 


Conclusion

- SVMs were originally proposed by Boser, Guyon, and Vapnik in 1992.
- Good performance on a number of classification tasks ranging from text to genomic data.
- SVMs can be applied to complex data types, e.g. graphs, trees, sequences, by designing kernel functions for such data.
  - Also to probability distributions – “Learning from Distributions via Support Measure Machines” [Muandet et al., NIPS 2012]
- Kernel trick has been extended to other methods such as Perceptron, PCA, kNN, etc.
- Popular optimization algorithms for SVMs use decomposition to hill-climb over a subset of $\alpha_n$’s at a time, e.g. SMO [Platt ‘99].
  - But training and testing with linear SVMs are much faster.
- Read Lin’s “Machine Learning Software: Design and Practical Use”
Convex Optimization

- Convex optimization problem in standard form (primal):

  minimize:
  
  \[ f_0(x) \]

  subject to:
  
  \[ f_i(x) \leq 0, \quad i = 1, \ldots, m \]
  
  \[ h_i(x) = 0, \quad i = 1, \ldots, p \]

  - \( f_i : \mathbb{R}^n \rightarrow \mathbb{R} \) are all **convex functions**, for \( i = 0, \ldots, m \)
  
  - \( h_i : \mathbb{R}^n \rightarrow \mathbb{R} \) are all **affine functions**, for \( i = 0, \ldots, p \) (e.g. \( h_i(x) = Ax + b \))

  solution \( x^* \)
Lagrange Multipliers

• Define Lagrangian function $L_P : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$:

$$L_P(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

• $\lambda_i \geq 0$, and $\nu_i$ are the Lagrange multipliers.

• Define Lagrange dual function $L_D : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$:

$$L_D(\lambda, \nu) = \inf_{x} L_P(x, \lambda, \nu)$$
Convex Optimization

- Lagrange Dual Problem:

\[
\begin{align*}
\text{maximize:} & \quad L_D(\lambda, \nu) \\
\text{subject to:} & \quad \lambda_i \geq 0, \quad i = 1, \ldots, m
\end{align*}
\]

\[
L_D(\lambda, \nu) = \inf_{x} L_P(x, \lambda, \nu)
\]

solution \((\lambda^*, \nu^*)\)
Strong Duality

- Optimum for primal problem = optimum for dual problem:
  \[ f_0(x^*) = L_D(\lambda^*, \nu^*) \]
Karush–Kuhn–Tucker (KKT) conditions

Assume \((x, \lambda, \nu)\) are the primal & dual solutions. Then \((x, \lambda, \nu)\) satisfy the following constraints:

1. **primal constraints:**
   \[
   \begin{align*}
   f_i(x) &\leq 0, \quad i = 1, \ldots, m \\
   h_i(x) &= 0, \quad i = 1, \ldots, p
   \end{align*}
   \]

2. **dual constraints:**
   \[\lambda_i \geq 0, \quad i = 1, \ldots, m\]

3. **complementary slackness:**
   \[\lambda_i f_i(x) = 0, \quad i = 1, \ldots, m\]

4. **gradient of Lagrangian with respect to** \(x\) **vanishes:**
   \[
   \nabla L_P(x) = \nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0
   \]
Max-Margin: Quadratic Optimization

• Constrained optimization problem:

\[
\begin{align*}
\text{minimize:} & \quad J(w, b) = \frac{1}{2} \|w\|^2 \\
\text{subject to:} & \quad t_n (w^T \varphi(x_n) + b) \geq 1, \quad \forall n \in \{1, \ldots, N\}
\end{align*}
\]

• Let’s solve it using the technique of Lagrange Multipliers.
Max-Margin: Quadratic Optimization

- Lagrangian function:
  
  \[ L_P(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N} \alpha_n \left\{ t_n (w^T \varphi(x_n) + b) - 1 \right\} \]

  - \( \alpha_n \geq 0 \) are the Lagrangian multipliers.

- Lagrangian dual function:
  
  \[ L_D(\alpha) = \inf_{w, b} L_P(w, b, \alpha) \]

- Solve:
  
  \[ \begin{align*}
  \frac{\partial L_P}{\partial w} &= 0 \\
  \frac{\partial L_P}{\partial b} &= 0
  \end{align*} \]

  \[ \Rightarrow \quad \begin{cases} 
  w = \sum_{n=1}^{N} \alpha_n t_n \varphi(x_n) \\
  \sum_{n=1}^{N} \alpha_n t_n = 0 
  \end{cases} \]