Support Vector Machines
Maximum Margin Classifiers

Marek Petrik

3/30/2017
Classifiers

- Which classifiers do you know? (5+)
Classifiers

- Which classifiers do you know? (5+)

- Which ones are generative/discriminative?
Classifiers

- Which classifiers do you know? (5+)

- Which ones are generative/discriminative?

- Do we need any more? Why?
Separating Hyperplane

Hyperplane: $\beta_0 + x^\top \beta = 0$
Separating Hyperplane

Blue: $\beta_0 + x^\top \beta > 0$
Separating Hyperplane

Red: $\beta_0 + x^T \beta < 0$
Question

- Which other classification methods classify using a separating hyperplane?
Best Separating Hyperplane

- Data is separable
- Why would either one be better than others?
Separating Hyperplane Methods

How is it computed?

- **Logistic regression:**
Separating Hyperplane Methods

How is it computed?

- **Logistic regression**: Maximum likelihood
How is it computed?

- **Logistic regression**: Maximum likelihood

- **LDA**: 
Separating Hyperplane Methods

How is it computed?

- **Logistic regression**: Maximum likelihood

- **LDA**: Maximum likelihood
Separating Hyperplane Methods

How is it computed?

- **Logistic regression**: Maximum likelihood
- **LDA**: Maximum likelihood
- **Support vector machines**: Maximum margin
Maximum Margin Hyperplane
Class labels: $y_i \in \{-1, +1\}$ (not $\{0, 1\}$)

Solve a quadratic program (assume one of the features is a constant to get the equivalent of an intercept)

$$\begin{align*}
\max_{\beta, M} & \quad M \\
\text{s.t.} & \quad y_i(\beta^\top x) \geq M \\
& \quad \|\beta\|_2 = 1
\end{align*}$$
Non-separable Case

- Rarely lucky enough to get separable classes
Almost Unseparable Cases

- Maximum margin can be brittle even when classes are separable
Introducing Slack Variables

- **Maximum margin classifier**

\[
\max_{\beta, M} \quad M \\
\text{s.t.} \quad y_i(\beta^T x) \geq M \\
\quad \|\beta\|_2 = 1
\]

- **Support Vector Classifier a.k.a Linear SVM**

\[
\max_{\beta, M, \epsilon \geq 0} \quad M \\
\text{s.t.} \quad y_i(\beta^T x) \geq (M - \epsilon_i) \\
\quad \|\beta\|_2 = 1 \\
\quad \|\epsilon\|_1 \leq C
\]

- Slack variables: \(\epsilon\)
- Parameter: \(C\)
Introducing Slack Variables

▶ Maximum margin classifier

\[
\begin{align*}
\max_{\beta, M} & \quad M \\
\text{s.t.} & \quad y_i (\beta^\top x) \geq M \\
& \quad \|\beta\|_2 = 1
\end{align*}
\]

▶ Support Vector Classifier a.k.a Linear SVM

\[
\begin{align*}
\max_{\beta, M, \epsilon \geq 0} & \quad M \\
\text{s.t.} & \quad y_i (\beta^\top x) \geq (M - \epsilon_i) \\
& \quad \|\beta\|_2 = 1 \\
& \quad \|\epsilon\|_1 \leq C
\end{align*}
\]

▶ Slack variables: \( \epsilon \)

▶ Parameter: \( C \) What if \( C = 0 \)?
Effect of Decreasing Parameter $C$
What About Nonlinearity?
Dealing with Nonlinearity

- Introduce more features, just like with logistic regression
- It is possible to do better with SVMs
- **Primal Quadratic Program**

  \[
  \max_{\beta, M} \quad M \\
  \text{s.t.} \quad y_i(\beta^\top x) \geq M \\
  \quad \|\beta\|_2 = 1
  \]

- Equivalent **Dual Quadratic Program** (usually max-min, not here)

  \[
  \max_{\alpha \geq 0} \sum_{l=1}^{M} \alpha_l - \frac{1}{2} \sum_{j,k=1}^{M} \alpha_j \alpha_k y_j y_k \langle x_j, x_k \rangle \\
  \text{s.t.} \quad \sum_{l=1}^{M} \alpha_l y_l = 0
  \]
SVM Dual Representation

- **Dual Quadratic Program** (usually max-min, not here)

\[
\begin{align*}
\max_{\alpha \geq 0} & \quad \sum_{l=1}^{M} \alpha_l - \frac{1}{2} \sum_{j,k=1}^{M} \alpha_j \alpha_k y_j y_k \langle x_j, x_k \rangle \\
\text{s.t.} & \quad \sum_{l=1}^{M} \alpha_l y_l = 0
\end{align*}
\]

- **Representer theorem**: (classification test):

\[
f(z) = \sum_{l=1}^{M} \alpha_l y_l \langle z, x_l \rangle > 0
\]
SVM Dual Representation

- **Dual Quadratic Program** (usually max-min, not here)

\[
\begin{align*}
\max_{\alpha \geq 0} & \quad \sum_{l=1}^{M} \alpha_l - \frac{1}{2} \sum_{j,k=1}^{M} \alpha_j \alpha_k y_j y_k \langle x_j, x_k \rangle \\
\text{s.t.} & \quad \sum_{l=1}^{M} \alpha_l y_l = 0
\end{align*}
\]

- **Representer theorem**: (classification test):

\[
f(z) = \sum_{l=1}^{M} \alpha_l y_l \langle z, x_l \rangle > 0
\]

- Only need the inner product between data points
SVM Dual Representation

- **Dual Quadratic Program** (usually max-min, not here)

\[
\max_{\alpha \geq 0} \sum_{l=1}^{M} \alpha_l - \frac{1}{2} \sum_{j,k=1}^{M} \alpha_j \alpha_k y_j y_k \langle x_j, x_k \rangle
\]

s.t. \[\sum_{l=1}^{M} \alpha_l y_l = 0\]

- **Representer theorem**: (classification test):

\[
f(z) = \sum_{l=1}^{M} \alpha_l y_l \langle z, x_l \rangle > 0
\]

- Only need the inner product between data points

- Define a **kernel function** by projecting data to higher dimensions:

\[
k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle
\]
Kernelized SVM

- **Dual Quadratic Program** (usually max-min, not here)

\[
\max_{\alpha \geq 0} \sum_{l=1}^{M} \alpha_l - \frac{1}{2} \sum_{j,k=1}^{M} \alpha_j \alpha_k y_j y_k k(x_j, x_k)
\]

s.t. \(\sum_{l=1}^{M} \alpha_l y_l = 0\)

- **Representer theorem**: (classification test):

\[
f(z) = \sum_{l=1}^{M} \alpha_l y_l k(z, x_l) > 0
\]
Kernels

- Polynomial kernel

\[ k(x_1, x_2) = \left(1 + x_1^\top x_2\right) \]

- Radial kernel

\[ k(x_1, x_2) = \exp\left(-\gamma\|x_1 - x_2\|^2_{2}\right) \]

- Many many more
Polynomial and Radial Kernels
SVM vs LDA: Train

![ROC Curves for SVM and LDA](image)

- **Support Vector Classifier**
  - $\gamma = 10^{-3}$
  - $\gamma = 10^{-2}$
  - $\gamma = 10^{-1}$

- **LDA**

**False positive rate** vs **True positive rate**

- The ROC curves illustrate the trade-off between the true positive rate and the false positive rate.
- SVM outperforms LDA across different values of $\gamma$.
SVM vs LDA: Test

Support Vector Classifier
SVM: $\gamma = 10^{-3}$
SVM: $\gamma = 10^{-2}$
SVM: $\gamma = 10^{-1}$

Support Vector Classifier
LDA
Multiple Classes

- One-vs-one

- One-vs-all
**SVM vs Logistic Regression**

- **Logistic regression**: Minimize negative log likelihood
- **SVM**: Minimize hinge loss

Bottom line: use SVM when classes are better separated or there is a good *kernel*