STATISTICAL CLASSIFICATION

(5 Lectures)

@Yiming Yang, Lecture on SVM

5 Lectures

- Introduction (Lec 11) + Evaluation (Lec 14)
- SVM with Stochastic Gradient Descent (Lec 12)
- Logistic Regression (Lec 13)
- Large Scale Classification (Lec 15)

@Yiming Yang, Lecture on SVM
Task Definition

▪ In English
  ▪ Given an (arbitrary) document and a predefined set (or hierarchy) of class labels, assign the labels to the document based on relevance.

▪ In mathematical terms
  ▪ Given a d-dimensional real-valued vector, map it onto a K-dimensional space of binary-valued vectors.
    \[ f: X \rightarrow Y \text{ where } X \in \mathbb{R}^d, \ Y \in \{0,1\}^k \]
  ▪ Usually, we obtain a real-valued output vector first, and then obtain a yes/no decision for each class by applying a threshold to each cell of the output vector.

Terminology

▪ \( K = 2 \), binary classification (e.g., yes/no with respect to "red")
▪ \( K > 2 \), multi-label classification (e.g., "politics", "economics" and "China" as the category labels of a news story)
▪ \( K > 2 \) constrained on one label per instance, multi-class classification (e.g., choosing among “apple, orange, banana, ...” for a fruit)
### Terminology

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input variables</td>
<td>Output variables</td>
</tr>
<tr>
<td>Predictors</td>
<td>Responses</td>
</tr>
<tr>
<td>Independent variables</td>
<td>Dependent variables</td>
</tr>
<tr>
<td>Features</td>
<td>Classes or categories</td>
</tr>
<tr>
<td>Factors</td>
<td>Outcomes</td>
</tr>
</tbody>
</table>

---

**Linear Decision Boundary**

Figure 2.1: A classification example in two dimensions. The classes are coded as a binary variable—GREEN = 0, RED = 1—and then fit by linear regression. The line is the decision boundary defined by $X^T \beta = 0.5$.  

2/23/2019
Non-linear Decision Boundaries of kNN

Figure 2.2: The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable (GREEN = 0, RED = 1) and then fit by 15-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the 15-nearest neighbors.

2/23/2019 @Yiming Yang, Lecture on SVM

LDA (linear) vs. QDA (non-linear)

Figure 4.1: The left plot shows some data from three classes, with linear decision boundaries found by linear discriminant analysis. The right plot shows quadratic decision boundaries. These were obtained by finding linear boundaries in the five-dimensional space \(X_1, X_2, X_{12}, X_1^2, X_2^2\). Linear inequalities in this space are quadratic inequalities in the original space. (Hastie et al., ESL)
Binary Classification Problem

- Given $D = \{(x_i, y_i): i = 1 \text{ to } n\}$ the labeled training instances
- We learn the mapping
  \[ f_w: X \rightarrow Y \quad \text{where} \quad X \in \mathbb{R}^{d+1}, w \in \mathbb{R}^{d+1}, Y \in \{0,1\} \]
- By solving the optimization problem
  \[ \hat{w} = \arg\min_w \{\text{Loss}(D, w) + C\|w\|\} \]
- The 1\textsuperscript{st} term is the training-set loss (measuring how well the model fits the data)
- The 2\textsuperscript{nd} term is the regularization term (controlling the model complexity)

Linearly separable data in 2-d Space

We want the classifier (a line) to separate the two classes of dots. But, there are so many ways to draw the line ...

Which one is the best?
How would we measure the “risk”?

Heuristic: the larger distance b/w the two dashed parallel lines the smaller the risk

Hyperplane Properties

1) A hyperplane is defined as \( h = \{ x : w^T x = b \} \).
2) Each \( h \) is specified by its normal vector \( w \) and its intercept \( d(\text{origin}, h) = \frac{b}{\| w \|} \).
3) The distance from any point to \( h \) is \( d(x, h) = \frac{y_i f(x)}{\| w \|} \).

References:
- C.M. Bishop, Pattern Recognition and Machine Learning, page 181-182
- http://www.youtube.com/watch?v=AUzQg79gKjQ
Support Vectors & Margin

- Support vectors (for linearly separable problems)
  - \( yf(x) = 1 \)
- Margin: the distance from any support vector to \( h \)
  \[ \gamma \equiv \frac{yf(x)}{\|w\|} = \frac{1}{\|w\|} \]

Maximizing the margin is equivalent to minimize \( \|w\| \).

Soft Margin Linear SVM:

to cope with linearly non-separable data

- The objective is relaxed as
  \[
  \min_{w,b,\xi} \left\{ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \right\} \quad C > 0
  \]
  subject to:
  \[
  y_i (w^T x_i - b) \geq 1 - \xi_i \quad \text{for } i = 1, \ldots, n
  \]
  \[ \xi_i \geq 0 \quad \text{("slack variable")} \]

- Hinge Loss
  \[
  \{ \xi_i \geq 0 \} \wedge \{ \xi_i \geq 1 - y_i (w^T x_i - b) \} \\
  \Rightarrow \xi_i \geq \max \left\{ 0, 1 - y_i (w^T x_i - b) \right\} = \left( 1 - y_i (w^T x_i - b) \right)_+
  \]
**Extended Def. of Support Vectors**

- Support Vectors are those receiving a non-zero weights in the trained SVM.
- KKT conditions of SVM for every training data point $i$:
  $$
  \alpha_i (y_i f(x_i) - 1 + \xi_i) = 0, \quad \xi_i \geq 0
  $$

- For $\alpha_i > 0$ we need to satisfy:
  $$
  y_i f(x_i) - 1 + \xi_i = 0, \quad \xi_i \geq 0
  $$

  i.e. $\xi_i = \max(0, y_i f(x_i) - 1)$.

---

**Loss Functions of Classifiers**

(Hastie, lecture 2003)

Assume classification threshold at $f_i = 0$:

- $y_i f_i \geq 0$ correct classification
- $y_i f_i < 0$ wrong classification
- $y_i f_i \geq 1$ reinforcing a large margin
- $f_i \geq 1 - \xi_i$ reinforcing a soft margin

$$
\xi_i \geq 1 - y_i f_i \quad \text{equivalent constraint}
$$

Logistic Regression (rescaled by a factor of $\frac{1}{\ln(2)}$)
The trained SVM Solution

- a linear function of the input variables
  \[ f(x) = \hat{\omega}^T x - \hat{b} \]
  where \( \hat{\omega} = \sum_i \alpha_i y_i x_i \)
  \( \hat{b} = \hat{\omega}^T x_i - y_i \) using any support vector with \( y_i \hat{\omega}^T x_i = 1 \)

- Classification Rule
  \[ \hat{y}(x) = \text{sign}(\hat{\omega}^T x - \hat{b}) = \text{sign} \left( \sum_{i: \alpha_i \neq 0} \alpha_i y_i (x_i \cdot x - \hat{b}) \right) \]
  primal form          dual form

From Linear SVM to Non-linear SVM

- Linear SVM
  \[ h(x) = \text{sign} \left\{ w \cdot x - b \right\} = \text{sign} \left\{ \sum_{i=1}^{n} \alpha_i y_i (x_i \cdot x) - b \right\} \]
  Linear Kernel

- Non-linear SVM
  \[ h(x) = \text{sign} \left\{ \sum_{i=1}^{n} \alpha_i y_i (\phi(x_i) \cdot \phi(x)) - b \right\} \]
  Non-linear Kernel
Example of a non-linear Kernel

A kernel function is defined as:

$$K(u, x) = \phi(u) \cdot \phi(x)$$

where $u \in R^m$, $x \in R^m$, and $\phi(.) \in R^M$, $M > m$

Example:

$$K(u, x) = (u \cdot x + 1)^2 \quad u = (u_1, u_2, u_3), \quad x = (x_1, x_2, x_3)$$

$$\phi(u) = ?$$

$$\phi(x) = ?$$
Non-linear SVM Solution

\[ h(\mathbf{x}) = \text{sign} \left\{ \sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) - b \right\} \]

where \( K(u, x) = (u \cdot x + 1)^2 \)

\[ = \sum_{i=1}^{3} u_i^2 x_i^2 + 2 \sum_{i=1}^{2} \sum_{j=i+1}^{3} u_i u_j x_i x_j + \sum_{i=1}^{3} u_i x_i + 1 \]

quadratic part linear part

Ex. Data are not linearly separable in 1-d

Harder 1-dimensional dataset

That’s wiped the smirk off SVM’s face.

What can be done about this?
L2H: Making it easier to separate the data

Harder 1-dimensional dataset

Remember how permitting non-linear basis functions made linear regression so much nicer?

Let's permit them here too

\[ z_k = (x_k, x_k^2) \]

Dimension from L to H

\[ K(u, v) = (u \cdot v)^2 = \phi(u) \cdot \phi(v), \quad u, v \in \mathbb{R}^m \]

\[
K(u, v) = (\underbrace{u_1 v_1 + u_2 v_2 + \cdots + u_m v_m}_{= z_1} )^2 \\
= (z_1 + z_2 + \cdots + z_m)^2 = \sum_{j=1}^{m} z_j^2 + 2 \sum_{j=1}^{m} \sum_{j \neq j} z_j z_{j'}
\]

\[ \phi(u) = (\underbrace{u_1^2, u_2^2, \cdots, u_m^2}_{m \choose 1}, \sqrt{2} u_1 u_2, \sqrt{2} u_1 u_3, \cdots, \sqrt{2} u_1 u_{m-1}) \]

\[ \dim(\phi) = m + \frac{1}{2} m(m-1) \]
Dimension from L to H (cont'd)

\[ K(u, v) = (u \cdot v)^3 = (z_1 + z_2 + \cdots + z_m)^3, \quad z_j \equiv u_j v_j \]

\[
= \sum_{j=1}^{m} z_j^3 + a \sum_{j=1}^{m} \sum_{j' \neq j} z_j z_{j'}^2 + b \sum_{j=1}^{m} \sum_{j' \neq j} \sum_{j'' \neq j, j'' \neq j'} z_j z_{j'} z_{j''}
\]

\[
\phi(u) = (u_1^3, u_2^3, \cdots, \sqrt{a} u_1^2 u_2, \sqrt{a} u_1^2 u_3, \cdots, \sqrt{b} u_1 u_2 u_3, \sqrt{b} u_1 u_2 u_4, \cdots)
\]

\[
\dim(\phi) = m + \frac{m(m-1)}{2!} + \frac{m(m-1)(m-2)}{3!}
\]

---

Dimension from L to H (cont'd)

\[ K(u, v) = (u \cdot v)^d = (z_1 + z_2 + \cdots + z_m)^d, \quad z_j \equiv u_j v_j \]

\[
= \sum_{j=1}^{m} z_j^d + a \sum_{j=1}^{m} \sum_{j' \neq j} z_j z_{j'}^{d-1} z_{j''} + \cdots + c \sum_{j_1 \neq j_2 \neq \cdots \neq j_m} z_{j_1} z_{j_2} \cdots z_{j_m}
\]

\[
\phi(u) = (u_1^d, u_2^d, \cdots, \sqrt{a} u_1^{d-1} u_2, \sqrt{a} u_1^{d-1} u_3, \cdots, \sqrt{b} u_1 u_2 u_3, \cdots, \sqrt{b} u_1 u_2 u_4, \cdots)
\]

\[
\dim(\phi) = m + \frac{m(m-1)}{2!} + \cdots + \frac{m!}{d!(d-m)!} = O\left(\frac{m^d}{d!}\right) \quad (m \geq d)
\]
Time Complexity of Kernel Computation

<table>
<thead>
<tr>
<th>Degree of Polynomial</th>
<th>Quadratic</th>
<th>Cubic</th>
<th>Quartic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dim(φ(χ))</td>
<td>m² / 2</td>
<td>m³ / 6</td>
<td>m⁴ / 24</td>
</tr>
<tr>
<td>Cost to compute φₜ, φᵥ in H</td>
<td>m² / 2</td>
<td>m³ / 6</td>
<td>m⁴ / 24</td>
</tr>
<tr>
<td>K( u, v)</td>
<td>(uᵀv+1)²</td>
<td>(uᵀv+1)³</td>
<td>(uᵀv+1)⁴</td>
</tr>
<tr>
<td>Cost of directly computing K( u, v)</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Time Saving (row 2 / row 4)</td>
<td>m² / 2</td>
<td>m² / 6</td>
<td>m³ / 24</td>
</tr>
<tr>
<td>Time saving if m = 1000</td>
<td>500</td>
<td>167,000</td>
<td>417,000,000</td>
</tr>
</tbody>
</table>

Time saving if m = 1000:
- Quadratic: 500
- Cubic: 167,000
- Quartic: 417,000,000

Commonly Used Kernels

- Linear: \( K(\tilde{u}, \tilde{v}) = \tilde{u} \cdot \tilde{v} \)
- Polynomial of degree d: \( K(\tilde{u}, \tilde{v}) = (\tilde{u} \cdot \tilde{v} + c)^d \)
- Gaussian Radial Basis Function (RBF):
  \( K(\tilde{u}, \tilde{v}) = e^{-γ||\tilde{u} - \tilde{v}||^2} \)
- Two-layer sigmoidal neural network
  \( K(\tilde{u}, \tilde{v}) = \tanh(k\tilde{u} \cdot \tilde{v} + c) \)
Summary of Kernels for SVM

- Mapping data from a low dimensional space (L) to a high dimensional space (H)
- Making hard problems (non-linear decision boundaries) easy to solve in H, hopefully
- Using the same optimization algorithms as in solving a linear problem
- Choosing kernels which can be computed efficiently

Typical Optimization Methods for SVM (next lecture)

<table>
<thead>
<tr>
<th>Method</th>
<th>Objective</th>
<th>Large # of Examples</th>
<th>Non-linear Modeling</th>
<th>Large # of Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interior Point</td>
<td>Primal</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Subgradient descent</td>
<td>Primal</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Subgradient stochastic descent with projection</td>
<td>Primal</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Dual co-ordinate descent with projection (for linear SVM)</td>
<td>Dual</td>
<td>~</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Dual QP with active set</td>
<td>Dual</td>
<td>~+</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Primal-Dual solver</td>
<td>~++</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Pegasos: SGD solver for SVMs (Shalev-Shwartz et al., 2011), 10 lines!
References


@Yiming Yang, Lecture on SVM

References


@Yiming Yang, Lecture on SVM