Lecture 5: SVM and Logistic Regression

Hsuan-Tien Lin (林軒田)
htlin@csie.ntu.edu.tw

Department of Computer Science & Information Engineering
National Taiwan University (國立台灣大學資訊工程系)
Lecture 5: SVM and Logistic Regression

- Soft-Margin SVM as Regularization
- SVM versus Logistic Regression
- SVM for Soft Classification
- Kernel Logistic Regression
**Wrap-Up**

**Hard-Margin Primal**

\[
\begin{align*}
\min_{b,w} & \quad \frac{1}{2} w^T w \\
\text{s.t.} & \quad y_n(w^T z_n + b) \geq 1
\end{align*}
\]

**Soft-Margin Primal**

\[
\begin{align*}
\min_{b,w,\xi} & \quad \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n \\
\text{s.t.} & \quad y_n(w^T z_n + b) \geq 1 - \xi_n
\end{align*}
\]

**Hard-Margin Dual**

\[
\begin{align*}
\min_{\alpha} & \quad \frac{1}{2} \alpha^T Q \alpha - 1^T \alpha \\
\text{s.t.} & \quad y^T \alpha = 0 \\
& \quad 0 \leq \alpha_n
\end{align*}
\]

**Soft-Margin Dual**

\[
\begin{align*}
\min_{\alpha} & \quad \frac{1}{2} \alpha^T Q \alpha - 1^T \alpha \\
\text{s.t.} & \quad y^T \alpha = 0 \\
& \quad 0 \leq \alpha_n \leq C
\end{align*}
\]

*soft-margin preferred in practice;*  
linear: LIBLINEAR; non-linear: LIBSVM
Slack Variables $\xi_n$

- record ‘margin violation’ by $\xi_n$
- penalize with margin violation

$$
\min_{b,w,\xi} \frac{1}{2} w^T w + C \cdot \sum_{n=1}^{N} \xi_n \\
\text{s.t.} \quad y_n(w^T z_n + b) \geq 1 - \xi_n \text{ and } \xi_n \geq 0 \text{ for all } n
$$

on any $(b, w)$, $\xi_n = \text{margin violation} = \max(1 - y_n(w^T z_n + b), 0)$

- $(x_n, y_n)$ violating margin: $\xi_n = 1 - y_n(w^T z_n + b)$
- $(x_n, y_n)$ not violating margin: $\xi_n = 0$

‘unconstrained’ form of soft-margin SVM:

$$
\min_{b,w} \frac{1}{2} w^T w + C \sum_{n=1}^{N} \max(1 - y_n(w^T z_n + b), 0)
$$
Unconstrained Form

\[
\min_{b,w} \frac{1}{2} w^T w + C \sum_{n=1}^{N} \max(1 - y_n(w^T z_n + b), 0)
\]

familiar? :-) 

\[
\min \frac{1}{2} w^T w + C \sum \hat{\text{err}}
\]

just L2 regularization

\[
\min \frac{\lambda}{N} w^T w + \sum \text{err}
\]

with shorter \(w\), another parameter, and special \(\text{err}\)

why not solve this? :-) 

- not QP, no (?) kernel trick
- \(\max(\cdot, 0)\) not differentiable, harder to solve
SVM as Regularization

<table>
<thead>
<tr>
<th></th>
<th>minimize</th>
<th>constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>regularization by constraint</td>
<td>$E_{in}$</td>
<td>$w^T w \leq C$</td>
</tr>
<tr>
<td>hard-margin SVM</td>
<td>$w^T w$</td>
<td>$E_{in} = 0$ [and more]</td>
</tr>
<tr>
<td>L2 regularization</td>
<td>$\frac{\lambda}{N} w^T w + E_{in}$</td>
<td></td>
</tr>
<tr>
<td>soft-margin SVM</td>
<td>$\frac{1}{2} w^T w + C \hat{E}_{in}$</td>
<td></td>
</tr>
</tbody>
</table>

large margin $\iff$ fewer hyperplanes $\iff$ L2 regularization of short $w$

soft margin $\iff$ special $\hat{\text{err}}$

larger $C$ or $C$ $\iff$ smaller $\lambda$ $\iff$ less regularization

viewing SVM as regularization:

allows *extending/connecting* to other learning models
Fun Time
Error Function of SVM

\[
\min_{b, w} \quad \frac{1}{2} w^T w + C \sum_{n=1}^{N} \max (1 - y_n (w^T z_n + b), 0)
\]

linear score \( s = w^T z_n + b \)

- \( \text{err}_{0/1}(s, y) = [ys \neq 1] \)
- \( \hat{\text{err}}_{\text{SVM}}(s, y) = \max(1 - ys, 0) \): upper bound of \( \text{err}_{0/1} \)
  —often called **hinge error measure**

\( \hat{\text{err}}_{\text{SVM}} \): **algorithmic error measure**

by **convex upper bound** of \( \text{err}_{0/1} \)
Error Function of SVM

\[
\begin{align*}
\min_{b,w} & \quad \frac{1}{2} w^T w + C \sum_{n=1}^{N} \max(1 - y_n(w^T z_n + b), 0) \\
\end{align*}
\]

linear score \( s = w^T z_n + b \)

- \( \text{err}_{0/1}(s, y) = [y s \neq 1] \)
- \( \hat{\text{err}}_{\text{SVM}}(s, y) = \max(1 - y s, 0) \): upper bound of \( \text{err}_{0/1} \)
  —often called **hinge error measure**

\( \hat{\text{err}}_{\text{SVM}} \): algorithmic error measure by convex upper bound of \( \text{err}_{0/1} \)
Connection between SVM and Logistic Regression

linear score \( s = w^T z_n + b \)

- \( \text{err}_{0/1}(s, y) = [ys \neq 1] \)
- \( \hat{\text{err}}_{\text{SVM}}(s, y) = \max(1 - ys, 0) \): upper bound of \( \text{err}_{0/1} \)
- \( \text{err}_{\text{SCE}}(s, y) = \log_2(1 + \exp(-ys)) \): another upper bound of \( \text{err}_{0/1} \) used in logistic regression

\[
\begin{align*}
\hat{\text{err}}_{\text{SVM}}(s, y) &\approx -ys \\
(\ln 2) \cdot \text{err}_{\text{SCE}}(s, y) &\approx 0 \\
\end{align*}
\]

**SVM** \( \approx \) **L2-regularized logistic regression**
### Linear Models for Binary Classification

<table>
<thead>
<tr>
<th>PLA</th>
<th>soft-margin SVM</th>
<th>regularized logistic regression for classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimize $\text{err}_{0/1}$ specially</td>
<td>minimize regularized $\hat{\text{err}}_{\text{SVM}}$ by QP</td>
<td>minimize regularized $\text{err}_{\text{SCE}}$ by GD/SGD/...</td>
</tr>
<tr>
<td>• pros: efficient if lin. separable</td>
<td>• pros: ‘easy’ optimization &amp; theoretical guarantee</td>
<td>• pros: ‘easy’ optimization &amp; regularization guard</td>
</tr>
<tr>
<td>• cons: works only if lin. separable, otherwise needing pocket</td>
<td>• cons: loose bound of $\text{err}_{0/1}$ for very negative $y$s</td>
<td>• cons: loose bound of $\text{err}_{0/1}$ for very negative $y$s</td>
</tr>
</tbody>
</table>

regularized LogReg $\Rightarrow$ approximate SVM

SVM $\Rightarrow$ approximate LogReg (?)
Fun Time
SVM for Soft Classification

Naïve Idea 1

1. run SVM and get 
   \((b_{SVM}, w_{SVM})\)
2. return 
   \(g(x) = \theta(w_{SVM}^T x + b_{SVM})\)

- ‘direct’ use of similarity — works reasonably well
- no LogReg flavor

Naïve Idea 2

1. run SVM and get 
   \((b_{SVM}, w_{SVM})\)
2. run LogReg with 
   \((b_{SVM}, w_{SVM})\) as \(w_0\)
3. return LogReg solution as 
   \(g(x)\)

- not really ‘easier’ than original LogReg
- SVM flavor (kernel?) lost

want: flavors from both sides
A Possible Model: Two-Level Learning

\[ g(x) = \theta (A \cdot (w_{SVM}^T \Phi(x) + b_{SVM}) + B) \]

- **SVM flavor**: fix hyperplane direction by \( w_{SVM} \)—kernel applies
- **LogReg flavor**: fine-tune hyperplane to match maximum likelihood by scaling (\( A \)) and shifting (\( B \))
  - often \( A > 0 \) if \( w_{SVM} \) reasonably good
  - often \( B \approx 0 \) if \( b_{SVM} \) reasonably good

new LogReg Problem:

\[
\min_{A,B} \frac{1}{N} \sum_{n=1}^{N} \log \left( 1 + \exp \left( -y_n (A \cdot (w_{SVM}^T \Phi(x_n) + b_{SVM}) + B) \right) \right)
\]

two-level learning: LogReg on SVM-transformed data
Platt’s Model of Probabilistic SVM for Soft Classification

1. run **SVM** on $\mathcal{D}$ to get $(b_{\text{SVM}}, w_{\text{SVM}})$ [or the equivalent $\alpha$], and transform $\mathcal{D}$ to $z_n' = w_{\text{SVM}}^T \Phi(x_n) + b_{\text{SVM}}$.
   —actual model performs this step more sophisticatedally

2. run **LogReg** on $\{(z_n', y_n)\}_{n=1}^N$ to get $(A, B)$.
   —actual model adds some special regularization here

3. return $g(x) = \theta(A \cdot (w_{\text{SVM}}^T \Phi(x) + b_{\text{SVM}}) + B)$

- **soft classifier** not having the same boundary as **SVM classifier**
  —because of $B$

- how to solve **LogReg**: GD/SGD/or better
  —because only **two variables**

**kernel SVM** $\mapsto$ approx. LogReg in $\mathcal{Z}$-space

exact LogReg in $\mathcal{Z}$-space?
Fun Time
**Key behind Kernel Trick**

One key behind kernel trick: optimal $\mathbf{w}_* = \sum_{n=1}^{N} \beta_n \mathbf{z}_n$

because $\mathbf{w}_*^T \mathbf{z} = \sum_{n=1}^{N} \beta_n \mathbf{z}_n^T \mathbf{z} = \sum_{n=1}^{N} \beta_n K(\mathbf{x}_n, \mathbf{x})$

**SVM**

$$\mathbf{w}_\text{SVM} = \sum_{n=1}^{N} (\alpha_n y_n) \mathbf{z}_n$$

$\alpha_n$ from dual solutions

**PLA**

$$\mathbf{w}_\text{PLA} = \sum_{n=1}^{N} (\alpha_n y_n) \mathbf{z}_n$$

$\alpha_n$ by # mistake corrections

**LogReg by SGD**

$$\mathbf{w}_\text{LOGREG} = \sum_{n=1}^{N} (\alpha_n y_n) \mathbf{z}_n$$

$\alpha_n$ by total SGD moves

When can optimal $\mathbf{w}_*$ be represented by $\mathbf{z}_n$?
Representer Theorem

Claim: for any L2-regularized linear model

\[
\min_w \frac{\lambda}{N} w^T w + \frac{1}{N} \sum_{n=1}^{N} \text{err}(y, w^T z_n)
\]

Optimal \( w_* = \sum_{n=1}^{N} \beta_n z_n \).

- Let optimal \( w_* = w_\parallel + w_\perp \), where \( w_\parallel \in \text{span}(z_n) \) & \( w_\perp \perp \text{span}(z_n) \)
  —want \( w_\perp = 0 \)

- What if not? Consider \( w_\parallel \)
  - of same err as \( w_*: \text{err}(y, w_*^T z_n) = \text{err}(y, (w_\parallel + w_\perp)^T z_n) \)
  - of smaller regularizer as \( w_*: w_*^T w_* = w_\parallel^T w_\parallel + 2w_\parallel^T w_\perp + w_\perp^T w_\perp > w_\parallel^T w_\parallel \)
  —\( w_\parallel \) ‘more optimal’ than \( w_* \) (contradiction!)

Any L2-regularized linear model can be kernelized!
Kernel Logistic Regression

Solving L2-regularized logistic regression

\[
\min_w \frac{\lambda}{N} w^T w + \frac{1}{N} \sum_{n=1}^{N} \log \left( 1 + \exp \left( -y_n w^T z_n \right) \right)
\]

Yields optimal solution \( w_* = \sum_{n=1}^{N} \beta_n z_n \)

With out loss of generality, can solve for optimal \( \beta \) instead of \( w \)

\[
\min_{\beta} \frac{\lambda}{N} \sum_{n=1}^{N} \sum_{m=1}^{N} \beta_n \beta_m K(x_n, x_m) + \frac{1}{N} \sum_{n=1}^{N} \log \left( 1 + \exp \left( -y_n \sum_{m=1}^{N} \beta_m K(x_m, x_n) \right) \right)
\]

—How? GD/SGD/… for unconstrained optimization

Kernel logistic regression:

Use representer theorem for kernel trick on L2-regularized logistic regression
Kernel Logistic Regression: Another View

\[
\min_{\beta} \frac{1}{N} \sum_{n=1}^{N} \sum_{m=1}^{N} \beta_n \beta_m K(x_n, x_m) + \frac{1}{N} \sum_{n=1}^{N} \log \left( 1 + \exp \left( -y_n \sum_{m=1}^{N} \beta_m K(x_m, x_n) \right) \right)
\]

- \( \sum_{m=1}^{N} \beta_m K(x_m, x_n) \): inner product between variables \( \beta \) and transformed data \((K(x_1, x_n), K(x_2, x_n), \ldots, K(x_N, x_n))\)
- \( \sum_{n=1}^{N} \sum_{m=1}^{N} \beta_n \beta_m K(x_n, x_m) \): a special regularizer \( \beta^T K \beta \)
- KLR = linear model of \( \beta \) with kernel transform & kernel regularizer;  
  = linear model of \( w \)  
  with kernel-embedded transform & L2 regularizer
- similar for SVM

\textbf{different routes to the same destination}  
—allows extension from different views
SVM and Logistic Regression

Fun Time
Summary

Lecture 5: SVM and Logistic Regression

- Soft-Margin SVM as Regularization
  L2-regularization with hinge error measure
- SVM versus Logistic Regression
  \( \approx \) L2-regularized logistic regression
- SVM for Soft Classification
  common approach: two-level learning
- Kernel Logistic Regression
  representer theorem on L2-regularized LogReg