ECE544NA: Logistic Regression and Multivariate Logistic Regression

Raymond Yeh

University of Illinois at Urbana Champaign

yeh17@illinois.edu

September 8, 2016
Overview

1. Support Vector Machine
2. Supervised Learning Example
3. Logistic Regression
4. Multinomial Logistic Regression
5. Review
From last lecture, we formulated the SVM optimization as

\[
\vec{w}^* = \arg \min_{\vec{w}} \|\vec{w}\|^2
\]  

subject to \( y_i \vec{w}^\top \vec{x}_i \geq 1, \forall i \in \{1, ..., N\} \)

Consider the following example,

Figure: Example by A. Zisserman
Intuitively there should be a trade off between the margin and classification accuracy.

Introduce a slack variable, $\xi_i$, to control the trade off, by allowing some examples to be within the margin or misclassified.

Then the optimization problem becomes,

$$\vec{w}^* = \arg \min_{\vec{w}, \xi_i \in \mathbb{R}^+} \| \vec{w} \|_2^2 + C \sum_i \xi_i$$

subject to $y_i \vec{w}^\top \vec{x}_i \geq 1 - \xi_i, \forall i \in \{1, ..., N\}$

Observe that when $0 < \xi_i < 1$, $x_i$ is within the margin, and when $\xi_i > 1$, the $x_i$ is misclassified.
Observe that we can rewrite the constraint to $\xi_i \geq 1 - y_i \vec{w}^\top \vec{x}_i$.

Combining with constraint $\xi_i \geq 0$, we can write $\xi_i = \max(0, 1 - y_i \vec{w}^\top \vec{x}_i)$

Finding $\vec{w}^*$ becomes an unconstrained optimization problem.

$$\vec{w}^* = \arg\min_{\vec{w}} \|\vec{w}\|^2 + C \sum_i \max(0, 1 - y_i \vec{w}^\top \vec{x}_i)$$

Observe that the first term controls the margin, while the second term controls the classification accuracy.
Let’s say we want to predict whether a student will get an A on the ECE544NA final exam. We have data from previous semester,

<table>
<thead>
<tr>
<th>id</th>
<th>Hours studied</th>
<th>HW grade</th>
<th>Favorite animal</th>
<th>Final grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>90</td>
<td>dog</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>100</td>
<td>elephant</td>
<td>A+</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>50</td>
<td>zebra</td>
<td>B</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

1. How to obtain the labels $y_i$? 
2. How to construct the feature vector $\vec{x}_i$? 
3. Which model to choose and how to optimize it?
Supervised Learning Problem

1. Training Examples \( D = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots (\mathbf{x}_N, y_N) \)
2. Model \( g : \mathcal{R}^d \mapsto \{0, 1\} \),
3. Denote the prediction as \( \hat{y}_i = g(\mathbf{x}_i; \mathbf{w}) \)
4. For binary classification, \( E[min_w \mathbf{1}[\hat{y}! = y]] \)
Given a training examples, $D = (\vec{x}_1, y_1), (\vec{x}_2, y_2), ... (\vec{x}_N, y_N)$, where $\vec{x} \in \mathcal{R}^d$, and $y \in \{0, 1\}$. We hope to learn a function $g : \mathcal{R}^d \mapsto \{0, 1\}$, where $g$ is a “good” predictor.

Recall, we have defined the logistic regression to have the form

$$\hat{y}_i = g(\vec{x}_i; \vec{w}) = \frac{1}{1 + e^{-\vec{w}^T \vec{x}_i}}$$

where $\hat{y}_i$ is the prediction given input $\vec{x}_i$, and $\vec{w} \in \mathcal{R}^d$ is model parameter.
1. Why use a sigmoid function? (Hint: what is the range of $y$)

2. Assume $P[Y = 1|X = \bar{x}_i] = \hat{y}_i$, and $P[Y = 0|X = \bar{x}_i] = 1 - \hat{y}_i$, then we can compute the likelihood:

$$
\prod_{i=1}^{N} P[Y = y_i|X = \bar{x}_i] = \prod_{i=1}^{N} (\hat{y}_i)^{y_i} \cdot (1 - \hat{y}_i)^{(1-y_i)} 
$$

Then the log likelihood is:

$$
\log(\prod_{i=1}^{N} P[Y = y_i|X = \bar{x}_i]) = \sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)
$$
1. We want to find the model parameters, \( \vec{w} \), such that the likelihood of training examples, \( D \), given the model is maximized.

2. Converting the likelihood maximization problem to a minimization problem. Simply minimize the negative log likelihood.

\[
\vec{w}^* = \arg \min_{\vec{w}} \left( - \sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \right) \tag{7}
\]
Logistic Regression: Convexity

1. We will show that the negative log likelihood,
\[ \sum_{i=1}^{N} y_i \cdot -\log(\hat{y}_i) + (1 - y_i) \cdot -\log(1 - \hat{y}_i) \] 

, is convex with respect to \( \vec{w} \).

2. \( f \) is called convex if:
\[ \forall \vec{x}_1, \vec{x}_2, t \in [0, 1] : f(t\vec{x}_1 + (1 - t)\vec{x}_2) \leq t \cdot f(\vec{x}_1) + (1 - t) \cdot f(\vec{x}_2) \] 

3. A twice differentiable function of several variables is convex on a convex set if and only if its Hessian matrix is positive semidefinite.

4. Linear combination of convex functions with nonnegative coefficients is also convex.

5. Therefore, showing \(-\log(\hat{y}_i)\) and \(-\log(1 - \hat{y}_i)\) are convex, proves the overall negative log likelihood is convex.
Recall,

\[- \log(\hat{y}_i) = - \log\left(\frac{1}{1 + e^{-\vec{w}^\top \vec{x}_i}}\right) = \log(1 + e^{-\vec{w}^\top \vec{x}_i}) \]  

(10)

Gradient:

\[
\nabla_{\vec{w}} [\log(1 + e^{-\vec{w}^\top \vec{x}_i})] = \frac{-e^{-\vec{w}^\top \vec{x}_i}}{1 + e^{-\vec{w}^\top \vec{x}_i}} \cdot \vec{x}_i = \left(\frac{1}{1 + e^{-\vec{w}^\top \vec{x}_i}} - 1\right) \cdot \vec{x}_i \]  

(11)

Hessian:

\[
\nabla_{\vec{w}}^2 (- \log(\hat{y}_i)) = \nabla_{\vec{w}}((\hat{y}_i - 1) \cdot \vec{x}_i) = \frac{e^{-\vec{w}^\top \vec{x}_i}}{(1 + e^{-\vec{w}^\top \vec{x}_i})^2} \vec{x} \vec{x}^\top = (\hat{y}_i)(1 - \hat{y}_i)\vec{x}_i\vec{x}_i^\top \]  

(12)
Logistic Regression: Convexity

1. Prove Hessian is positive semi-definite: \( \forall \vec{z}, \)

\[
\vec{z}^T \left[ \nabla^2_w (\log \hat{y}_i) \vec{x}_i^T \right] \vec{z} = \vec{z}^T \left[ (\hat{y}_i)(1 - \hat{y}_i) \vec{x}_i \vec{x}_i^T \right] \vec{z} \quad (13)
\]

\[
= (\hat{y}_i)(1 - \hat{y}_i)(\vec{z}^T \vec{x}_i)(\vec{x}_i^T \vec{z}) \quad (14)
\]

\[
= (\hat{y}_i)(1 - \hat{y}_i)(\vec{x}_i^T \vec{z})^T(\vec{x}_i^T \vec{z}) \quad (15)
\]

\[
= (\hat{y}_i)(1 - \hat{y}_i)(\vec{x}_i^T \vec{z})^2 \geq 0 \quad (16)
\]

2. Convexity proof for \(- \log(1 - \hat{y}_i)\) is left as an exercise.
Given a training examples, \( D = (\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_N, y_N) \), where \( \vec{x} \in \mathbb{R}^d \), and \( y \in \{1, 2, \ldots, K\} \). We want to estimate \( P[Y = 1|X] \), \ldots, \( P[Y = K|X] \).

The multinomial logistic regression has the form

\[
\hat{\vec{y}}_i = g(\vec{x}_i; W) = \begin{bmatrix} \hat{y}_i[1] \\ \hat{y}_i[2] \\ \vdots \\ \hat{y}_i[K] \end{bmatrix} = \frac{1}{\sum^K \exp(\vec{w}_i^T \vec{x}_i)} \begin{bmatrix} \exp(\vec{w}_1^T \vec{x}_i) \\ \exp(\vec{w}_2^T \vec{x}_i) \\ \vdots \\ \exp(\vec{w}_k^T \vec{x}_i) \end{bmatrix}
\] (17)
Assume $P[Y = k|X = \bar{x}_i] = \hat{y}_i[k]$, then we can compute the likelihood as follows:

$$\prod_{i}^{N} P[Y = y_i|X = \bar{x}_i] = \prod_{i}^{N} \prod_{k}^{K} \hat{y}_i[k]^{1[y_i = k]}$$  \hspace{1cm} (18)$$

The log likelihood is

$$\sum_{i}^{N} \log(\prod_{k}^{K} \hat{y}_i[k]^{1[y_i = k]}) = \sum_{i}^{N} \sum_{k}^{K} 1[y_i = k] \cdot \log(\hat{y}_i[k])$$  \hspace{1cm} (19)$$
Binary classification problem
Zero-one loss: $1[y_i \neq \hat{y}_i]$ (Intractable, not differentiable, not convex)

Linear regression
Square loss: $||y_i - \hat{y}_i||^2$

Logistic Regression
Log loss: $-(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i))$ Note: $y \in \{0, 1\}$

Log loss: $\frac{1}{\ln(2)} \ln(1 + e^{-t_i \cdot \hat{t}})$ Note: $t \in \{-1, 1\}$, $y = (1 + t)/2$.

Perceptron
Hinge loss: $\max(0, -y_i \hat{y}_i)$

SVM
Hinge loss: $\max(0, 1 - y_i \hat{y}_i)$
Model Comparison
1 Principle Component Analysis
2 Python + Tensorflow Tutorial