ECE544NA: Logistic Regression and Multivariate Logistic Regression

Raymond Yeh

University of Illinois at Urbana Champaign

yeh17@illinois.edu

September 8, 2016

Overview

1. From last lecture, we formulated the SVM optimization as

$$\vec{w}^* = \arg\min_{w} ||\vec{w}||^2 \quad (1)$$

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

2. Consider the following example,

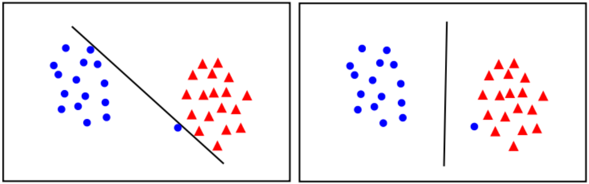


Figure : Example by A. Zisserman

1. Intuitively there should be a trade off between the margin and classification accuracy.
2. Introduce a slack variable, ξ_i, to control the trade off, by allowing some examples to be within the margin or misclassified.
3. Then the optimization problem becomes,

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

subject to $y_i \vec{w}^\intercal \vec{x_i} \geq 1 - \xi_i, \forall i \in \{1, ..., N\}$
4. Observe that when $0 < \xi_i < 1$, x_i is within the margin, and when $\xi_i > 1$, the x_i is misclassified.

Soft Margin SVM

1. Observe that we can rewrite the constraint to $\xi_i \geq 1 - y_i \vec{w}^\intercal \vec{x_i}$.
2. Combining with constraint $\xi_i \geq 0$, we can write $\xi_i = max(0, 1 - y_i \vec{w}^\intercal \vec{x_i})$
3. Finding $\vec{w}^*$ becomes an unconstrained optimization problem.

$$\vec{w}^* = \arg\min_w ||\vec{w}||^2 + C \sum_i max(0, 1 - y_i \vec{w}^\intercal \vec{x_i}) \quad (3)$$

4. Observe that the first term controls the margin, while the second term controls the classification accuracy.

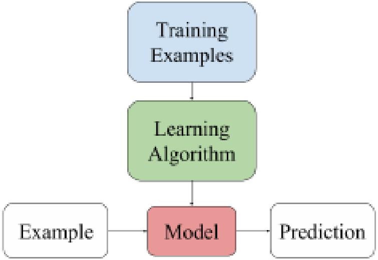
Supervised Learning Example

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,

id	Hours studied	HW grade	Favorite animal	Final grade
1	10	90	dog	A
2	20	100	elephant	A+
3	0	50	zebra	B
⋮	⋮	⋮	⋮	⋮

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

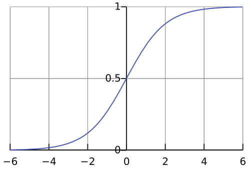


1. Training Examples $D = (\vec{x}_1, y_1), (\vec{x}_2, y_2), ...(\vec{x}_N, y_N)$
2. Model $g : \mathcal{R}^d \mapsto \{0, 1\}$,
3. Denote the prediction as $\hat{y}_i = g(\vec{x}_i; \vec{w})$
4. For binary classification, $\mathbf{E}[min_w \mathbf{1}[\hat{y}! = y]]$

1. Given a training examples, $D = (\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots (\vec{x}_N, y_N)$, where $\vec{x} \in \mathcal{R}^d$, and $y \in \{0, 1\}$. We hope to learn a function $g : \mathcal{R}^d \mapsto \{0, 1\}$, where g is a "good" predictor.
2. Recall, we have defined the logistic regression to have the form

$$\hat{y}_i = g(\vec{x}_i; \vec{w}) = \frac{1}{1 + e^{-\vec{w}^\intercal \vec{x}_i}} \quad (4)$$

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

$$\prod_{i=1}^{N} P[Y=y_i|X=\vec{x}_i]=\prod_{i=1}^{N}(\hat{y}_i)^{y_i}\cdot(1-\hat{y}_i)^{(1-y_i)} \tag{5}$$

Then the log likelihood is:

$$log(\prod_{i=1}^{N} P[Y=y_i|X=\vec{x}_i])=\sum_{i=1}^{N} y_i\cdot log(\hat{y}_i)+(1-y_i)\cdot log(1-\hat{y}_i) \tag{6}$$

1. We want 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.

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

1. We will show that the negative log likelihood,

$$\sum_{i=1}^{N} y_i \cdot -log(\hat{y}_i) + (1 - y_i) \cdot -log(1 - \hat{y}_i) \tag{8}$$

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

2. f is called convex if:

$$\forall \vec{x_1}, \vec{x_2}, t \in [0, 1] : f(t\vec{x_1} + (1 - t)\vec{x_2}) \leq t * f(\vec{x}_1) + (1 - t)f(\vec{x}_2) \tag{9}$$

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.

Logistic Regression: Convexity

1. Recall,

$$-\log(\hat{y}_i) = -\log(\frac{1}{1+e^{-\vec{w}^\intercal \vec{x_i}}}) = log(1+e^{-\vec{w}^\intercal \vec{x_i}}) \quad (10)$$

2. Gradient:

$$\nabla_{\vec{w}}[log(1+e^{-\vec{w}^\intercal \vec{x_i}})] = \frac{-e^{-\vec{w}^\intercal \vec{x_i}}}{1+e^{-\vec{w}^\intercal \vec{x_i}}} \cdot \vec{x_i} = (\frac{1}{1+e^{-\vec{w}^\intercal \vec{x_i}}} - 1) \cdot \vec{x_i} \quad (11)$$

3. Hessian:

$$\nabla^2_{\vec{w}}(-\log(\hat{y}_i)) = \nabla_{\vec{w}}((\hat{y}_i - 1) \cdot \vec{x_i}) = \frac{e^{-\vec{w}^\intercal \vec{x_i}}}{(1+e^{-\vec{w}^\intercal \vec{x_i}})^2} \vec{x}\vec{x}^\intercal = (\hat{y}_i)(1-\hat{y}_i)\vec{x_i}\vec{x_i}^\intercal \quad (12)$$

ILLINOIS

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

$$\vec{z}^{\intercal}[\nabla^2_{\vec{w}}(-\log(\hat{y}_i))\vec{x}^{\intercal}]\vec{z} = \vec{z}^{\intercal}[(\hat{y}_i)(1-\hat{y}_i)\vec{x}_i\vec{x}_i^{\intercal}]\vec{z} \quad (13)$$

$$= (\hat{y}_i)(1-\hat{y}_i)(\vec{z}^{\intercal}\vec{x})(\vec{x}^{\intercal}\vec{z}) \quad (14)$$

$$= (\hat{y}_i)(1-\hat{y}_i)(\vec{x}^{\intercal}\vec{z})^{\intercal}(\vec{x}^{\intercal}\vec{z}) \quad (15)$$

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

2. Convexity proof for $-log(1-\vec{y_i})$ is left as an exercise.

1. 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 $y \in \{1, 2, ...K\}$. We want to estimate $P[Y = 1|X]$, ..., $P[Y = K|X]$.
2. The multinomial logistic regression has the form

$$\hat{\vec{y}}_i = g(\vec{x}_i; W) = \begin{bmatrix} \hat{y}_i[1] \\ \hat{y}_i[2] \\ \vdots \\ \hat{y}_i[K] \end{bmatrix} = \frac{1}{\sum_l^K \exp(\vec{w}_l^\intercal \vec{x}_i)} \begin{bmatrix} \exp(\vec{w}_1^\intercal \vec{x}_i) \\ \exp(\vec{w}_2^\intercal \vec{x}_i) \\ \vdots \\ \exp(\vec{w}_k^\intercal \vec{x}_i) \end{bmatrix} \tag{17}$$

1. Assume $P[Y = k|X = \vec{x}_i] = \hat{y}_i[k]$, then we can compute the likelihood as follows:

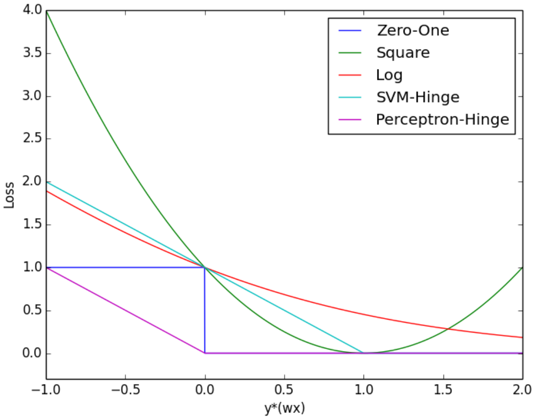
$$\prod_{i}^{N} P[Y = y_i|X = \vec{x}_i] = \prod_{i}^{N} \prod_{k}^{K} \hat{\vec{y}}_i[k]^{\mathbf{1}[y_i=k]} \tag{18}$$

2. The log likelihood is

$$\sum_{i}^{N} \log(\prod_{k}^{K} \hat{\vec{y}}_i[k]^{\mathbf{1}[y_i=k]}) = \sum_{i}^{N} \sum_{k}^{K} \mathbf{1}[y_i = k] \cdot \log(\hat{\vec{y}}_i[k]) \tag{19}$$

1. Binary classification problem
   Zero-one loss: $\mathbf{1}[y_i! = \hat{y}_i]$ (Intractable, not differentiable, not convex)
2. Linear regression
   Square loss: $||y_i - \hat{y}_i||^2$
3. 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: $\frac{1}{\ln(2)} \ln(1 + e_i^{-t_i \cdot \hat{t}})$ **Note:** $t \in \{-1, 1\}$, $y = (1 + t)/2$.
4. Perceptron
   Hinge loss: $\max(0, -y_i \hat{y}_i)$
5. SVM
   Hinge loss: $\max(0, 1 - y_i \hat{y}_i)$

Model Comparison



Next Lecture

ILLINOIS

1. Principle Component Analysis
2. Python + Tensorflow Tutorial