Lecture 9 - Feb 7

Logistic Regression
Assessing Classifiers

Reading

Week 5 Notes in GitHub

Data Mining and Machine Learning
24 Logistic Regression

Elements of Statistical Learning
4.4 Logistic Regression

Upcoming Deadlines

Homework 2 (Feb 18)

Given independent predictors $X_1, \ldots, X_d$ + categorical output Y, logistic regression tries to predict a probability distribution for Y

$P(c_i | x_j)$

$$X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nd} \end{pmatrix} \begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{matrix} \quad \text{with} \quad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \text{where } y_i \in \{0,1\}$$

$X_1 \dots X_d$ independent predictors

Bernoulli RV predictors

Let $X_1, \dots, X_d$ be independent predictors

$Y \in \{0,1\}$ Bernoulli RV.

D - dataset of $x_i \in \mathbb{R}^{d+1}$, $x_i = (1, x_{i1}, \dots, x_{id})$, y_i labels

$$\pi(x) = \mathbb{P}(Y=1 \mid X=x) \Rightarrow 1-\pi(x) = \mathbb{P}(Y=0 \mid X=x)$$

$\pi(x)$: unknown true probability

Goal: predict $\pi(x)$

A linear regression model $\pi(x) = \theta^T x$ is not appropriate since $\pi(x)$ is a probability in $[0,1]$.

$\Rightarrow$ we feed $\theta^T x$ into a function squishing it into $[0,1]$

The logistic (sigmoid) function $\sigma: \mathbb{R} \to (0,1)$ is

$$\sigma(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{1+e^z}$$

Properties:

(1) $0 < \sigma(z) < 1$ since $0 < \frac{1}{1+e^{-z}} < 1$, $e^{-z} > 0$

(2) $\sigma'(z) = \frac{e^z(1+e^z) - e^z(e^z)}{(1+e^z)^2} = \frac{e^z}{1+e^z} \cdot \frac{1}{1+e^z}$

$= \sigma(z) \cdot \sigma(-z) = \sigma(z)(1-\sigma(z)) > 0$ (σ is increasing)

(3) $\lim_{z \to -\infty} \sigma(z) = \lim_{z \to -\infty} \frac{1}{1+\frac{1}{e^z}} = 0$

(4) $\lim_{z \to \infty} \sigma(z) = \lim_{z \to \infty} \frac{1}{1+\frac{1}{e^z}} = \frac{1}{1+0} = 1$

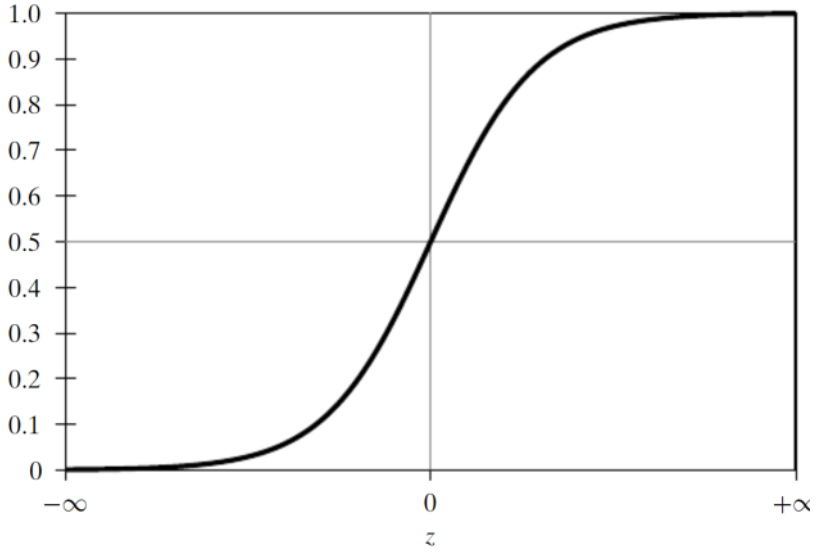


Figure 24.1. Logistic function.

Logistic regression model:

$\theta^T x$

Logistic regression

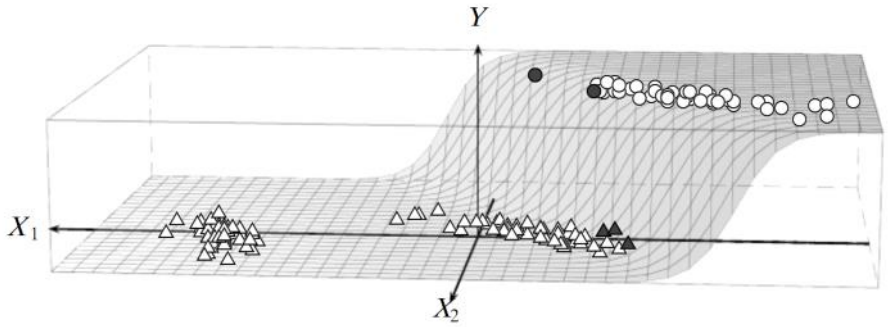
$$\left.\begin{aligned} \mathbb{P}(Y=1 \mid X=x) &= \sigma(\theta^T x) = \frac{e^{\theta^T x}}{1+e^{\theta^T x}} \\ \mathbb{P}(Y=0 \mid X=x) &= \sigma(-\theta^T x) = \frac{1}{1+e^{\theta^T x}} \end{aligned}\right\} \mathbb{P}(Y \mid X=x) = \sigma(\theta^T x)^Y \sigma(\theta^T x)^{1-Y}$$

The log-odds ratio for $Y=1$ is

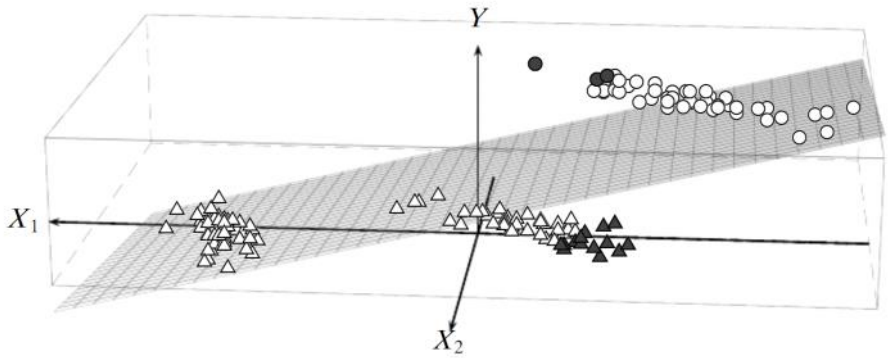
$$\log\left(\frac{\hat{\pi}(x)}{1-\hat{\pi}(x)}\right) = \log\left(e^{\theta^T x}\right) = \theta^T x$$

$\Rightarrow$ Logistic regression assumes log-odds are linear in θ.

prediction rule: $\hat{y} = \begin{cases} 1, & \text{if } \sigma(\theta^T x) \geq 0.5 \\ 0, & \text{if } \sigma(\theta^T x) < 0.5 \end{cases}$



(a) Logistic Regression



(b) Linear Regression

Figure 24.2. Logistic versus linear regression: Iris principal components data. Misclassified point are shown in dark gray color. Circles denote `Iris-virginica` and triangles denote the other two Iris types.

Likelihood is the probability of the observed data given the estimated parameters

$$L(\theta) = \mathbb{P}(Y|\theta) = \prod_{i=1}^{n} \mathbb{P}(y_i|x_i) = \prod_{i=1}^{n} \sigma(\theta^T x_i)^{y_i} \sigma(-\theta^T x_i)^{(1-y_i)}$$

We will solve $\max_\theta L(\theta)$ or, equivalently $\max_\theta \log(L(\theta))$ (log-likelihood)

$$\log(L(\theta)) = \log\left(\prod_{i=1}^{n} \sigma(\theta^T x_i)^{y_i} \sigma(-\theta^T x_i)^{(1-y_i)}\right)$$

$$= \sum_{i=1}^{n} y_i \log(\sigma(\theta^T x_i)) + (1-y_i)\log(\sigma(-\theta^T x_i))$$

Equivalently, minimize the cross-entropy error function

$$E(\theta) = -\log(L(\theta)) = -\sum_{i=1}^{n}\left(y_i \log(\underbrace{\sigma(\theta^T x_i)}_{\hat{y}_i(x_i)}) + (1-y_i)\log(\underbrace{\sigma(-\theta^T x_i)}_{1-\hat{y}_i(x_i)})\right)$$

Typically, we solve $\nabla E(\theta) = 0$ to try to solve $\min_\theta E(\theta)$, but there is no closed-form solution for θ, so we use gradient descent. An explicit gradient ∇E speeds it up, so let's find it.

$$\frac{\partial}{\partial \sigma(z_i)}\{\log(\sigma(z_i))\} = \frac{1}{\sigma(z_i)}$$

$$\frac{\partial}{\partial \sigma(z_i)}\{\log(\sigma(-z_i))\} = \frac{\partial}{\partial \sigma(z_i)}\{\log(1-\sigma(z_i))\} = \frac{-1}{1-\sigma(z_i)}$$

$$\frac{\partial \sigma(z_i)}{\partial z_i} = \sigma(z_i)(1-\sigma(z_i)) = \sigma(z_i)\sigma(-z_i)$$

$$\nabla_\theta(z_i) = \nabla_\theta(\theta^T x_i) = x_i$$

$$\Rightarrow \nabla_\theta(\log(\sigma(\theta^T x_i))) = \frac{\partial \log(\sigma(\theta^T x_i))}{\partial \sigma(\theta^T x_i)} \cdot \frac{\partial \sigma(\theta^T x_i)}{\partial \theta^T x_i} \cdot \frac{\partial \theta^T x_i}{\partial \theta}$$

$$= \frac{1}{\sigma(\theta^T x_i)} \sigma(\theta^T x_i)\sigma(-\theta^T x_i) \cdot x_i$$

$$= \sigma(-\theta^T x_i) x_i$$

$$\nabla_\theta(\log(\sigma(-\theta^T x_i))) = \frac{\partial \log(\sigma(-\theta^T x_i))}{\partial \sigma(\theta^T x_i)} \cdot \frac{\partial \sigma(\theta^T x_i)}{\partial \theta^T x_i} \cdot \frac{\partial \theta^T x_i}{\partial \theta}$$

$$= \frac{-1}{1-\sigma(\theta^T x_i)} \cdot \sigma(\theta^T x_i)(1-\sigma(\theta^T x_i)) \cdot x_i$$

$$= -\sigma(\theta^T x_i) x_i$$

$$\Rightarrow \nabla_\theta E(\theta) = -\sum_{i=1}^{n} y_i \sigma(-\theta^T x_i) x_i - (1-y_i)\sigma(\theta^T x_i) x_i \quad \star$$

$$= -\sum_{i=1}^{n} y_i \underbrace{(\sigma(-\theta^T x_i) + \sigma(\theta^T x_i))}_{1} x_i - \sigma(\theta^T x_i) x_i$$

$$= \sum_{i=1}^{n} \left(\sigma(\theta^T x_i) - y_i \right) x_i$$

Algorithm 24.1: Logistic Regression: Stochastic Gradient Ascent

LOGISTICREGRESSION-SGA $(\mathbf{D}, \eta, \epsilon)$:

1 **foreach** $\mathbf{x}_i \in \mathbf{D}$ **do** $\tilde{\mathbf{x}}_i^T \leftarrow \begin{pmatrix} 1 & \mathbf{x}_i^T \end{pmatrix}$ // map to $\mathbb{R}^{d+1}$
2 $t \leftarrow 0$ // step/iteration counter
3 $\boldsymbol{\theta}^t \leftarrow (0, \ldots, 0)^T \in \mathbb{R}^{d+1}$ // initial weight vector
4 **repeat**
5 $\boldsymbol{\theta}^{old} \leftarrow \boldsymbol{\theta}^t$ // make a copy of $\boldsymbol{\theta}^t$
6 **foreach** $\tilde{\mathbf{x}}_i \in \tilde{\mathbf{D}}$ *in random order* **do**
7 $\nabla(\boldsymbol{\theta}, \tilde{\mathbf{x}}_i) \leftarrow \left(y_i - \sigma(\boldsymbol{\theta}^T \tilde{\mathbf{x}}_i)\right) \cdot \tilde{\mathbf{x}}_i$ // compute gradient at $\tilde{\mathbf{x}}_i$
8 $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \cdot \nabla(\boldsymbol{\theta}, \tilde{\mathbf{x}}_i)$ // update estimate for $\boldsymbol{\theta}$
9 $\boldsymbol{\theta}^{t+1} \leftarrow \boldsymbol{\theta}$ // update $\boldsymbol{\theta}^{t+1}$
10 $t \leftarrow t + 1$
11 **until** $\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^{t-1}\| \leq \epsilon$

Suppose $Y \in \{c_1, \dots, c_k\}$

We model Y as a k-dim Multivariate Bernoulli R.V. taking values

$c_1 \rightarrow e_1 = (1, 0, \dots, 0)$

$c_2 \rightarrow e_2 = (0, 1, \dots, 0)$

$\vdots$

$c_k \rightarrow e_k = (0, 0, \dots, 1)$

} one-hot encoding

Denote $\mathbb{P}(Y = e_i | X = x) = \pi_i(x)$ for $i = 1, \dots, k$

unknown probability of class c_i given datapoint x_i

$$\Rightarrow \sum_{i=1}^{k} \pi_i(x) = \sum_{i=1}^{k} \mathbb{P}(Y = e_i | X = x) = 1$$

And, $$\mathbb{P}(Y | X = x) = \prod_{j=1}^{k} (\pi_j(x))^{y_j}$$

In multiclass logistic regression, select c_k as a reference class and let log-odds of others w.r.t. c_k be linear in X

$$\log\left(\frac{\mathbb{P}(Y=e_i|X=x)}{\mathbb{P}(Y=e_k|X=x)}\right)=\log\left(\frac{\hat{\pi}_i(x)}{\hat{\pi}_k(x)}\right)=\theta_i^T x=\theta_{i0}+\theta_{i1}x_1+\dots+\theta_{id}x_d$$

$$\Rightarrow \hat{\pi}_i(x)=\exp(\theta_i^T x)\hat{\pi}_k(x), \quad i=1,\dots,k-1$$

$$\sum_{j=1}^{k-1}\hat{\pi}_j(x)+\hat{\pi}_k(x)=1$$

$$\sum_{j=1}^{k-1}\exp(\theta_j^T x)\hat{\pi}_k(x)+\hat{\pi}_k(x)=1$$

$$\hat{\pi}_k(x)\left(1+\sum_{j=1}^{k-1}\exp(\theta_j^T x)\right)=1$$

$$\hat{\pi}_k(x)=\frac{1}{1+\sum_{j=1}^{k-1}\exp(\theta_j^T x)}$$

$$\hat{\pi}_i(x)=\frac{\exp(\theta_i^T x)}{1+\sum_{j=1}^{k-1}\exp(\theta_j^T x)}$$

Setting $\theta_k=0$ so $\exp(\theta_k^T x)=1$, so

$$\hat{\pi}_i(x)=\frac{\exp(\theta_i^T x)}{\sum_{j=1}^{k}\exp(\theta_j^T x)}$$

Softmax function

prediction criteria $\hat{y}=\arg\max_{C_i}\{\hat{\pi}_i(x)\}=\arg\max_{C_i}\left\{\frac{\exp(\theta_i^T x)}{\sum_{j=1}^{k}\exp(\theta_j^T x)}\right\}$

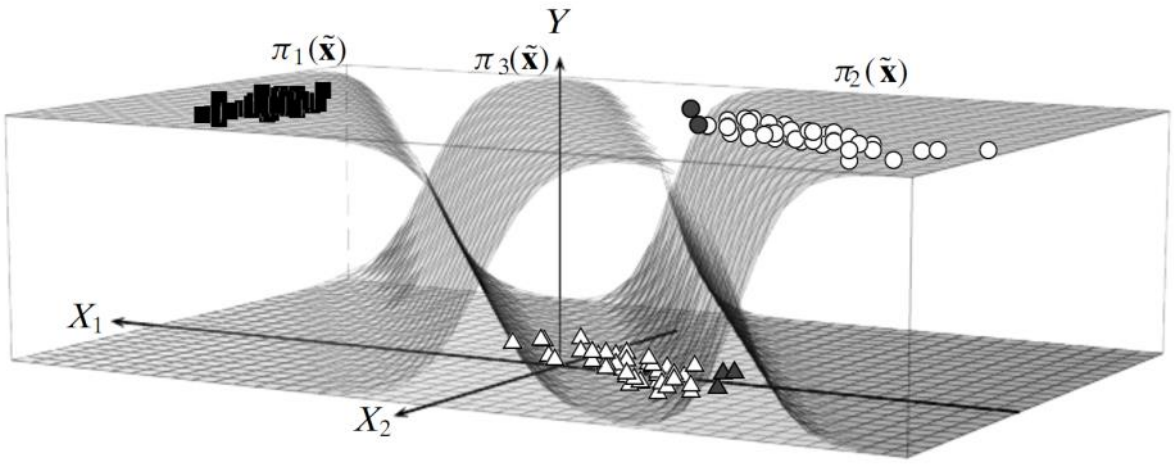


Figure 24.3. Multiclass logistic regression: Iris principal components data. Misclassified point are shown in dark gray color. All the points actually lie in the (X_1, X_2) plane, but c_1 and c_2 are shown displaced along Y with respect to the base class c_3 purely for illustration purposes.

Suppose $M: \mathbb{R}^d \to \{c_1, \ldots, c_k\}$ is a classifier

To build a classifier requires a training set.
We assess its performance on a testing set.

22.1 Classification Performance Measures

D - testing set of n points $x_i \in \mathbb{R}^d$ with labels y_i

$\hat{y}_i = M(x_i)$ - predicted label in $\{c_1, \ldots, c_k\}$

Error Rate: $\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{y_i \neq \hat{y}_i\}$

Accuracy: $\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{y_i = \hat{y}_i\}$

These metrics are global and do not consider how different classes contribute to the error.

Let $D_j = \{x_i \mid y_i = c_j\}$ points in class j

$n_j = |D_j|$

$D_j = \{x_i \mid \hat{y}_i = c_j\}$ points with predicted class j

$R_j = \{x_i \mid \hat{y}_i = c_j\}$ points with predicted class j

$M_j = |R_j|$

A <u>confusion matrix</u> is a $k \times k$ table N with

$N_{ij} = n_{ij}$ - number of points with predicted class c_i with true label c_j

$= |R_i \cap D_j| = \{x_a \in D \mid \hat{y}_a = c_i \text{ and } y_a = c_j\}$

predicted class \ true class	c_1	c_2	c_3
c_1	n_{11}	n_{12}	n_{13}
c_2	n_{21}	n_{22}	n_{23}
c_3	n_{31}	n_{32}	n_{33}

<u>Precision</u>: the class-specific precision of M for class c_i is

$$prec_i = \frac{n_{ii}}{m_i} = \frac{\text{true class } i \text{ predictions}}{\text{class } i \text{ predictions}} \quad (1 = \max)$$

i.e. the <u>fraction</u> of our class i <u>predictions</u> that are correct.

Global precision is

$$\text{precision} = \sum_{i=1}^{k} \left(\frac{m_i}{n}\right) \text{prec}_i = \sum_{i=1}^{k} \frac{n_{ii}}{n} \quad (= \text{accuracy})$$

Recall:

$$\text{recall}_i = \frac{n_{ii}}{n_i} \quad (1 = \max)$$

the fraction of class i points we correctly identify

There is a trade-off between precision and recall...
for example, if $M(x_i) = y_2$ for all x_i, then recall_2 is

$$\frac{n_2}{n_2} = 1$$

but precision is $\frac{n_2}{n}$, which is likely low.

The class-specific F-measure tries to balance them

$$F_i = \frac{2}{\frac{1}{\text{prec}_i} + \frac{1}{\text{recall}_i}} = \frac{2\,\text{prec}_i \cdot \text{recall}_i}{\text{prec}_i + \text{recall}_i} = \frac{2 \cdot \frac{n_{ii}}{m_i} \cdot \frac{n_{ii}}{n_i}}{\frac{n_{ii}}{m_i} + \frac{n_{ii}}{n_i}}$$

$$\frac{2n_{ii}^2}{\quad} \qquad \frac{2n_{ii}}{\quad}$$

$$= \frac{2n_{ii}^2}{n_{ii}n_i + n_{ii}m_i} = \frac{2n_{ii}}{n_i + m_i}$$

for F_i to be near 1, we need both to be near 1.

Global F-score $= \frac{1}{k}\sum_{i=1}^{k} F_i$ AKA F1-score

Binary Classification: suppose there are only 2 classes, we can call

Predicted class \ True Class	pos.	neg.
pos	n_{11} = True Positives	n_{12} = False Positives
neg	n_{21} = False Negatives	n_{22} = True Negatives

$$prec_p = \frac{TP}{TP+FP}$$ % of pos. predictions that are correct

$$recall_p = \frac{TP}{TP+FN}$$ % of pos. points that are correct

Receiver Operating Characteristic (ROC)

Receiver Operating Characteristic (ROC)

Popular strategy for assessing a binary classifier which outputs a score value for the positive class for each point in the test set, e.g. Bayes classifier outputs a posterior probability $\hat{P}(c_1|x_i)$.

Typically, $M(x_i) = \begin{cases} c_1, & \text{if score} > \varsigma \\ c_2, & \text{else} \end{cases}$ but ς is a somewhat arbitrary threshold

ROC plots false positive rate vs. true positive rate for each value of ς from ς^{min} to ς^{max}

$\frac{FP}{\text{neg. points}}$ $\min_i S(x_i)$ $\max_i S(x_i)$ $\frac{TP}{\text{pos. points}}$

S - score for pos. class

At ς^{min} ...

pred. \ truth	P	N
P	0	0
N	FN	TN

At ς^{max} ...

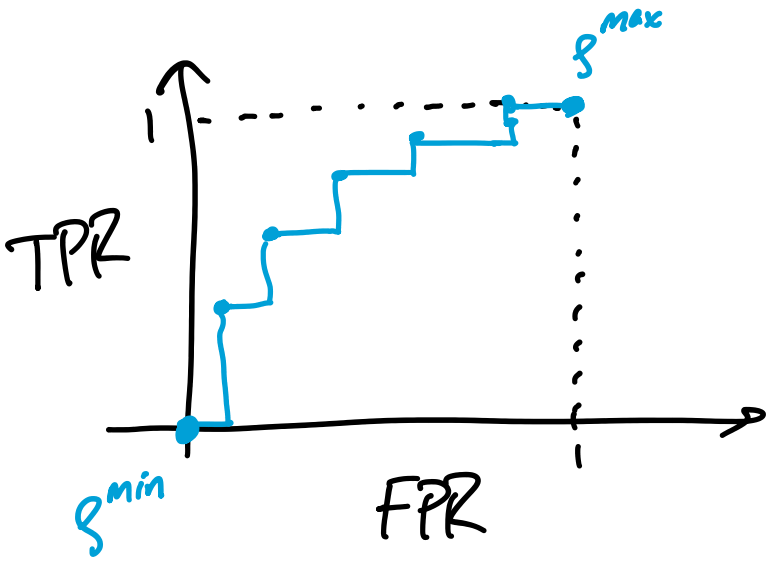
pred. \ truth	P	N
P	TP	FP
N	0	0

Ideal Classifier

	P	N
P	TP	0
N	0	TN

between...

$R_i(s) = \{x_i \in D \mid S(x_i) > s\}$ - points classified pos. for threshold s



Ideally, the curve should grow to 1 as quickly as possible

Area under ROC curve (AUC) is a one-number performance metric in $[0,1]$ - the prob. classifier will rank a random pos. test point higher than a random neg. point.

imp: ROC/AUC is not sensitive to class imbalance