Логистическая регрессия

А.Г. Трофимов
к.т.н., доцент, НИЯУ МИФИ

lab@neuroinfo.ru
http://datalearning.ru

Курс “Машинное обучение”

Сентябрь 2017

Problem Statement

Given:

$\mathscr{D} = \{(x^{(1)}, y^{(1)}), ..., (x^{(n)}, y^{(n)})\}$ — available data sample

$(x^{(i)}, y^{(i)}) \in \mathscr{X} \times \mathscr{Y}, \quad i = 1, ..., n$

$\mathscr{X} = \mathbb{R}^M$ — feature space, $\quad \mathscr{Y} = \{-1, 1\}$ — class labels

We consider that y is a response of unknown mapping
$F : \mathscr{X} \to \mathscr{Y}, \quad y^{(i)} = F(x^{(i)}), \quad i = 1, ..., n$

Build:

Learning algorithm $h \in \mathscr{H}$, $\mathscr{H} = \{h : h(x), h(x) \in \mathscr{Y}\}$, that estimates the unknown mapping F

Assumption:

Class of hypotheses $\mathscr{H} = \{h : h(x) = \operatorname{sign} \varphi(x, w)\}$,

where $\varphi(x, w) \in \mathbb{R}$ — classification score for object $x \in \mathscr{X}$

$w \in \mathbb{R}^L$ — vector of parameters

Linear Classifier

Feature vector $x = (x_0, x_1, ..., x_M)^T \in \mathbb{R}^{M+1}$, $x_0 \equiv 1$

Definition

Linear classifier is a classifier that makes the decision $h(x)$ based on the value of a linear combination of the features $x_0, ..., x_M$

Classification score for object $x \in \mathscr{X}$:

$$\varphi(x, w) = w_0 + w_1 x_1 + ... + w_M x_M = w^T x$$

where $w = (w_0, w_1, ..., w_M)^T$ — vector of parameters

Class of hypotheses:

$\mathscr{H} = \{h : h(x) = \text{sign}(w_0 + w_1 x_1 + ... + w_M x_M)\}$

Example:

Normal Bayes classifier with shared covariance matrix

Probabilistic Classifier

Definition

Probabilistic classifier is a classifier that is able to predict a probability distribution over a set of classes for the observation $x_0, ..., x_M$, rather than only outputting the most likely class that the observation should belong to

Examples of probabilistic classifiers:
Bayes classifier, logistic regression

Examples of non-probabilistic classifiers:
Support vector machines, LDA

Some non-probabilistic classifiers can be modified to be able to predict probabilities (e.g., support vector machines)

Logistic regression is a linear probabilistic classifier

Link Function

Can classification score $\varphi(x, w)$ be used as a measure of probability $P(Y = 1|x)$?

Intuitively:
The greater $\varphi(x, w) \Rightarrow$ the greater probability $P(Y = 1|x)$
The lower $\varphi(x, w) \Rightarrow$ the greater probability $P(Y = -1|x)$
$\varphi(x, w) = 0 \Rightarrow P(Y = -1|x) = P(Y = 1|x) = 0.5$

To use the classification score $\varphi(x, w) \in (-\infty; \infty)$ as a measure of probability $P(Y = 1|x)$ we need to map it monotonically into $[0; 1]$

Definition

Link function is a function $f : [0; 1] \to \mathbb{R}$ which defines relationship between probabilities $P(Y = 1|x)$ and classification scores $\varphi(x, w)$:

$$\varphi(x, w) = f(P(Y = 1|x, w))$$
$$P(Y = 1|x, w) = f^{-1}(\varphi(x, w))$$

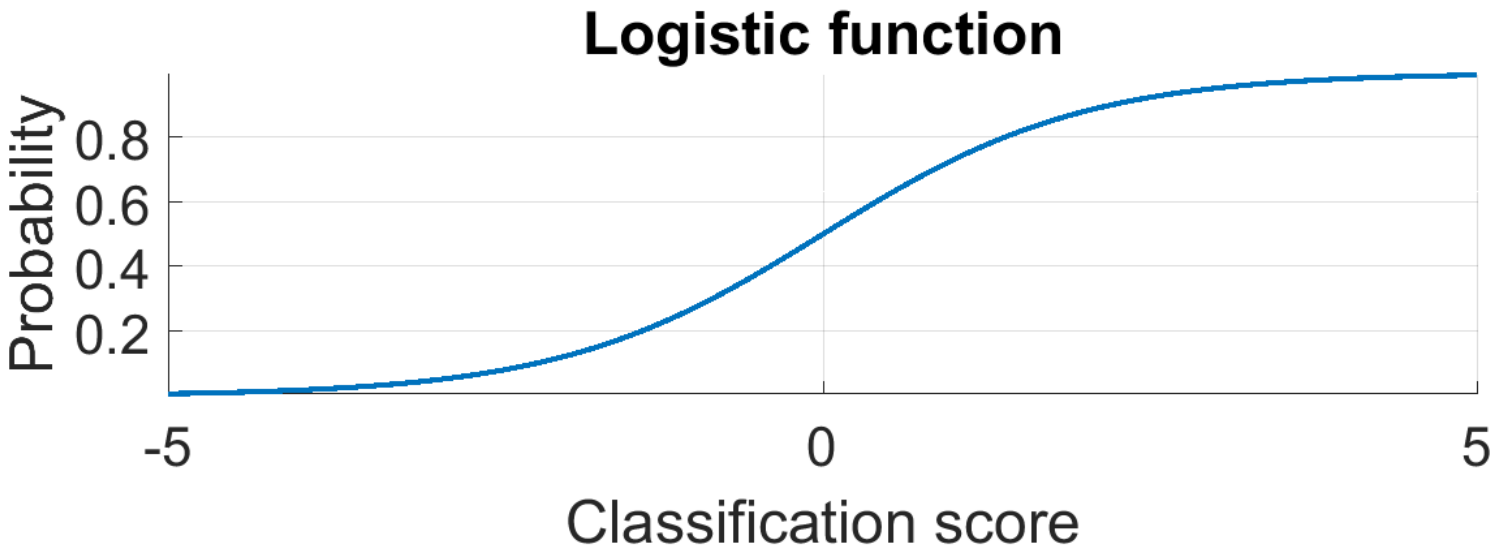
Logit Function

$$P(Y = 1|x, w) = p, \quad P(Y = -1|x, w) = 1 - p$$

Logit link function: $f(p) = logit(p) = \ln \frac{p}{1-p}$

Inverse logit function: $f^{-1}(\varphi) = sigmoid(\varphi) = \frac{1}{1 + \exp(-\varphi)}$

Inverse logit function is a logistic function



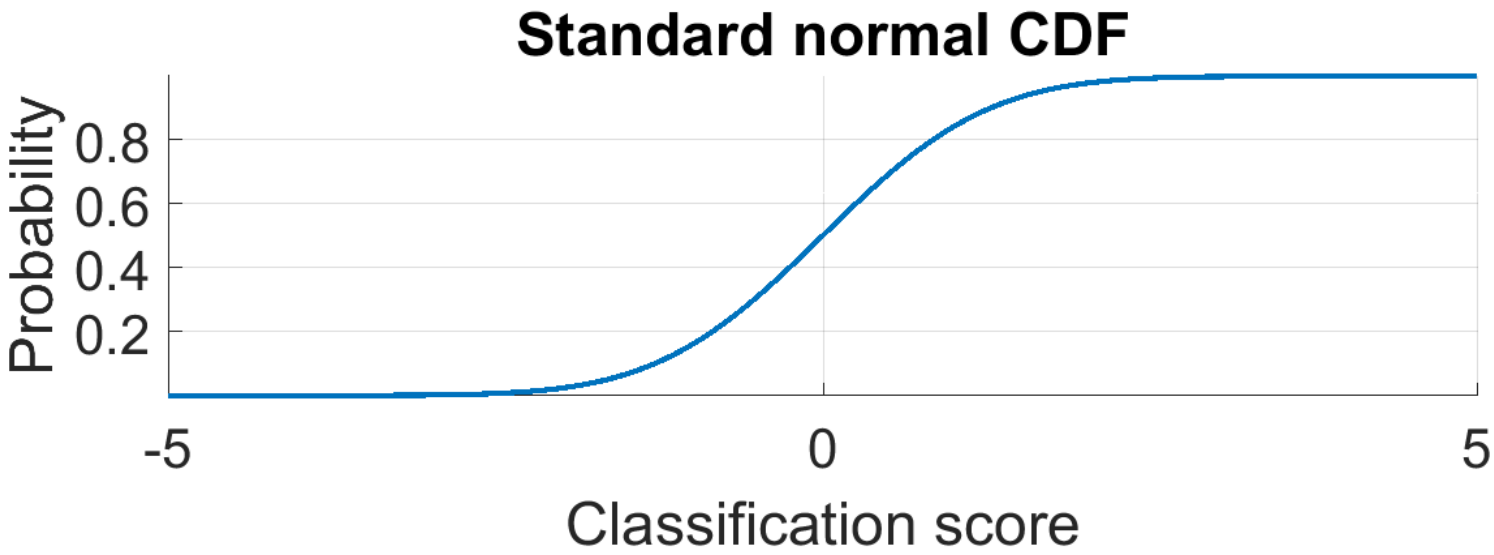
Probit Function

$$P(Y = 1|x, w) = p, \quad P(Y = -1|x, w) = 1 - p$$

Probit link function: $f(p) = probit(p) = \Phi^{-1}(p)$

Inverse probit function: $f^{-1}(\varphi) = \Phi(\varphi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\varphi} \exp\left(-\frac{u^2}{2}\right) du$

Inverse probit function is a cumulative distribution function Φ of the standard normal distribution



Logit vs Probit Link Functions

Logistic function has slightly flatter tails than the standard normal CDF, i.e probit curve approaches the axes more quickly than logit curve

Logit has better interpretation than probit:

$$\varphi(x,w) = logit(p) = \ln\frac{p}{1-p} = \ln\frac{P(Y=1|x,w)}{P(Y=-1|x,w)}$$

The ratio $\frac{p}{1-p} = \frac{P(Y=1|x,w)}{P(Y=-1|x,w)}$ is named odds ratio

For logit link function:

The classification score $\varphi(x,w)$ can be interpreted as log odds ratio for object $x \in \mathscr{X}$

Sigmoid Posteriors

Linear model for classification scores (by assumption):

$$\varphi(x, w) = w^T x$$

Logit link function means linear model for log odds ratio:

$$\varphi(x, w) = logit(p) = \ln \frac{p}{1-p} = w^T x$$

Given x, posterior probability p of positive class:

$$p = sigmoid(\varphi(x, w)) = \frac{1}{1 + e^{-\varphi(x,w)}} = \frac{1}{1 + e^{-w^T x}}$$

Inverse logit function is a sigmoid function: $sigmoid = logit^{-1}$

Logit link function means sigmoid model for posteriors of classes

Have you encountered the sigmoid posteriors before?

Odds Ratio for Bayes Classifier

Definitions:

$P(Y = k|X = x) = p_Y(k|x)$ — posterior probability class k
$P(Y = k) = p_Y(k)$ — prior probability of class k
$P(X = x|Y = k) = p_X(x|k)$ — likelihood of class k
$P(X = x) = p_X(x)$ — evidence of x
$p = p_Y(+1|x)$ — posterior probability of positive class
$1 - p = p_Y(-1|x)$ — posterior probability of negative class

Bayes' theorem:

$$p_Y(k|x) = \frac{p_X(x|k)p_Y(k)}{p_X(x)}$$

Log odds ratio:

$$\ln \frac{p}{1-p} = \ln \frac{p_Y(+1|x)}{p_Y(-1|x)} = \ln \frac{p_X(x|_{+1})p_Y(+1)}{p_X(x|_{-1})p_Y(-1)}$$

Odds Ratio for Normal Naive Bayes Classifier

For normal naive Bayes classifier with shared covariance matrix:

$$p_X(x|k) = \prod_{j=1}^{M} p_{X_j}(x_j|k) = \prod_{j=1}^{M} \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left(-\frac{(x_j - m_{j|k})^2}{2\sigma_j^2}\right)$$

Log odds ratio:

$$\ln\frac{p}{1-p} = \ln\frac{p_Y(+1|x)}{p_Y(-1|x)} = \ln\frac{p_X(x|_{+1})p_Y(+1)}{p_X(x|_{-1})p_Y(-1)}$$

Odds Ratio for Normal Naive Bayes Classifier

For normal naive Bayes classifier with shared covariance matrix:

$$p_X(x|k) = \prod_{j=1}^{M} p_{X_j}(x_j|k) = \prod_{j=1}^{M} \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left(-\frac{(x_j - m_{j|k})^2}{2\sigma_j^2}\right)$$

Log odds ratio:

$$\ln\frac{p}{1-p} = \ln\frac{p_Y(+1|x)}{p_Y(-1|x)} = \ln\frac{p_X(x|_{+1})p_Y(+1)}{p_X(x|_{-1})p_Y(-1)}$$

$$= \ln\frac{p_Y(+1)\prod_{j=1}^{M}\frac{1}{\sigma_j\sqrt{2\pi}}\exp\left(-\frac{(x_j-m_{j|+1})^2}{2\sigma_j^2}\right)}{p_Y(-1)\prod_{j=1}^{M}\frac{1}{\sigma_j\sqrt{2\pi}}\exp\left(-\frac{(x_j-m_{j|-1})^2}{2\sigma_j^2}\right)}$$

$$= \ln\frac{p_Y(+1)}{p_Y(-1)} + \sum_{j=1}^{M}\frac{(x_j - m_{j|-1})^2 - (x_j - m_{j|+1})^2}{2\sigma_j^2}$$

Odds Ratio for Normal Naive Bayes Classifier

$$\ln \frac{p}{1-p} = \ln \frac{p_Y(+1)}{p_Y(-1)} + \\ + \sum_{j=1}^{M} \frac{(x_j^2 - 2x_j m_{j|-1} + m_{j|-1}^2) - (x_j^2 - 2x_j m_{j|+1} + m_{j|+1}^2)}{2\sigma_j^2}$$

Odds Ratio for Normal Naive Bayes Classifier

$$\ln \frac{p}{1-p} = \ln \frac{p_Y(+1)}{p_Y(-1)} +$$

$$+ \sum_{j=1}^{M} \frac{(x_j^2 - 2x_j m_{j|-1} + m_{j|-1}^2) - (x_j^2 - 2x_j m_{j|+1} + m_{j|+1}^2)}{2\sigma_j^2}$$

$$= \ln \frac{p_Y(+1)}{p_Y(-1)} + \sum_{j=1}^{M} \frac{2x_j(m_{j|+1} - m_{j|-1}) + (m_{j|-1}^2 - m_{j|+1}^2)}{2\sigma_j^2}$$

$$= w_0 + \sum_{j=1}^{M} w_j x_j$$

where $w_0 = \ln \frac{p_Y(+1)}{p_Y(-1)} + \sum_{j=1}^{M} \frac{m_{j|-1}^2 - m_{j|+1}^2}{2\sigma_j^2}$

$$w_j = \frac{m_{j|+1} - m_{j|-1}}{\sigma_j^2}, \quad j = 1, ..., M$$

Logistic Regression and Bayesian Classification

Log odds ratio for normal naive Bayes classifier with shared covariance matrix is linear:

$$\ln \frac{p}{1-p} = w_0 + \sum_{j=1}^{M} w_j x_j = w^T x$$

It means that the posteriors are logistic:

$$p = sigmoid(w^T x) = \frac{1}{1 + e^{-w^T x}}$$

Under assumptions of normal naive Bayesian classification (the features $x_1, ..., x_M$ are independent and normally distributed) with shared covariance matrix the parameters $w_0, ..., w_M$ of logistic regression can be written in closed form

The value $w^T x$ can be considered as the classification score $\varphi(x)$ for object $x \in \mathscr{X}$

Discriminative Approach: Logistic Regression

Logistic model for posteriors:

$$\varphi(x, w) = logit(p(x, w)) = \ln \frac{p(x, w)}{1 - p(x, w)} = w^T x$$

$$p(x, w) = P(Y = 1|x, w) = sigmoid(\varphi(x, w)) = \frac{1}{1 + e^{-w^T x}}$$

How to estimate parameters $w_0, ..., w_M$ from the data, without any assumptions about underlying distributions?

Discriminative approach: we model directly classification scores $\varphi(x, w)$ without modelling the underlying joined distribution $f_{XY}(x, y)$

Logistic regression realizes discriminative approach: we model posteriors $p(x, w)$ explicitly related to classification scores $\varphi(x, w)$ via logit link function

Logistic Model of Classes

$\mathscr{D} = \{(x^{(1)}, y^{(1)}), ..., (x^{(n)}, y^{(n)})\}$ — available data sample

Let's re-label: $y^{(i)} := \dfrac{y^{(i)} + 1}{2}, \quad i = 1, ..., n, \quad$ so $y \in \{0, 1\}$

Assume that $y^{(i)}$ is drawn from Bernoulli distribution:

$Y_i \sim B(1, p(x^{(i)}, w))$, where $p(x^{(i)}, w) = P(Y_i = 1 | x^{(i)}, w)$

$P(Y_i = 0 | x^{(i)}, w) = 1 - p(x^{(i)}, w)$

$P(Y_i = k | x^{(i)}, w) = p(x^{(i)}, w)^k (1 - p(x^{(i)}, w))^{1-k}$

To estimate the vector of parameters w the maximum likelihood method (MLE) is used

The sample likelihood:

$$\mathscr{L}(y^{(1)}, ..., y^{(n)}, w) = \prod_{i=1}^{n} p(x^{(i)}, w)^{y^{(i)}} (1 - p(x^{(i)}, w))^{1-y^{(i)}} \to \max_w$$

Maximum Likelihood Estimation of Logistic Regression

The sample likelihood:

$$\mathscr{L}(y^{(1)}, ..., y^{(n)}, w) = \prod_{i=1}^{n} p(x^{(i)}, w)^{y^{(i)}} (1 - p(x^{(i)}, w))^{1-y^{(i)}} \to \max_{w}$$

Negative log-likelihood:

$$E(w) = -\sum_{i=1}^{n} \left(y^{(i)} \ln p(x^{(i)}, w) + (1 - y^{(i)}) \ln(1 - p(x^{(i)}, w)) \right) \to \min_{w}$$

Logistic model for $p(x, w)$:

$$p(x, w) = sigmoid(w^T x) = \frac{1}{1 + e^{-w^T x}}$$

MLE: Optimization Problem

Because of the non-linearity of the sigmoid function, we cannot find minimum directly and we use gradient descent:

$$w(t+1) = w(t) - \eta \frac{\partial E(t)}{\partial w}, \quad w(0) = w^0$$

where $\frac{\partial E(t)}{\partial w}$ is gradient, t is iteration, $\eta > 0$ is step size

Derivative of logistic function:

$$p = sigmoid(\varphi) = \frac{1}{1+e^{-\varphi}}, \qquad \frac{dp}{d\varphi} = p(1-p)$$

Derivatives of negative log-likelihood:

$$\frac{\partial E}{\partial w_j} = -\sum_{i=1}^{n} \left(y^{(i)} \frac{p(1-p)x_j^{(i)}}{p} - (1-y^{(i)}) \frac{p(1-p)x_j^{(i)}}{1-p} \right)$$

$$= -\sum_{i=1}^{n} \left(y^{(i)}(1-p) - (1-y^{(i)})p \right) x_j^{(i)} = \sum_{i=1}^{n} \left(p(x^{(i)}, w) - y^{(i)} \right) x_j^{(i)}$$

ERM Principle for Logistic Regression

Assume $\mathscr{Y} = \{-1, 1\}$

ERM principle:

$$R^*(w) = \sum_{i=1}^{n} L\left(m\left((x^{(i)}, y^{(i)}), w\right)\right) \to \min_w$$

Logistic loss function:

$$L(m) = \ln\left(1 + e^{-m}\right)$$

where

$$m\left((x^{(i)}, y^{(i)}), w\right) = y^{(i)}\varphi(x^{(i)}) = y^{(i)}w^T x^{(i)}$$

is a margin of object $x^{(i)}$, $i = 1, ..., n$

Empirical risk:

$$R^*(w) = \sum_{i=1}^{n} \ln\left(1 + e^{-y^{(i)}w^T x^{(i)}}\right)$$

ERM: Optimization Problem

Empirical risk: $R^*(w) = \sum_{i=1}^{n} \ln\left(1 + e^{-y^{(i)} w^T x^{(i)}}\right)$

Derivatives: $\dfrac{\partial R^*}{\partial w_j} = \sum_{i=1}^{n} \dfrac{1}{1 + e^{-y^{(i)} w^T x^{(i)}}} e^{-y^{(i)} w^T x^{(i)}} (-y^{(i)} x_j^{(i)})$

For $y^{(i)} = -1$:

$$\frac{\partial R^*}{\partial w_j} = \sum_{i=1}^{n} \frac{e^{w^T x^{(i)}}}{1 + e^{w^T x^{(i)}}} x_j^{(i)} = \sum_{i=1}^{n} \frac{1}{1 + e^{-w^T x^{(i)}}} x_j^{(i)} = \sum_{i=1}^{n} p(x^{(i)}, w) x_j^{(i)}$$

For $y^{(i)} = 1$:

$$\frac{\partial R^*}{\partial w_j} = \sum_{i=1}^{n} \frac{-e^{-w^T x^{(i)}}}{1 + e^{-w^T x^{(i)}}} x_j^{(i)} = \sum_{i=1}^{n} \left(\frac{1}{1 + e^{-w^T x^{(i)}}} - 1\right) x_j^{(i)}$$

$$= \sum_{i=1}^{n} (p(x^{(i)}, w) - 1) x_j^{(i)}$$

Logistic Regression Learning Problem

Derivatives of empirical risk ($\mathscr{Y} = \{-1, 1\}$):

$$\frac{\partial R^*}{\partial w_j} = \begin{cases} \sum\limits_{i=1}^{n} p(x^{(i)}, w) x_j^{(i)}, & y^{(i)} = -1 \\ \sum\limits_{i=1}^{n} (p(x^{(i)}, w) - 1) x_j^{(i)}, & y^{(i)} = 1 \end{cases}$$

Derivatives of negative log-likelihood ($\mathscr{Y} = \{0, 1\}$):

$$\frac{\partial E}{\partial w_j} = \sum_{i=1}^{n} \left(p(x^{(i)}, w) - y^{(i)} \right) x_j^{(i)} = \begin{cases} \sum\limits_{i=1}^{n} p(x^{(i)}, w) x_j^{(i)}, & y^{(i)} = 0 \\ \sum\limits_{i=1}^{n} \left(p(x^{(i)}, w) - 1 \right) x_j^{(i)}, & y^{(i)} = 1 \end{cases}$$

Logistic regression learning problem is empirical risk minimization with logistic loss function

Regularized Logistic Regression

If we impose some prior distribution $f_W(w)$ then we define a regularizer $r(w) = -\ln f_W(w)$ (doesn't depend on data) (see previous topics)

Regularized empirical risk:

$$R'(h) = R^*(h) + r(w)$$

The assumption about independent and normally distributed parameters with zero mean and variance σ^2 leads to L_2-regularizer:

$$r(w) = -\ln f_W(w) = \frac{1}{2\sigma^2}\sum_{j=1}^{M} w_j^2 + const$$

Derivatives of regularized empirical risk ($\mathscr{Y} = \{-1, 1\}$):

$$\frac{\partial R'}{\partial w_j} = \frac{\partial R^*}{\partial w_j} + \frac{1}{\sigma^2} w_j, \quad j = 1, ..., M$$

Problem Statement

Given:

$\mathscr{D} = \{(x^{(1)}, y^{(1)}), ..., (x^{(n)}, y^{(n)})\}$ — available data sample

$(x^{(i)}, y^{(i)}) \in \mathscr{X} \times \mathscr{Y}, \quad i = 1, ..., n$

$\mathscr{X} = \mathbb{R}^M$ — feature space, $\quad \mathscr{Y} = \{1, ..., K\}$ — class labels

Assumption:

Class of hypotheses $\mathscr{H} = \left\{ h : h(x) = \arg \max\limits_{k=\overline{1,K}} \varphi(x, w_k) \right\}$,

where $\varphi(x, w_k) \in \mathbb{R}$ — classification score for object x with respect to class k, $k = 1, ..., K$, $x = (x_0, x_1, ..., x_M)^T \in \mathbb{R}^{M+1}$, $x_0 \equiv 1$

$w_k \in \mathbb{R}^{M+1}$ — vector of parameters associated with k-th class, $k = 1, ..., K$

Classifications scores with respect to each class are linear:

$$\varphi(x, w_k) = w_{0k} + w_{1k}x_1 + ... + w_{Mk}x_M = w_k^T x$$

Relation to Posterior Probabilities

Intuitively:
The greater $\varphi(x, w_k) \Rightarrow$ the greater probability $P(Y = k|x)$

$$\varphi(x, w_{k^*}) = \max_{k=\overline{1,K}} \varphi(x, w_k) \Rightarrow P(Y = k^*) = \max_{k=\overline{1,K}} P(Y = k|x)$$

To use the classification scores $\varphi(x, w_1), ..., \varphi(x, w_K) \in (-\infty; \infty)$ as measures of probabilities $P(Y = 1|x), ..., P(Y = K|x)$ we need to map them monotonically into $[0; 1]$ such as $\sum_{k=1}^{K} P(Y = k|x) = 1$

In multiclass logistic regression to perform this mapping the softmax function is used:

$$(p_1, ..., p_K) = softmax(\varphi(x, w_1), ..., \varphi(x, w_K))$$

$$p_k = P(Y = k|x), \quad \varphi(x, w_k) = w_k^T x, \quad k = 1, ..., K$$

Softmax Function

Definition

Softmax function is a function $\mathbb{R}^K \to [0;1]^K$ defined as:

$$(p_1, ..., p_K) = softmax(\varphi_1, ..., \varphi_K) \Leftrightarrow p_k = \frac{e^{\varphi_k}}{\sum_{i=1}^{K} e^{\varphi_i}}, \; k = 1, ..., K$$

For $K = 2$ and $\varphi_1 = \frac{\varphi}{2}$, $\varphi_2 = -\frac{\varphi}{2}$:

$$softmax\left(\frac{\varphi}{2}, -\frac{\varphi}{2}\right) = \left(\frac{e^{\varphi/2}}{e^{\varphi/2} + e^{-\varphi/2}}, \frac{e^{-\varphi/2}}{e^{\varphi/2} + e^{-\varphi/2}}\right)$$

$$= \left(\frac{1}{1 + e^{-\varphi}}, \frac{e^{-\varphi}}{1 + e^{-\varphi}}\right) = (sigmoid(\varphi), 1 - sigmoid(\varphi)) = (p, 1 - p)$$

Softmax function is a generalization of the sigmoid (logistic) function for $K > 2$

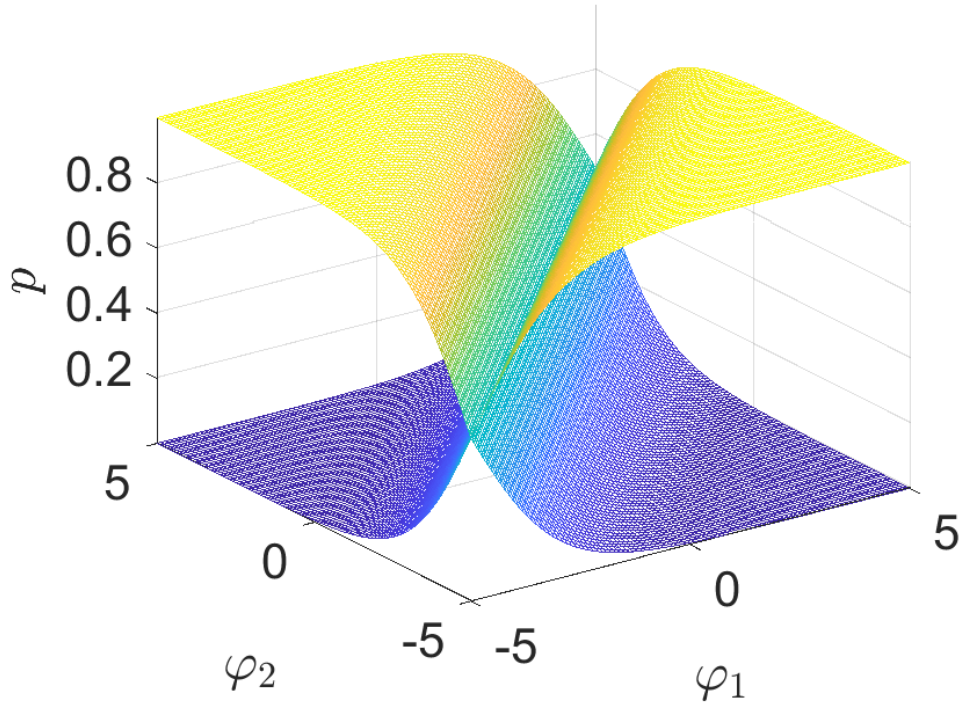
Softmax Function. Illustration

Hard max:

$$(p_1, p_2) = \arg\max(\varphi_1, \varphi_2)$$
$$\varphi_1 > \varphi_2 \Rightarrow (1, 0)$$
$$\varphi_1 < \varphi_2 \Rightarrow (0, 1)$$

Softmax:

$$(p_1, p_2) = softmax(\varphi_1, \varphi_2)$$
$$\varphi_1 > \varphi_2 \Rightarrow (p_1, p_2),\ 0 \geq p_2 > p_1 \geq 1$$
$$\varphi_1 < \varphi_2 \Rightarrow (p_1, p_2),\ 0 \geq p_1 > p_2 \geq 1$$



Softmax is a smooth approximation of indicator max function

Reference Class

$$(p_1, ..., p_K) = softmax(\varphi_1, ..., \varphi_K), \quad \varphi_k = w_k^T x,\ k = 1, ..., K$$

As soon as $\sum_{k=1}^{K} p_k = 1$, p_K is not independent: $p_K = 1 - \sum_{k=1}^{K-1} p_k$

It means that it is not necessary to learn vector of parameters for one of classes, e.g. w_K, it can be defined as reference

To prove it, add a constant vector C to all vectors $w_1, ..., w_K$:

$$\frac{e^{(w_k+C)^T x}}{\sum_{l=1}^{K} e^{(w_l+C)^T x}} = \frac{e^{w_k^T x} e^{C^T x}}{\sum_{l=1}^{K} e^{w_l^T x} e^{C^T x}} = \frac{e^{w_k^T x} e^{C^T x}}{e^{C^T x} \sum_{l=1}^{K} e^{w_l^T x}} = \frac{e^{w_k^T x}}{\sum_{l=1}^{K} e^{w_l^T x}}$$

The result of softmax function remains the same

Posterior Probabilities

Let $C = -w_K$: $p_k = \dfrac{e^{(w_k - w_K)^T x}}{\sum\limits_{l=1}^{K} e^{(w_l - w_K)^T x}}, \quad k = 1, ..., K$

Define:

$$w_1 := w_1 - w_K$$

$$...$$

$$w_{K-1} := w_{K-1} - w_K$$

$$w_K := 0$$

We need only $K - 1$ vectors of parameters to learn

Posterior probabilities:

$$p_1 = \frac{e^{w_1^T x}}{1 + \sum\limits_{l=1}^{K-1} e^{w_l^T x}}, ..., \; p_{K-1} = \frac{e^{w_{K-1}^T x}}{1 + \sum\limits_{l=1}^{K-1} e^{w_l^T x}}, \; p_K = \frac{1}{1 + \sum\limits_{l=1}^{K-1} e^{w_l^T x}}$$

Logistic Model of Classes

$\mathscr{D} = \{(x^{(1)}, y^{(1)}), ..., (x^{(n)}, y^{(n)})\}$ — available data sample

Assume that $y^{(i)}$ is drawn from multinomial distribution:

$$Y_i \sim Mult(1, p_1(x^{(i)}, w), ..., p_K(x^{(i)}, w))$$

where $p_k(x^{(i)}, w) = P(Y_i = k | x^{(i)}, w)$

Let's re-label: $y^{(i)} := \left(y_1^{(i)}, ..., y_K^{(i)}\right), \quad y_k^{(i)} = \begin{cases} 1, & y^{(i)} = k \\ 0, & otherwise \end{cases}$

So $y^{(i)}$ is a binary vector that contains one 1 at k-th position, other elements are 0, $i = 1, ..., n$

Probabilities:

$$P(Y_i = k | x^{(i)}, w) = p_k(x^{(i)}, w) = \prod_{l=1}^{K} \left(p_l(x^{(i)}, w)\right)^{y_l^{(i)}}$$

Maximum Likelihood Estimation of Logistic Regression

To estimate the vector of parameters w the maximum likelihood method (MLE) is used

The sample likelihood:

$$\mathscr{L}(y^{(1)}, ..., y^{(n)}, w) = \prod_{i=1}^{n} \prod_{l=1}^{K} \Big(p_l(x^{(i)}, w)\Big)^{y_l^{(i)}} \to \max_w$$

Negative log-likelihood:

$$E(w) = -\sum_{i=1}^{n} \sum_{l=1}^{K} y_l^{(i)} \ln p_l(x^{(i)}, w) \to \min_w$$

Logistic model for $p(x, w)$:

$$(p_1(x, w), ..., p_K(x, w)) = softmax(w_1^T x, ..., w_{K-1}^T x, 0)$$

where $w = (w_1^T, ..., w_{K-1}^T)$ is a $(K-1) * M$ matrix of parameters

MLE: Optimization Problem

Because of the non-linearity of the sigmoid function, we cannot find minimum directly and we use gradient descent:

$$w_k(t+1) = w_k(t) - \eta \frac{\partial E(t)}{\partial w_k}, \quad w_k(0) = w_k^0, \quad k = 1, ..., K-1$$

where $\frac{\partial E(t)}{\partial w_k}$ is gradient, t is iteration, $\eta > 0$ is step size

Derivatives of softmax function:

$$(p_1, ..., p_K) = softmax(\varphi_1, ..., \varphi_K), \qquad \frac{dp_l}{d\varphi_k} = p_l(\delta_{kl} - p_k)$$

where $\delta_{kl} = \begin{cases} 1, & k = l \\ 0, & otherwise \end{cases}$ is the Kronecker delta

MLE: Optimization Problem

Derivatives of negative log-likelihood:

$$\frac{\partial E}{\partial w_{kj}} = -\sum_{i=1}^{n}\sum_{l=1}^{K} y_l^{(i)} \ln p_l(x^{(i)}, w)$$

$$= -\sum_{i=1}^{n}\sum_{l=1}^{K} y_l^{(i)} \frac{p_l(x^{(i)}, w)(\delta_{kl} - p_k(x^{(i)}, w))}{p_l(x^{(i)}, w)} x_j^{(i)}$$

$$= -\sum_{i=1}^{n}\sum_{l=1}^{K} y_l^{(i)} (\delta_{kl} - p_k(x^{(i)}, w)) x_j^{(i)}$$

$$= -\sum_{i=1}^{n} \left(\sum_{l=1}^{K} y_l^{(i)} \delta_{kl} - p_k(x^{(i)}, w) \sum_{l=1}^{K} y_l^{(i)} \right) x_j^{(i)}$$

$$= -\sum_{i=1}^{n} \left(y_k^{(i)} - p_k(x^{(i)}, w) \right) x_j^{(i)}, \quad k = 1, ..., K-1, \; j = 1, ..., M$$

Logistic Regression vs Bayesian Classification

Logistic Regression:

- Assumes the model for posterior probabilities $p_Y(y|x)$ of classes and trains its parameters
- Can still be used when the class-conditional densities are non-normal or when they are not unimodal as long as classes are linearly separable

Parametric Bayesian classification:

- Assumes the model for conditional distributions of features $p_X(x|y)$ and class priors $p_Y(y)$, the posterior probabilities are derived using Bayes' rule
- The assumptions about underlying distributions can be wrong that leads to classification errors

When data are normally distributed, the logistic discriminant has a comparable error rate to the parametric, normal-based linear discriminant