CS 340 Lec. 16: Logistic Regression

AD

March 2011

Introduction

- Assume you are given some training data $\left\{ \mathbf{x}^i, y^i \right\}_{i=1}^{N}$ where $\mathbf{x}^i \in \mathbb{R}^d$ and y^i can take C different values.
- Given an input test data $\mathbf{x}$, you want to predict/estimate the output y associated to $\mathbf{x}$.
- Previously we have followed a probabilistic approach

$$p\left(y = k \middle| \mathbf{x} \right) = \frac{p\left(\mathbf{x} \middle| y = k \right) p\left(y = k \right)}{\sum_{j=0}^{C-1} p\left(\mathbf{x} \middle| y = j \right) p\left(y = j \right)}.$$

- This requires modelling and learning the parameters of the class conditional density of features $p\left(\mathbf{x} \middle| y = k \right)$.

Logistic Regression

- Discriminative model: we model and learn directly $p(y=k|\mathbf{x})$ and bypassing the introduction of $p(\mathbf{x}|y=k)$.
- Consider the following model for $C=2$ (binary classification)

$$\begin{aligned} p(y=1|\mathbf{x},\mathbf{w}) &= 1-p(y=0|\mathbf{x},\mathbf{w}) \\ &= g\left(\mathbf{w}^{\mathsf{T}}\mathbf{x}\right) \end{aligned}$$

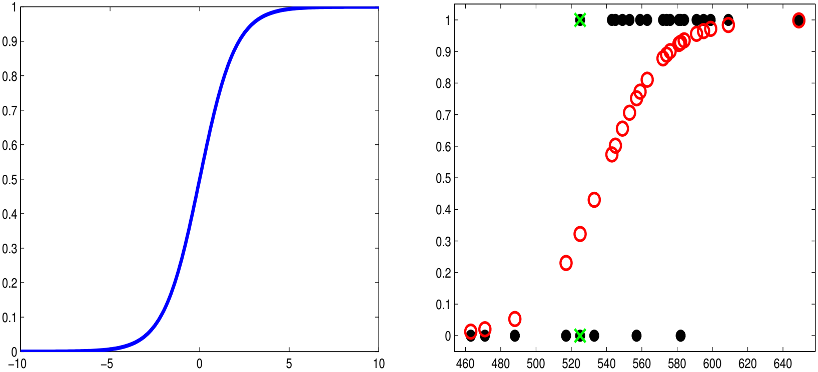
  where $\mathbf{w}=(w_0 \ \cdots \ w_d)^{\mathsf{T}}$, $\mathbf{x}=(x_0 \ \cdots \ x_d)^{\mathsf{T}}$ so

$$z=\mathbf{w}^{\mathsf{T}}\mathbf{x}=w_0+\sum_{j=1}^{d} w_j x_j$$

  and g is a "squashing" function: $g:\mathbb{R}\rightarrow[0,1]$.
- Logistic regression corresponds to

$$g(z)=\frac{1}{1+\exp(-z)}=\frac{\exp(z)}{1+\exp(z)}.$$

Logistic Function



(Left) logistic or sigmoid function (Right) logistic regression for x=SAT score and y=pass/fail class (solid black dots are the data), open red circles are predicted probabilities.

Logistic Regression

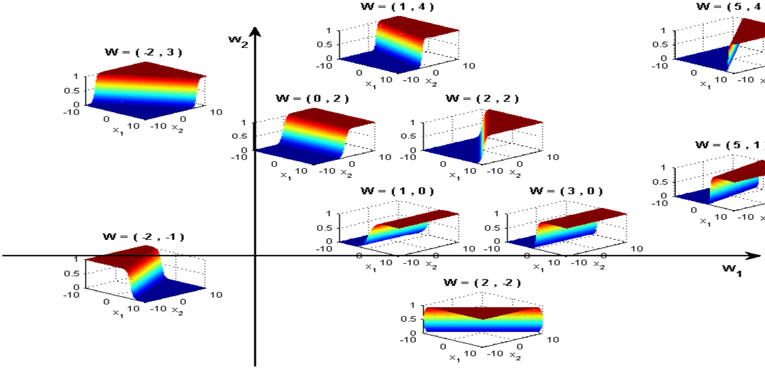
- The log odds ratio satisfies

$$LOR(\mathbf{x}) = \log \frac{p(y = 1|\,\mathbf{x}, \mathbf{w})}{p(y = 0|\,\mathbf{x}, \mathbf{w})} = \mathbf{w}^{\mathsf{T}}\mathbf{x}$$

  so the logistic parameters are easily interpretable.

- If $w_j > 0$, then increasing x_j makes $y = 1$ more likely while decreasing x_j makes $y = 0$ more likely (and opposite if $w_j = 0$). $w_j = 0$ means x_j has no impact on the outcome.
- Logistic regression partitions the input space into two regions whose decision boundary is $\{\mathbf{x} : LOR(\mathbf{x}) = 0\} = \{\mathbf{x} : \mathbf{w}^{\mathsf{T}}\mathbf{x} = 0\}$
- Simple model of a neuron: it forms a weighted sum of its inputs and the "fires" an output pulse if this sum exceeds a threshold. Logistic regression mimics this as you can sort of think of it as a process which "fires" if $p(y = 1|\,\mathbf{x}, \mathbf{w}) > p(y = 0|\,\mathbf{x}, \mathbf{w})$ equivalently if $LOR(\mathbf{x}) > 0$.

Logistic Function in Two Dimensions



Plots of $p\left(y=1|\, w_1x_1+w_2x_2\right)$. Here $\mathbf{w}=\left(w_1,w_2\right)$ define the normal to the decision boundary. Points to the right have $\mathbf{w}^{\mathsf{T}}\mathbf{x}>0$ and to the left have $\mathbf{w}^{\mathsf{T}}\mathbf{x}<0$.

MLE Parameter Learning for Logistic Regression

- To learn the parameters $\mathbf{w}$, we can maximize w.r.t $\mathbf{w}$ the (conditional) log-likelihood function

$$
\begin{aligned}
l(\mathbf{w}) &= \log\ p\left(\left\{y^i\right\}_{i=1}^N \middle| \left\{\mathbf{x}^i\right\}_{i=1}^N, \mathbf{w}\right) = \log \prod_{i=1}^N p\left(y^i \middle| \mathbf{x}^i, \mathbf{w}\right) \\
&= \sum_{i=1}^N \log p\left(y^i \middle| \mathbf{x}^i, \mathbf{w}\right)
\end{aligned}
$$

- We have

$$
\begin{aligned}
l(\mathbf{w}) &= \sum_{i=1}^N y^i \log p\left(y^i = 1 \middle| \mathbf{x}^i, \mathbf{w}\right) + \left(1 - y^i\right) \log p\left(y^i = 0 \middle| \mathbf{x}^i, \mathbf{w}\right) \\
&= -\sum_{i=1}^N \left(1 - y^i\right) \mathbf{w}^\mathsf{T}\mathbf{x}^i - \sum_{i=1}^N \log\left(1 + \exp\left(-\mathbf{w}^\mathsf{T}\mathbf{x}^i\right)\right)
\end{aligned}
$$

- **Good news**: $l(\mathbf{w})$ is concave so there is no local maxima.
- **Bad news:** there is no-closed form solution for $\widehat{\mathbf{w}}_{\mathsf{MLE}}$.

Gradient Ascent

- Gradient ascent is one of the most basic method to maximize a function.
- It is an iterative procedure such that at iteration t :

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \eta \;\; \nabla_{\mathbf{w}} l\left(\mathbf{w}\right)\big|_{\mathbf{w}^{(t-1)}}$$

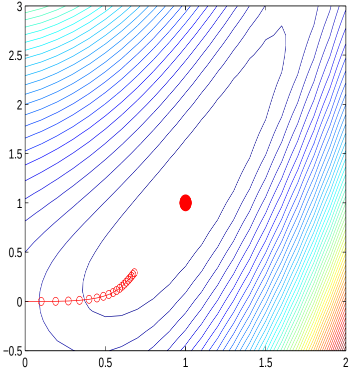
  where the gradient is

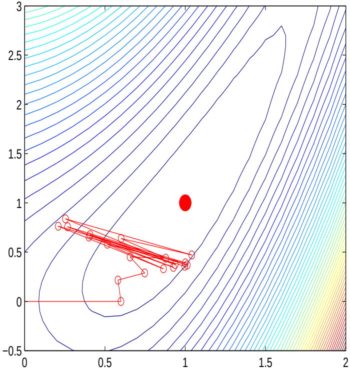
$$\nabla_{\mathbf{w}} l\left(\mathbf{w}\right) = \begin{bmatrix} \frac{\partial l(\mathbf{w})}{\partial w_0} & \cdots & \frac{\partial l(\mathbf{w})}{\partial w_d} \end{bmatrix}^{\mathsf{T}}$$

  and $\eta > 0$ is the learning rate.
- To minimize a function $f\left(\mathbf{w}\right)$, simply use the gradient descent

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} - \eta \;\; \nabla_{\mathbf{w}} f\left(\mathbf{w}\right)\big|_{\mathbf{w}^{(t-1)}}$$

Gradient Descent Example





Gradient descent on a simple function, starting from (0,0) for 20 steps using $\eta = 0.1$ (left) and $\eta = 0.6$ (right)

Gradient Ascent for Logistic Regression

- We have

$$\frac{\partial l(\mathbf{w})}{\partial w_k} = -\sum_{i=1}^{N}\left(1-y^i\right)x_k^i + \sum_{i=1}^{N} x_k^i \frac{\exp\left(-\mathbf{w}^{\mathsf{T}}\mathbf{x}^i\right)}{1+\exp\left(-\mathbf{w}^{\mathsf{T}}\mathbf{x}^i\right)}$$

- Hence we have

$$\begin{aligned} \frac{\partial l(\mathbf{w})}{\partial w_k} &= \sum_{i=1}^{N} x_k^i \left\{ p\left(y^i = 0 \middle| \mathbf{x}^i, \mathbf{w}\right) - \left(1-y^i\right)\right\} \\ &= \sum_{i=1}^{N} x_k^i \left\{ y^i - p\left(y^i = 1 \middle| \mathbf{x}^i, \mathbf{w}\right)\right\} \end{aligned}$$

- So in vector-form, we will do

$$\begin{aligned} \mathbf{w}^{(t)} &= \mathbf{w}^{(t-1)} + \eta \left. \nabla_{\mathbf{w}} l(\mathbf{w})\right|_{\mathbf{w}^{(t-1)}} \\ &= \mathbf{w}^{(t-1)} + \eta \sum_{i=1}^{N} \left\{ y^i - p\left(y^i = 1 \middle| \mathbf{x}^i, \mathbf{w}^{(t-1)}\right)\right\} \mathbf{x}^i \end{aligned}$$

Regularized Logistic Regression

- Similarly to regression, we can regularize the solution by assigning a Gaussian prior to **w**

$$p(\mathbf{w}) = \prod_{j=0}^{d} p(w_j) = \prod_{j=0}^{d} \mathcal{N}(w_j; 0, \lambda)$$

- This pushes the parameters **w** towards zero and can prevent overfitting. In this case, we have

$$\begin{aligned} \widehat{\mathbf{w}}_{MAP} &= \arg\max \; p\left(\mathbf{w} \middle| \left\{\mathbf{x}^i, y^i\right\}_{i=1}^{N}\right) \\ &= \arg\max \; l(\mathbf{w}) - \frac{\mathbf{w}^{\mathsf{T}}\mathbf{w}}{2\lambda}. \end{aligned}$$

- $\widehat{\mathbf{w}}_{MAP}$ can be computed iteratively using

$$\begin{aligned} &\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \eta \;\; \nabla_{\mathbf{w}} \left(l(\mathbf{w}) - \tfrac{\mathbf{w}^{\mathsf{T}}\mathbf{w}}{2\lambda} \right)\Big|_{\mathbf{w}^{(t-1)}} \\ &= \mathbf{w}^{(t-1)} + \eta \;\; \left\{ -\lambda^{-1}\mathbf{w} + \textstyle\sum_{i=1}^{N} \left\{ y^i - p\left(y^i = 1 \middle| \mathbf{x}^i, \mathbf{w}^{(t-1)} \right) \right\} \;\; \mathbf{x}^i \right\} \end{aligned}$$

Using Basis Functions for Logistic Regression

- Similarly to regression, we can use basis functions; i.e.

$$p\left(y=1|\, \mathbf{x},\mathbf{w}\right)=g\left(\mathbf{w}^{\mathsf{T}}\Phi\left(\mathbf{x}\right)\right)$$

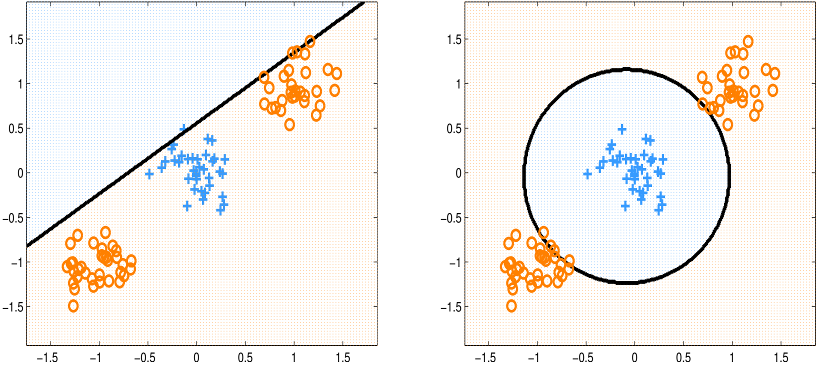
  where $\mathbf{w}=\left(w_1 \;\cdots\; w_m\right)^{\mathsf{T}}$, $\Phi\left(\mathbf{x}\right)=\left(\Phi_1\left(\mathbf{x}\right) \;\cdots\; \Phi_m\left(\mathbf{x}\right)\right)^{\mathsf{T}}$.
- For example, if $\mathbf{x}\in\mathbb{R}$ then we can pick

$$\Phi\left(\mathbf{x}\right)=\left(1,x,\ldots,x^m\right)$$

- For $\mathbf{x}\in\mathbb{R}^d$, we can pick some radial basis functions

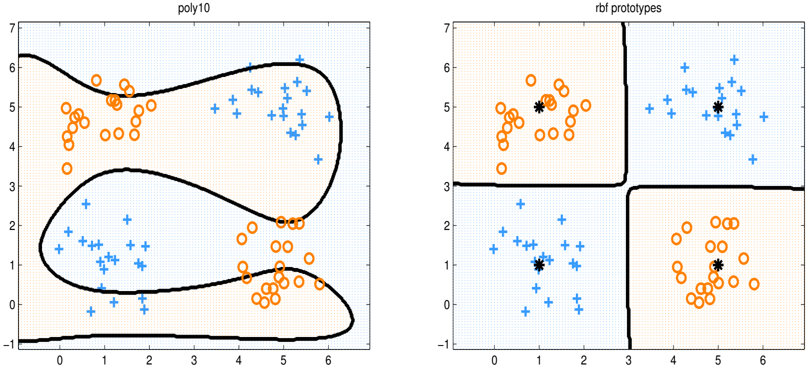
$$\Phi_j\left(\mathbf{x}\right)=\exp\left(-\frac{\left(\mathbf{x}-\boldsymbol{\mu}_j\right)^{\mathsf{T}}\left(\mathbf{x}-\boldsymbol{\mu}_j\right)}{2\sigma^2}\right).$$

Example



(left) Logistic regression in the original feature space $\mathbf{x} = (x_1, x_2)$. (right) Logistic regression obtained after performing a 2nd degree poly expansion $\Phi(\mathbf{x}) = \left(1, x_1, x_2, x_1^2, x_2^2\right)$.

Example



(left) Logistic regression for $\Phi(\mathbf{x}) = \left(1, x_1, x_2, \ldots, x_1^{10}, x_2^{10}\right)$. (right) Logistic regression using 4 radial basis functions with centers $\boldsymbol{\mu}_j$ specified by black crosses.

Multinomial Logistic Regression

- Consider now the case where $C > 2$. We could consider the following generalization

$$p\left(y = c \middle| \mathbf{x}, \{\mathbf{w}_c\}_{c=1}^{C}\right) = \frac{\exp\left(\mathbf{w}_c^{\mathsf{T}}\mathbf{x}\right)}{\sum_{k=1}^{C}\exp\left(\mathbf{w}_k^{\mathsf{T}}\mathbf{x}\right)} \text{ for } c = 1, ..., C$$

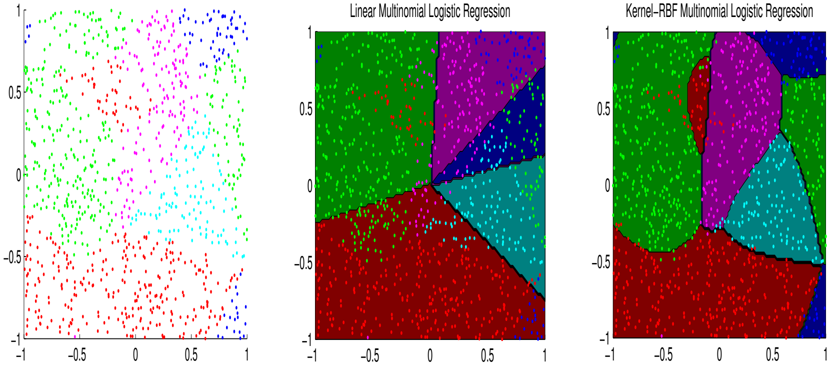
  but this is not identifiable as $p\left(y = c \middle| \mathbf{x}, \{\mathbf{w}_c + \mathbf{w}'\}_{c=1}^{C}\right) = p\left(y = c \middle| \mathbf{x}, \{\mathbf{w}_c\}_{c=1}^{C}\right)$.

- Hence we set $\mathbf{w}_C = \begin{pmatrix} 0 & \cdots & 0 \end{pmatrix}^{\mathsf{T}}$ to obtain

$$\begin{aligned} p\left(y = c \middle| \mathbf{x}, \{\mathbf{w}_c\}_{c=1}^{C-1}\right) &= \frac{\exp\left(\mathbf{w}_c^{\mathsf{T}}\mathbf{x}\right)}{1 + \sum_{k=1}^{C-1}\exp\left(\mathbf{w}_k^{\mathsf{T}}\mathbf{x}\right)} \text{ for } c = 1, ..., C-1 \\ p\left(y = C \middle| \mathbf{x}, \{\mathbf{w}_c\}_{c=1}^{C-1}\right) &= \frac{1}{1 + \sum_{k=1}^{C-1}\exp\left(\mathbf{w}_k^{\mathsf{T}}\mathbf{x}\right)}. \end{aligned}$$

- The (conditional) log-likelihood is concave w.r.t $\{\mathbf{w}_c\}_{c=1}^{C-1}$ so MLE estimates can be computed using gradient.

Example



(left) Some 5 class data in 2d (center) Multinomial logistic regression in the original feature space $\mathbf{x} = (x_1, x_2)$ (right) RBF basis functions with bandwidth 1 using $m = 1 + N$. We use all the data points as centers.