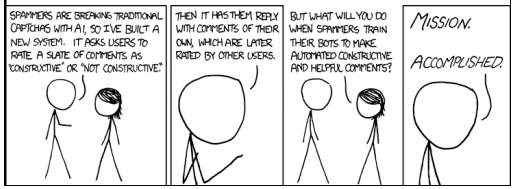
Linear models - continued
Logistic regression

Chapter 3.3



Predicting probabilities

Objective: learn to predict a probability P(y | x) for a binary classification problem using a linear classifier

The target function: $f(\mathbf{x}) = \mathbb{P}[y = +1 \mid \mathbf{x}].$

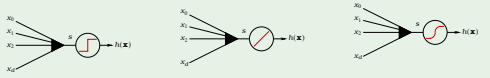
$$P(y \mid \mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{for } y = +1; \\ 1 - f(\mathbf{x}) & \text{for } y = -1. \end{cases}$$

For positive examples P(y = +1 | x) = 1 whereas P(y = +1 | x) = 0 for negative examples.

Another linear model

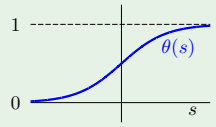
$$s = \mathbf{w}^{\mathsf{T}}\mathbf{x}$$

linear classification	linear regression	logistic regression
$h(\mathbf{x}) = \text{sign}(s)$	$h(\mathbf{x}) = s$	$h(\mathbf{x}) = \theta(s)$



The logistic function (aka squashing function):

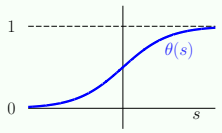
$$\theta(s) = \frac{e^s}{1 + e^s}$$



Properties of the logistic function

$$\theta(s) = \frac{e^s}{1+e^s} = \frac{1}{1+e^{-s}}.$$

$$\theta(-s) = \frac{e^{-s}}{1+e^{-s}} = \frac{1}{1+e^s} = 1 - \theta(s).$$



Predicting probabilities

Fitting the data means finding a good hypothesis h

h is good if:

$$\begin{cases} h(\mathbf{x}_n) \approx 1 & \text{whenever } y_n = +1; \\ h(\mathbf{x}_n) \approx 0 & \text{whenever } y_n = -1. \end{cases}$$

Suppose that $h(\mathbf{x}) = \theta(\mathbf{w}^{\mathsf{T}}\mathbf{x})$ closely captures $\mathbb{P}[+1|\mathbf{x}]$:

$$P(y \mid \mathbf{x}) = \begin{cases} \theta(\mathbf{w}^{\mathsf{T}}\mathbf{x}) & \text{for } y = +1; \\ 1 - \theta(\mathbf{w}^{\mathsf{T}}\mathbf{x}) & \text{for } y = -1. \end{cases}$$

Predicting probabilities

Fitting the data means finding a good hypothesis h

h is good if:

$$\begin{cases} h(\mathbf{x}_n) \approx 1 & \text{whenever } y_n = +1; \\ h(\mathbf{x}_n) \approx 0 & \text{whenever } y_n = -1. \end{cases}$$

Suppose that $h(\mathbf{x}) = \theta(\mathbf{w}^{\mathsf{T}}\mathbf{x})$ closely captures $\mathbb{P}[+1|\mathbf{x}]$:

$$P(y \mid \mathbf{x}) = \begin{cases} \theta(\mathbf{w}^{\mathsf{T}}\mathbf{x}) & \text{for } y = +1; \\ \theta(-\mathbf{w}^{\mathsf{T}}\mathbf{x}) & \text{for } y = -1. \end{cases}$$

More compactly: $P(y \mid \mathbf{x}) = \theta(y \cdot \mathbf{w}^{\mathsf{T}}\mathbf{x})$

Is logistic regression really linear?

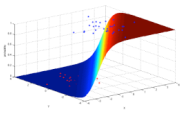
$$P(y=+1|\mathbf{x}) = \frac{\exp(\mathbf{w}^\intercal\mathbf{x})}{\exp(\mathbf{w}^\intercal\mathbf{x})+1}$$

$$P(y=-1|\mathbf{x}) = 1 - P(y=+1|\mathbf{x}) = \frac{1}{\exp(\mathbf{w}^\intercal\mathbf{x})+1}$$

To figure out how the decision boundary looks like consider:

$$\ln\frac{P(y=+1|\mathbf{x})}{P(y=-1|\mathbf{x})} = \mathbf{w}^\intercal\mathbf{x}$$

i.e. linear!



Maximum likelihood

We will find $\mathbf{w}$ using the principle of maximum likelihood.

Likelihood:
The probability of getting the $y_1,\ldots,y_N$ in $\mathcal{D}$ from the corresponding $\mathbf{x}_1,\ldots,\mathbf{x}_N$:

$$P(y_1,\ldots,y_N \mid \mathbf{x}_1,\ldots,\mathbf{x}_N) = \prod_{n=1}^{N} P(y_n \mid \mathbf{x}_n).$$

Valid since $(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_N,y_N)$ are independently generated

Maximizing the likelihood

$$\begin{aligned}
& \max && \textstyle\prod_{n=1}^{N} P(y_n \mid \mathbf{x}_n) \\
\Leftrightarrow\ & \max && \ln\left(\textstyle\prod_{n=1}^{N} P(y_n \mid \mathbf{x}_n)\right) \\
\equiv\ & \max && \textstyle\sum_{n=1}^{N} \ln P(y_n \mid \mathbf{x}_n) \\
\Leftrightarrow\ & \min && -\frac{1}{N}\textstyle\sum_{n=1}^{N} \ln P(y_n \mid \mathbf{x}_n) \\
\equiv\ & \min && \frac{1}{N}\textstyle\sum_{n=1}^{N} \ln\frac{1}{P(y_n|\mathbf{x}_n)} \\
\equiv\ & \min && \frac{1}{N}\textstyle\sum_{n=1}^{N} \ln\frac{1}{\theta(y_n\cdot\mathbf{w}^\intercal\mathbf{x}_n)} \\
\equiv\ & \min && \frac{1}{N}\textstyle\sum_{n=1}^{N} \ln(1+e^{-y_n\cdot\mathbf{w}^\intercal\mathbf{x}_n})
\end{aligned}$$

Maximizing the likelihood

Summary: maximizing the likelihood is equivalent to

$$\text{minimize } -\frac{1}{N}\ln\left(\prod_{n=1}^{N}\theta(y_n\,\mathbf{w}^\intercal\,\mathbf{x}_n)\right)$$

$$= \frac{1}{N}\sum_{n=1}^{N}\ln\left(\frac{1}{\theta(y_n\,\mathbf{w}^\intercal\,\mathbf{x}_n)}\right) \qquad \left[\theta(s)=\frac{1}{1+e^{-s}}\right]$$

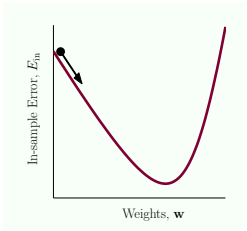
$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\underbrace{\ln\left(1+e^{-y_n\mathbf{w}^\intercal\mathbf{x}_n}\right)}_{e(h(\mathbf{x}_n),y_n)}$$

Cross entropy error

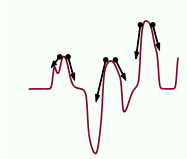
Gradient descent

We will use gradient descent to minimize our error function.

Fortunately, the logistic regression error function has a single global minimum:



So we don't need to worry about getting stuck in local minima



Gradient descent

Gradient descent is an iterative process

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta\hat{\mathbf{v}}$$

How to pick $\hat{\mathbf{v}}$?

Solution using gradient descent

Remember: gradient descent is a general purpose method for maximizing/minimizing a cost function

Approximating the change in E_{in}

$$\begin{aligned}\Delta E_{\text{in}} &= E_{\text{in}}(\mathbf{w}(t+1)) - E_{\text{in}}(\mathbf{w}(t)) \\ &= E_{\text{in}}(\mathbf{w}(t) + \eta\hat{\mathbf{v}}) - E_{\text{in}}(\mathbf{w}(t)) \\ &= \eta \nabla E_{\text{in}}(\mathbf{w}(t))^{\mathrm{T}}\hat{\mathbf{v}} + O(\eta^2)\end{aligned}$$

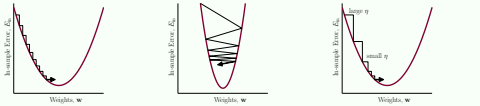
minimized at $\hat{\mathbf{v}} = -\frac{\nabla E_{\text{in}}(\mathbf{w}(t))}{\|\nabla E_{\text{in}}(\mathbf{w}(t))\|}$

13

Choosing the step size

The choice of the step size affects the rate of convergence:

η too small | η too large | variable η_t – just right



Let's take: $\eta_t = \eta \cdot \|\nabla E_{\text{in}}(\mathbf{w}(t))\|$

$$\begin{aligned}\hat{\mathbf{v}} &= -\eta_t \cdot \frac{\nabla E_{\text{in}}(\mathbf{w}(t))}{\|\nabla E_{\text{in}}(\mathbf{w}(t))\|} \\ &= -\eta \cdot \cancel{\|\nabla E_{\text{in}}(\mathbf{w}(t))\|} \cdot \frac{\nabla E_{\text{in}}(\mathbf{w}(t))}{\cancel{\|\nabla E_{\text{in}}(\mathbf{w}(t))\|}}\end{aligned}$$

$\|\nabla E_{\text{in}}(\mathbf{w}(t))\| \to 0$ when closer to the minimum.

$$\hat{\mathbf{v}} = -\eta \cdot \nabla E_{\text{in}}(\mathbf{w}(t))$$

14

Logistic regression using gradient descent

Putting it all together:

1: Initialize at step $t = 0$ to $\mathbf{w}(0)$.
2: **for** $t = 0, 1, 2, \ldots$ **do**
3: Compute the gradient

$$\mathbf{g}_t = \nabla E_{\text{in}}(\mathbf{w}(t)).$$

4: Move in the direction $\mathbf{v}_t = -\mathbf{g}_t$.
5: Update the weights:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta\mathbf{v}_t.$$

6: Iterate 'until it is time to stop'.
7: **end for**
8: Return the final weights.

$$\nabla E_{\text{in}} = -\frac{1}{N}\sum_{n=1}^{N}\frac{y_n\mathbf{x}_n}{1+e^{y_n\mathbf{w}^{\mathrm{T}}(t)\mathbf{x}_n}}$$

15

Logistic regression

Comments:

- Assumptions: i.i.d. data and form of P(y | x). Alternative to the assumption of P(y | x): Features have a Gaussian distribution for each class, i.e.
  $P(x_i|Y = y_k)$ has a normal distribution
- In practice logistic regression is solved by faster methods than gradient descent
- There is an extension to multi-class classification

16

Stochastic gradient descent

Variation on gradient descent that considers the error for a single training example:

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\ln(1+e^{-y_n\mathbf{w}^{\mathrm{T}}\mathbf{x}_n}) = \frac{1}{N}\sum_{n=1}^{N}e(\mathbf{w},\mathbf{x}_n,y_n)$$

Pick a random data point $(\mathbf{x}_*, y_*)$

Run an iteration of GD on $e(\mathbf{w}, \mathbf{x}_*, y_*)$

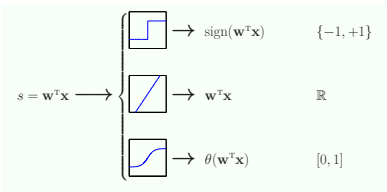
$$\mathbf{w}(t+1) \leftarrow \mathbf{w}(t) - \eta\nabla_{\mathbf{w}}e(\mathbf{w},\mathbf{x}_*,y_*)$$

$$\mathbf{w}(t+1) \leftarrow \mathbf{w}(t) + y_*\mathbf{x}_*\left(\frac{\eta}{1+e^{y_*\mathbf{w}^{\mathrm{T}}\mathbf{x}_*}}\right)$$

17

Summary of linear models

Linear methods for classification and regression:



18