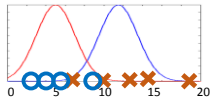
Logistic Classifier

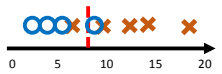
CISC 5800

Professor Daniel Leeds

Classification strategy: generative vs. discriminative



- Generative, e.g., Bayes/Naïve Bayes:
  - Identify probability distribution for each class
  - Determine class with maximum probability for data example
- Discriminative, e.g., Logistic Regression:
  - Identify boundary between classes
  - Determine which side of boundary new data example exists on



2

Linear algebra: data features

- Vector – list of numbers: each number describes a data **feature**
- Matrix – list of lists of numbers: features for each data point

	Document 1	Document 2	Document 3
Wolf	12	8	0
Lion	16	10	2
Monkey	14	11	1
Broker	0	1	14
Analyst	1	0	10
Dividend	1	1	12
⋮	⋮	⋮	⋮

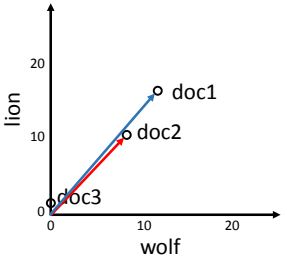
# of word occurrences

3

Feature space

- Each data feature defines a dimension in space

	Document1	Document2	Document3
Wolf	12	8	0
Lion	16	10	2
Monkey	14	11	1
Broker	0	1	14
Analyst	1	0	10
Dividend	1	1	12
⋮	⋮	⋮	⋮

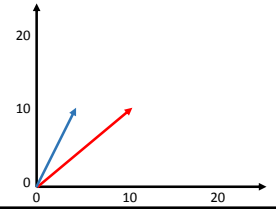


4

The dot product

The dot product compares two vectors:

• $\boldsymbol{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$ $\quad \boldsymbol{a} \cdot \boldsymbol{b} = \sum_{i=1}^{n} a_i b_i \quad = \boldsymbol{a}^T \boldsymbol{b}$



$$\begin{bmatrix} 5 \\ 10 \end{bmatrix} \cdot \begin{bmatrix} 10 \\ 10 \end{bmatrix} = 5 \times 10 + 10 \times 10$$

$$= 50 + 100 = 150$$

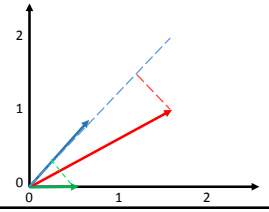
5

The dot product, continued $\boldsymbol{a} \cdot \boldsymbol{b} = \sum_{i=1}^{n} a_i b_i$

Magnitude of a vector is the sum of the squares of the elements

$|\boldsymbol{a}| = \sqrt{\sum_i a_i^2}$

If $\boldsymbol{a}$ has unit magnitude, $\boldsymbol{a} \cdot \boldsymbol{b}$ is the "projection" of $\boldsymbol{b}$ onto $\boldsymbol{a}$



$$\begin{bmatrix} 0.6 \\ 0.8 \end{bmatrix} \cdot \begin{bmatrix} 1.5 \\ 1 \end{bmatrix} = 0.6 \times 1.5 + 0.8 \times 1$$

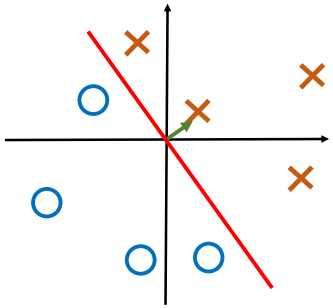
$$= 0.9 + 0.8 = 1.7$$

$$\begin{bmatrix} 0.6 \\ 0.8 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} = 0.6 \times 0 + 0.8 \times 0.5$$

$$= 0 + .4 = 0.4$$

7

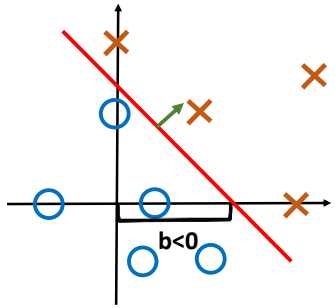
Separating boundary, defined by **w**



- Separating **hyperplane** splits **class 0** and **class 1**
- Plane is defined by line **w** perpendicular to plan
- Is data point **x** in class 0 or class 1? $\mathbf{w}^T\mathbf{x}+b > 0$ class **1**
  $\mathbf{w}^T\mathbf{x}+b < 0$ class **0**

8

Separating boundary, defined by **w** and b



Example:

$\mathbf{w} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ b=-4

$\mathbf{x}^1 = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$ $\mathbf{w}^T\mathbf{x}^1 + b = 1$

$\mathbf{x}^2 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$ $\mathbf{w}^T\mathbf{x}^2 + b = -2$

10

Notational simplification

Recall: $\boldsymbol{w}^T\boldsymbol{x} = \boldsymbol{w}\cdot\boldsymbol{x} = \sum_{i=1}^{n} w_i x_i$

Define $x'_{1:n} = x_{1:n}$ and $x'_{n+1} = 1$ for all inputs $\boldsymbol{x}$ and $w'_{1:n} = w_{1:n}$ and $w'_{n+1} = b$

Now $\boldsymbol{w'}^T\boldsymbol{x'} = \boldsymbol{w}^T\boldsymbol{x} + b$

Let's assume x_{n+1}=1 always, and w_{n+1}=b always

From real-number projection to 0/1 label

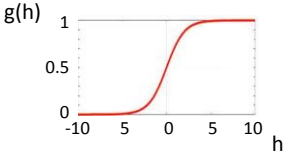
Binary classification: 0 is class A, 1 is class B

Sigmoid function stands in for p(x|y)

Sigmoid: $g(h) = \frac{1}{1+e^{-h}}$

$p(y=0|x;\theta) = 1 - g(w^T x) = \frac{e^{-w^T x}}{1+e^{-w^T x}}$

$p(y=1|x;\theta) = g(w^T x) = \frac{1}{1+e^{-w^T x}}$



$$w^T x = \sum_j w_j x_j + b$$

12

Learning parameters for classification

Similar to MLE for Bayes classifier

"Likelihood" for data points $y^1, \ldots, y^n$ (different from Bayesian likelihood)

- If y^i in class A, y^i =0, multiply (1-g(x^i;w))
- If y^i in class B, y^i=1, multiply (g(x^i;w))

$$\underset{w}{\operatorname{argmax}}\, L(y|x;w) = \prod_i \left(1 - g(x^i;w)\right)^{(1-y^i)} g(x^i;w)^{y^i}$$

13

Learning parameters for classification

The long derivation

- Similar to MLE for Bayes classifier
- "Likelihood" for data points $y^1, \ldots, y^n$ (different from Bayesian likelihood)
  - If y^i in class A, y^i =0, multiply (1-g(x^i;w))
  - If y^i in class B, y^i=1, multiply (g(x^i;w))

$$\underset{w}{\operatorname{argmax}}\, L(y|x;w) = \prod_i \left(1 - g(x^i;w)\right)^{(1-y^i)} g(x^i;w)^{y^i}$$

$$LL(y|x;w) = \sum_i (1-y^i)\log\left(1 - g(x^i;w)\right) + y^i \log\left(g(x^i;w)\right)$$

$$LL(y|x;w) = \sum_i y^i \log\frac{g(x^i;w)}{1-g(x^i;w)} + \log\left(1 - g(x^i;w)\right)$$

14

Learning parameters for classification

The long derivation

$$g(h) = \frac{1}{1+e^{-h}}$$

$$LL(y|x;w) = \sum_i y^i \log \frac{g(x^i;w)}{1-g(x^i;w)} + \log\left(1-g(x^i;w)\right)$$

$$LL(y|x;w) = \sum_i y^i \log \frac{\frac{1}{1+e^{-w^T x^i}}}{1-\frac{1}{1+e^{-w^T x^i}}} + \log\left(\frac{e^{-w^T x^i}}{1+e^{-w^T x^i}}\right)$$

$$LL(y|x;w) = \sum_i y^i \log \frac{1}{1+e^{-w^T x^i}-1} + \log\left(\frac{e^{-w^T x^i}}{1+e^{-w^T x^i}}\right)$$

$$LL(y|x;w) = \sum_i y^i w^T x^i - w^T x^i - \log\left(1+e^{-w^T x^i}\right)$$

15

Learning parameters for classification

The long derivation

$$w^T x = \sum_j w_j x_j$$

$$g'(h) = \frac{e^{-h}}{(1+e^{-h})^2}$$

$$LL(y|x;w) = \sum_i y^i w^T x^i - w^T x^i + \log\left(g(w^T x^i)\right)$$

$$\frac{\partial}{\partial w_j} LL(y|x;w) = \sum_i y^i x_j^i - x_j^i + \frac{x_j^i e^{-w^T x^i}}{1+e^{-w^T x^i}}$$

$$\frac{\partial}{\partial w_j} LL(y|x;w) = \sum_i x_j^i \left(y^i - (1-(1-g(w^T x^i)))\right)$$

$$\frac{\partial}{\partial w_j} LL(y|x;w) = \sum_i x_j^i \left(y^i - g(w^T x^i)\right)$$

16

Iterative gradient ascent

y^i – true data label
$g(w^T x^i)$ – computed data label

Begin with initial guessed weights w

For each data point (y^i, x^i), update each weight w_j

$$w_j \leftarrow w_j + \varepsilon x_j^i \left(y^i - g(w^T x^i)\right)$$

Choose ε so change is not too big or too small – **"step size"**

Intuition

$x_j^i \left(y^i - g(w^T x^i)\right)$

- If y^i=1 and $g(w^T x^i)$=0, and x_j^i>0, make w_j larger and push $w^T x^i$ to be larger
- If y^i=0 and $g(w^T x^i)$=1, and x_j^i>0, make w_i smaller and push $w^T x^i$ to be smaller

17

Iterative gradient ascent – big picture

Initialize **w** with random values

- Loop across all training data x^i for each feature x^i_j
- Repeat this loop many times (100x, or 1000x, etc.)

Repeat process 10-100 times, each time with new random initial **w**

Find **w** with high training set classification performance

- Goal of repeat initialization: avoid settling on local maximum

MAP for discriminative classifier

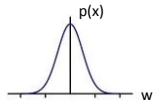
MLE: P(y=1|x;w) ~ $g(w^T x)$

MAP: P(y=1,w|x) ∝ P(y=1|x;w) P(w) ~ $g(w^T x)$???
(different from Bayesian posterior)

P(w) priors
- L2 regularization – minimize all weights
- L1 regularization – minimize number of non-zero weights

19

MAP – L2 regularization



- P(y=1,w|x) ∝ P(y=1|x;w) P(w):

$$L(y,w|x) = \prod_i \left(1 - g(x^i;w)\right)^{(1-y^i)} g(x^i;w)^{y^i} \prod_j e^{-\frac{w_j^2}{2\lambda}}$$

$$LL(y,w|x) = \sum_i y^i w^T x^i - w^T x^i + \log\left(g(w^T x^i)\right) - \sum_j \frac{w_j^2}{2\lambda}$$

$$\frac{\partial}{\partial w_j} LL(y,w|x) = \sum_i x_j^i \left(y^i - g(w^T x^i)\right) - \frac{w_j}{\lambda}$$

20

Thinking about your data: numeric and non-numeric features

Data to be classified can have multiple features $\boldsymbol{x}^i = \begin{bmatrix} x_1^i \\ \vdots \\ x_n^i \end{bmatrix}$

Each feature could be:
- Numeric: Loudness of music, from 0 to 30 decibels
- Non-numeric: Action, including Laugh, Cry, Jump, Dance

Classifier choice

Logistic regression only makes sense for numeric data

Gaussian Bayes only makes sense for numeric data

Multinomial Bayes makes sense for non-numeric data

Non-numeric features -> numeric

You may map non-numeric features to continuous space

Example:

- Mood={Depressed, Disappointed, Neutral, Happy, Excited}
- Switch to: HappinessLevel = {-2, -1, 0, 1, 2}