Bayesian classification

CISC 5800

Professor Daniel Leeds

Introduction to classifiers

- Goal: learn function C to maximize correct labels (Y) based on features (X)

C(x)=y

lion: 16
wolf: 12
monkey: 14
broker: 0
analyst: 1
dividend: 1

C → jungle

lion: 0
wolf: 2
monkey: 1
broker: 14
analyst: 10
dividend: 12

C → wallStreet

Giraffe detector

- Label X : height
- Class Y : True or False ("is giraffe" or "is not giraffe")

Learn optimal classification parameter(s)

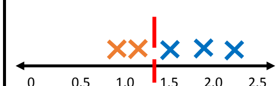
- Parameter: x^{thresh}

Example function:

$$C(x) = \begin{cases} True & \text{if } x > x^{thresh} \\ False & \text{otherwise} \end{cases}$$

Learning our classifier parameter(s)

- Adjust parameter(s) based on observed data
- Training set: contains features and corresponding labels

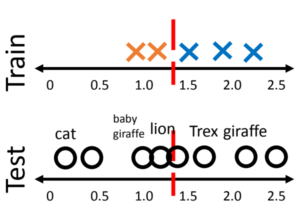


X	Y
1.5	True
2.2	True
1.8	True
1.2	False
0.9	False

The testing set

Testing set should be distinct from training set!

- Does classifier correctly label new data?



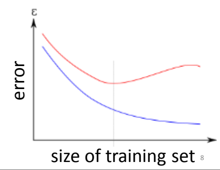
Example "good" performance:
90% correct labels

Be careful with your training set

- What if we train with only baby giraffes and ants?
- What if we train with only T rexes and adult giraffes?

Training vs. testing

- **Training**: learn parameters from set of data in each class
- **Testing**: measure how often classifier correctly identifies new data
- More training reduces classifier error ε
- Too much training data causes worse testing error – overfitting



8

Quick probability review

- P(G=C|H=True)
- P(G=C,H=True)
- P(H=True)
- P(H=True|G=C)

G	H	P(G,H)
A	False	0.05
B	False	0.05
C	False	0.05
D	False	0.1
A	True	0.3
B	True	0.2
C	True	0.15
D	True	0.1

9

Bayes rule

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Typically:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

where **D is the observed data** and **θ are the parameters** to describe that data
Our job is to find the most likely parameters for given data

- A posteriori probability: Probability of Parameters p for data d: $P(\theta|D)$
- Likelihood: Probability of data d given it is from Parameters p: $P(D|\theta)$
- Prior: Probability of observing Parameters p: $P(\theta)$

Parameters may be treated as analogous to class

10

Typical classification approaches

- MAP – Maximum A Posteriori: Determine parameters/class that has maximum probability

$$\underset{\theta}{\operatorname{argmax}} P(\theta|D)$$

- MLE – Maximum Likelihood: Determine parameters/class which maximize probability of the data

$$\underset{\theta}{\operatorname{argmax}} P(D|\theta)$$

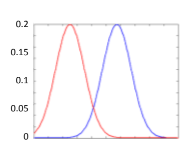
11

Likelihood: $P(D|\theta)$

- Each parameter has own distribution of possible data
- Distribution described by **parameter(s)** in θ

Example

- Classes: {Horse, Dog}
- Feature: RunningSpeed: [0 20]
- Model as Gaussian with fixed σ
- $\mu_{horse} = 11.5$, $\mu_{dog} = 5$



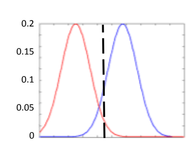
12

The prior: $P(\theta)$

$$P(\theta|D) \propto P(D|\theta)P(\theta)$$

- Certain parameters/classes are more common than others
- Classes: {Horse, Dog}
- P(Horse)=0.05, P(Dog)=0.95
- High likelihood may not mean high posterior

Which is higher?
P(Horse|D=9)
P(Dog|D=9)



13

Review

Classify by finding class with max posterior or max likelihood

- $\underset{\theta}{\text{argmax}}\, P(\theta|D) \propto P(D|\theta)\boldsymbol{P(\theta)}$
- Posterior ∝ Likelihood x Prior   ∝ - means proportional

We "ignore" the P(D) denominator because D stays same while comparing different classes (θ)

Learning probabilities

- We have a coin biased to favor one side
- How can we calculate the bias?
- Data (D): {HHTH, TTHH, TTTT, HTTT}   Bias (θ): p probability of H
- $P(D|\theta) = p^{|H|}(1-p)^{|T|}$ |H| - # heads, |T| - # tails

Optimization: finding the maximum likelihood

$\underset{\theta}{\text{argmax}}\, P(D|\theta) =$

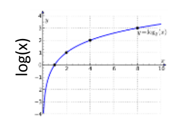
$\underset{p}{\text{argmax}}\, p^{|H|}(1-p)^{|T|}$

p - probability of Head

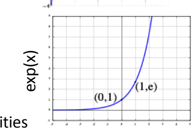
Equivalently, maximize $\log P(D|\theta)$

$\underset{p}{\text{argmax}}\, |H|\log p + |T|\log(1-p)$

The properties of logarithms



- $e^a = b \leftrightarrow \log b = a$
- $a < b \leftrightarrow \log a < \log b$
- $\log ab = \log a + \log b$
- $\log a^n = n \log a$



Convenient when dealing with small probabilities

- 0.0000454 x 0.000912 = 0.0000000414 -> -10 + -7 = -17

Optimization: finding the maximum likelihood

$\underset{\theta}{\text{argmax}}\, P(D|\theta) =$

$\underset{p}{\text{argmax}}\, p^{|H|}(1-p)^{|T|}$

p - probability of Head

Equivalently, maximize $\log P(D|\theta)$

$\underset{p}{\text{argmax}}\, |H|\log p + |T|\log(1-p)$

Optimization: finding zero slope

- Location of maximum has slope 0

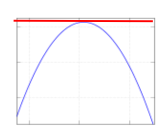
p - probability of Head

maximize $\log P(D|\theta)$

$\underset{p}{\text{argmax}}\, |H|\log p + |T|\log(1-p):$

$\frac{d}{dp}|H|\log p + |T|\log(1-p) = 0$

$\frac{|H|}{p} - \frac{|T|}{1-p} = 0$



Intuition of the MLE result

$$p = \frac{|H|}{|H| + |T|}$$

- Probability of getting heads is # heads divided by # total flips

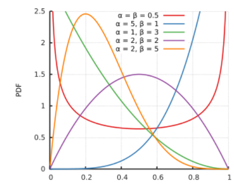
Finding the maximum **a posteriori**

- $P(\theta|D) \propto P(D|\theta)P(\theta)$
- Incorporating the Beta prior:

$$P(\theta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)}$$

$\underset{\theta}{\operatorname{argmax}} P(D|\theta)P(\theta) =$

$\underset{\theta}{\operatorname{argmax}} \log P(D|\theta) + \log P(\theta)$



MAP: estimating θ (estimating p)

$\underset{\theta}{\operatorname{argmax}} \log P(D|\theta) + \log P(\theta)$

$\underset{p}{\operatorname{argmax}} |H| \log p + |T| \log(1-p) +$

$(\alpha - 1)\log p + (\beta - 1)\log(1-p) - \log(\mathrm{B}(\alpha,\beta))$

Set derivative to 0

$$\frac{|H|}{p} - \frac{|T|}{1-p} + \frac{(\alpha-1)}{p} - \frac{(\beta-1)}{1-p} = 0$$

$$(1-p)|H| - p|T| + (1-p)(\alpha-1) - p(\beta-1) = 0$$

$$|H| + (\alpha-1) = (|H| + |T| + (\alpha-1) + (\beta-1))p$$

Intuition of the MAP result

$$p = \frac{|H| + (\alpha-1)}{|H| + (\alpha-1) + |T| + (\beta-1)}$$

- Prior has strong influence when |H| and |T| small
- Prior has weak influence when |H| and |T| large

Multiple features

Dr. Lyon's lecture:

- Position coordinates: x, y, angle
- Pictures: pixels, sonar

Sometimes multiple features provide new information

- Robot localization: (2,4) different from (2,2) and from (4,4)

Sometimes multiple features redundant:

- Super-hero fan: Watch Batman? Watch Superman?

Assuming independence: Is there a storm?

- P(storm | lightning, wind) : $P(S|L,W)$
- $P(S|L,W) = \frac{P(L,W|S)P(S)}{P(L,W)} \propto \boldsymbol{P(L,W|S)}P(S)$
- Let's assume L and W are independent given S
- $P(L,W|S)$ =?

Estimating P(Lightning|Storm)

- Is there Lightning? Yes or No (Binary variable like Heads or Tails)
- P(L=yes|S=yes) – Probability of lightning given there's a storm
- P(L=no|S=yes) = ?
- What is MLE of P(L=yes|S=yes)?
- What is MLE of P(L=yes|S=no)?

26

MLE – counting data points

Updated Oct 1:

- $P(A = a_i | C = c_j) = \frac{\#D\{A=a_i \wedge C=c_j\}}{\#D\{C=c_j\}}$

Note: both A and C can take on multiple values (binary **and** beyond)

- $P(A = a_i, B = b_k | C = c_j) = \frac{\#D\{A=a_i \wedge B=b_k \wedge C=c_j\}}{\#D\{C=c_j\}}$

27

P(L,W|S)

Non-independent, estimate:

- P(L=yes,W=yes|S=yes)
- P(L=yes,W=no|S=yes)
- P(L=no,W=yes|S=yes)
- Deduce P(L=no,W=no|S=yes):

$$1 - \sum_{(L,W)\neq(no,no)} P(L, W | S = yes)$$

- Repeat for S=no

$P(A_1,\ldots,A_n|C)$

Number of parameters to estimate:

- For each class find 2^n-1
- In total: 2 (2^n-1)

Updated Oct 1:

Note: in this slide, all variables are binary

28

P(L,W|S)=P(L|S)P(W,S)

Independent, estimate:

- P(L=yes|S=yes)
- Deduce P(L=no|S=yes): 1-P(L=yes|S=yes)
- P(W=yes|S=yes)
- Deduce P(W=no|S=yes): 1-P(W=yes|S=yes)
- Repeat for S=no

$P(A_1,\ldots,A_n|C)$

Number of parameters to estimate:

- For each class find n
- In total: 2 n

Updated Oct 1:

Note: in this slide, all variables are binary

29

Naïve Bayes: Classification + Learning

Updated Oct 1:

- Want to know $P(Y|X_1,X_2,\ldots,X_n)$
- Compute $P(X_1,X_2,\ldots,X_n|Y)$ and P(Y)
  - Compute $P(X_1, X_2, \ldots, X_n|Y) = \prod P(X_i|Y)$

Note: both X and Y can take on multiple values (binary **and** beyond)

Learning:

- Estimate each $P(X_i|Y)$ (through MLE)

$$P(X_i = x_k | Y = y_j) = \frac{\#D(X_i = x_k \wedge Y = y_j)}{\#D(Y = y_j)}$$

- Estimate P(Y)

$$P(Y = y_j) = \frac{\#D(Y = y_j)}{|D|}$$

30

Shortcoming of MLE

Updated Oct 1:

$$P(X_i = x_k | Y = y_j) = \frac{\#D(X_i = x_k \wedge Y = y_j)}{\#D(Y = y_j)}$$

Note: both X and Y can take on multiple values (binary **and** beyond)

- What if $X_i = x_k \wedge Y = y_j$ is very rare, but possible?

Example – classify articles:

- X_i – does $word_i$ appear in article?
- Y={jungle, wallStreet}
- X_i=broker very unlikely in jungle:
  - MLE $P(X_i$=broker|Y=jungle)=0

lion: 16
wolf: 12
monkey: 14
broker: 0
analyst: 0
dividend: 0

C → jungle

- $P(X_1 = x_{11}, \ldots, X_n = x_{n1} | Y = y_j) = \prod_i P(X_i = x_{i1} | Y = y_j)$

31

Estimate each P(X$_i$|Y) through **MAP**

Incorporating prior for each class β_j

$$P(X_i = x_k | Y = y_j) = \frac{\#D(X_i = x_k \wedge Y = y_j) + (\beta_j - 1)}{\#D(Y = y_j) + \sum_m (\beta_m - 1)}$$

$$P(Y = y_j) = \frac{\#D(Y = y_j) + (\beta_j - 1)}{|D| + \sum_m (\beta_m - 1)}$$

Extra note:

Updated Oct 1:

$(\boldsymbol{\beta_j} - \mathbf{1})$ **– "frequency" of class j**
$\boldsymbol{\sum_m (\beta_m - 1)}$ **– "frequencies" of all classes**

Note: both X and Y can take on multiple values (binary and beyond)

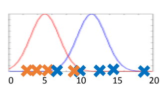
32

Benefits of Naïve Bayes

- Very fast learning and classifying:
  - 2n+1 parameters, not 2x(2^n-1)+1 parameters
- Often works even if features are NOT independent

33

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

34

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
⁝	⁝	⁝	⁝

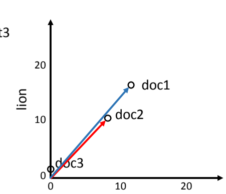
# of word occurrences

35

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
⁝	⁝	⁝	⁝



36

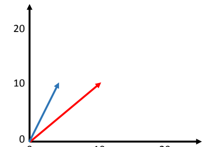
The dot product

The dot product compares two vectors:

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

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

$$= \mathbf{50 + 100 = 150}$$



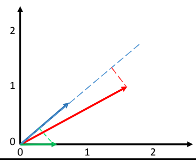
37

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 $\boldsymbol{a}$



$$\begin{bmatrix}0.71\\0.71\end{bmatrix} \cdot \begin{bmatrix}1.5\\1\end{bmatrix} = .71 \times 1.5 + .71 \times 1$$

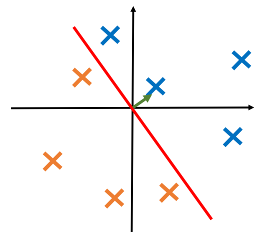
$$\approx 1.07 + .71 = 1.78$$

$$\begin{bmatrix}0.71\\0.71\end{bmatrix} \cdot \begin{bmatrix}0\\0.5\end{bmatrix} = .71 \times 0 + .71 \times 0.5$$

$$\approx 0 + .35 = 0.35$$

38

Separating boundary, defined by w

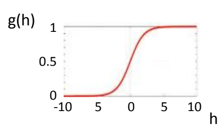


- Separating **hyperplane** splits **class 0** and **class 1**
- Plane is defined by line **w** perpendicular to plan
- Is data point **x** in class 0 or class 1? $\mathbf{w}^T\mathbf{x} > 0$ class 0
  $\mathbf{w}^T\mathbf{x} < 0$ class 1

39

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



$$\boldsymbol{w}^T\boldsymbol{x} = \sum_j w_j x_j$$

40

Learning parameters for classification

- Similar to MLE for Bayes classifier
- "Likelihood" for data points $y^1, \ldots, y^n$ (really framed as posterior y|x)
  - 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 $(g(x^i;w))$

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

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

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

41

Learning parameters for classification

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

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

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

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

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

42

Learning parameters for classification

$$\boldsymbol{w}^T\boldsymbol{x} = \sum_j w_j x_j$$

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

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

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

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

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

43

Iterative gradient descent

y^i – true data label
$g(w^Tx^i)$ – 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

Intuition

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

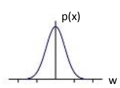
44

MAP for discriminative classifier

- MLE: P(x|y=1;w) ~ $g(w^Tx)$
- MAP: P(y=1|x) = P(x|y=1;w) P(w) ~ $g(w^Tx)$???
- P(w) priors
  - L2 regularization – minimize all weights
  - L1 regularization – minimize number of non-zero weights

45

MAP – L2 regularization



- P(y=1|x,w) = P(x|y=1;w) P(w):

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

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

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

46