10-701 Introduction to Machine Learning (PhD)
Lecture 7: Decision Trees

Leila Wehbe
Carnegie Mellon University
Machine Learning Department

Slides based on Tom Mitchell's
10-701 Spring 2016 material

Logistic Regression

Idea:

- Naïve Bayes allows computing P(Y|X) by learning P(Y) and P(X|Y)
- Why not learn P(Y|X) directly?

Derive form for P(Y|X) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

Derive form for P(Y|X) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

Derive form for P(Y|X) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$$= \frac{1}{1 + \exp(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

Derive form for P(Y|X) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$$= \frac{1}{1 + \exp(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

$$= \frac{1}{1 + \exp(\ (\ln \frac{1-\pi}{\pi}) + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}$$

Derive form for P(Y|X) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$$= \frac{1}{1 + \exp(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

$$= \frac{1}{1 + \exp(\ (\ln \frac{1-\pi}{\pi}) + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}$$

$$P(X_i = x_i|Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{\frac{-(x_i-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

Derive form for P(Y|X) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$$= \frac{1}{1 + \exp(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

$$= \frac{1}{1 + \exp(\ (\ln \frac{1-\pi}{\pi}) + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}$$

$$P(X_i = x_i|Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{\frac{-(x_i-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

$$\ln P(X_i = x_i|Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} + \frac{-(x_i-\mu_{ik})^2}{2\sigma_{ik}^2}$$

Derive form for P(YIX) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$$= \frac{1}{1 + \exp(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

$$= \frac{1}{1 + \exp(\ (\ln \frac{1-\pi}{\pi}) + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}$$

$$P(X_i = x_i|Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{\frac{-(x_i-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

$$\ln P(X_i = x_i|Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} + \frac{-(x_i-\mu_{ik})^2}{2\sigma_{ik}^2}$$

$$\ln \frac{P(X_i = x_i|Y=0)}{P(X_i = x_i|Y=1)} \propto + \frac{-(x_i-\mu_{i0})^2}{2\sigma_{i0}^2} - \frac{-(x_i-\mu_{i1})^2}{2\sigma_{i1}^2}$$

$$= \frac{x_i^2 - 2x_i\mu_{i1} + \mu_{i1}^2}{2\sigma_{i1}^2} - \frac{x_i^2 - 2x_i\mu_{i0} + \mu_{i0}^2}{2\sigma_{i0}^2}$$

Derive form for P(YIX) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$$= \frac{1}{1 + \exp(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

$$= \frac{1}{1 + \exp(\ (\ln \frac{1-\pi}{\pi}) + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}$$

$$P(X_i = x_i|Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{\frac{-(x_i-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

$$\ln P(X_i = x_i|Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} + \frac{-(x_i-\mu_{ik})^2}{2\sigma_{ik}^2}$$

$$\ln \frac{P(X_i = x_i|Y=0)}{P(X_i = x_i|Y=1)} \propto + \frac{-(x_i-\mu_{i0})^2}{2\sigma_{i0}^2} - \frac{-(x_i-\mu_{i1})^2}{2\sigma_{i1}^2}$$

$$= \frac{x_i^2 - 2x_i\mu_{i1} + \mu_{i1}^2}{2\sigma_{i1}^2} - \frac{x_i^2 - 2x_i\mu_{i0} + \mu_{i0}^2}{2\sigma_{i0}^2}$$

Now assume $\sigma_{ik} = \sigma_i$

Derive form for P(YIX) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$$= \frac{1}{1 + \exp(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

$$= \frac{1}{1 + \exp(\ (\ln \frac{1-\pi}{\pi}) + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}$$

Linear function!

$$\sum_i \left(\frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2} X_i + \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2} \right)$$

$$P(Y=1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^n w_i X_i)}$$

Very convenient!

$$P(Y=1|X =< X_1, \ldots X_n >) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

$$P\big(Y=0|X=(X_1,\ldots,X_n)\big) = \frac{exp(w_0 + \sum_i w_i X_w)}{1 + exp(w_0 + \sum_i w_i X_w)}$$

implies

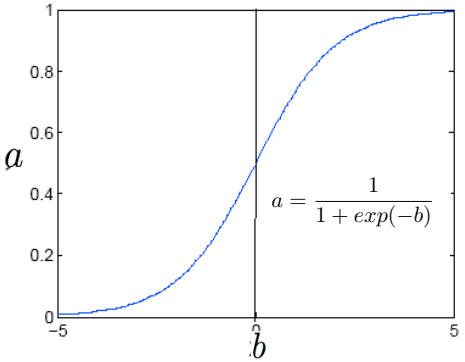
$$\frac{P(Y=0|X)}{P(Y=1|X)} = exp(w_0 + \sum_i w_i X_i)$$

linear classification rule!

implies

$$\ln \frac{P(Y=0|X)}{P(Y=1|X)} = w_0 + \sum_i w_i X_i \qquad < \text{or} > 0$$

Logistic function



$$a = \frac{exp(b)}{1+exp(b)}$$

$$P(Y=1|X) = \frac{1}{1+\exp(w_0+\sum_{i=1}^{n} w_i X_i)}$$

Maximize Conditional Log Likelihood: Gradient Ascent

$$l(W) \equiv \ln \prod_l P(Y^l|X^l,W) = \sum_l \ln P(Y^l|X^l,W)$$
$$= \sum_l Y^l(w_0 + \sum_i^n w_i X_i^l) - \ln(1+exp(w_0+\sum_i^n w_i X_i^l))$$

$$\frac{\partial l(W)}{\partial w_i} = \sum_l X_i^l(Y^l - \hat{P}(Y^l=1|X^l,W))$$

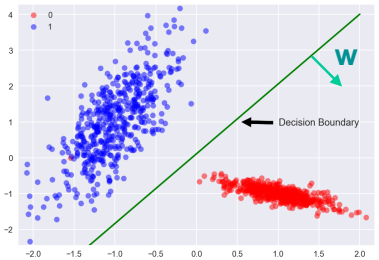
Gradient ascent algorithm: iterate until change < ε

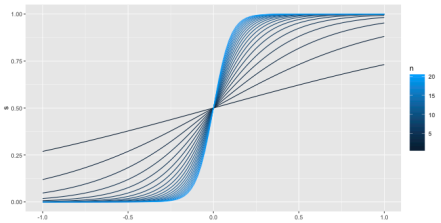
For all i, repeat

$$w_i \leftarrow w_i + \eta \sum_l X_i^l(Y^l - \hat{P}(Y^l=1|X^l,W))$$

Need to regularize the weights

- $w \to \infty$ to maximize the probability of the data, if data linearly separable





From https://stats.stackexchange.com/questions/224863/understanding-complete-separation-for-logistic-regression

MAP estimates and Regularization

- Maximum a posteriori estimate with prior

$$W \leftarrow \arg\max_W \ln[P(W) \prod_l P(Y^l|X^l,W)]$$

$$w_i \leftarrow w_i - \eta\lambda w_i + \eta \sum_l X_i^l(Y^l - \hat{P}(Y^l=1|X^l,W))$$

called a "regularization" term

- helps reduce overfitting
- if P(W) is Gaussian, then encourages W to be near the mean of P(W) : zero here, but can easily use any mean
- used very frequently in Logistic Regression

G.Naïve Bayes vs. Logistic Regression

Recall two assumptions deriving form of LR from GNBayes:

1. X_i conditionally independent of X_k given Y
2. $P(X_i \mid Y = y_k) = N(\mu_{ik},\sigma_i)$, ← not $N(\mu_{ik},\sigma_{ik})$

[Ng & Jordan, 2002]

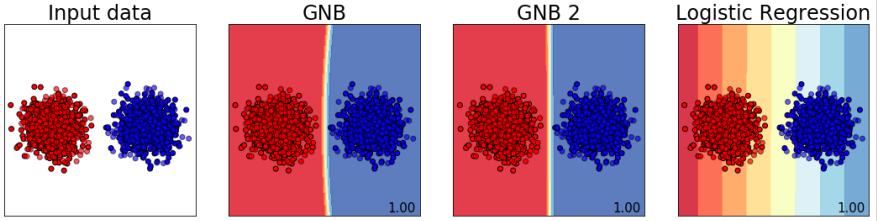
Consider three learning methods:

- •GNB (assumption 1 only) -- decision surface can be non-linear
- •GNB2 (assumption 1 and 2) – decision surface linear
- •LR -- decision surface linear, trained without assumption 1.

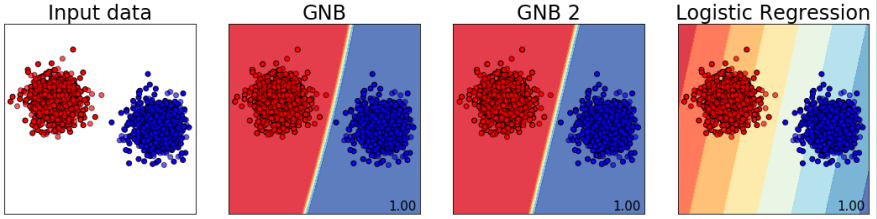
How do these methods perform if we have plenty of data and:

- •Both (1) and (2) are satisfied:

Assumptions 1 and 2 are satisfied



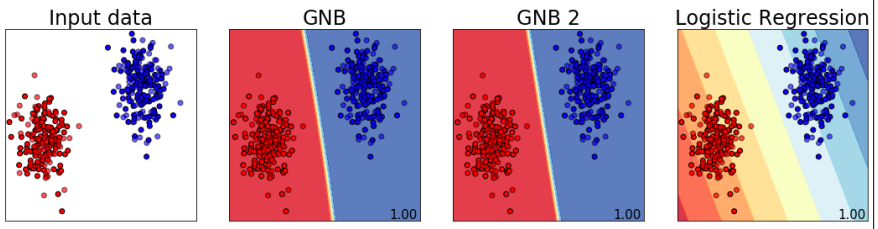
X_i's conditionally independent and variance is shared



In these cases, LR, GNB2 and GNB perform similarly

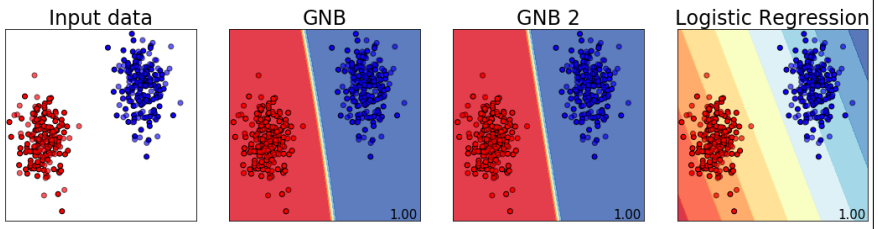
Assumptions 1 and 2 satisfied

The decision boundary of GNB and GNB2 is sensitive to the locations of the means (since the variances are the same)



Assumptions 1 and 2 satisfied

The decision boundary of GNB and GNB2 is sensitive to the locations of the means (since the variances are the same)

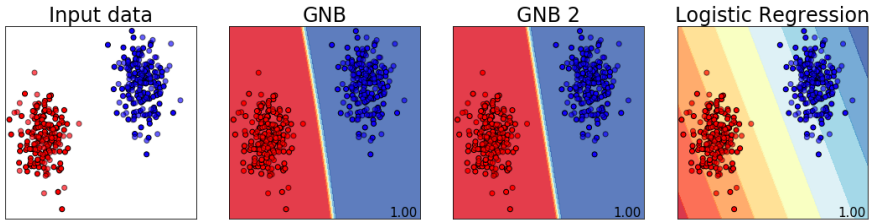


$$\ln \frac{P(X_i = x_i|Y=0)}{P(X_i = x_i|Y=1)} = \frac{1}{\sigma_{i0}\sqrt{2\pi}} + \frac{-(x_i-\mu_{i0})^2}{2\sigma_{i0}^2} - \frac{1}{\sigma_{i1}\sqrt{2\pi}} + \frac{-(x_i-\mu_{i1})^2}{2\sigma_{i1}^2}$$

Now assume $\sigma_{ik} = \sigma_i$

Assumptions 1 and 2 satisfied

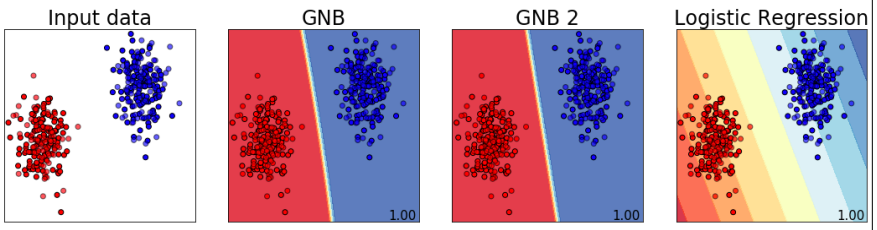
The decision boundary of GNB and GNB2 is sensitive to the locations of the means (since the variances are the same)



$$\sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)} = \sum_i \frac{(x_i-\mu_{i1})^2 - (x_i-\mu_{i2})^2}{2\sigma_i^2}$$

Assumptions 1 and 2 satisfied

The decision boundary of GNB and GNB2 is sensitive to the locations of the means (since the variances are the same)



$$\sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)} = \sum_i \frac{(x_i-\mu_{i1})^2 - (x_i-\mu_{i2})^2}{2\sigma_i^2}$$

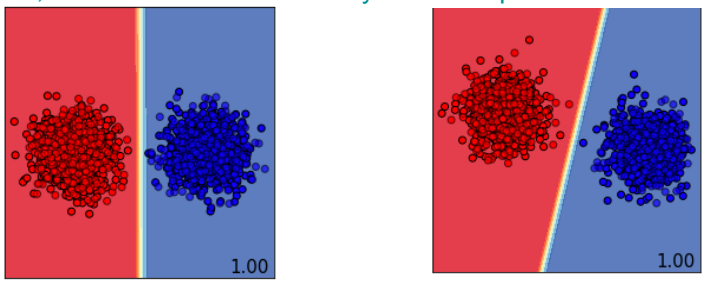
Distance between centers, weighted by variance on each dimension

Assumptions 1 and 2 satisfied

If the variances of the Xi are the same (across classes and across i), the decision boundary of GNB2 and GNB is determined by the distance to the mean (perpendicular bisector)

$$\sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)} = \sum_i \frac{(x_i-\mu_{i1})^2 - (x_i-\mu_{i2})^2}{2\sigma_i^2}$$

Independently, if one of the coordinates of the two means are the same, then the decision boundary becomes parallel to that axis



G.Naïve Bayes vs. Logistic Regression

Recall two assumptions deriving form of LR from GNBayes:

1. X_i conditionally independent of X_k given Y
2. $P(X_i \mid Y = y_k) = N(\mu_{ik},\sigma_i)$, ← not $N(\mu_{ik},\sigma_{ik})$

[Ng & Jordan, 2002]

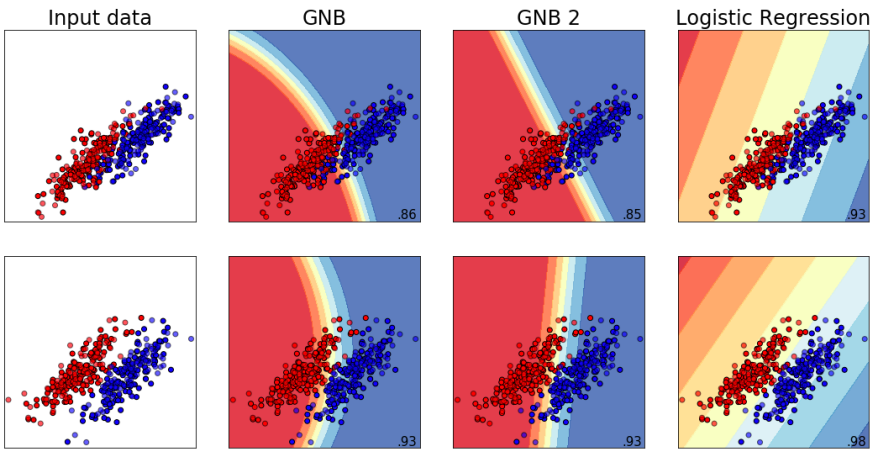
Consider three learning methods:

- GNB (assumption 1 only) -- decision surface can be non-linear
- GNB2 (assumption 1 and 2) – decision surface linear
- LR -- decision surface linear, trained without assumption 1.

How do these methods perform if we have plenty of data and:

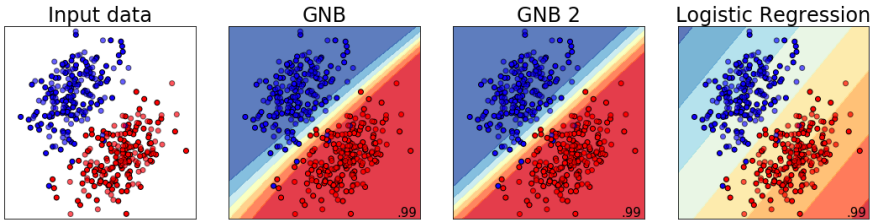
- Both (1) and (2) are satisfied
- (2) is satisfied, but not (1)

Assumption 2 satisfied and not 1



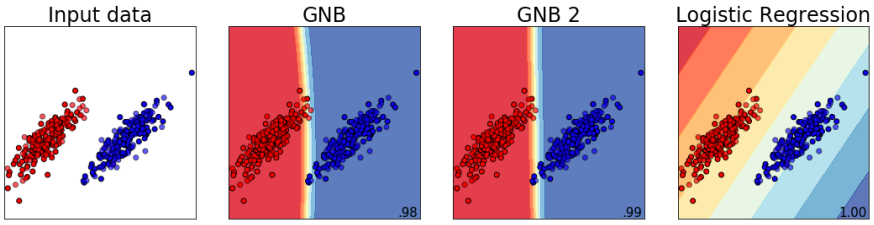
GNB2 can break (also GNB)

Assumption 2 satisfied and not 1



GNB2 and GNB can also work well

Assumption 2 satisfied and not 1



The decision boundary of GNB2 and GNB is also dependent on the means of the two classes. If one of the coordinates of the two means is the same, again, we have a decision boundary parallel to that axis

G.Naïve Bayes vs. Logistic Regression

Recall two assumptions deriving form of LR from GNBayes:

1. X_i conditionally independent of X_k given Y
2. $P(X_i \mid Y = y_k) = N(\mu_{ik},\sigma_i)$, ← not $N(\mu_{ik},\sigma_{ik})$

[Ng & Jordan, 2002]

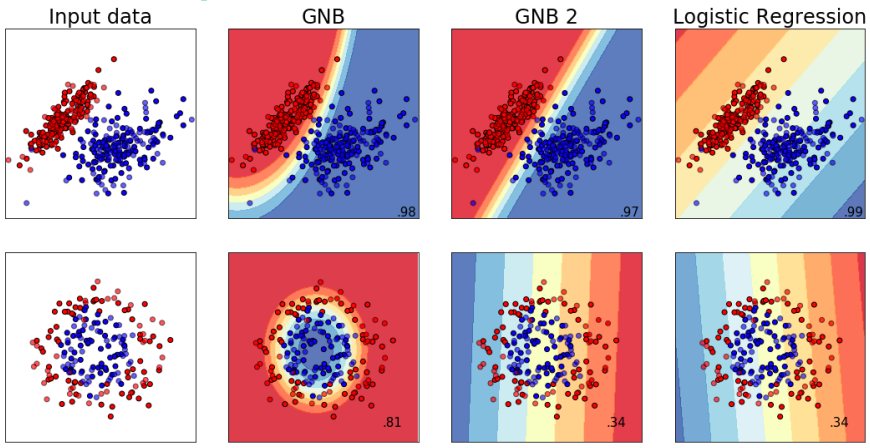
Consider three learning methods:

- GNB (assumption 1 only) -- decision surface can be non-linear
- GNB2 (assumption 1 and 2) – decision surface linear
- LR -- decision surface linear, trained without assumption 1.

How do these methods perform if we have plenty of data and:

- Both (1) and (2) are satisfied
- (2) is satisfied, but not (1)
- Neither (1) nor (2) is satisfied

Assumptions 1 and 2 are not satisfied



Depending on the dataset, GNB and LR have different performances. Even though LR and GNB2 can be expressed in the same way, LR has more flexibly to learn parameters that fit the data, and they are don't have to be tied to the marginal means and variance

Naïve Bayes vs. Logistic Regression

The bottom line:

GNB2 and LR both use linear decision surfaces, GNB need not

Given infinite data, LR is better or equal to GNB2 because *training procedure* does not make assumptions 1 or 2 (though our derivation of the form of P(Y|X) did).

But GNB2 converges more quickly to its perhaps-less-accurate asymptotic error. (more bias than LR)

And GNB is both more biased (assumption1) and less (no assumption 2) than LR, so either might outperform the other.

Decision Trees

Function approximation

Problem Setting:

- Set of possible instances X
- Unknown target function $f : X \rightarrow Y$
- Set of function hypotheses $H=\{ h \mid h : X \rightarrow Y \}$

Input:

- Training examples $\{(x^{(i)},y^{(i)})\}$ of unknown target function f

Output:

- Hypothesis $h \in H$ that best approximates target function f

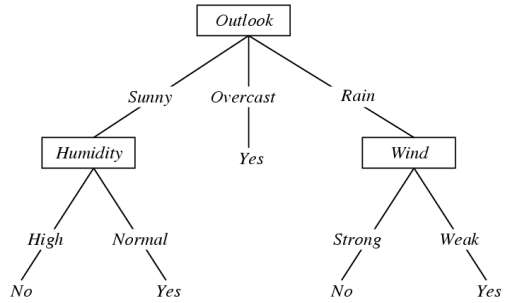
Simple Training Data Set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

A Decision tree for

f: (Outlook, Temperature, Humidity, Wind) → PlayTennis?

(X_1 X_2 X_3 X_4) → Y

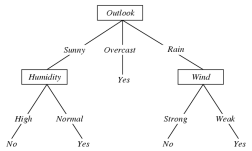


Each internal node: test one discrete-valued attribute X_i

Each branch from a node: selects one value for X_i

Each leaf node: predict Y (or P(Y|X ∈ leaf))

Decision Tree Learning



Problem Setting:

- Set of possible instances X
  - each instance x in X is vector of discrete-valued features $x = < x_1, x_2 \ldots x_n >$
- Unknown target function $f : X \rightarrow Y$
  - Y is discrete-valued
- Set of function hypotheses $H = \{ h \mid h : X \rightarrow Y \}$
  - each hypothesis h is a decision tree

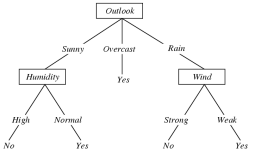
Input:

- Training examples $\{<x^{(i)}, y^{(i)}>\}$ of unknown target function f

Output:

- Hypothesis $h \in H$ that best approximates target function f

Decision Trees



Suppose $X = <X_1, \ldots X_n>$

where X_i are boolean-valued variables

How would you represent $Y = X_2 X_5$? $Y = X_2 \vee X_5$

How would you represent $X_2 X_5 \vee X_3 X_4 (\neg X_1)$

A Tree to Predict C-Section Risk

Learned from medical records of 1000 women

Negative examples are C-sections

```
[833+,167-] .83+ .17-
Fetal_Presentation = 1: [822+,116-] .88+ .12-
| Previous_Csection = 0: [767+,81-] .90+ .10-
| | Primiparous = 0: [399+,13-] .97+ .03-
| | Primiparous = 1: [368+,68-] .84+ .16-
| | | Fetal_Distress = 0: [334+,47-] .88+ .12-
| | | | Birth_Weight < 3349: [201+,10.6-] .95+ .
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+
| | | Fetal_Distress = 1: [34+,21-] .62+ .38-
| Previous_Csection = 1: [55+,35-] .61+ .39-
Fetal_Presentation = 2: [3+,29-] .11+ .89-
Fetal_Presentation = 3: [8+,22-] .27+ .73-
```

Top-Down Induction of Decision Trees

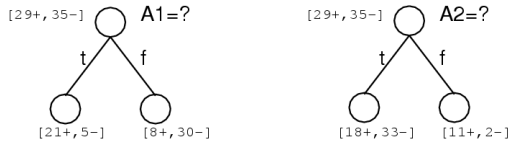
[ID3, C4.5, Quinlan]

node = Root

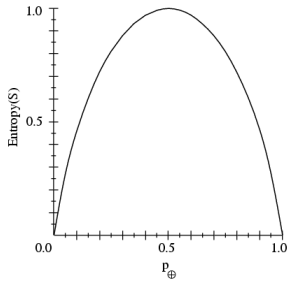
Main loop:

1. $A \leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?



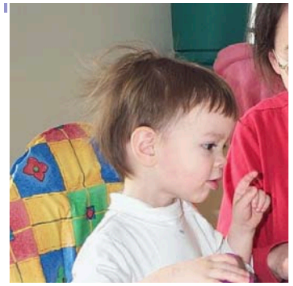
Sample Entropy



- S is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in S
- $p_\ominus$ is the proportion of negative examples in S
- Entropy measures the impurity of S

$$H(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

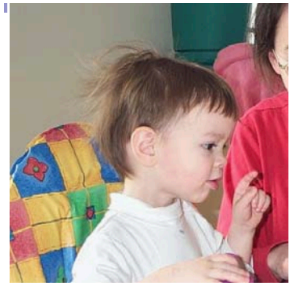
Andrew Moore's entropy example



Low Entropy

High Entropy

Andrew Moore's entropy example



Low Entropy

..the values (locations of soup) sampled entirely from within the soup bowl

High Entropy

..the values (locations of soup) unpredictable... almost uniformly sampled throughout our dining room

Entropy

Entropy $H(X)$ of a random variable X

$$H(X) = -\sum_{i=1}^{n} P(X=i)\log_2 P(X=i)$$

# of possible values for X

$H(X)$ is the expected number of bits needed to encode a randomly drawn value of X (under most efficient code)

Why? Information theory:

- Most efficient possible code assigns $-\log_2 P(X=i)$ bits to encode the message $X=i$
- So, expected number of bits to code one random X is:

$$\sum_{i=1}^{n} P(X=i)(-\log_2 P(X=i))$$

Entropy

Entropy $H(X)$ of a random variable X

$$H(X) = -\sum_{i=1}^{n} P(X=i)\log_2 P(X=i)$$

Specific conditional entropy $H(X|Y=v)$ of X given $Y=v$:

$$H(X|Y=v) = -\sum_{i=1}^{n} P(X=i|Y=v)\log_2 P(X=i|Y=v)$$

Conditional entropy $H(X|Y)$ of X given Y :

$$H(X|Y) = \sum_{v \in values(Y)} P(Y=v)H(X|Y=v)$$

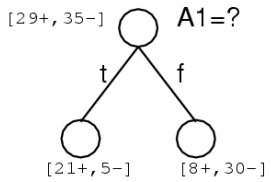
Mutual information (aka Information Gain) of X and Y :

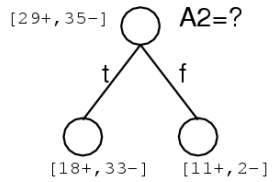
$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Information Gain is the mutual information between input attribute A and target variable Y

Information Gain is the expected reduction in entropy of target variable Y for data sample S, due to sorting on variable A

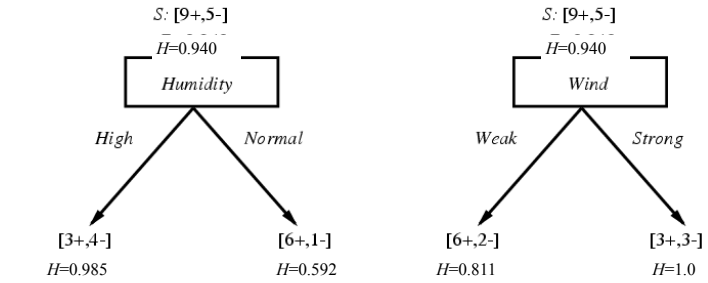
$$Gain(S,A) = I_S(A,Y) = H_S(Y) - H_S(Y|A)$$





Selecting the Next Attribute

Which attribute is the best classifier?



Gain (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151

Gain (S, Wind)
= .940 - (8/14).811 - (6/14)1.0
= .048

Example in class

- Try to find IG(Y,X1), IG(Y,X2) and IG(Y,X3)

X1	X2	X3	Y
1	1	1	+
1	1	0	+
0	0	1	-
1	0	0	-

Example in class

- H(Y) = 1
- H(Y|X1=1) = -1/3 log2(1/3) - 2/3 log2(2/3) = 0.92
- H(Y|X1=0) = -1log2(1) = 0
- H(Y|X1) = 3/4 * H(Y|X1=1) +1/4* H(Y|X1=0) ~ 0.92
- IG(Y,X1) ~ 0.31

X1	X2	X3	Y
1	1	1	+
1	1	0	+
0	0	1	-
1	0	0	-

Example in class

- H(Y) = 1
- H(Y|X2=1) = -1log2(1) = 0
- H(Y|X2=0) = -1log2(1) = 0
- H(Y|X2) = 1/2 * H(Y|X2=1) +1/2* H(Y|X2=0)=0
- IG(Y,X2) = 1

- Pick X2!

X1	X2	X3	Y
1	1	1	+
1	1	0	+
0	0	1	-
1	0	0	-

Example in class

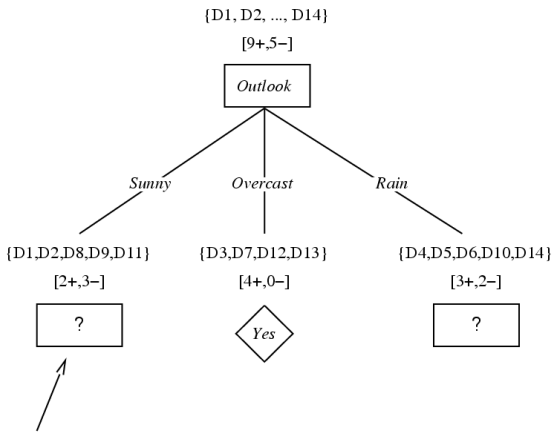
- H(Y) = 1
- H(Y|X3=1) = -1/2log2(1/2) -1/2log2(1/2) = 1
- H(Y|X3=0) = -1/2log2(1/2) -1/2log2(1/2) = 1
- H(Y|X3) = 1/2 * H(Y|X3=1) +1/2* H(Y|X3=0)=1
- IG(Y,X3) = 0

- X3 doesn't help at all at this step

X1	X2	X3	Y
1	1	1	+
1	1	0	+
0	0	1	-
1	0	0	-

Simple Training Data Set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Which attribute should be tested here?

S_{sunny} = {D1,D2,D8,D9,D11}

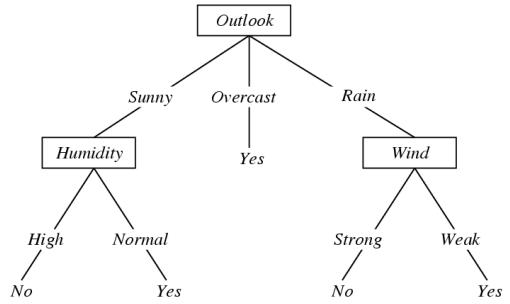
Gain (S_{sunny}, *Humidity*) = .970 − (3/5) 0.0 − (2/5) 0.0 = .970

Gain (S_{sunny}, *Temperature*) = .970 − (2/5) 0.0 − (2/5) 1.0 − (1/5) 0.0 = .570

Gain (S_{sunny}, *Wind*) = .970 − (2/5) 1.0 − (3/5) .918 = .019

Final Decision Tree for

f: <Outlook, Temperature, Humidity, Wind> → PlayTennis?



Each internal node: test one discrete-valued attribute X_i

Each branch from a node: selects one value for X_i

Each leaf node: predict Y

Continuous Valued Attributes

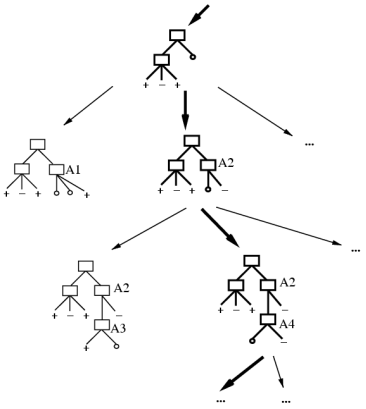
Create a discrete attribute to test continuous

- $Temperature = 82.5$
- $(Temperature > 72.3) = t, f$

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

Which Tree Should We Output?

- ID3 performs heuristic search through space of decision trees
- It stops at smallest acceptable tree. Why?



Occam's razor: prefer the simplest hypothesis that fits the data

Why Prefer Short Hypotheses? (Occam's Razor)

Arguments in favor:

Arguments opposed:

Why Prefer Short Hypotheses? (Occam's Razor)

Argument in favor:

- Fewer short hypotheses than long ones

→ a short hypothesis that fits the data is less likely to be a statistical coincidence

→ highly probable that a sufficiently complex hypothesis will fit the data

Argument opposed:

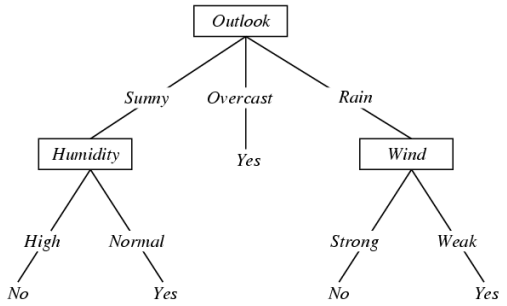
- Also fewer hypotheses with prime number of nodes and attributes beginning with "Z"
- What's so special about "short" hypotheses?

Overfitting in Decision Trees

Consider adding noisy training example #15:

$Sunny,\ Hot,\ Normal,\ Strong,\ PlayTennis = No$

What effect on earlier tree?



Overfitting

Consider a hypothesis h and its

- Error rate over training data: $error_{train}(h)$
- True error rate over all data: $error_{true}(h)$

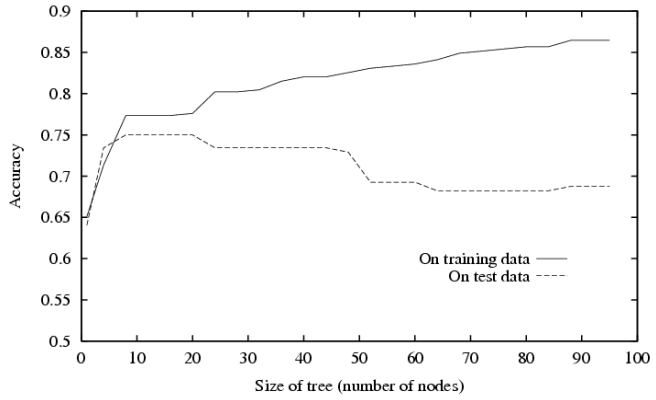
We say h overfits the training data if

$$error_{true}(h) > error_{train}(h)$$

Amount of overfitting =

$$error_{true}(h) - error_{train}(h)$$

Overfitting in Decision Tree Learning



Avoiding Overfitting

How can we avoid overfitting?

- stop growing when data split not statistically significant
- grow full tree, then post-prune

Avoiding Overfitting

How can we avoid overfitting?

- stop growing when data split not statistically significant
- grow full tree, then post-prune

How to select "best" tree:

- Measure performance over training data
- Measure performance over separate validation data set
- MDL: minimize
  $size(tree) + size(misclassifications(tree))$

Reduced-Error Pruning

Split data into *training* and *validation* set

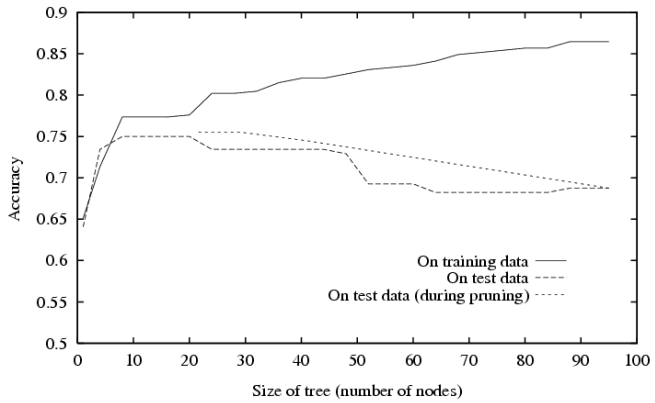
Create tree that classifies *training* set correctly

Do until further pruning is harmful:

1. Evaluate impact on *validation* set of pruning each possible node (plus those below it)
2. Greedily remove the one that most improves *validation* set accuracy

- produces smallest version of most accurate subtree
- What if data is limited?

Effect of Reduced-Error Pruning



Random Forests

Key idea:

1. learn a collection of many trees
2. classify by taking a weighted vote of the trees

Empirically successful. Widely used in industry.

- human pose recognition in Microsoft kinect
- medical imaging – cortical parcellation
- classify disease from gene expression data

How to train different trees

1. Train on different random subsets of data
2. Randomize the choice of decision nodes

Random Forests

Key idea:

1. learn a collection of many trees
2. classify by taking a weighted vote of the trees

more to come

later lecture on boosting

Emp

- hu
- m
- classify disease from gene expression data

How to train different trees

- Train on different random subsets of data
- Randomize the choice of decision nodes

Questions to think about (1)

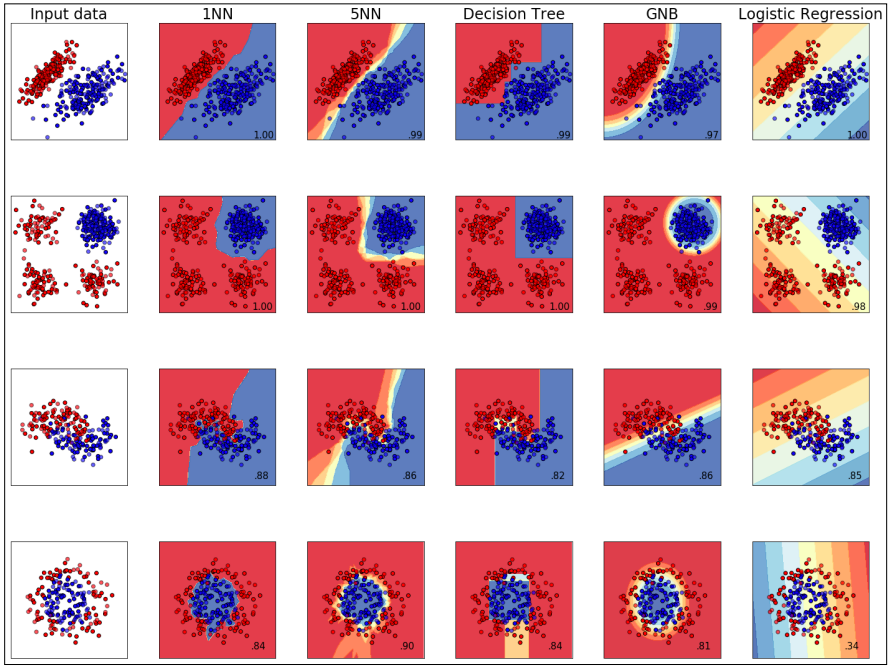
- Consider target function f: (x1,x2) → y, where x1 and x2 are real-valued, y is boolean. What is the set of decision surfaces describable with decision trees that use each attribute at most once?

Questions to think about (2)

- ID3 and C4.5 are heuristic algorithms that search through the space of decision trees. Why not just do an exhaustive search?

Questions to think about (3)

- Why use Information Gain to select attributes in decision trees? What other criteria seem reasonable, and what are the tradeoffs in making this choice?


Input data
1NN
5NN
Decision Tree
GNB
Logistic Regression
1.00
.99
.99
.97
1.00
1.00
1.00
1.00
.99
.98
.88
.86
.82
.86
.85
.84
.90
.84
.81
.34