BAYESIAN LEARNING

Bayesian Learning

- A powerful approach in machine learning
- Combine data seen so far with prior beliefs
  - This is what has allowed us to do machine learning, have good inductive biases, overcome "No free lunch", and obtain good generalization on novel data
- We use it in our own decision making all the time
  - You hear a word which which could equally be “Thanks” or “Hanks”, which would you go with?
    - Combine sound likelihood and your prior knowledge
  - Texting Suggestions on phone
  - Spell checkers, speech recognition, etc.
  - Many applications

Bayesian Classification

- $P(c|x)$ - Posterior probability of output class c given the input vector x
- The discriminative learning algorithms we have learned so far try to "approximate" this directly
- Bayes Rule – A true probability: $P(c|x) = P(x|c)P(c) / P(x)$
- $P(c)$ - Prior probability of class c – How do we know?
  - Just count up and get the probability for the Training Set – Easy!
- $P(x|c)$ - Probability "likelihood" of data vector x given that the output class is c
  - We will discuss ways to calculate this *likelihood*
- $P(x)$ - Prior probability of the data vector x
  - This is a normalizing term to get an actual probability. In practice we drop it because it is the same for each class c (i.e. independent), and we are just interested in which class c maximizes $P(c|x)$.

Bayesian Intuition

- Bayesian vs. Frequentist
- Bayesian allows us to talk about probabilities/beliefs even when there is little data, because we can use the prior
  - What is the probability of a nuclear plant meltdown?
  - What is the probability that BYU will win the national championship?
- As the amount of data increases, Bayes shifts confidence from the prior to the likelihood
- Requires reasonable priors in order to be helpful
- We use priors all the time in our decision making
  - Unknown coin: probability of heads?

Bayesian Learning of ML Models

- Assume H is the hypothesis space, h a specific hypothesis from H, and D is all the training data
- $P(h|D)$ - Posterior probability of h, this is what we usually want to know in a learning algorithm – (i.e. model selection)
- $P(h|D) = P(D|h)P(h)/P(D)$ Bayes Rule
- $P(h)$ - Prior probability of the hypothesis/model independent of D - do we usually know?
  - Could assign equal probabilities
  - *Could assign probability based on inductive bias* (e.g. simple hypotheses have higher probability) – Thus regularization already in the equation!
- $P(D|h)$ - Probability "likelihood" of data given the hypothesis
- $P(D)$ - Prior probability of the data
- $P(h|D)$ increases with $P(D|h)$ *and* $P(h)$. In learning when seeking to discover the best h given a particular D, $P(D)$ is the same and can be dropped.

Naïve Bayes
Revisit Bayesian Classification

- $P(c|x) = P(x|c)P(c)/P(x)$

- $P(c)$ - Prior probability of class c – How do we know?
  - Just count up and get the probability for the Training Set – Easy!

- $P(x|c)$ - Probability "likelihood" of data vector x given that the output class is c
  - We use $P(x1,\ldots,xn \mid cj)$ as short for $P(x1 = val1,\ldots,xn = valn \mid cj)$
  - How do we really do this?
  - If x is real valued?
  - If x is nominal we can just look at the training set and count to see the probability of x given the output class c but how often will all x's be the same?
    - Which will also be the problem if we bin real valued inputs

Naïve Bayes Classifier

- Note we are not considering $h \in H$, rather just collecting statistics from the data set
- Given a training set, $P(c_j)$ is easy to calculate
- How about $P(x_1, \ldots, x_n|c_j)$? Most cases would be either 0 or 1
- Key "Naïve" leap: Assume conditional independence of the attributes

$$P(x1, \ldots, xn \mid cj) = \prod_i P(x_i \mid c_j)$$

$$cNB = \underset{cj \in C}{\operatorname{argmax}} P(c_j) \prod_i P(x_i \mid c_j)$$

  - P(Thin, Red, Meat | Good) = P(Thin | Good) * P(Red | Good) * P(Meat | Good)
  - There is usually sufficient data to get accurate values for independent terms

Naïve Bayes Classifier

- While conditional independence is not typically a reasonable assumption… (heart rate and blood pressure)
  - Low complexity simple approach
  - Need only store all $P(c_j)$ and $P(x_i|c_j)$ terms
  - Assume nominal features for the moment
  - Easy to calculate the |*attribute values*| × |*classes*| terms
  - There is often enough data to make the independent terms reasonably accurate
  - Effective and common for many large applications (Document classification, etc.)

Naïve Bayes (cont.)

- Can normalize to get the actual naïve Bayes probability

- Continuous data? - Can discretize a continuous feature into bins, thus changing it into a nominal feature and then gather statistics normally
  - How many bins? - More bins is good, but need sufficient data to make statistically significant bins. Thus, base it on data available
  - Could also assume data is Gaussian and compute the mean and variance for each feature given the output class, then each $P(x_i|c_j)$ becomes $\mathcal{N}(x_i|\mu_{xi|cj}, \sigma^2_{xi|cj})$
    - Not good if data is multi-modal

Naïve Bayes (cont.)

- NB uses just 1st order features - assumes conditional independence
  - calculate statistics for all $P(x_i|c_j)$
  - $|attributes| \times |attribute\ values| \times |output\ classes|$
- nth order - $P(x_i, \ldots, x_n|c_j)$ - assumes full conditional dependence
  - $|attributes|^n \times |attribute\ values| \times |output\ classes|$
  - Too computationally expensive - exponential
  - Not enough data to get reasonable statistics - most cases occur 0 or 1 time
- 2nd order? - compromise - $P(x_i x_k|c_j)$ - assume only low order dependencies
  - $|attributes|^2 \times |attribute\ values| \times |output\ classes|$
  - More likely to have cases where number of $x_i x_k|c_j$ occurrences are 0 or few, could just use the higher order features which occur often in the data
  - 3rd order, etc.
- How might you test if a problem is conditionally independent?
  - Could just compare against 2nd or higher order. How far off on average is our assumption

$$P(x_i x_k|c_j) = P(x_i|c_j)\ P(x_k|c_j)?$$

Naïve Bayes (cont.)

- No training - Just gather the statistics from the data set and then apply the Naïve Bayes classification equation to any new instance
- Easier to have many attributes since not building a net, etc. and the amount of statistics gathered grows linearly with the number of attribute values (# attribute values × # *classes*) - Thus natural for applications like text classification which can be represented with huge numbers of input attributes.
- Though Naïve Bayes is limited by the first order assumptions, it is still often used and gives reasonable results in many large real-world applications

Naïve Bayes Example

Size (L, S)	Color (R,G,B)	Output (P,N)
L	R	N
S	B	P
S	G	N
L	R	N
L	G	P

For the given training set:

1. Create a table of the statistics needed to do Naïve Bayes
2. What would be the best output for a new instance which is Large and Blue? (e.g. the class which wins the argmax)
3. What is the true probability for each output class (P or N) for Large and Blue?

$$cNB = \underset{cj \in C}{\operatorname{argmax}} P(c_j) \prod_i P(x_i \mid c_j)$$

Size (L, S)	Color (R,G,B)	Output (P,N)
L	R	N
S	B	P
S	G	N
L	R	N
L	G	P

$$cNB = \underset{cj \in C}{\operatorname{argmax}} P(c_j) \prod_i P(x_i \mid c_j)$$

P(P)	
P(N)	
P(Size=L|P)	
P(Size=S|P)	
P(Size=L|N)	
P(Size=S|N)	
P(Color=R|P)	
P(Color=G|P)	
P(Color=B|P)	
P(Color=R|N)	
P(Color=G|N)	
P(Color=B|N)	

$P(c_j)$

$P(x_i \mid c_j)$

Challenge Question

Finish and give the true Probabilities of P and N for Size = L and Color = B

Size (L, S)	Color (R,G,B)	Output (P,N)
L	R	N
S	B	P
S	G	N
L	R	N
L	G	P

$$cNB = \underset{cj \in C}{\operatorname{argmax}} P(c_j) \prod_i P(x_i \mid c_j)$$

P(P)	2/5
P(N)	3/5
P(Size=L|P)	1/2
P(Size=S|P)	1/2
P(Size=L|N)	2/3
P(Size=S|N)	
P(Color=R|P)	
P(Color=G|P)	
P(Color=B|P)	
P(Color=R|N)	
P(Color=G|N)	
P(Color=B|N)	

$P(c_j)$

$P(x_i \mid c_j)$

P(P) = 2/5 * 1/2 * P(Color = B|P) = ?   P(N) = ?

Challenge Question

Finish and give the true Probabilities of P and N for Size = L and Color = B

Size (L, S)	Color (R,G,B)	Output (P,N)
L	R	N
S	B	P
S	G	N
L	R	N
L	G	P

$$cNB = \underset{cj \in C}{\operatorname{argmax}} P(c_j) \prod_i P(x_i \mid c_j)$$

P(P)	2/5
P(N)	3/5
P(Size=L|P)	1/2
P(Size=S|P)	1/2
P(Size=L|N)	2/3
P(Size=S|N)	1/3
P(Color=R|P)	0/2
P(Color=G|P)	1/2
P(Color=B|P)	1/2
P(Color=R|N)	2/3
P(Color=G|N)	1/3
P(Color=B|N)	0/3

$P(c_j)$

$P(x_i \mid c_j)$

<u>True Probabilities</u>

P(P) = 1/(1+0) = 1

P(N) = 0/(1+0) = 0

P(P) = 2/5 * 1/2 * 1/2 = 1/10

P(N) = 3/5 * 2/3 * 0/3 = 0

Naïve Bayes Homework

Size (L, S)	Color (R,G,B)	Output (P,N)
L	R	P
S	B	P
S	B	N
L	R	N
L	B	P
L	G	N
S	B	P

For the given training set:

1. Create a table of the statistics needed to do Naïve Bayes
2. What would be the best output for a new instance which is Small and Blue? (e.g. the class which wins the argmax)
3. What is the true probability for each output class (P or N) for Small and Blue?

$$cNB = \underset{cj \in C}{\operatorname{argmax}} P(c_j) \prod_i P(x_i \mid c_j)$$