NAIVE BAYES

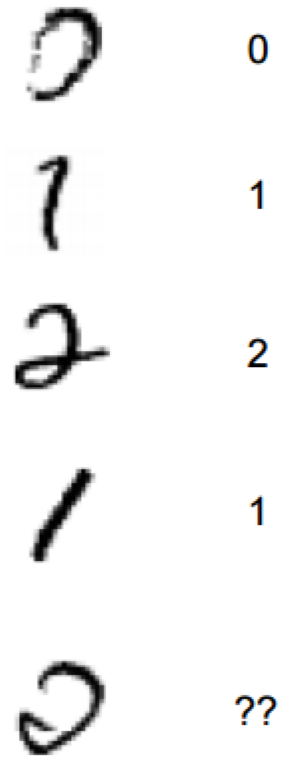
Slide adapted from *learning from data* book and course, and Berkeley cs188 by Dan Klein, and Pieter Abbeel

Machine Learning Recap

- Learning from data
- Tasks:
  - Prediction
  - Classification
  - Recognition
- Focus on Supervised Learning only
- Classification: Naïve Bayes

Example: Digit Recognition

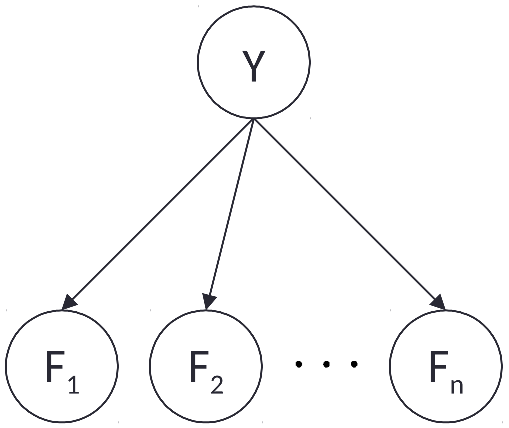
- Input: images/ pixel grids
- Output: a digit 0-9
- Setup:
  - Get a large collection of example images, each label with a digit
  - Note: someone has to hand label all this data
  - Want to learn to predict labels of new, future digit images



Model-Based Classification

- Model-Based approach
  - Build a model (e.g. Bayes' net) where both the label and features are random variables
  - Instantiate any observed features
  - Query for the distribution of the label conditioned on the features
- Challenges (solution components)
  - How to answer the query
  - How should we learn its parameters?
  - What structure should the BN have?

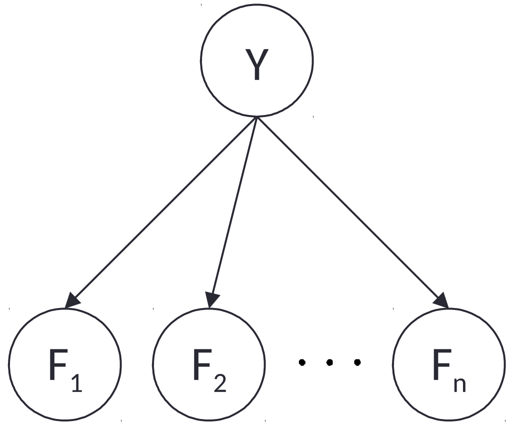
Naïve Bayes for Digits



- Naïve Bayes: Assume all features are independent effects of the label
- In other word: features are conditional independent given the class/label
- Simple digit recognition version:
  - One feature (variable) F_{ij} for each grid position <i,j>
  - Feature vales are on/off, based on whether intensity is more or less than 0.5 in underlying image
  - Each input maps to feature vector, e.g.
  -  › $< F_{0,0} = 0, F_{0,1} = 0, \ldots, F_{15,15} = 0>$
- Naïve Bayes model: $P(Y|F_{0,0} \ldots F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)$

General Naïve Bayes

- A general Naïve Bayes Model:



- |Y| parameters

$$P(Y, F_1 \ldots F_n) = P(Y) \prod_i P(F_i | Y)$$

|Y| x $|F|^n$ values

|Y| x $|F|^n$ values

- We only have to specify how each feature depends on the class
- Total number of parameters is linear in n
- Model is very simplistic, but often work anyway.

Inference for Naïve Bayes

- Goal: compute posterior distribution over label variable Y
  - Step 1: get joint probability of label and evidence for each label

$$P(Y, f_1 \ldots f_n) = \begin{bmatrix} P(y_1, f_1 \ldots f_n) \\ P(y_2, f_1 \ldots f_n) \\ \vdots \\ P(y_k, f_1 \ldots f_n) \end{bmatrix} \Rightarrow \frac{\begin{bmatrix} P(y_1) \prod_i P(f_i|y_1) \\ P(y_2) \prod_i P(f_i|y_2) \\ \vdots \\ P(y_k) \prod_i P(f_i|y_k) \end{bmatrix}}{P(f_1 \ldots f_n)}$$

$+$

$$\Downarrow$$

$$P(Y|f_1 \ldots f_n)$$

  - Step 2: sum to get probability of evidence
  - Step 3: normalize by dividing Step 1 by Step 2

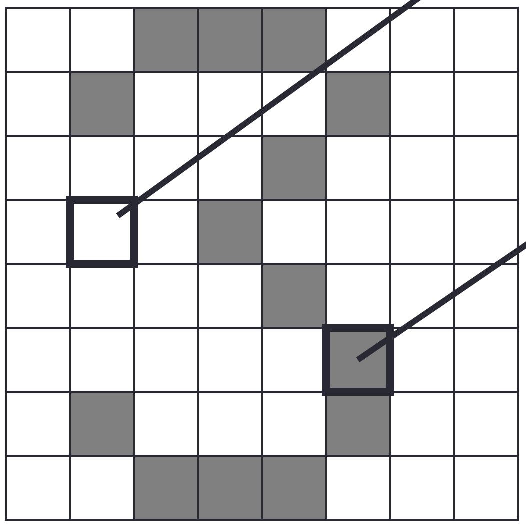
General Naïve Bayes

- What do we need in order to use Naïve Bayes?
  - Inference method (we just saw this part)
    - Start with a bunch of probabilities: P(Y) and the P(F_i|Y) tables
    - Use standard inference to compute P(Y|F_1…F_n)
    - Nothing new here
  - Estimates of local conditional probability tables
    - P(Y), the prior over labels
    - P(F_i|Y) for each feature (evidence variable)
    - These probabilities are collectively called the *parameters* of the model and denoted by ⚠
    - Up until now, we assumed these appeared by magic, but…
    - …they typically come from training data counts

Example: Conditional Probabilities

$P(Y)$

1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
0	0.1



$P(F_{3,1} = on|Y)$

1	0.01
2	0.05
3	0.05
4	0.30
5	0.80
6	0.90
7	0.05
8	0.60
9	0.50
0	0.80

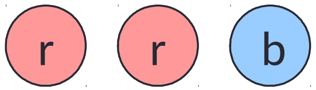
$P(F_{5,5} = on|Y)$

1	0.05
2	0.01
3	0.90
4	0.80
5	0.90
6	0.90
7	0.25
8	0.85
9	0.60
0	0.80

Parameter Estimation

- Estimating the distribution of a random variable (CPTs)
- Elicitation: ask a human (why is this hard?)
- Empirically: use training data (learning!)
  - E.g.: for each outcome x, look at the empirical rate of that value:

$$P_{\mathsf{ML}}(x) = \frac{\mathsf{count}(x)}{\mathsf{total\ samples}}$$



$$P_{\mathsf{ML}}(\mathsf{r}) = 2/3$$

  - This is the estimate that maximizes the likelihood of the data

$$L(x, \theta) = \prod_i P_\theta(x_i)$$

  - Relative frequencies are the maximum likelihood estimate

Unseen Events and Laplace Smoothing

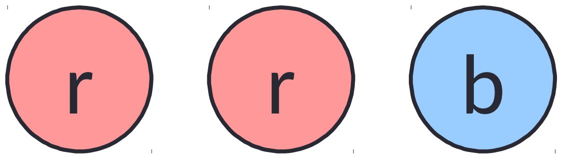
- What happen if you've never seen an event or feature for a given class?
- Laplace's estimate:
  - Pretend you saw every outcome once more than you actually did

$$P_{LAP}(x) = \frac{c(x)+1}{\sum_x [c(x)+1]}$$

$$= \frac{c(x)+1}{N+|X|}$$

|X| = #class

$$P_{LAP,k}(x) = \frac{c(x)+k}{N+k|X|}$$



$$P_{ML}(X) =$$

$$P_{LAP}(X) =$$

Summary

- Bayes rule lets us do diagnostic queries with causal probabilities
- The naïve Bayes assumption takes all features to be independent given the class label
- We can build classifiers out of a naïve Bayes model using training data
- Smoothing estimates is important in real systems