DS 4400

Machine Learning and Data Mining I
Spring 2024

David Liu

Khoury College of Computer Science

Northeastern University

February 27 2024

Announcements

- Final Project
  - Project proposals due this Friday on Gradescope
    - Each member should submit a copy.
  - Expect further details on final project report (6-8 pages) and video (~4 minutes) after spring break.
- Homework 3
  - Released on February 26, due on March 18
- Midterm grades will be released after Spring Break.

Where We Are

Regression	Simple, Multiple, Polynomial
Classification	Logistic Regression, kNN
Generative Models	LDA, Naïve Bayes
Ensemble Learning	Bagging, Boosting
Neural Networks	Multi-layer Perceptron, Convolutional Neural Networks

Tools

Optimization:
- Closed form solution
- Gradient descent
- Backpropogation

Preventing Overfitting:
- Regularization
- Cross Validation

Evaluation of ML:
- Precision, Recall, AUC

Outline

- Generative classifiers
- Naïve Bayes classifiers [focus for today]
  - Naïve Bayes assumption
  - Laplace smoothing
  - Comparison to LDA
- Decision trees [preview]
  - Information gain / entropy measures
  - Training algorithm

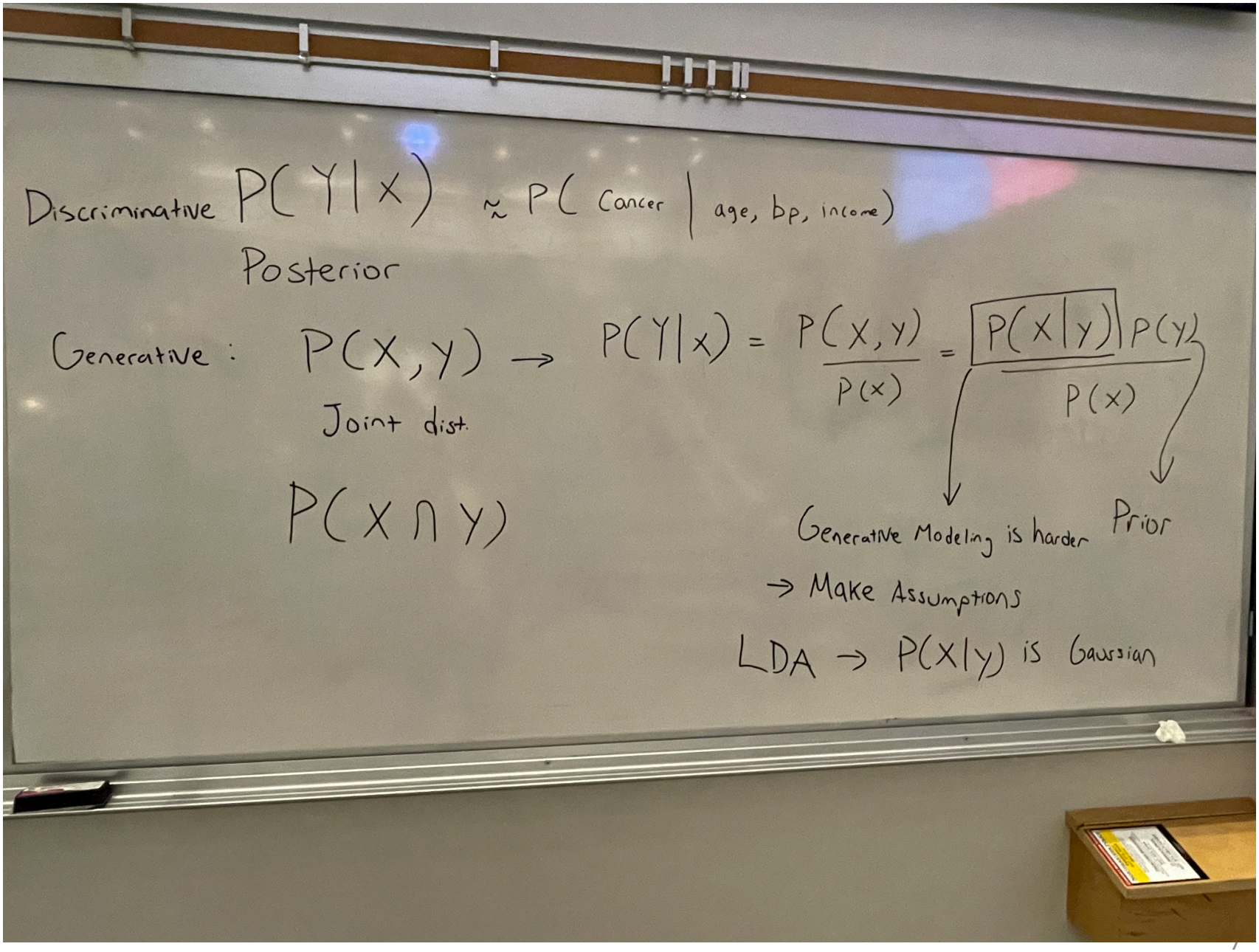
Recap: ML Concepts

- Supervised vs unsupervised learning
- Classification vs regression
- Linear vs non-linear classifiers
- Generative vs discriminative classifiers
- Loss functions
- Metrics for evaluation

Generative vs Discriminative

- Generative model
  - Given X and Y, learns the joint probability $P(X, Y)$
  - Can generate more examples from distribution
  - Examples: LDA, Naïve Bayes, language models (GPT-2, GPT-3, BERT)

- Discriminative model
  - Given X and Y, learns a decision function for classification


Discriminative P(Y|X) ≈ P(Cancer | age, bp, income)
Posterior
Generative: P(X,Y) → P(Y|X) = P(X,Y)/P(X) = P(X|Y)P(Y)/P(X)
Joint dist.
P(X ∩ Y)
Generative Modeling is harder
Prior
→ Make Assumptions
LDA → P(X|Y) is Gaussian

Generative classifiers based on Bayes Theorem

$$P(A \mid B) = \frac{P(B \mid A) \times P(A)}{P(B)}$$

- Exactly the process we just used
- The most important formula in probabilistic machine learning

(Super Easy) Derivation:

$$P(A \wedge B) = P(A \mid B) \times P(B)$$
$$P(B \wedge A) = P(B \mid A) \times P(A)$$

these are the same

Just set equal...

$$P(A \mid B) \times P(B) = P(B \mid A) \times P(A)$$

and solve...

Bayes, Thomas (1763) An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London,* **53:370-418**

LDA Training and Testing

Given training data $(x_i, y_i), i = 1, \ldots, n, y_i \in \{1, \ldots, K\}$

1. Estimate mean and variance

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

2. Estimate prior

$$\hat{\pi}_k = n_k/n.$$

Given testing point x, predict k that maximizes:

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

Naïve Bayes Classifier

Idea: Use the training data to estimate

$P(X \mid Y)$ and $P(Y)$.

Then, use Bayes rule to infer $P(Y|X_{\text{new}})$ for new data

---

Easy to estimate from data

Impractical, but necessary

$$\mathrm{P}[Y = k|X = x] = \frac{\mathrm{P}[Y = k]\mathrm{P}[X_1 = x_1 \wedge \cdots \wedge X_d = x_d|Y = k]}{\mathrm{P}[X_1 = x_1 \wedge \cdots \wedge X_d = x_d]}$$

Unnecessary, as it turns out

Learning Joint Distributions

Step 1:

Build a JD table for your attributes in which the probabilities are unspecified

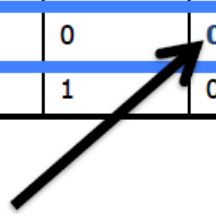
A	B	C	Prob
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

Step 2:

Then, fill in each row with:

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



Fraction of all records in which A and B are true but C is false

Example – Learning Joint Probability Distribution

This Joint PD was obtained by learning from three attributes in the UCI "Adult" Census Database [Kohavi 1995]

gender	hours_worked	wealth		
Female	v0:40.5-	poor	0.253122	
		rich	0.0245895	
	v1:40.5+	poor	0.0421768	
		rich	0.0116293	
Male	v0:40.5-	poor	0.331313	
		rich	0.0971295	
	v1:40.5+	poor	0.134106	
		rich	0.105933	

Naïve Bayes Classifier

Problem: estimating the joint density isn't practical

However, if we make the assumption that the attributes are independent given the class label, estimation is easy!

Naïve Bayes Classifier

Problem: estimating the joint PD or CPD isn't practical

- Severely overfits, as we saw before

However, if we make the assumption that the attributes are independent given the class label, estimation is easy!

$$\mathrm{P}[X_1 = x_1 \wedge \cdots \wedge X_d = x_d | Y = k] = \prod_{j=1}^{d} P\left[X_j = x_j \middle| Y = k\right]$$

- In other words, we assume all attributes are *conditionally independent* given Y
- Often this assumption is violated in practice, but more on that later…

Using the Naïve Bayes Classifier

- Now, we have

$$\mathrm{P}[Y = k|X = x] = \frac{\mathrm{P}[Y = k]\mathrm{P}[X_1 = x_1 \wedge \cdots \wedge X_d = x_d|Y = k]}{\mathrm{P}[X_1 = x_1 \wedge \cdots \wedge X_d = x_d]}$$

This is constant for a given instance, and so irrelevant to our prediction

Using the Naïve Bayes Classifier

- Now, we have

$$\mathrm{P}[Y = k|X = x] = \frac{\mathrm{P}[Y = k]\mathrm{P}[X_1 = x_1 \wedge \cdots \wedge X_d = x_d|Y = k]}{\mathrm{P}[X_1 = x_1 \wedge \cdots \wedge X_d = x_d]}$$

This is constant for a given instance, and so irrelevant to our prediction

  - In practice, we use log-probabilities to prevent underflow

- To classify a new point $\mathbf{x}$,

$$h(\mathbf{x}) = \arg\max_{y_k} \; P(Y = k \;) \prod_{j=1}^{d} P(X_j = x_j \mid Y = k \;)$$

x_j: j^{th} attribute value of $\mathbf{x}$

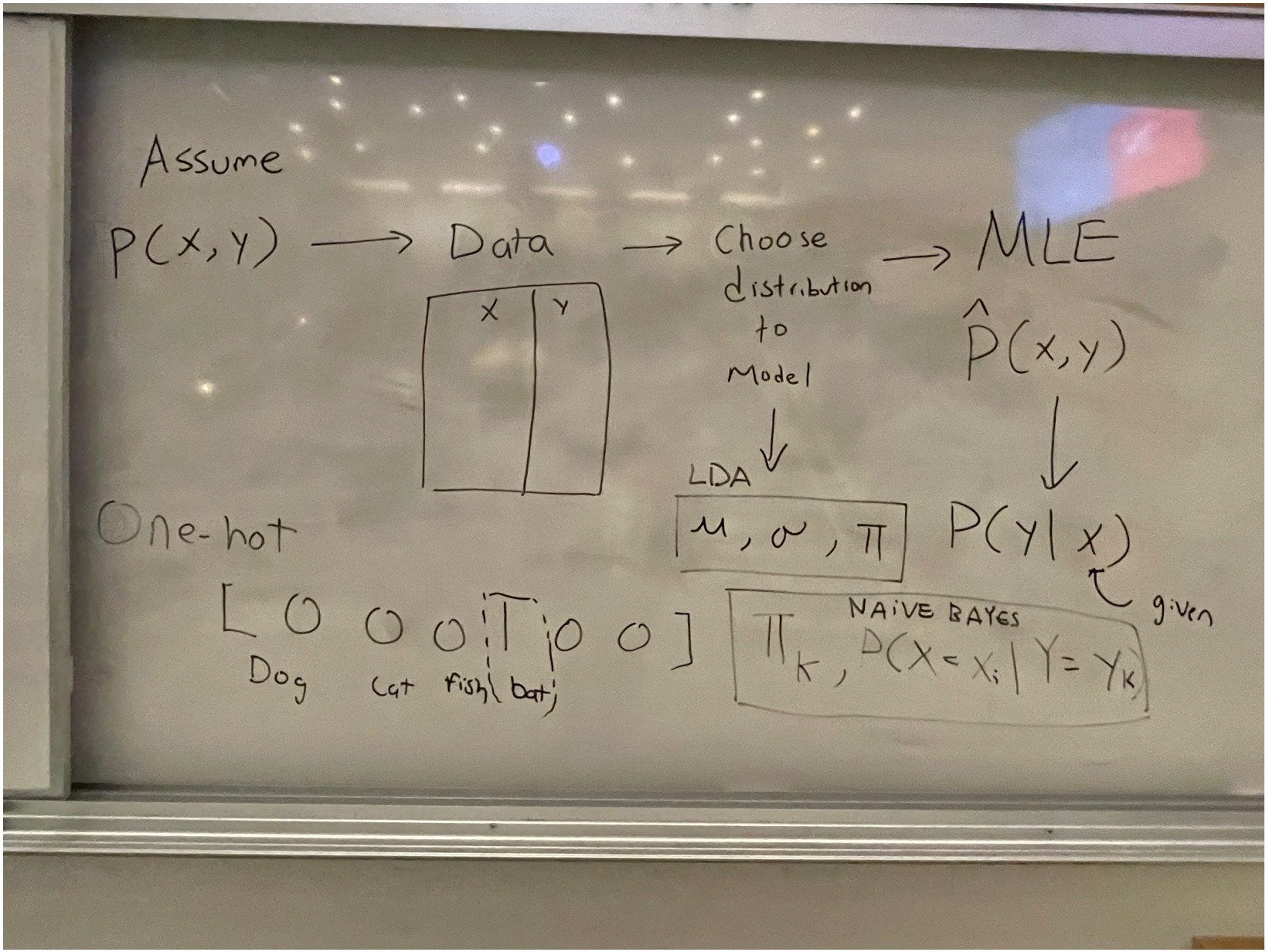
Naïve Bayes Classifier

TRAIN

- For each class label k
  1. Estimate prior $\pi_k = P[Y = k]$ from the data
  2. For each value v of attribute X_j
     - Estimate $\mathrm{P}[X_j = v | Y = k]$

TEST on INPUT $x = (x_1, \ldots, x_d)$

- For every k, compute the probabilities
  - $p_k = P[Y = k] \prod_{j=1}^{d} P[X_j = x_j \,|Y = k]$
- Classify x to the class k that maximizes p_k


Assume
P(x, y) → Data → Choose distribution to Model → MLE
X
Y
LDA
μ, σ, π
P̂(x, y)
P(y | x)
given
One-hot
[0 0 0 1 0 0]
Dog Cat fish bat
NAIVE BAYES
π_k, P(X = x_i | Y = y_k)

Training Naïve Bayes

Estimate $P[X_j = x_j | Y = k]$ and $P[Y = k]$ directly from the training data by counting!

Sky	Temp	Humid	Play?
sunny	warm	normal	*yes*
sunny	cold	high	*yes*
rainy	cold	high	*no*
sunny	warm	high	*yes*

Prior: P(Play = Yes) =

P(Play = No) =

Conditional feature distributions

P(Sky = sunny| Play = Yes)=

P(Temp = warm|Play = Yes) =

P(Humid = high|Play = Yes) =

P(Sky = sunny| Play = No)=

P(Temp = warm|Play = No) =

P(Humid = high|Play = No) =

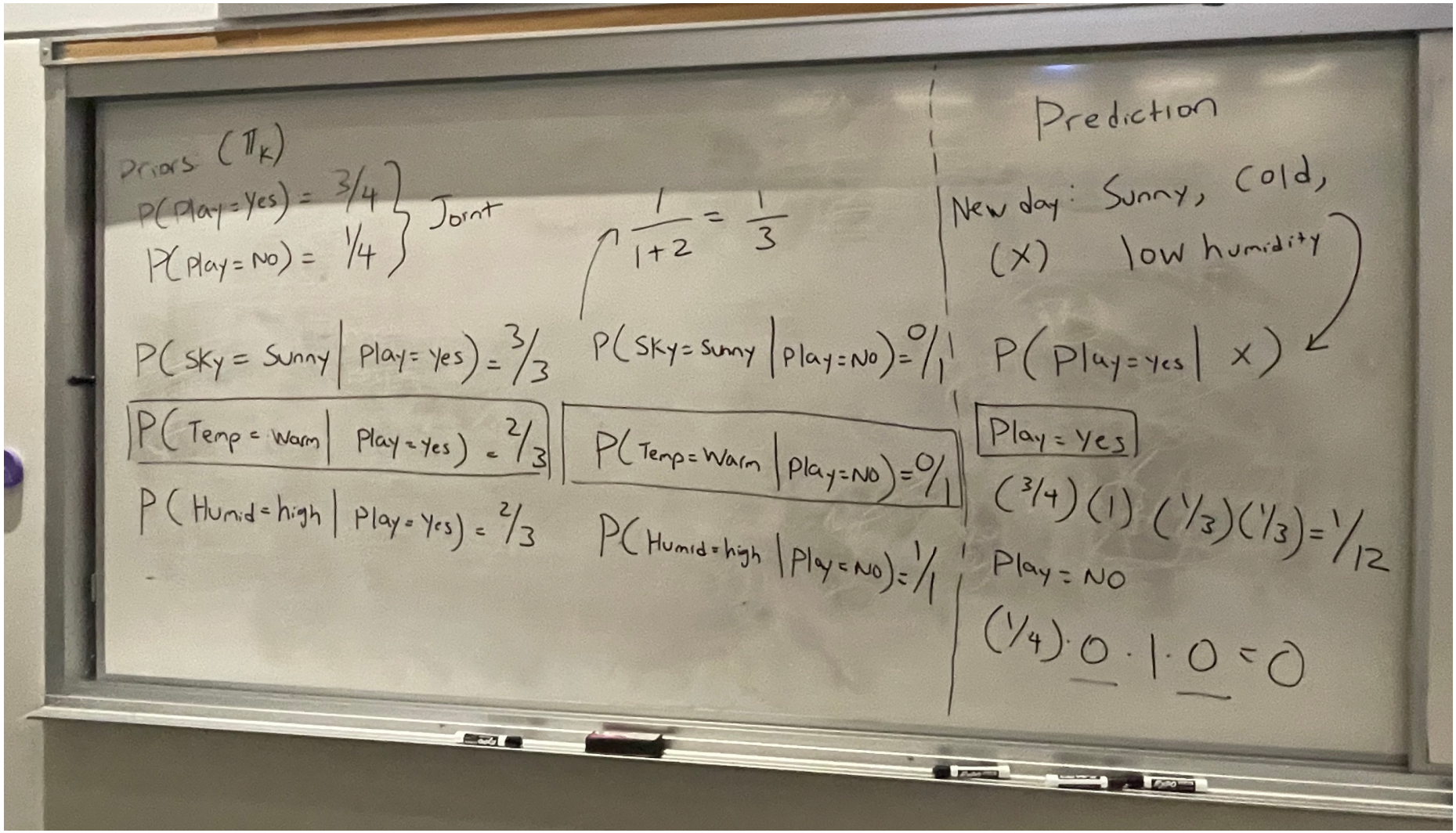
Training Naïve Bayes

Estimate $P[X_j = x_j | Y = k]$ and $P[Y = k]$ directly from the training data by counting!

Sky	Temp	Humid	Play?
sunny	warm	normal	*yes*
sunny	cold	high	*yes*
rainy	cold	high	*no*
sunny	warm	high	*yes*

Classify new point:

X: Sky = sunny, Temp = cold, Humid = high


Priors (Π_K)
P(Play=Yes) = 3/4
P(Play=No) = 1/4
Joint
P(Sky=Sunny | Play=Yes) = 3/3
P(Temp=Warm | Play=Yes) = 2/3
P(Humid=high | Play=Yes) = 2/3
1/(1+2) = 1/3
P(Sky=Sunny | Play=No) = 0/1
P(Temp=Warm | Play=No) = 0/1
P(Humid=high | Play=No) = 1/1
Prediction
New day: Sunny, Cold, (X) low humidity
P(Play=Yes | X)
Play=Yes
(3/4)(1)(1/3)(1/3) = 1/12
Play=No
(1/4)·0·1·0 = 0

Laplace Smoothing

- Notice that some probabilities estimated by counting might be zero
  - Possible overfitting!

Laplace Smoothing

- Notice that some probabilities estimated by counting might be zero
  - Possible overfitting!

- Fix by using Laplace smoothing:
  - Adds 1 to each count

$$P(X_j = v \mid Y = k\ ) = \frac{\mathrm{c}_v + 1}{\sum_{v' \in \mathrm{values}(X_j)} \mathrm{c}_{v'} + |\mathrm{values}(X_j)|}$$

where

- c_v is the count of training instances with a value of v for attribute j and class label k
- $|\mathrm{values}(X_j)|$ is the number of values X_j can take on

Naïve Bayes Classifier

TRAIN

- For each class label k
  1. Estimate prior $\pi_k = P[Y = k]$ from the data
  2. For each value v of attribute X_j
     - Estimate $\mathrm{P}\left[X_j = v \middle| Y = k\right]$ with Laplace smoothing

TEST on INPUT $x = (x_1, \ldots, x_d)$

- For every k, compute the probabilities
  - $P[Y = k] \prod_{j=1}^{d} P\left[X_j = x_j \middle| Y = k\right]$
- Classify x to the class k that maximizes the above product

Continuous Features

- Naïve Bayes can be extended to continuous features
- Gaussian Naïve Bayes
  - Here an additional assumption is that each distribution $P\left[X_j | Y = k\right]$ is Gaussian $N(\mu_j, \sigma_j)$
  - It estimates the mean and standard deviation from training data
- This leads to a linear classifier

Comparison to LDA

- Similarity to LDA
  - Both are generative models
  - They both estimate:

    $$P[X = x \text{ and } Y = k] = P[X = x|Y = k]P[Y = k]$$

    Using Bayes Theorem
- Difference from LDA
  - Naïve Bayes can handle discrete data
  - LDA uses multi-variate normal
  - LDA assumes same variances for all classes
  - Naïve Bayes make the conditional independence assumption
  - LDA is linear, while Naïve Bayes is usually not linear

Naïve Bayes Summary

Advantages:

- Fast to train (single scan through data)
- Fast to classify
- Not sensitive to irrelevant features
- Handles real and discrete data
- Handles streaming data well

Disadvantages:

- Assumes independence of features