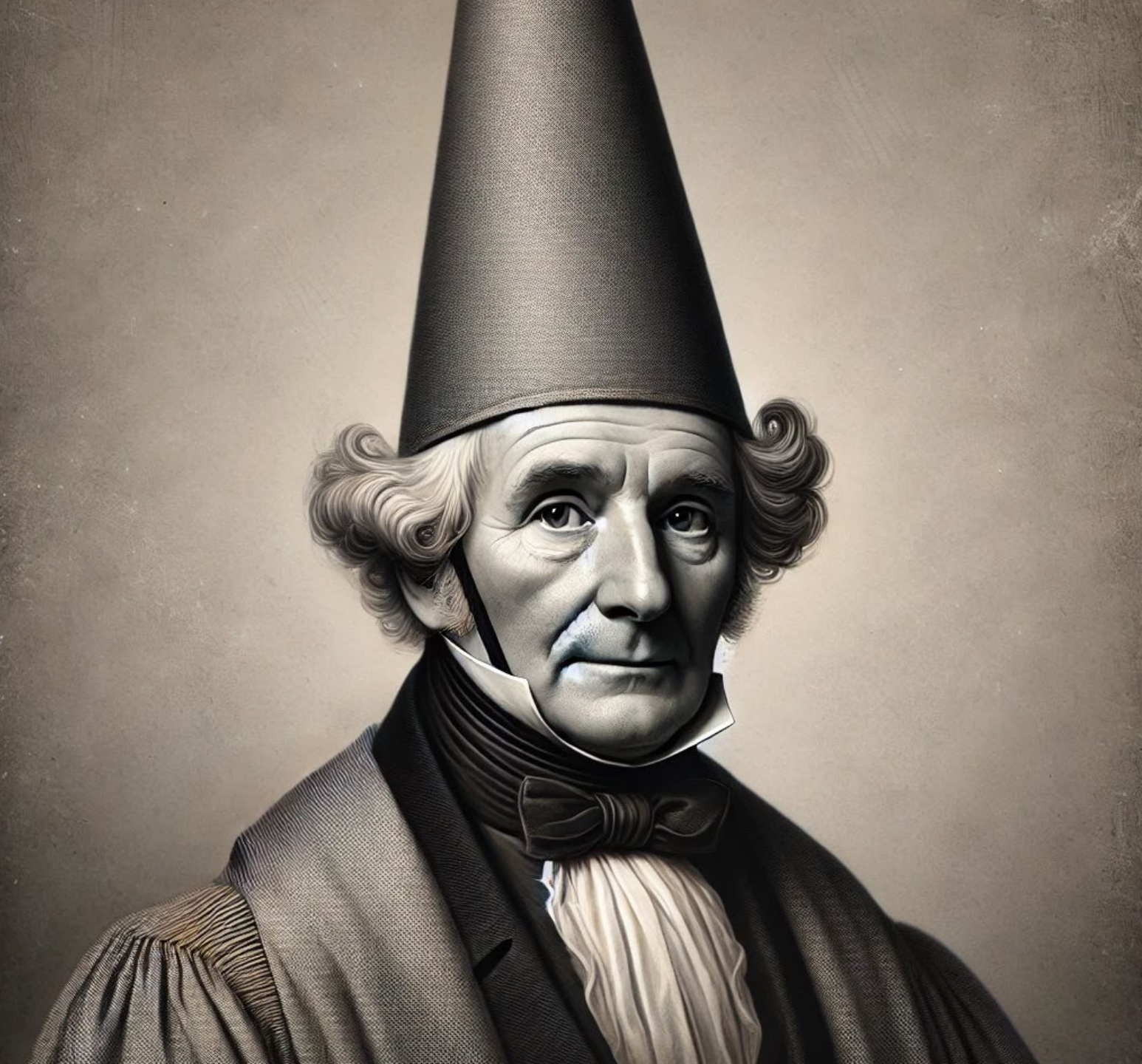




Naïve Bayes Classifier

Applied Machine Learning
Derek Hoiem

Dall-E: portrait of Thomas Bayes with a Dunce Cap
on his head

Approaches to classification and regression

- We've learned about two of the most widely used
 - Nearest neighbor
 - Linear models
- Next, we'll learn about probabilistic models and trees
- Everything else is representation learning

Today's Lecture

- Introduce probabilistic models
- Naïve Bayes Classifier
 - Assumptions / model
 - How to estimate from data
 - How to predict given new features
 - Cases of discrete and continuous variables
- “Semi-naïve Bayes” object detector

What is a probability

- A belief, a confidence, a likelihood
- “There’s a 60% chance it will rain tomorrow.”
 - Based on the information I have, if we were to simulate the future 100 times, I’d expect it to rain 60 of them.
 - I think it’s a little more likely to rain than not
- You have a $1/18$ chance of rolling a 3 with two dice.
 - If you roll an infinite number of pairs of dice, 1 out of 18 of them will sum to 3.
- Probabilities are expectations, according to some information and assumptions.
 - E.g., it will either rain tomorrow or not

Joint and conditional probability

$$P(x, y) = P(x|y)P(y) = P(y|x)P(x)$$

$$P(a, b, c) = P(a|b, c)P(b|c)P(c)$$

Bayes Rule:
$$P(x|y) = \frac{P(x, y)}{P(y)} = \frac{P(y|x)P(x)}{P(y)}$$

Law of total probability

$$\left[\sum_{v \in \mathcal{X}} P(x = v) \right] = 1$$

Marginalization

$$\left[\sum_{v \in \mathcal{X}} P(x = v, y) \right] = P(y)$$

For continuous variables, replace sum over possible values with integral over domain

Estimate probabilities of discrete variables by counting

$$P(x = v) = \frac{1}{|N|} \sum_n \delta(x_n = v)$$

Example

Rained

1

0

0

0

1

0

0

1

$$P(\text{rains}) = ?$$

Example

Rained

1

0

0

0

1

0

0

1

$$P(\text{rains}) = 3/8$$

Example, converting counts to probabilities

		x : Larger than 10 lbs?	
		F	T
y	Cat	15	25
	Dog	5	40

$$P(y = \textit{Cat}) =$$

$$P(y = \textit{Cat} | x = F) =$$

$$P(x = F | y = \textit{Cat}) =$$

Example

		x : Larger than 10 lbs?	
		F	T
y	Cat	15	25
	Dog	5	40

$$P(y = Cat) = 40 / 85$$

$$P(y = Cat|x = F) = 15/20$$

$$P(x = F|y = Cat) = 15/40$$

A is independent of B if (and only if)

$$P(A, B) = P(A)P(B)$$

$$P(A|B) = P(A), \quad P(B|A) = P(B)$$

What if you have 100 variables? How can you count all combinations?

Fully modeling dependencies between many variables (more than 3 or 4) is challenging and requires a lot of data

Probabilistic model

$$y^* = \operatorname{argmax}_y P(y|\mathbf{x})$$

Or equivalently...

$$y^* = \operatorname{argmax}_y P(\mathbf{x}|y)P(y)$$

$$\operatorname{argmax}_y P(y|\mathbf{x}) = \operatorname{argmax}_y P(y|\mathbf{x})P(\mathbf{x}) = \operatorname{argmax}_y P(y, \mathbf{x}) = \operatorname{argmax}_y P(\mathbf{x}|y)P(y)$$

Notation

- x_i is the i th feature variable
 - i indicates the feature index
- $\mathbf{x}_n$ is the n th feature vector
 - n indicates the sample index
- y_n is the n th label
- x_{ni} is the i th feature of the n th sample
- $\delta(x_{ni} = v)$ returns 1 if $x_{ni} = v$; 0 otherwise
 - v indicates a feature value
 - δ is an indicator function, mapping from true/false to 1/0

Suppose you want to classify whether a text message is spam

ham

Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got a...

ham

Ok lar... Joking wif u oni...

spam

Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entr...

ham

U dun say so early hor... U c already then say...

ham

Nah I don't think he goes to usf, he lives around here though

spam

FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it s...

Suppose you want to classify whether a text message is spam or not

$P(\text{"Ok lar... Joking wif u oni..."})$ probably is 0 in the training set because you might not get exactly the same message twice

How to model?

Naïve Bayes Model

Assume features $x_1 \dots x_m$ are independent given the label y :

$$P(\mathbf{x}|y) = \prod_i P(x_i|y)$$

Then

$$\begin{aligned} y^* &= \operatorname{argmax}_y \prod_i P(x_i|y)P(y) \\ &= \operatorname{argmax}_y [\log P(y) + \sum_i \log P(x_i|y)] \end{aligned}$$

Naïve Bayes Classifier

<https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset>

1. Estimate $P(y = \text{spam})$ and $P(y = \text{ham})$
2. Estimate $P(\text{word} = j \mid \text{spam})$ and $P(\text{word} = j \mid \text{ham})$

$$\begin{aligned} &P(\text{"Ok lar ... Joking wif u oni ..."}, \text{spam}) \\ &= P(\text{spam})P(\text{"Ok"} \mid \text{spam})P(\text{"lar"} \mid \text{spam}) \dots P(\text{"oni"} \mid \text{spam}) \end{aligned}$$

$$\begin{aligned} &P(\text{"Ok lar ... Joking wif u oni ..."}, \text{ham}) \\ &= P(\text{ham})P(\text{"Ok"} \mid \text{ham})P(\text{"lar"} \mid \text{ham}) \dots P(\text{"oni"} \mid \text{ham}) \end{aligned}$$

If $P(\text{message}, \text{spam}) > P(\text{message}, \text{ham})$, it's more likely spam.

Naïve Bayes Algorithm

- Training

1. Estimate parameters for $P(x_i|y)$ for each i
2. Estimate parameters for $P(y)$

- Prediction

1. Solve for y that maximizes $P(x, y)$

$$y^* = \operatorname{argmax}_y \prod_i P(x_i|y)P(y)$$

Naïve Bayes Algorithm

- Training
 1. Estimate parameters for $P(x_i|y)$ for each i
 2. Estimate parameters for $P(y)$
- Prediction
 1. Solve for y that maximizes $P(x, y)$

But generally, $P(x, y)$ will get vanishingly small if x contains many features, so instead compute $\log P(x, y)$

For spam detection, the spam score is
 $\log P(x, y = \text{spam}) - \log P(x, y = \text{ham})$

$$y^* = \operatorname{argmax}_y \prod_i P(x_i|y)P(y)$$

$$y^* = \operatorname{argmax}_y [\log P(y) + \sum_i \log P(x_i|y)]$$

How to estimate $P(x_i|y)$ from data?

1. MLE (maximum likelihood estimation): Choose the parameter that maximizes the likelihood of the data
2. MAP (maximum a priori): Choose the parameter that maximizes the data likelihood and its own prior

MLE (maximum likelihood estimation)

- MLE: Choose the parameter that maximizes the likelihood of the data

- Bernoulli (x is binary; y is discrete)

$$P(x_i | y = k) = \theta_{ki}^{x_i} (1 - \theta_{ki})^{1-x_i}$$

$$\theta_{ki} = \sum_n \delta(x_{ni}=1, y_n=k) / \sum_n \delta(y_n=k)$$

```
theta_ki[k,i] = np.sum((X[:,i]==1) & (y==k)) / np.sum(y==k)
```

- Categorical (x has multiple discrete values, y is discrete)

$$\theta_{kiv} = \sum_n \delta(x_{ni}=v, y_n=k) / \sum_n \delta(y_n=k)$$

```
theta_kiv[k,i,v] = np.sum((X[:,i]==v) & (y==k)) / (np.sum(y==k))
```

Priors and MAP (maximum a priori)

- MAP: Choose the parameter that maximizes the data likelihood and its own prior
- Priors on the likelihood parameters prevent a single feature from having zero or extremely low likelihood due to insufficient training data
 - As Warren Buffet says, it's not just about maximizing expected return – it's about making sure there are no zeros.
- Discrete: initialize counts with α (e.g. $\alpha = 1$)

$$P(x_i = v | y = k) = \frac{\alpha + \text{count}(x_i = v, y = k)}{\sum_v [\alpha + \text{count}(x_i = v, y = k)]}$$

```
theta_kiv[k,i,v] = (np.sum((X[:,i]==v) & (y==k))+alpha) / (np.sum(y==k)+alpha*num_v)
```

MLE and MAP estimates of binary variable likelihoods

- MLE (maximize data likelihood)

$$P(x = 1|y = 1) = \frac{\sum_n \delta(x_n = 1, y_n = 1)}{\sum_n \delta(x_n = 0, y_n = 1) + \sum_n \delta(x_n = 1, y_n = 1)}$$

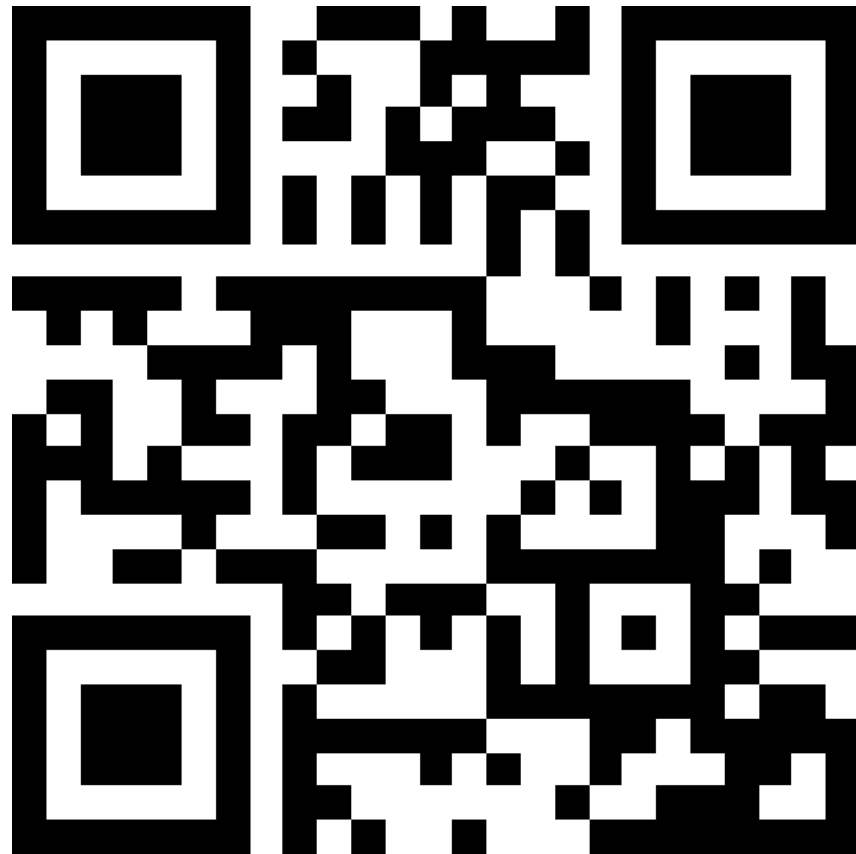
- MAP (maximum a posteriori) with prior α

$$P(x = 1|y = 1) = \frac{\alpha + \sum_n \delta(x_n = 1, y_n = 1)}{(\alpha + \sum_n \delta(x_n = 0, y_n = 1)) + (\alpha + \sum_n \delta(x_n = 1, y_n = 1))}$$

- This is a Bayesian prior that implies $P(x = 0|y) \approx P(x = 1|y)$, unless data tells us differently
- Similar concept to regularization that we saw in linear regression and classification
- Important because it avoids zeros that could dominate the overall likelihood and provides a more stable estimate with limited data
- With more data, the prior has less effect

Q1-Q3

<https://tinyurl.com/AML441-L8r>



What if x is continuous? Can we still use Naïve Bayes?

- E.g., estimate whether a person is male or female based on height and weight

$$P(m = 1|h, w) = \frac{P(m = 1, h, w)}{P(h, w)}$$

$$P(m = 1, h, w) = P(m = 1)P(h|m = 1)P(w|m = 1)$$

$$P(m = 0, h, w) = P(m = 0)P(h|m = 0)P(w|m = 0)$$

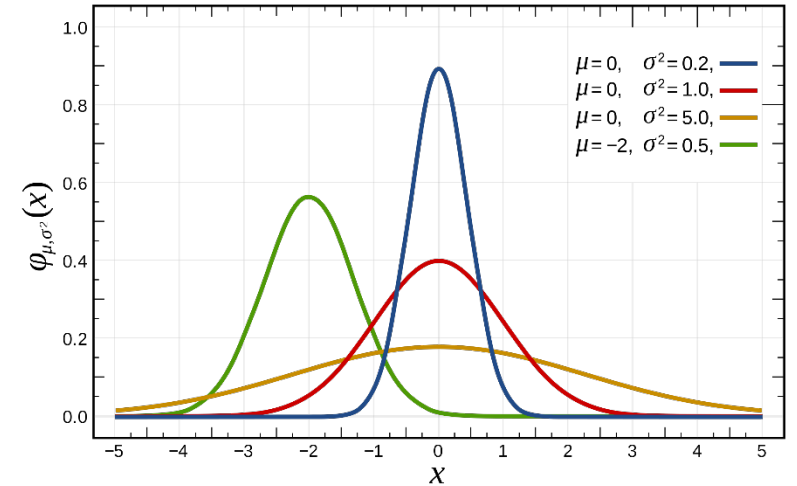
$$P(h, w) = P(m = 1, h, w) + P(m = 0, h, w)$$

Suppose x_i is Gaussian, and y is discrete

$$P(x_i | y=k) = \frac{1}{\sqrt{2\pi} \sigma_{ki}} \cdot \exp\left(-\frac{1}{2} \frac{(x_i - \mu_{ki})^2}{\sigma_{ki}^2}\right)$$

$$\mu_{ki} = \sum_n [X_{ni} \cdot \delta(y_n=k)] / \sum_n \delta(y_n=k)$$

$$\sigma_{ki}^2 = \sum_n [(X_{ni} - \mu_{ki})^2 \cdot \delta(y_n=k)] / \sum_n \delta(y_n=k)$$



```
mu[k,i] = np.mean(X[y==k,i], axis=0)
sigma[k, i] = np.std(X[y==k,i], axis=0)
```

Prior for Gaussian distributions

- Add some ϵ to the variance (e.g. $\epsilon = 0.1/N$)
 - For multivariate, add to diagonal of covariance

```
sigma[k, i] = np.std(X[y==k,i], axis=0) + np.sqrt(0.1/len(X))
```

Suppose we want to predict weight w given height h and sex m ?

$P(w, h, m) = P(h|w)P(m|w)P(w)$ according to naïve Bayes assumption

Can still model with a mix of 1D and 2D Gaussians and categorical distribution:

- $P(h|w) = P(h, w)/P(w)$ can be modeled with a 2D Gaussian and a 1D Gaussian
- $P(m = 1|w) = P(w|m = 1)P(m = 1)/P(w)$, which can each be modeled with 1D Gaussian or categorical
- $w^* = \operatorname{argmax}_w P(w, h, m)$ can be solved by $0 = \frac{\partial}{\partial w} P(w, h, m)$, which will have a closed form solution in this case

How to predict y from x when $(y - x_i)$ is Gaussian

If y is continuous,

$$\frac{\partial}{\partial y} \sum_i \log P(x_i | y) + \log P(y) = 0$$

General formulation (set partial derivative wrt y of $\log P(x, y)$ to 0)

$$\frac{\partial}{\partial y} \sum_i \frac{-\frac{1}{2}(y - x_i - \mu_i)^2}{\sigma_i^2} - \frac{1}{2} \frac{(y - \mu_y)^2}{\sigma_y^2} = 0$$

Example of Temperature regression:
 $y - x_i$ is Gaussian

$$\frac{\partial}{\partial y} \sum_i \frac{-\frac{1}{2}y^2}{\sigma_i^2} + \frac{yx_i}{\sigma_i^2} + \frac{y\mu_i}{\sigma_i^2} - \frac{1}{2} \frac{y^2}{\sigma_y^2} + \frac{y\mu_y}{\sigma_y^2} = 0$$

$$\sum_i \left(\frac{-y}{\sigma_i^2} + \frac{x_i}{\sigma_i^2} + \frac{\mu_i}{\sigma_i^2} \right) - (y - \mu_y) / \sigma_y^2 = 0$$

$$y \left(\sum_i \frac{1}{\sigma_i^2} + \frac{1}{\sigma_y^2} \right) = \sum_i \frac{x_i + \mu_i}{\sigma_i^2} + \frac{\mu_y}{\sigma_y^2}$$

$$y = \frac{1}{\sum_i \frac{1}{\sigma_i^2} + \frac{1}{\sigma_y^2}} \left[\sum_i \frac{x_i + \mu_i}{\sigma_i^2} + \frac{\mu_y}{\sigma_y^2} \right]$$

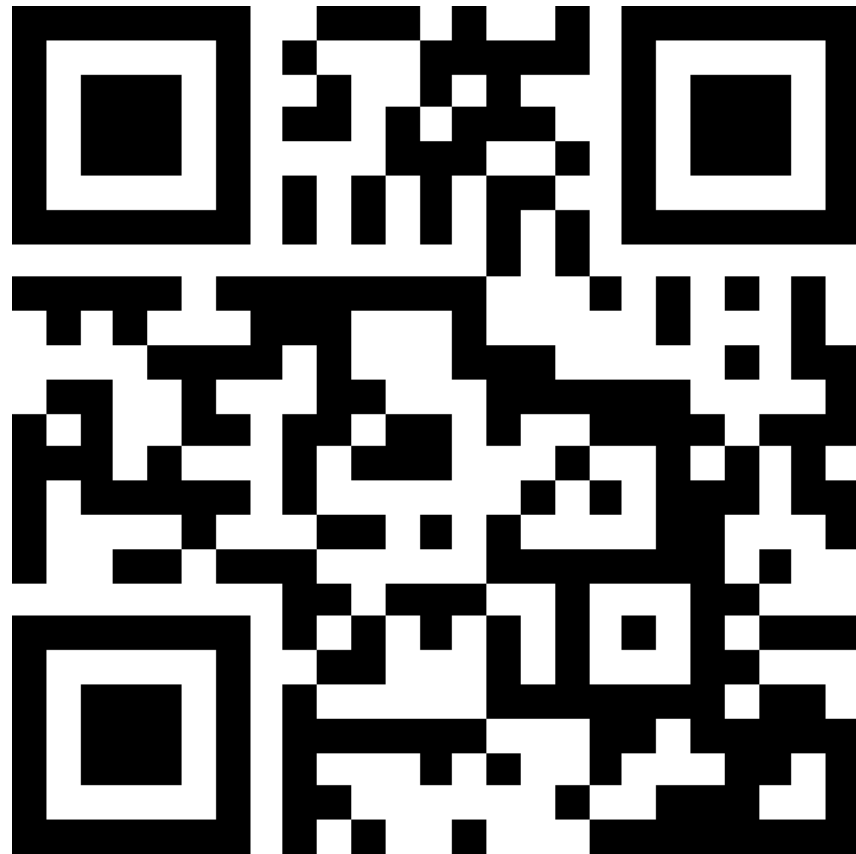
$$y = \frac{1}{\sum_i w_i + w_y} \left[\sum_i (x_i + \mu_i) w_i + \mu_y w_y \right]$$

Prediction is weighted average of means, where weights are inverse variance

$$P(x_i | y) \sim N(y - x_i, \sigma^2) = \frac{1}{\sqrt{2\pi} \sigma_i} \exp\left(-\frac{1}{2} \frac{(y - x_i - \mu_i)^2}{\sigma_i^2}\right)$$

Q4-Q6

<https://tinyurl.com/AML441-L8r>

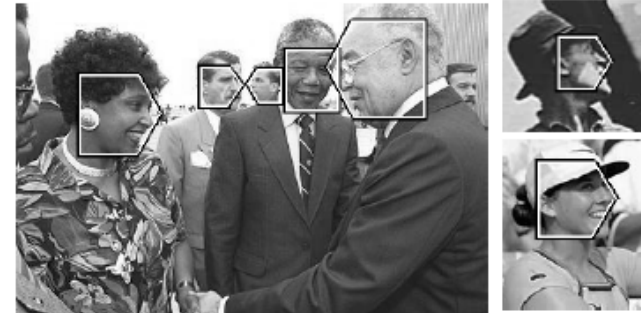


Use case: “Semi-naïve Bayes” object detection

A Statistical Method for 3D Object Detection Applied to Faces and Cars

Henry Schneiderman and Takeo Kanade

- Best performing face/car detector in 2000-2005
- Model probabilities of small groups of features (wavelet coefficients)
- Search for groupings, discretize features, estimate parameters



$$\frac{\prod_{x, y \in \text{region}} \prod_{k=1}^{17} P_k(\text{pattern}_k(x, y), x, y | \text{object})}{\prod_{x, y \in \text{region}} \prod_{k=1}^{17} P_k(\text{pattern}_k(x, y), x, y | \text{non-object})} > \lambda$$

Naïve Bayes

- Pros
 - Easy and fast to train
 - Fast inference
 - Can be used with continuous, discrete, or mixed features
- Cons
 - Does not account for feature interactions
 - Does not provide good confidence estimate
- Notes
 - Best when used with discrete variables, variables that are well fit by Gaussian, or kernel density estimation

Things to remember

- Probabilistic models are a large class of machine learning methods
- Naïve Bayes assumes that features are independent given the label
 - Easy/fast to estimate parameters
 - Less risk of overfitting when data is limited
- You can look up how to estimate parameters for most common probability models
 - Or take partial derivative of total data/label likelihood given parameter
- Prediction involves finding y that maximizes $P(x, y)$, either by trying all y or solving partial derivative
- Maximizing $\log P(x, y)$ is equivalent to maximizing $P(x, y)$ and often much easier

$$P(x, y) = \prod_i P(x_i | y) P(y)$$

$$\begin{aligned} y^* &= \underset{y}{\operatorname{argmax}} \prod_i P(x_i | y) P(y) \\ &= \underset{y}{\operatorname{argmax}} \sum_i \log P(x_i | y) + \log P(y) \end{aligned}$$

Next week

- EM and Density Estimation