Machine Learning Department
School of Computer Science
Carnegie Mellon University

MLE/MAP

+

Naïve Bayes

Matt Gormley
Lecture 17
Mar. 20, 2019

Reminders

- Homework 5: Neural Networks
  - Out: Fri, Mar 1
  - Due: Fri, Mar 22 at 11:59pm
- Homework 6: Learning Theory / Generative Models
  - Out: Fri, Mar 22
  - Due: Fri, Mar 29 at 11:59pm (1 week)

TIP: Do the readings!

- Today's In-Class Poll
  - http://p17.mlcourse.org

MLE AND MAP

Likelihood Function

One R.V.

- Suppose we have N **samples** $D = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ from a **random variable** X

- The **likelihood** function:
  - Case 1: X is **discrete** with *pmf* $p(x|\theta)$
    $$L(\theta) = p(x^{(1)}|\theta)\, p(x^{(2)}|\theta) \ldots\, p(x^{(N)}|\theta)$$
  - Case 2: X is **continuous** with *pdf* $f(x|\theta)$
    $$L(\theta) = f(x^{(1)}|\theta)\, f(x^{(2)}|\theta) \ldots\, f(x^{(N)}|\theta)$$

In both cases (discrete / continuous), the **likelihood** tells us how likely one sample is relative to another

- The **log-likelihood** function:
  - Case 1: X is **discrete** with *pmf* $p(x|\theta)$
    $$\ell(\theta) = \log p(x^{(1)}|\theta) + \ldots + \log p(x^{(N)}|\theta)$$
  - Case 2: X is **continuous** with *pdf* $f(x|\theta)$
    $$\ell(\theta) = \log f(x^{(1)}|\theta) + \ldots + \log f(x^{(N)}|\theta)$$

Likelihood Function

Two R.V.s

- Suppose we have N **samples** $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$ from a pair of **random variables** X, Y

- The **conditional likelihood** function:
  - Case 1: Y is **discrete** with *pmf* $p(y \mid x, \theta)$
    $L(\theta) = p(y^{(1)} \mid x^{(1)}, \theta) \ldots p(y^{(N)} \mid x^{(N)}, \theta)$
  - Case 2: Y is **continuous** with *pdf* $f(y \mid x, \theta)$
    $L(\theta) = f(y^{(1)} \mid x^{(1)}, \theta) \ldots f(y^{(N)} \mid x^{(N)}, \theta)$

- The **joint likelihood** function:
  - Case 1: X and Y are **discrete** with *pmf* $p(x,y|\theta)$
    $L(\theta) = p(x^{(1)}, y^{(1)}|\theta) \ldots p(x^{(N)}, y^{(N)}|\theta)$
  - Case 2: X and Y are **continuous** with *pdf* $f(x,y|\theta)$
    $L(\theta) = f(x^{(1)}, y^{(1)}|\theta) \ldots f(x^{(N)}, y^{(N)}|\theta)$

Likelihood Function

Two R.V.s

- Suppose we have N **samples** $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$ from a pair of **random variables** X, Y

- The **joint likelihood** function:
  - Case 1: X and Y are **discrete** with *pmf* $p(x,y|\theta)$
    $L(\theta) = p(x^{(1)}, y^{(1)}|\theta) \ldots p(x^{(N)}, y^{(N)}|\theta)$
  - Case 2: X and Y are **continuous** with *pdf* $f(x,y|\theta)$
    $L(\theta) = f(x^{(1)}, y^{(1)}|\theta) \ldots f(x^{(N)}, y^{(N)}|\theta)$
  - Case 3: Y is **discrete** with *pmf* $p(y|\beta)$ and
    X is **continuous** with *pdf* $f(x|y,\alpha)$
    $L(\alpha, \beta) = f(x^{(1)}| y^{(1)}, \alpha) p(y^{(1)}|\beta) \ldots f(x^{(N)}| y^{(N)}, \alpha) p(y^{(N)}|\beta)$
  - Case 4: Y is **continuous** with *pdf* $f(y|\beta)$ and
    X is **discrete** with *pmf* $p(x|y,\alpha)$
    $L(\alpha, \beta) = p(x^{(1)}| y^{(1)}, \alpha) f(y^{(1)}|\beta) \ldots p(x^{(N)}| y^{(N)}, \alpha) f(y^{(N)}|\beta)$

Mixed discrete/continuous!

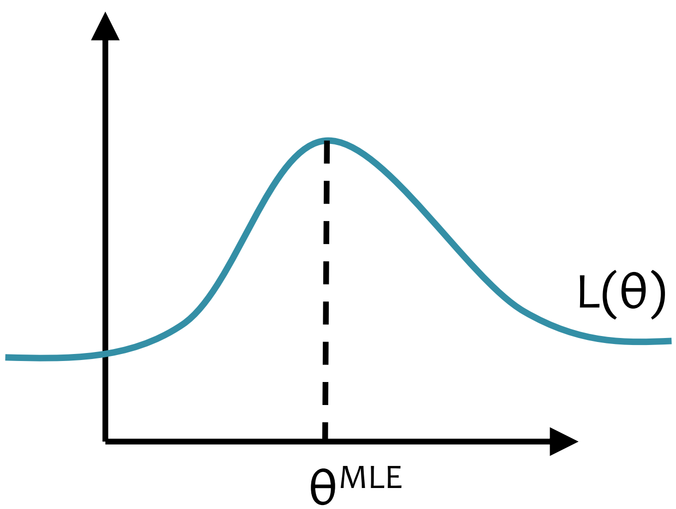
MLE

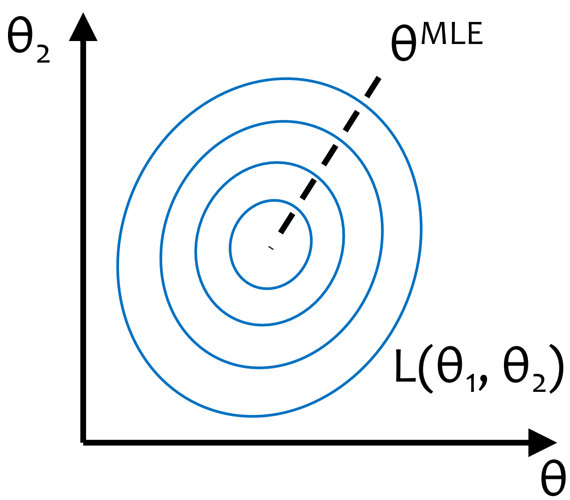
Suppose we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^N$

Principle of Maximum Likelihood Estimation:
Choose the parameters that maximize the likelihood of the data.

$$\boldsymbol{\theta}^{\text{MLE}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta})$$

Maximum Likelihood Estimate (MLE)





MLE

What does maximizing likelihood accomplish?

- There is only a finite amount of probability mass (i.e. sum-to-one constraint)
- MLE tries to allocate **as much** probability mass **as possible** to the things we have observed…

  … **at the expense** of the things we have **not** observed

Recipe for Closed-form MLE

1. Assume data was generated i.i.d. from some model (i.e. write the generative story)

   $x^{(i)} \sim p(x|\boldsymbol{\theta})$

2. Write log-likelihood

   $\ell(\boldsymbol{\theta}) = \log p(x^{(1)}|\boldsymbol{\theta}) + \ldots + \log p(x^{(N)}|\boldsymbol{\theta})$

3. Compute partial derivatives (i.e. gradient)

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_1 = \ldots$

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_2 = \ldots$

   …

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_M = \ldots$

4. Set derivatives to zero and solve for $\boldsymbol{\theta}$

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_m = 0$ for all $m \in \{1, \ldots, M\}$

   $\boldsymbol{\theta}^{MLE}$ = solution to system of M equations and M variables

5. Compute the second derivative and check that $\ell(\boldsymbol{\theta})$ is concave down at $\boldsymbol{\theta}^{MLE}$

MLE

Example: MLE of Exponential Distribution

Goal:

- pdf of Exponential(λ): $f(x) = \lambda e^{-\lambda x}$
- Suppose $X_i \sim$ Exponential(λ) for $1 \leq i \leq N$.
- Find MLE for data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$

Steps:

- First write down log-likelihood of sample.
- Compute first derivative, set to zero, solve for λ.
- Compute second derivative and check that it is concave down at λ^{MLE}.

MLE

- pdf of Exponential(λ): $f(x) = \lambda e^{-\lambda x}$
- Suppose $X_i \sim$ Exponential(λ) for $1 \leq i \leq N$.
- Find MLE for data $\mathcal{D} = \{x^{(i)}\}_{i=1}^N$

Example: MLE of Exponential Distribution

- First write down log-likelihood of sample.

$$\ell(\lambda) = \sum_{i=1}^{N} \log f(x^{(i)}) \tag{1}$$

$$= \sum_{i=1}^{N} \log(\lambda \exp(-\lambda x^{(i)})) \tag{2}$$

$$= \sum_{i=1}^{N} \log(\lambda) + -\lambda x^{(i)} \tag{3}$$

$$= N \log(\lambda) - \lambda \sum_{i=1}^{N} x^{(i)} \tag{4}$$

MLE

- pdf of Exponential(λ): $f(x) = \lambda e^{-\lambda x}$
- Suppose $X_i \sim$ Exponential(λ) for $1 \leq i \leq N$.
- Find MLE for data $\mathcal{D} = \{x^{(i)}\}_{i=1}^N$

Example: MLE of Exponential Distribution

- Compute first derivative, set to zero, solve for λ.

$$\frac{d\ell(\lambda)}{d\lambda} = \frac{d}{d\lambda} N \log(\lambda) - \lambda \sum_{i=1}^{N} x^{(i)} \qquad (1)$$

$$= \frac{N}{\lambda} - \sum_{i=1}^{N} x^{(i)} = 0 \qquad (2)$$

$$\Rightarrow \lambda^{\text{MLE}} = \frac{N}{\sum_{i=1}^{N} x^{(i)}} \qquad (3)$$

MLE

In-Class Exercise

Show that the MLE of parameter ϕ for N samples drawn from Bernoulli(ϕ) is:

$$\phi_{MLE} = \frac{\text{Number of } x_i = 1}{N}$$

Steps to answer:

1. Write log-likelihood of sample
2. Compute derivative w.r.t. ϕ
3. Set derivative to zero and solve for ϕ

MLE

Question:

Assume we have N samples $x^{(1)}$, $x^{(2)}$, …, $x^{(N)}$ drawn from a Bernoulli(ϕ).

What is the **log-likelihood** of the data $\ell(\phi)$?

Assume $N_1 = \#\text{ of } (x^{(i)} = 1)$

$N_0 = \#\text{ of } (x^{(i)} = 0)$

Answer:

A. $l(\phi) = N_1 \log(\phi) + N_0 (1 - \log(\phi))$

B. $l(\phi) = N_1 \log(\phi) + N_0 \log(1-\phi)$

C. $l(\phi) = \log(\phi)^{N_1} + (1 - \log(\phi))^{N_0}$

D. $l(\phi) = \log(\phi)^{N_1} + \log(1-\phi)^{N_0}$

E. $l(\phi) = N_0 \log(\phi) + N_1 (1 - \log(\phi))$

F. $l(\phi) = N_0 \log(\phi) + N_1 \log(1-\phi)$

G. $l(\phi) = \log(\phi)^{N_0} + (1 - \log(\phi))^{N_1}$

H. $l(\phi) = \log(\phi)^{N_0} + \log(1-\phi)^{N_1}$

I. $l(\phi)$ = the most likely answer

MLE

Question:

Assume we have N samples $x^{(1)}$, $x^{(2)}$, …, $x^{(N)}$ drawn from a Bernoulli(ϕ).

What is the **derivative** of the log-likelihood $\partial \ell(\boldsymbol{\theta})/\partial\theta$?

Assume $N_1 = \#\ of\ (x^{(i)} = 1)$
$N_0 = \#\ of\ (x^{(i)} = 0)$

Answer:

A. $\partial \ell(\boldsymbol{\theta})/\partial\theta = \phi^{N_1} + (1 - \phi)^{N_0}$

B. $\partial \ell(\boldsymbol{\theta})/\partial\theta = \phi / N_1 + (1 - \phi) / N_0$

C. $\partial \ell(\boldsymbol{\theta})/\partial\theta = N_1 / \phi + N_0 / (1 - \phi)$

D. $\partial \ell(\boldsymbol{\theta})/\partial\theta = \log(\phi) / N_1 + \log(1 - \phi) / N_0$

E. $\partial \ell(\boldsymbol{\theta})/\partial\theta = N_1 / \log(\phi) + N_0 / \log(1 - \phi)$

Learning from Data (Frequentist)

Whiteboard

- Optimization for MLE
- Examples: 1D and 2D optimization
- Example: MLE of Bernoulli
- Example: MLE of Categorical
- Aside: Method of Langrange Multipliers

MLE vs. MAP

Suppose we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^N$

Principle of Maximum Likelihood Estimation:
Choose the parameters that maximize the likelihood of the data.

$$\boldsymbol{\theta}^{\text{MLE}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)} | \boldsymbol{\theta})$$

Maximum Likelihood Estimate (MLE)

Principle of Maximum *a posteriori* (MAP) Estimation:
Choose the parameters that maximize the posterior of the parameters given the data.

$$\boldsymbol{\theta}^{\text{MAP}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{i=1}^{N} p(\boldsymbol{\theta} | \mathbf{x}^{(i)})$$

Maximum *a posteriori* (MAP) estimate

MLE vs. MAP

Suppose we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^N$

Principle of Maximum Likelihood Estimation:
Choose the parameters that maximize the likelihood of the data.

$$\boldsymbol{\theta}^{\text{MLE}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta})$$

Maximum Likelihood Estimate (MLE)

Principle of Maximum *a posteriori* (MAP) Estimation:
Choose the parameters that maximize the posterior of the parameters given the data.

Prior

$$\boldsymbol{\theta}^{\text{MAP}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) \underbrace{p(\boldsymbol{\theta})}_{\text{Prior}}$$

Maximum *a posteriori* (MAP) estimate

MLE vs. MAP

Suppose we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$

Principle of Maximum Likeli

Choose the parameters that

of the data.

$$\boldsymbol{\theta}^{\text{MLE}} = \text{argr}$$

Maximum Likelihood Estimate (MLE)

Important!

Usually the parameters are **continuous,** so the prior is a probability **density** function

Principle of Maximum *a posteriori* (MAP) Estimation:

Choose the parameters that maximize the posterior of the parameters given the data.

Prior

$$\boldsymbol{\theta}^{\text{MAP}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)} | \boldsymbol{\theta}) \underbrace{p(\boldsymbol{\theta})}_{\text{Prior}}$$

Maximum *a posteriori* (MAP) estimate

Learning from Data (Bayesian)

Whiteboard

- *maximum a posteriori* (MAP) estimation
- Optimization for MAP
- Example: MAP of Bernoulli—Beta

Takeaways

- One view of what ML is trying to accomplish is **function approximation**
- The principle of **maximum likelihood estimation** provides an alternate view of learning

- **Synthetic data** can help **debug** ML algorithms
- Probability distributions can be used to **model** real data that occurs in the world
(don’t worry we’ll make our distributions more interesting soon!)

Learning Objectives

MLE / MAP

You should be able to…

1. Recall probability basics, including but not limited to: discrete and continuous random variables, probability mass functions, probability density functions, events vs. random variables, expectation and variance, joint probability distributions, marginal probabilities, conditional probabilities, independence, conditional independence
2. Describe common probability distributions such as the Beta, Dirichlet, Multinomial, Categorical, Gaussian, Exponential, etc.
3. State the principle of maximum likelihood estimation and explain what it tries to accomplish
4. State the principle of maximum a posteriori estimation and explain why we use it
5. Derive the MLE or MAP parameters of a simple model in closed form

NAÏVE BAYES

Naïve Bayes Outline

- **Real-world Dataset**
  - Economist vs. Onion articles
  - Document → bag-of-words → binary feature vector
- **Naive Bayes: Model**
  - Generating synthetic "labeled documents"
  - Definition of model
  - Naive Bayes assumption
  - Counting # of parameters with / without NB assumption
- **Naïve Bayes: Learning from Data**
  - Data likelihood
  - MLE for Naive Bayes
  - MAP for Naive Bayes
- **Visualizing Gaussian Naive Bayes**

Naïve Bayes

- Why are we talking about Naïve Bayes?
  - It's **just another decision function** that fits into our "big picture" recipe from last time
  - But it's our first **example of a Bayesian Network** and provides a *clearer* picture of **probabilistic learning**
  - Just like the other Bayes Nets we'll see, it **admits a closed form solution** for MLE and MAP
  - So learning is **extremely efficient** (just counting)

Fake News Detector

Today's Goal: To define a generative model of emails of two different classes (e.g. real vs. fake news)

CNN


U.S. » Crime + Justice | Energy + Environment | Extreme Weather | Space + Science | Live TV | U.S. Edition +

People are petitioning the White House to move Halloween to Saturdays

By Doug Criss, CNN
Updated 10:18 AM ET, Wed October 31, 2018

Trick-or-treating would be a weekend-only affair if an unofficial presidential petition were granted.

(CNN) — Americans long have used petitions to try to solve the nation's thorniest problems, and now a really weighty issue has landed atop the stack: moving Halloween to the weekend.

The change.org petition, addressed to President Donald Trump, wants the US to celebrate Halloween on the last Saturday of October instead of October 31. Nearly 40,000 people had signed it by midmorning Tuesday.

"It's time for a safer, longer, stress-free celebration!"

More from CNN
Longtime conservative columnist tells why he's a 'Never Trumper'
A 5-year-old boy tested positive for meth after...

Paid Content
These Eight Luxury Vacation Tours And Packages Are Incredible! And Under $2000!
Travel Enthusiast | Sponsored Links
Wow - These New Luxury Cars Are Loaded
Faqeo
Watch Parkeon Smart Kiosks Transform City Parking

The Onion


NEWS

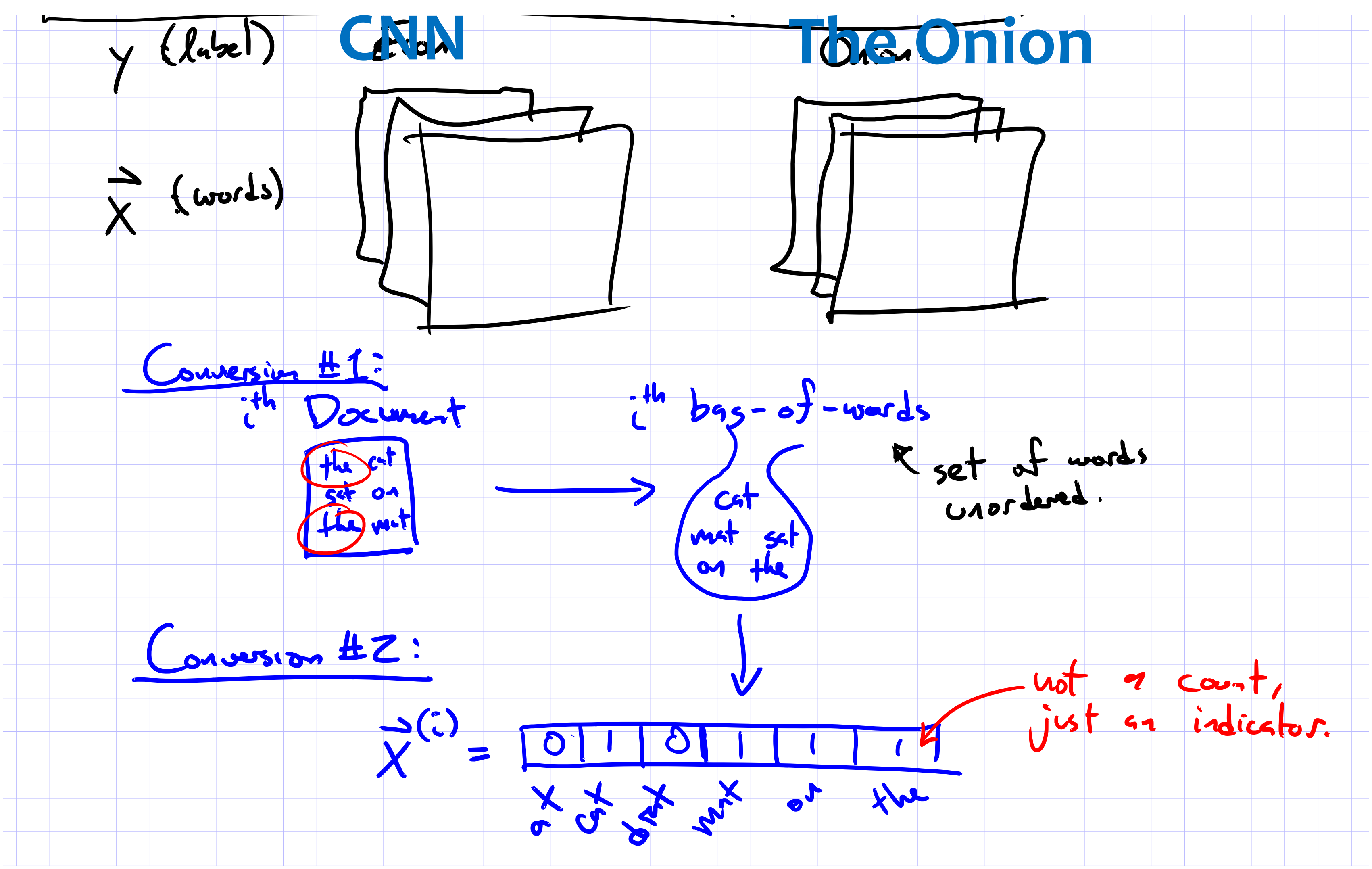
'Take This Grape For It Is The Witch's Eye, Take This Spaghetti For It Is The Witch's Brain,' Says Pope Francis During Halloween-Themed Communion

Today 12:00pm · SEE MORE: HALLOWEEN

Pope Francis administers the unholy sacrament to parishioners in the Haunted House of the Lord.

VATICAN CITY—Standing before his costumed congregants in St. Peter's Basilica, Pope Francis declared, "Take this grape for it is the witch's eye, take this spaghetti for it is the witch's brain," during a Halloween-themed Communion Wednesday, Vatican sources confirmed.

Fake News Detector



We can pretend the natural process generating these vectors is stochastic…

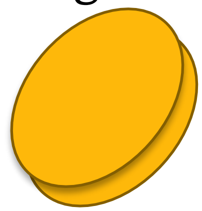
Naive Bayes: Model

Whiteboard

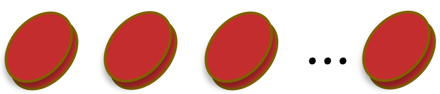
- Document → bag-of-words → binary feature vector
- Generating synthetic "labeled documents"
- Definition of model
- Naive Bayes assumption
- Counting # of parameters with / without NB assumption

Model 1: Bernoulli Naïve Bayes

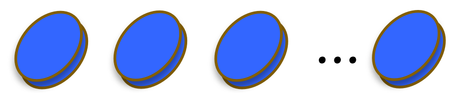
Flip weighted coin



If HEADS, flip each red coin



If TAILS, flip each blue coin



y	x_1	x_2	x_3	…	x_M
0	1	0	1	…	1
1	0	1	0	…	1
1	1	1	1	…	1
0	0	0	1	…	1
0	1	0	1	…	0
1	1	0	1	…	0

We can **generate** data in this fashion. Though in practice we never would since our data is **given.**

Instead, this provides an explanation of **how** the data was generated (albeit a terrible one).

Each red coin corresponds to an x_m