CPSC 440: Machine Learning

MAP Estimation

Winter 2022

Last Time: Bernoulli Distribution MLE

- The Bernoulli distribution for binary variables:
- We talked about difference inference tasks in Bernoulli models:
  - Compute likelihood of data, P(x^1, x^2,…, x^n | θ)
  - Find mode (decoding), argmax_x P(x | θ)
  - Generate samples $\tilde{x}$ from P(x | θ)
- We discussed learning with maximum likelihood estimation (MLE)
  - Find a $\hat{\theta}$ in argmax_θ P(x^1, x^2,…,x^n | θ)
  - Equivalent to finding $\hat{\theta}$ in argmax_θ log(P(x^1, x^2,…,x^n | θ)), "log-likelihood"
- For Bernoulli, equating derivative with respect to θ to 0 gives:
  - $\hat{\theta}$ = n_1/n (proportion of examples that are "1")

Derivation of MLE for Bernoulli

- We showed log-likelihood derivative is zero for $\theta = n_1/(n_1+n_0)$
  - Or $\theta = n_1/n$, since $n_1+n_0=n$

- We still need to convince ourselves this is a maximum:
  - You can verify that the second derivative of log-likelihood is negative
    - So the function is "curved downwards" and this is a maximum

- What about if $n_1=0$ or $n_0=0$?
  - In either case, our derivation would divide by zero
  - If $n_1 = 0$, MLE is $\theta = 0$; if $n_0 = 0$, MLE is $\theta = 1$
    - Can show that likelihood is increasing as it approaches 0/1 in these cases
    - So, the formula $\theta = n_1/n$ still works

Learning Task: Computing MLE

- Computing MLE for Bernoulli in code given data ‘X’:

Version 1:

```
n1 = sum(X)
n0 = n - n0
θ = n1/(n1 + n0)
```

Version 2:

```
θ = sum(X)/n
```

- Cost: O(n)
  - You need to sum up the *n* values (there’s a for loop hidden inside sum(X))

- You can then use this θ value for inference:
  - Compute likelihood of test data
  - Compute expected number of samples until first 1
  - Compute probability of seeing at least three 1 values in 10 samples

Next Topic: MAP Estimation

Problems with MLE

- In most settings, MLE is optimal as *n* goes to ∞.
  - It converges to the true parameter(s)
    - This is called "asymptotic consistency" (covered in honours/grad stats classes)
- However, it can be very sensitive for small *n*: $x^{(3)}$ $(x)^3$
  - Consider our example where $x^1=1$, $x^2=1$, $x^3=0$, and MLE was 0.67
  - If $x^4 = 1$, then MLE goes up to 0.75
  - If $x^4 = 0$, then MLE goes down to 0.5
    - If you get "unlucky" with your samples, the MLE might be really bad
- For Bernoullis, this sensitivity goes away quickly as we increase *n*
  - But for more complicated models, MLE tends to lead to overfitting

Problems with MLE

- Consider a different dataset consisting of $x^1=0$, $x^2=0$, $x^3=0$
  - In this case the MLE is $\theta = 0$
    - It assigns zero probability to events that do not occur in training data
- Causes problems if we have a '1' in test data:
  - Then likelihood of entire test set is 0, since:
    - A case of overfitting to the training data
    - If you test ten people and none have COVID, does that mean it's eradicated?
- It is common to add Laplace smoothing to the estimator:

$$\hat{\theta} = \frac{n_1 + 1}{(n_1+1)+(n_0+1)} = \frac{n_1 + 1}{n + 2}$$

  - MLE for a dataset with an extra "imaginary" 1 and 0 in the data.
    - This is a special case of "MAP estimation"

MLE and MAP Estimation

- In MLE we maximize the probability of the data given parameters:

$$\hat{\theta} \in \underset{\theta}{\operatorname{argmax}} \{ p(X \mid \theta) \}$$

- But this is kind of weird:
  - "Find the θ that makes **X** have the highest probability given θ"
  - Get overfitting, because data could be likely for an unlikely θ
    - For example, a complex model that overfits by memorizing the data

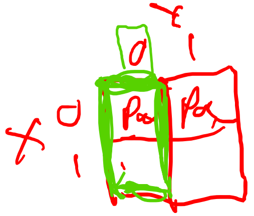
- What we really want if we are trying to find the "best" θ:
  - "Find the θ that has the highest probability given the data **X**."

$$\hat{\theta} \in \underset{\theta}{\operatorname{argmax}} \{ p(\theta \mid X) \}$$

reversed

  - This is called MAP estimation ("maximum a posteriori")

Digression: Super-Quick "Probability Rule" Review



= Pr(B|A) Pr(A)

- Product rule: Pr(A ∩ B) = Pr(A | B) Pr(B).
  - Re-arrange to get conditional probability formula: Pr(A | B) = Pr(A ∩ B)/ Pr(B)
  - Order doesn't matter in joint probabilities: Pr(A ∩ B) = Pr(B ∩ A)
  - Use product rule twice to get Bayes rule: Pr(A | B) = Pr(B | A) Pr(A) / Pr(B)
    - Conditional in terms of "reverse" conditional, and the "marginals" Pr(B) and Pr(A)

- Marginalization rule ("summing or integrating over a variable"):
  - Variable X with discrete domain: Pr(A) = $\sum_x \Pr(A \cap X = x)$
  - Variable X with continuous domain: Pr(A) = $\int p(A \cap X = x)\mathrm{d}x$

- These two rules are good friends and usually appear together:
  - $\mathrm{p(a)} = \sum_b p(a,b) = \sum_b p(a|b)p(b)$.
  - $\mathrm{p(a|b)} = \frac{p(b\,|a)p(a)}{p(b)} = \frac{p(b\,|a)p(a)}{\sum_a p(b\,|a)p(a)}$ (some people call this "Bayes rule").

- Rules still work if you add extra "conditioning" on the right:
  - p(a,b |c) = p(a | b, c)p(b | c).
  - p(a |c) = $\sum_b p(a,b|c)$.

MEMORIZE EVERYTHING ON THIS SLIDE

Maximum a Posteriori (MAP) Estimation

- Maximum a posteriori (MAP) estimate maximizes posterior probability:

$$\hat{\theta} \in \underset{\theta}{\operatorname{argmax}} \{ \underbrace{p(\theta \mid X)}_{\text{"posterior"}} \}$$

  - Bayesians would argue that this is reasonably what we want: the most likely θ given our data
- MLE and MAP are connected by Bayes rule:

$$\underbrace{p(\theta \mid X)}_{\text{(posterior)}} = \frac{p(X, \theta)}{p(X)} = \frac{p(X \mid \theta)\, p(\theta)}{p(X)} \propto \underbrace{p(X \mid \theta)}_{\text{(likelihood)}} \underbrace{p(\theta)}_{\text{(prior)}}$$

"proportional to"

definition of conditional probability

"product rule" $p(a,b) = p(a|b)p(b)$

constant that does not depend on θ

  - So posterior is proportional the likelihood $p(X|\theta)$ times the prior $p(\theta)$.
    - See "probability" notes on course webpage if equalities above aren't obvious (you need catch up fast).

The prior

- The prior p(θ) can encode our preference for different parameters
  - If we are flipping coins, we might think P(θ) is higher for values close to ½
    - We could make it really high for the exact value ½
  - In COVID-19 example, we might make P(θ) higher for values close to 0.05
    - Because, for example, we estimated a value of 0.05 from a similar population
  - In CPSC 340, you learned that priors correspond to regularizers
    - You often choose P(θ) to be lower for values that are likely to overfit

- Laplace smoothing corresponds to a particular p(θ)
  - We'll show this shortly

MAP Estimation for Bernoulli with Discrete Prior

- Consider our example where x_1=1, x_2=1, x_3=0 (and MLE was 0.67)
- Consider using a prior of:
  - $\Pr(\theta = 0.00)$ = 0.05
  - $\Pr(\theta = 0.25)$ = 0.2
  - $\Pr(\theta = 0.50)$ = 0.5
  - $\Pr(\theta = 0.75)$ = 0.2
  - $\Pr(\theta = 1.00)$ = 0.05

Posterior values are proportional to:
  - $\Pr(\theta = 0.00 \mid \mathbf{X}) \propto (0*0*1)*.05 = 0$
  - $\Pr(\theta = 0.25 \mid \mathbf{X}) \propto (.25*.25*.75)*.2 \approx 0.01$
  - $\Pr(\theta = 0.50 \mid \mathbf{X}) \propto (.5*.5*.5)*.5 \approx 0.06$
  - $\Pr(\theta = 0.75 \mid \mathbf{X}) \propto (.75*.75*.25)*.2 \approx 0.03$
  - $\Pr(\theta = 1.00 \mid \mathbf{X}) \propto (1*1*0)*.05 = 0$
- So our MAP estimate is θ = 0.5
  - Based on our prior “guesses for θ”, we think this is a fair coin
    - Notice that we don’t need P(**X**) in our calculations (since it’s the same for all θ)

Digression: "Proportional to" ($\propto$) Notation

- In math, the notation $f(\theta) \propto g(\theta)$
  means that $f(\theta) = \kappa g(\theta)$ for some number κ (for all θ)
  - But κ may not be known and/or may not be unique
    - For example, $f(\theta) \propto \theta^2$ for both $f(\theta) = 10\theta^2$ and $f(\theta) = -50\theta^2$

- For discrete probabilities, the constant κ is positive and unique
  - This is because probabilities are non-negative and sum to 1

- Consider a discrete variable θ with $p(\theta) = \kappa g(\theta) \propto g(\theta)$:
  - Since $\sum_{\theta'} P(\theta') = 1$, we have $\sum_{\theta'} \kappa g(\theta') = 1$
    - Solving for κ gives: $\kappa = \frac{1}{\sum_{\theta'} g(\theta')}$
  - Using this value for κ we have $p(\theta) = \kappa g(\theta) = \frac{g(\theta)}{\sum_{\theta'} g(\theta')}$
  - You can use this trick to get posterior probabilities on last slide: $p(\theta=0.5 \mid X) = \frac{0.06}{0+0.01+0.06+0.03+0}$

Values the posterior was proportional to.

Digression^2: “Probability” vs. “Probability Density”

- Recall that the value θ can be any number between 0 and 1
  - Instead of putting non-zero probability on a finite number of possible θ values, we could treat θ as a continuous random variable (to allow $\theta = 0.3452$)

- For continuous variables, we use a probability density function (PDF):
  - Function *p* that is non-negative and integrates to 1 over domain:

$$p(\theta) \geq 0 \text{ for all } \theta, \text{ and } \int_{-\infty}^{\infty} p(\theta)\,d\theta = 1$$

- We get probabilities from the PDF by integrating over ranges:

$$\Pr(0.45 \leq \theta \leq 0.55) = \int_{0.45}^{0.55} p(\theta)\,d\theta$$

  - If the PDF is continuous, probability of an individual θ is 0: $\text{prob}(\theta = 0.5) = \int_{0.5}^{0.5} p(\theta)\,d\theta = 0$

Digression^2: "Probability" vs. "Probability Density"

- Recall the relationship between posterior, likelihood, and prior:

$$\underbrace{p(\theta \mid X)}_{\text{(posterior)}} \propto \underbrace{p(X \mid \theta)}_{\text{(likelihood)}} \underbrace{p(\theta)}_{\text{(prior)}}$$

- What are these *p* functions in discrete and continuous case?
  - If θ is discrete: prior and posterior *p* functions are probabilities
  - If θ is continuous: prior and posterior *p* functions are PDFs
    - So p(θ) is not the "probability of θ", but the "probability density of θ"
- With our binary *X* values, likelihood p($\mathbf{X} \mid \theta$) is a probability
  - But when we later talk about continuous *X*, likelihood will be a PDF
- Important: Most ML people are really sloppy about this!
  - Say "probability of θ" for p(θ), even for continuous θ
  - I *try* to use P for probabilities and *p* for PDFs, but it's hard…

Digression: "Proportional to" (∝) Notation

- Consider a continuous variable θ with PDF $p(\theta) = \kappa g(\theta) \propto g(\theta)$:
  - Since $\int_{\theta'} p(\theta')d\theta' = 1$, we have $\int_{\theta'} \kappa g(\theta')d\theta' = 1$
    - Solving for κ gives: $\kappa = \frac{1}{\int_{\theta'} g(\theta')d\theta'}$
  - So we have $p(\theta) = \frac{g(\theta)}{\int_{\theta'} g(\theta')d\theta'}$

- For continuous θ in MAP estimation, we have $p(\theta \mid X) \propto p(X \mid \theta)p(\theta)$,
  - So we have $p(\theta \mid X) = \frac{p(X \mid \theta)p(\theta)}{\underbrace{\int_{\theta'} p(X \mid \theta')p(\theta')d\theta'}_{= p(X)}}$ $= p(X)$ by "marginalization rule": $p(a) = \sum_b p(a,b)$ (discrete) or $p(a) = \int_b p(a,b)db$ (continuous)

- You should memorize these "digression" slides
  - Knowing how to use "∝" simplifies a lot of things

Beta Distribution

- For Bernoulli likelihoods, most common prior is beta distribution:

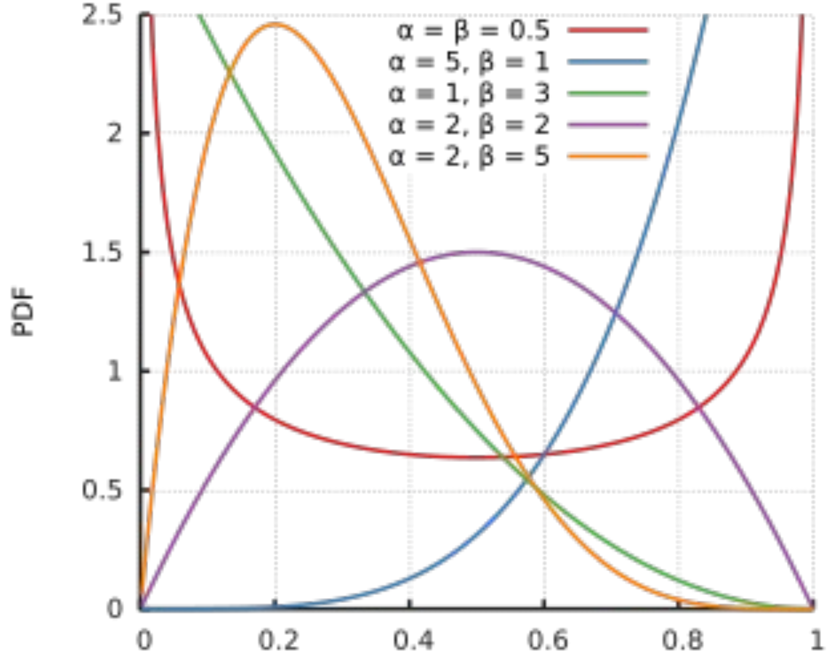
$$p(\theta \mid \alpha, \beta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1} \quad \text{for } \underbrace{0 \leq \theta \leq 1}_{p(\theta \mid \alpha,\beta)=0 \text{ if } \theta<0 \text{ or } \theta>1},\ \alpha>1,\ \beta>1$$

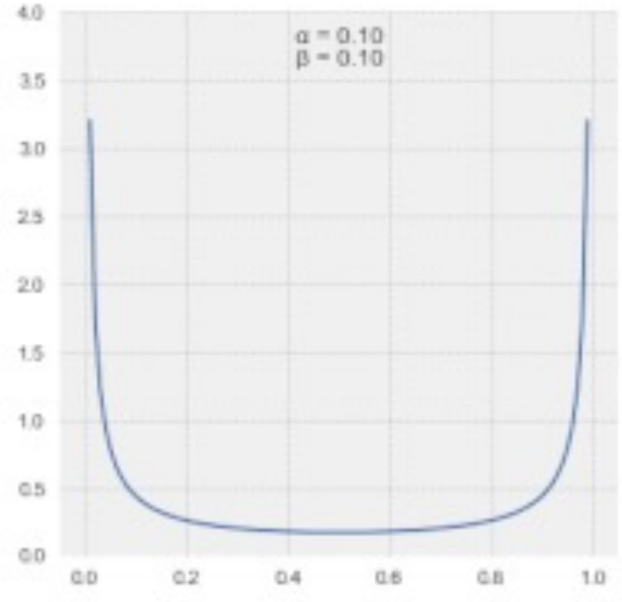
- Looks like a Bernoulli likelihood, with (α – 1) ones and (β-1) zeroes.
- Key difference with the Bernoulli is on the left side:
  - It defines a PDF over real numbers θ in the range 0 through 1.
    - Beta distribution is not assigning probabilities to binary values, but to θ
      - "Probability over probabilities"
- From the "digression", we can resolve what is hidden in the ∝ sign:

$$p(\theta \mid \alpha,\beta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\int \theta^{\alpha-1}(1-\theta)^{\beta-1}\,d\theta} = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)} \leftarrow \text{"beta" function}$$

Beta Distribution

- The beta distribution for different choices of α and β:





- Why is using the beta distribution as prior so popular?
  - Fake reason: it is quite flexible, so can encode a variety of priors.
    - Can represent bias towards 0.5, towards 1 or 0, towards 0.2, towards only 1, or uniform if $\alpha = \beta = 1$.
    - But it is still limited. For example, you can't say that "the exact value 0.5 is particularly likely".

https://en.wikipedia.org/wiki/Beta_distribution

Posterior for Bernoulli Likelihood and Beta Prior

- Real reason people use the beta: posterior and MAP have simple forms.
  - The posterior with a Bernoulli likelihood and beta prior:

$$p(\theta \mid X, \alpha, \beta) \propto \underbrace{p(X \mid \theta)}\, p(\theta \mid \alpha, \beta) \propto \theta^{n_1}(1-\theta)^{n_0}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$

We assume that X is independent of α and β given θ

$$= \theta^{(n_1+\alpha)-1}(1-\theta)^{(n_0+\beta)-1}$$

$$= \theta^{\tilde{\alpha}-1}(1-\theta)^{\tilde{\beta}-1}$$

  - This is another beta distribution with "updated" parameters $\tilde{\alpha}$ and $\tilde{\beta}$
    - Where $\tilde{\alpha} = n_1 + \alpha$ and $\tilde{\beta} = n_0 + \beta$.
  - How do we know that this is a beta distribution?
    - Because constant in $\propto$ is unique
      - "If you are proportional to a beta distribution, you are a beta distribution."
    - Make sure you understand why posterior is a beta distribution (important in this course)

MAP Estimation for Bernoulli-Beta Model

- The posterior with a Bernoulli likelihood and beta prior is a beta:

$$p(\theta \mid X, \alpha, \beta) = \frac{\theta^{\tilde{\alpha}-1}(1-\theta)^{\tilde{\beta}-1}}{B(\tilde{\alpha}, \tilde{\beta})}$$

"beta" function (which does not depend on θ)

  - Where $\tilde{\alpha} = n_1 + \alpha$ and $\tilde{\beta} = n_0 + \beta$.
- If $\tilde{\alpha} > 1$ and $\tilde{\beta} > 1$, taking log and setting derivative to 0 gives MAP of:

$$\hat{\theta} = \frac{n_1 + \alpha - 1}{(n_1 + \alpha - 1) + (n_0 + \beta - 1)}$$

$$\boxed{\theta = (\mathrm{sum}(X) + \alpha - 1)/(n + \alpha + \beta - 2)}$$

Cost: $O(n)$

  - If α = 1 and β = 1, we get the MLE
  - If α = 2 and β = 2, we get Laplace smoothing (which often overfits less)
  - If $\alpha = \beta > 2$, we get a stronger bias towards $\hat{\theta} = 0.5$ than Laplace smoothing
  - If $\alpha = \beta < 1$, we get a bias towards away from $\hat{\theta} = 0.5$ (towards 0 or 1)
  - You can also bias towards either 0 or 1; if α is large compared to β it biases towards $\hat{\theta}$=1
  - Notice that MAP converges to MLE n → ∞, so the data eventually "takes over" estimate
    - But we use a prior so our model does sensible things when we do not have enough data

review

Review: Hyper-Parameter and [Cross]-Validation

- We call the "parameters of the prior", α and β, the hyper-parameters.
  - We usually say that hyper-parameters are "parameters affecting the complexity of the model"
  - We usually also include "parameters of the learning algorithm" as hyper-parameters

- How can we choose hyper-parameters values?
  - Using the training likelihood does not work: it would make α and β arbitrarily small (ignoring prior)

- Usual CPSC 340 approach: use a validation set (or cross-validation)
  - Split your data **X** into a "training" set and a "validation" set
  - For different hyper-parameters of α and β:
    - Use the "training" examples to compute the MAP estimate
    - Use MAP estimate to compute the likelihood of the "validation" examples
  - Choose the hyper-parameters with the highest validation likelihood
    - But our final goal is to **not** optimize performance on the validation set
    - This is a surrogate for the test error (error on completely-new data), which you cannot measure.

- Take CPSC 340 to learn about many of the things that can go wrong
  - For example, if you are not careful you can overfit to the validation set
    - Happens all the time, even in UBC student's PhD theses and in top conference papers!
- Or take CPSC 532D to understand it more mathematically :)

* Blue: "Review:..." slides:
→ These are topics that covered in detail in CPSC 340, that I expect you to understand but that will not be covered in detail in this course.

Next Topic: Product of Bernoullis

Motivation: Modeling Traffic Congestion

- We want to model car “traffic congestion” in a big city.
- So we measure which intersections are busy on different days:

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

- We want to build a model of this data, to identify patterns/problems.
  - “Inter 4 is always busy”, “Inter 1 is rarely busy”.
  - “Inters 7+8 are always the same”, “Inter 2 is busy when Inter 7 is busy”.
  - “There is a 25% chance you will hit a busy intersection if you take Inter 1 and 8”.

Problem: Multivariate Binary Density Estimation

- We can view this as multivariate binary density estimation:
  - Input: n IID samples of binary vectors $x^1, x^2, x^3, \ldots, x^n$ from population.
  - Output: model that gives probability for any assignment of values $x \in \{0,1\}^d$.

X =

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

⇨ $\Pr(X_1 = 0, X_2 = 1, X_3 = 0, X_4 = 1, X_5 = 1, X_6 = 1, X_7 = 0, X_8 = 0, X_9 = 1) = 0.11$

(Estimates probability for all 2^9 values)

- Covid example: each feature could be "are covid cases >10% in area j?"
- Notation (please memorize):
  - We use n for the number of examples, d for the number of features
  - Notice that x^3 is a vector with d values, x_1^3 to x_d^3
  - X_3 is the third dimension of a random vector X; x_3 is a value X_3 might take

Product of Bernoullis Model

- There are **many** different models for binary density estimation
  - Each one makes different assumptions…we'll see lots of options!
- We'll start with the simple "product of Bernoullis" model
  - In this model we assume that the variables are "mutually independent"
    - If we have four variables, we assume $P(x_1, x_2, x_3, x_4) = P(x_1)\,P(x_2)\,P(x_3)\,P(x_4)$
  - As a picture, we treat multivariate problem as 'd' univariate problems:

X =

Inter 1	Inter 2	Inter 3	Inter 4
0	1	0	1
0	1	0	1
0	0	1	1
0	1	0	1
1	1	1	1
0	0	0	1
0	1	0	1

⇨

X_1 =

Inter 1
0
0
0
0
1
0
0

use to compute θ_1

X_2 =

Inter 2
1
1
0
1
1
0
1

use to compute θ_2

X_3 =

Inter 3
0
0
1
0
1
0
0

use to compute θ_3

X_4 =

Inter 4
1
1
1
1
1
1
1

compute θ_4

Product of Bernoullis Inference and Learning

- Key advantage of "product of Bernoullis" model: easy inference and learning
  - For most inference tasks: do inference on each variable, then combine the results
  - Compute joint probability
    - $\Pr(X_1 = 1, X_2 = 1, \ldots, X_d = 0) = \Pr(X_1 = 1)\, P(X_2 = 1) \cdots P(X_d = 0) = \theta_1\ \theta_2 \cdots (1 - \theta_d)$.
  - Compute marginal probabilities
    - $\Pr(X_2{=}1) = \theta_2$
    - $\Pr(X_2{=}1, X_3 = 1) = \Pr(X_2{=}1)\Pr(X_3{=}1) = \theta_2\theta_3$.
  - Compute conditional probabilities.
    - $P(x_2 \mid x_3) = P(x_2)$.

      def of cond. prob   ind.

      $$p(x_2 \mid x_3) = \frac{p(x_2, x_3)}{p(x_3)} = \frac{p(x_2)\cancel{p(x_3)}}{\cancel{p(x_3)}} = p(x_2)$$
    - $\Pr(X_2 = 0, X_3 = 1 \mid X_4 = 0) = \Pr(X_2{=}0, X_3{=}1) = (1 - \theta_2)\theta_3$.
  - Mode of $p(x_1, x_2, \ldots, x_d)$:
    - Set x_1 to argmax value of $P(x_1)$, set x_2 to argmax of $P(x_2)$,…, set x_d to argmax value of $P(x_d)$
  - Sampling:
    - Sample x_1 from $P(x_1)$, sample x_2 from $P(x_2)$,…, sample x_d from $P(x_d)$
- MLE (MAP is similar): $\hat{\theta}_1 = \frac{n_{11}}{n}$ ← number of times variable '1' is '1' $\hat{\theta}_2 = \frac{n_{21}}{n}$ . . . $\hat{\theta}_d = \frac{n_{d1}}{n}$

Product of Bernoullis Inference and Learning

- MLE in a product of Bernoullis:

```
θ = zeros(d)
for i in 1:n
    for j in 1:d
        if X[i,j] == 1
            θ[j] += 1
θ ./= n
```

count the number of times each $x_j^i = 1$

divide by 'n'

or

```
θ = sum(X, dims=1)./n
```

sum up columns of 'X'

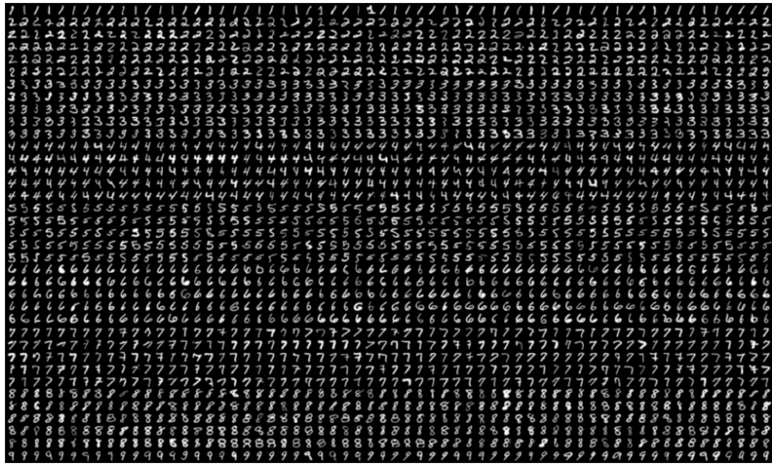
- Cost is $O(nd)$: do an $O(1)$ operation n*d times, then $O(n)$ division
  - If **X** is stored as a "sparse" matrix, can be implemented to only cost $O(z)$
    - z is the number of non-zero values ($z \leq nd$)
- Sampling code:

```
x = zeros(d)
for j in 1:d
    x[j] = sampleBinary(θ[j])
```

cost is $O(d)$ to generate a sample.

Running Example: MNIST Digits

- To illustrate density estimation, we will often use the MNIST digits:
  - 60,000 images, each a 28x28 pixel image of a number
  - Representing as binary density estimation:
    - Each image is one training example x^i
    - Each feature is one of the 784 pixels
    - Threshold each pixel to make it binary

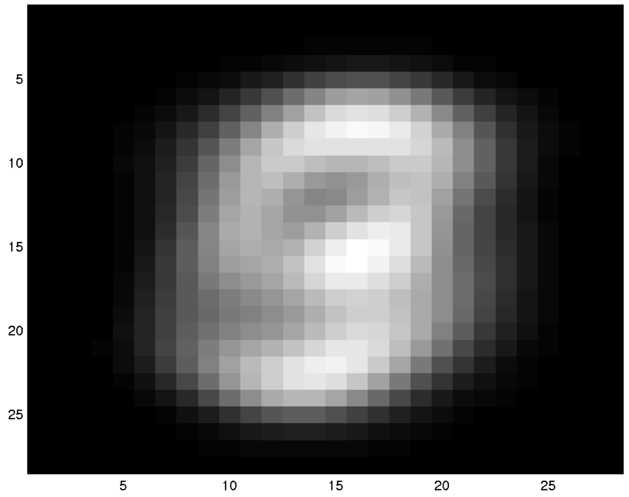


- CPSC 340 wanted to "recognize that this is a 4"
- In density estimation we want a probability distribution over images
  - Given one of the 2^{784} possible images, what is the probability it is a digit?
    - This is unsupervised; we're ignoring the class labels.
  - Sampling from the density should generate new images of digits.

https://www.kaggle.com/tarunkr/digit-recognition-tutorial-cnn-99-67-accuracy

Product of Bernoullis on MNIST Digits

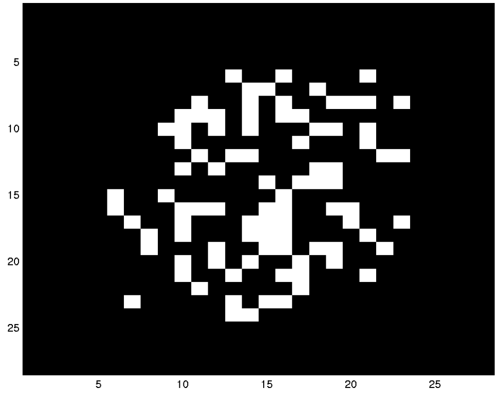
- Consider fitting the product of Bernoullis model to MNIST digits:
  - For each of the 784 pixels j, we have a parameter θ_j
    - A "position-specific Bernoulli" distribution
  - To compute MLE for θ_j, compute fraction of times pixel j was set to 1
    - Visualizing those MLE values as an image:

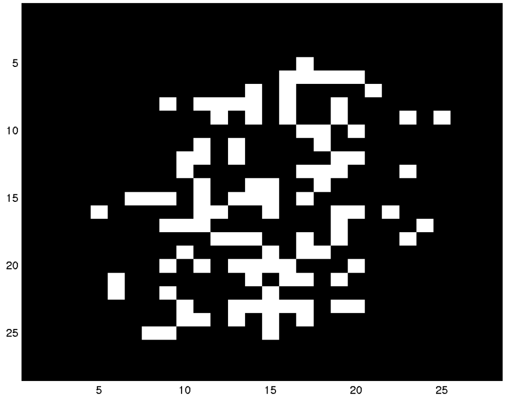


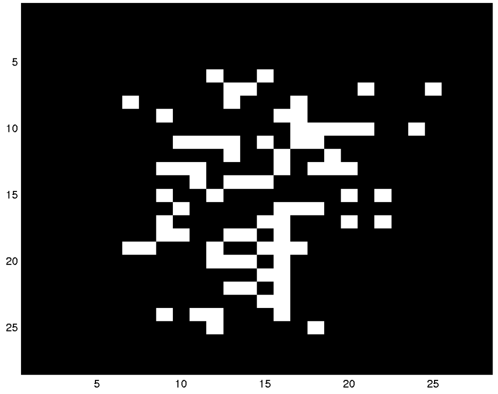
    - Pixels near the center are more likely to be 1 than pixels near the boundary

Product of Bernoullis on MNIST Digits

- Is product of Bernoullis a good model for the MNIST digits?
  - Samples generated from the model (independent sample from position-specific Bernoulli for each pixel):







  - This is a terrible model: these samples do not look like the data at all
  - Why is this a terrible model?
    - In the dataset, the pixels are not independent
    - For example, pixels that are "next" to each other in the image are highly correlated
  - Even it is a bad model, product of Bernoullis is often "good enough to be useful"
    - Usually when combined with other ideas, that we'll see shortly
    - In practice, I think it is actually the most-used method for binary density estimation even though it is one of the worst
  - Later in the course we'll cover several ways to relax the independence assumption

Summary

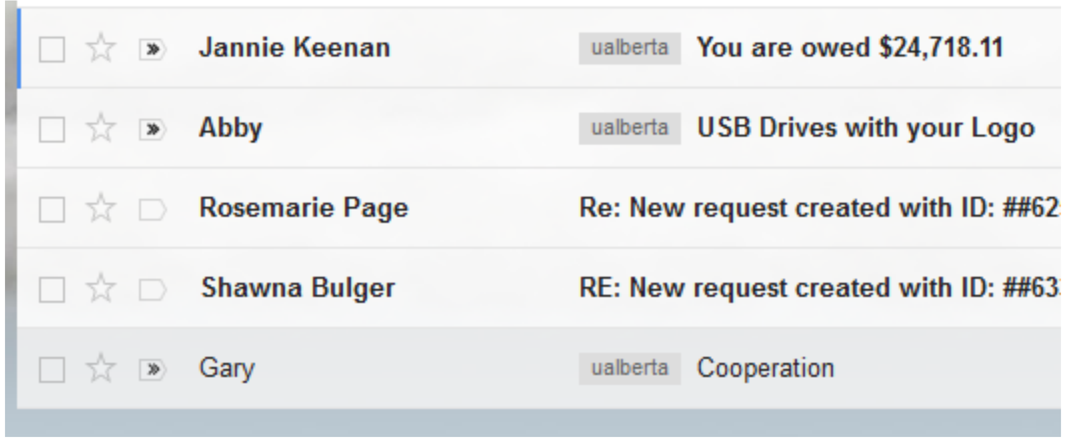
- MAP Estimation:
  - Find parameters maximizing posterior probability of parameters given data
  - Requires prior distribution on parameters: bias towards parameters that overfit less
- Probability review:
  - Product rule, marginalization rule, Bayes rule
  - Continuous "probabilities" and how "∝" has a restricted meaning for probabilities
- Beta distribution:
  - Prior for Bernoulli that yields a closed-form posterior (another beta distribution)
- Product of Bernoullis:
  - Method for multivariate binary density estimation.
  - Assumes all variables are independent.
  - Inference and learning are easy, but cannot accurate model many densities.

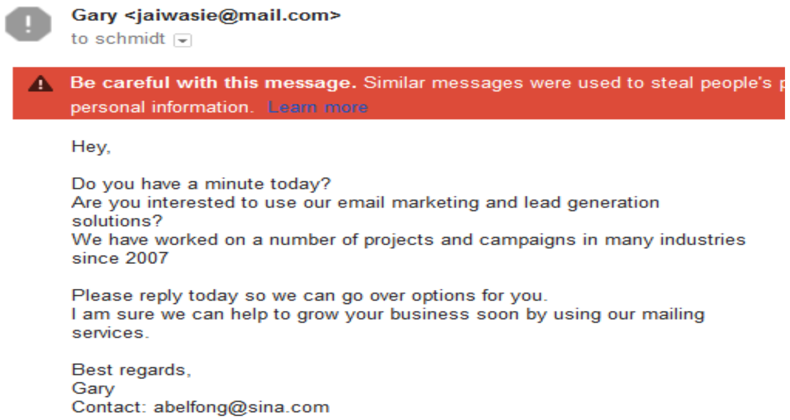
Next Topic: Generative Classifiers

Might not get to this in class,
and if not will skip for now!
I’ll delete from “final slides” if so

Motivation: E-mail Spam Filtering

- Want a build a system that detects spam e-mails.
  - Context: spam used to be a big problem.





- We can write this as a supervised learning problem:
  - Want to learn to map from “input” (e-mail) to “output” (spam or not).

review

Review: Data Collection and Feature Extraction

- Collect a large number of e-mails, gets users to label them.

$	Hi	CPSC	340	Vicodin	Offer	…		Spam?
1	1	0	0	1	0	…	⟹	1
0	0	0	0	1	1	…	⟹	1
0	1	1	1	0	0	…	⟹	0
…	…	…	…	…	…	…	⟹	…

- We can use ($y^i = 1$) if e-mail 'i' is spam, ($y^i = 0$) if e-mail is not spam.
- Extract features of each e-mail (like "bag of words").
  - ($x^i_j = 1$) if word/phrase 'j' is in e-mail 'i', ($x^i_j = 0$) if it is not.
    - See CPSC 330 (or 340) for different ways to extract features from text data.

review

Review: Supervised Learning Notation

- Our notation for supervised learning:

$X =$

$	Hi	CPSC	340	Vicodin	Offer	…
1	1	0	0	1	0	…
0	0	0	0	1	1	…
0	1	1	1	0	0	…
…	…	…	…	…	…	…

x_6^2

x^3

$y =$

Spam?
1
1
0
…

y^i

- X is matrix of all features, y is vector of all labels.
  - We use y^i for the label of example ‘i’ (element ‘i’ of ‘y’).
  - We use x^i_j for feature ‘j’ of example ‘i‘.
  - We use x^i as the list of features of example ‘i’ (row ‘i’ of ‘X’).
    - So in the above x^3 = [0 1 1 1 0 0 …].
    - In practice, store x^i in some “sparse” format (like a list of non-zeroes, smaller memory).

Generative Classifiers

- In early 2000s, best spam filtering methods used generative classifiers.
  - Generative classifiers treat supervised learning as density estimation.
- How can we do supervised learning with density estimation?
  - Learning: use a density estimator to estimate $p(x_1, x_2,\ldots,x_d,y)$.
    - Generative classifiers model "how the features and label were generated".
  - Inference: compute conditionals $p(y \mid x_1,x_2,\ldots,x_d)$ to make predictions.
    - For example, is $p(y = 1 \mid x_1,x_2,\ldots,x_d) > p(y = 0 \mid x_1,x_2,\ldots,x_d)$?
- Can we use a product of Bernoullis as the density estimator?
  - You could, but it would do terrible!
  - If 'y' is independent of the features, predictions would ignore features.
  - A simple model that does assume 'y' is independent of features is naïve Bayes.

$$\boxed{\begin{aligned} & p(y \mid x_1, x_2, \ldots, x_d) \\ & \propto p(x_1, x_2, \ldots, x_d, y) \\ & = p(x_1)p(x_2)\cdots p(x_d)p(y) \\ & = p(y) \end{aligned}}$$

Existence of MAP Estimate under Beta Prior

bonus!

- The MAP estimate for Bernoulli likelihood and beta prior:

$$\hat{\theta} = \frac{n_1 + \alpha - 1}{(n_1 + \alpha - 1) + (n_0 + \beta - 1)}$$

  - This assumes that $n_1 + \alpha > 1$ and $n_0 + \beta > 1$.

- Other cases:
  - $n_1 + \alpha > 1$ and $n_0 + \beta \leq 1$: $\hat{\theta} = 1$.
  - $n_1 + \alpha \leq 1$ and $n_0 + \beta > 1$: $\hat{\theta} = 0$.
  - $n_1 + \alpha < 1$ and $n_0 + \beta < 1$: $\hat{\theta}$ can be 0 or 1.
  - $n_1 + \alpha = 1$ and $n_0 = 1$: $\hat{\theta}$ can be anything between 0 and 1.