Machine Learning
4771

Instructor: Tony Jebara

Topic 8

- Discrete Probability Models
- Independence
- Bernoulli
- Text: Naïve Bayes
- Multinomial
- Text: Bag of Words

Discrete Probability Models

- Bernoulli: recall binary (coin flip) probability, just 1x2 table

$$p\left(x\right) = \alpha^{x}\left(1-\alpha\right)^{1-x} \qquad \alpha \in \left[0,1\right] \;\; x \in \left\{0,1\right\}$$

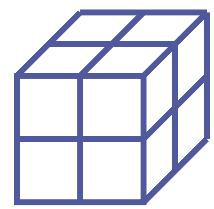
x=0	x=1
0.73	0.27

- Multidimensional Bernoulli: multiple binary events

$$p\left(x_1, x_2\right)$$

	x_2=0	x_2=1
x_1=0	0.4	0.1
x_1=1	0.3	0.2

$$p\left(x_1, x_2, x_3\right)$$



- Why do we write these as an equations instead of tables?

Discrete Probability Models

- Bernoulli: recall binary (coin flip) probability, just 1x2 table

$$p(x) = \alpha^x (1-\alpha)^{1-x} \qquad \alpha \in [0,1] \quad x \in \{0,1\}$$

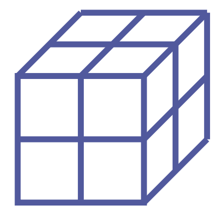
x=0	x=1
0.73	0.27

- Multidimensional Bernoulli: multiple binary events

$$p(x_1, x_2)$$

	x_2=0	x_2=1
x_1=0	0.4	0.1
x_1=1	0.3	0.2

$$p(x_1, x_2, x_3)$$



- Why do we write these as an equations instead of tables?

- To do things like… maximum likelihood…
- Fill in the table so that it matches real data…
- Example: coin flips H,H,T,T,T,H,T,H,H,H ???

x=T	x=H

Discrete Probability Models

- Bernoulli: recall binary (coin flip) probability, just 1x2 table

$$p(x) = \alpha^{x}(1-\alpha)^{1-x} \qquad \alpha \in [0,1] \quad x \in \{0,1\}$$

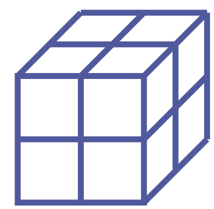
x=0	x=1
0.73	0.27

- Multidimensional Probability Table: multiple binary events

$$p(x_1, x_2)$$

	x_2=0	x_2=1
x_1=0	0.4	0.1
x_1=1	0.3	0.2

$$p(x_1, x_2, x_3)$$



- Why do we write these as an equations instead of tables?

- To do things like… maximum likelihood…
- Fill in the table so that it matches real data…
- Example: coin flips H,H,T,T,T,H,T,H,H,H
- Why is this correct?

x=T	x=H
0.4	0.6

Bernoulli Probability (ML)

- Bernoulli: $$p\left(x\right) = \alpha^{x}\left(1-\alpha\right)^{1-x} \qquad \alpha \in \left[0,1\right] \;\; x \in \left\{0,1\right\}$$
- Log-Likelihood (IID): $$\sum\nolimits_{i=1}^{N} \log p\left(x_i \mid \alpha\right) = \sum\nolimits_{i=1}^{N} \log \alpha^{x_i}\left(1-\alpha\right)^{1-x_i}$$
- Gradient=0:

$$\frac{\partial}{\partial\alpha}\sum\nolimits_{i=1}^{N} \log \alpha^{x_i}\left(1-\alpha\right)^{1-x_i} = 0$$

$$\frac{\partial}{\partial\alpha}\sum\nolimits_{i=1}^{N} x_i \log\alpha + \left(1-x_i\right)\log\left(1-\alpha\right) = 0$$

$$\frac{\partial}{\partial\alpha}\sum\nolimits_{i\in class1} \log\alpha + \sum\nolimits_{i\in class0} \log\left(1-\alpha\right) = 0$$

$$\sum\nolimits_{i\in class1} \tfrac{1}{\alpha} - \sum\nolimits_{i\in class0} \tfrac{1}{1-\alpha} = 0$$

$$N_1 \tfrac{1}{\alpha} - N_0 \tfrac{1}{1-\alpha} = 0$$

$$N_1\left(1-\alpha\right) - N_0\alpha = 0$$

$$N_1 - \left(N_1 + N_0\right)\alpha = 0$$

$$\alpha = \tfrac{N_1}{N_1+N_0}$$

x=0	x=1
$\frac{N_0}{N_0+N_1}$	$\frac{N_1}{N_0+N_1}$

Text: Naïve Bayes

- Text classification: simplest model

$$\in \left\{ \begin{array}{c} religion \\ politics \end{array} \right.$$

- There are about 50,000 words in English
- Each document is D=50,000 dimensional binary vector $\vec{x}_i$
- Each dimension is a word, set to 1 if word in the document

  Dim1: "the" = 1
  Dim2: "hello" = 0
  Dim3: "and" = 1
  Dim4: "happy" = 1
  …

- Naïve Bayes: assumes each word is independent

$$p\left(\vec{x}\right) = p\left(\vec{x}(1),\ldots,\vec{x}\left(D\right)\right) = \prod_{d=1}^{D} p\left(\vec{x}\left(d\right)\right)$$

$$= \prod_{d=1}^{D} \vec{\alpha}\left(d\right)^{\vec{x}\left(d\right)} \left(1 - \vec{\alpha}\left(d\right)\right)^{\left(1-\vec{x}\left(d\right)\right)}$$

- Each 1 dimensional alpha(d) is a Bernoulli parameter
- The whole alpha vector is multivariate Bernoulli

Text: Naïve Bayes

- Maximum likelihood: assume we have several IID vectors
- Have N documents, each a 50,000 dimension binary vector
- Each dimension is a word, set to 1 if word in the document

		$\vec{x}_1$	$\vec{x}_2$	$\vec{x}_3$	$\vec{x}_4$
Dim1:	**"the" =**	**1**	**0**	**1**	**1**
Dim2:	**"hello" =**	**0**	**1**	**0**	**1**
Dim3:	**"and" =**	**1**	**1**	**0**	**1**
Dim4:	**"happy" =**	**1**	**0**	**0**	**1**

- Likelihood $= \prod_{i=1}^{N} p\left(\vec{x}_i \mid \vec{\alpha}\right) = \prod_{i=1}^{N} \prod_{d=1}^{50000} \vec{\alpha}\left(d\right)^{\vec{x}_i(d)} \left(1 - \vec{\alpha}\left(d\right)\right)^{\left(1-\vec{x}_i(d)\right)}$
- Max likelihood solution: for each word d count number of documents it appears in divided by total N documents $\quad \vec{\alpha}\left(d\right) = \frac{N_d}{N}$
- To classify a new document x, build two models α_{+1} α_{-1} & compare $\quad prediction = \arg\max_{y=\{\pm 1\}} p\left(\vec{x} \mid \vec{\alpha}_y\right)$

Multinomial Probability Models

- Multinomial: beyond binary multi-category event (dice)

1	2	3	4	5	6
$\vec{\alpha}(1)$	$\vec{\alpha}(2)$	$\vec{\alpha}(3)$	$\vec{\alpha}(4)$	$\vec{\alpha}(5)$	$\vec{\alpha}(6)$

$$p(x) = \prod_{m=1}^{M} \vec{\alpha}(m)^{\vec{x}(m)} \qquad \sum_m \vec{\alpha}(m) = 1 \qquad \vec{x} \in \mathbb{B}^M \; ; \sum_m \vec{x}(m) = 1$$

$\vec{x}(1)$	$\vec{x}(2)$	$\vec{x}(3)$	$\vec{x}(4)$	$\vec{x}(5)$	$\vec{x}(6)$

- Maximum Likelihood (IID):

$$\sum_{i=1}^{N} \log p(\vec{x}_i \mid \vec{\alpha}) = \sum_{i=1}^{N} \log \prod_{m=1}^{M} \vec{\alpha}(m)^{\vec{x}_i(m)} = \sum_{i=1}^{N} \sum_{m=1}^{M} \vec{x}_i(m) \log\left(\vec{\alpha}(m)\right)$$

- Can't just take gradient, constraint: $\sum_m \vec{\alpha}(m) - 1 = 0$
- Try using Lagrange multipliers:

$$\frac{\partial}{\partial \alpha_q} \sum_{i=1}^{N} \sum_{m=1}^{M} \vec{x}_i(m) \log\left(\vec{\alpha}(m)\right) - \lambda\left(\sum_{m=1}^{M} \vec{\alpha}(m) - 1\right) = 0$$

$$\sum_{i=1}^{N} \left(\vec{x}_i(q) \frac{1}{\vec{\alpha}(q)} \right) - \lambda = 0$$

$$\vec{\alpha}(q) = \tfrac{1}{\lambda} \sum_{i=1}^{N} \vec{x}_i(q)$$

Multinomial Probability (ML)

- Taking the gradient with Lagrangian gives this formula for each q: $\vec{\alpha}(q) = \frac{1}{\lambda}\sum_{i=1}^{N}\vec{x}_i(q)$
- Recall the constraint: $\sum_m \vec{\alpha}(m) - 1 = 0$
- Plug in α's solution: $\sum_m \frac{1}{\lambda}\sum_{i=1}^{N}\vec{x}_i(m) - 1 = 0$
- Gives the lambda: $\lambda = \sum_m \sum_{i=1}^{N}\vec{x}_i(m)$
- Final answer: $$\vec{\alpha}(q) = \frac{\sum_{i=1}^{N}\vec{x}_i(q)}{\sum_m \sum_{i=1}^{N}\vec{x}_i(m)} = \frac{N_q}{N}$$
- Example: Rolling dice
  1,6,2,6,3,6,4,6,5,6

x=1	x=2	x=3	x=4	x=5	x=6
0.1	0.1	0.1	0.1	0.1	0.5

Text: Multinomial Counts

- Multinomial: can also *count many* multi-category events
  Dice: 1,3,1,4,6,1,1 Word Dice: the, dog, jumped, the
- Document i: has W_i=2000 words, each an IID dice roll

$$p(doc_i) = p\left(\vec{x}_i^1, \vec{x}_i^2, \ldots, \vec{x}_i^{W_i}\right) = \prod_{w=1}^{W_i} p\left(\vec{x}_i^w\right) = \prod_{w=1}^{W_i} \prod_{d=1}^{D} \vec{\alpha}\left(d\right)^{\vec{x}_i^w\left(d\right)}$$

- Get count of each time an event occurred

$$p(doc_i) = \prod_{w=1}^{W_i} \prod_{d=1}^{D} \vec{\alpha}\left(d\right)^{\vec{x}_i^w\left(d\right)} = \prod_{d=1}^{D} \vec{\alpha}\left(d\right)^{\sum_{w=1}^{W_i} \vec{x}_i^w\left(d\right)} = \prod_{d=1}^{D} \vec{\alpha}\left(d\right)^{\vec{X}_i\left(d\right)}$$

- BUT: order shouldn't matter when "counting" so multiply by # of possible choosings. Choosing X(1),…X(D) from N

$$\begin{pmatrix} W_i \\ \vec{X}_i\left(1\right), \ldots, \vec{X}_i\left(D\right) \end{pmatrix} = \frac{W_i!}{\prod_{d=1}^{D} \vec{X}_i\left(d\right)!} = \frac{\left(\sum_{d=1}^{D} \vec{X}_i\left(d\right)\right)!}{\prod_{d=1}^{D} \vec{X}_i\left(d\right)!}$$

- Bag-of-words model (only # of words matters, not order):

$$p\left(doc_i\right) = p\left(\vec{X}_i\right) = \frac{\left(\sum_{d=1}^{D} \vec{X}_i\left(d\right)\right)!}{\prod_{d=1}^{D} \vec{X}_i\left(d\right)!} \prod_{d=1}^{D} \vec{\alpha}\left(d\right)^{\vec{X}_i\left(d\right)} \qquad \sum_d \vec{\alpha}\left(d\right) = 1 \quad X \in \mathbb{Z}_+^D$$

Text: Multinomial Counts

$$\in \left\{ \begin{array}{c} religion \\ politics \end{array} \right.$$

- Text classification: bag-of-words model

- Each document is 50,000 dimensional vector
- Each dimension is a word, set to # times word in doc

		X_1	X_2	X_3	X_4
Dim1:	**“the” =**	**9**	**3**	**1**	**0**
Dim2:	**“hello” =**	**0**	**5**	**3**	**0**
Dim3:	**“and” =**	**6**	**2**	**2**	**2**
Dim4:	**“happy” =**	**2**	**5**	**1**	**0**
…					

- Each document is a vector of multinomial counts

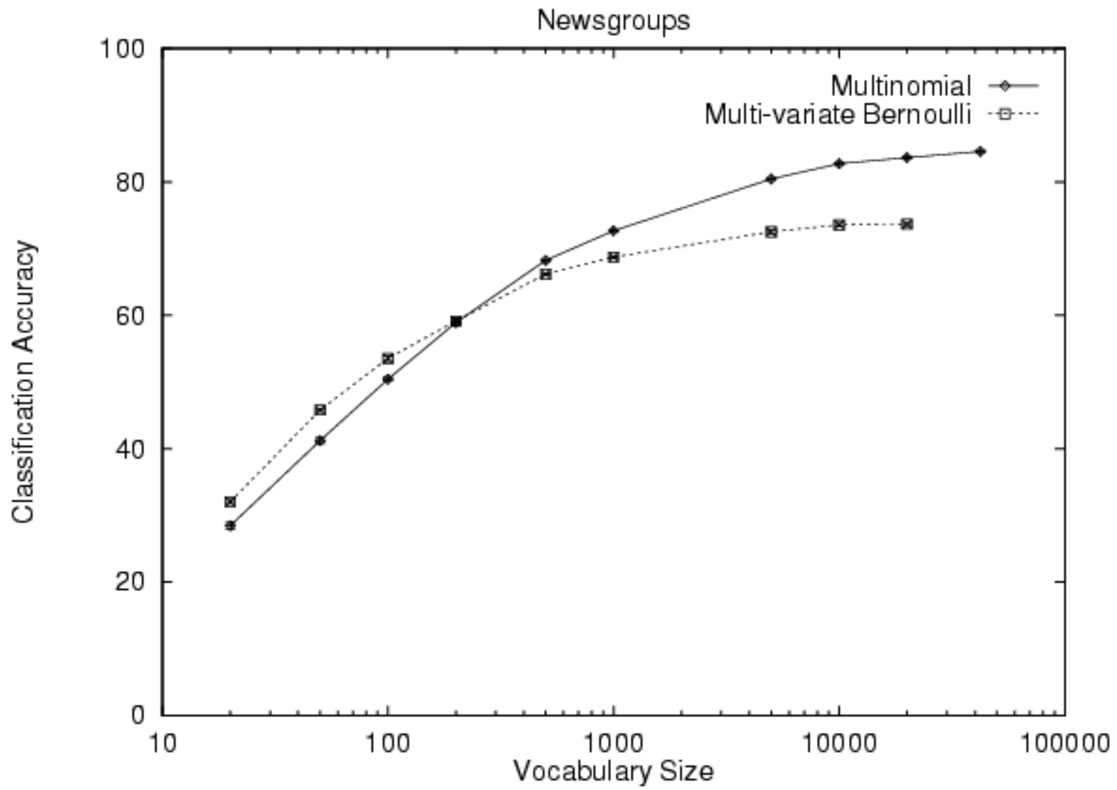
$$p\left(doc_i\right) = p\left(\vec{X}_i\right) = \frac{\left(\sum_{d=1}^{D} \vec{X}_i\left(d\right)\right)!}{\prod_{d=1}^{D} \vec{X}_i\left(d\right)!} \prod_{d=1}^{D} \vec{\alpha}\left(d\right)^{\vec{X}_i\left(d\right)} \qquad \sum_d \vec{\alpha}\left(d\right) = 1 \quad X \in \mathbb{Z}_+^D$$

- Likelihood: $l\left(\vec{\alpha}\right) = \sum_{i=1}^{N} \log p\left(\vec{X}_i\right) = \sum_{i=1}^{N} \log \frac{\left(\sum_{d=1}^{D} \vec{X}_i\left(d\right)\right)!}{\prod_{d=1}^{D} \vec{X}_i\left(d\right)!} \prod_{d=1}^{D} \vec{\alpha}\left(d\right)^{\vec{X}_i\left(d\right)}$

$$\propto \sum_{i=1}^{N} \sum_{d=1}^{D} \vec{X}_i\left(d\right) \log \vec{\alpha}\left(d\right)$$ **same formula as Multinomial ML**

Text: Models Comparison

- For text modeling (McCallum & Nigam ‘98)
  - Bernoulli better for small vocabulary
  - Multinomial better for large vocabulary



Text: Newsgroup Recognition

- Model text from 12 newsgroups each with a multinomial
- Use speech rec to make a document of past 200 words
- IBM ViaVoice speech recognizer only obtains 50% accuracy

- But can tell topic of the newsgroup which best aligns with conversation at 95% accuracy

- Probabilities for each topic in real-time