Machine Learning

4771

Instructor: Tony Jebara

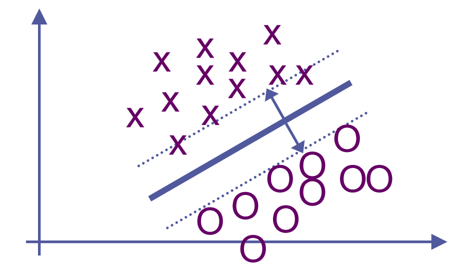
Topic 7

- Unsupervised Learning
- Statistical Perspective
- Probability Models
- Discrete & Continuous: Gaussian, Bernoulli, Multinomial
- Maximum Likelihood → Logistic Regression
- Conditioning, Marginalizing, Bayes Rule, Expectations
- Classification, Regression, Detection
- Dependence/Independence
- Maximum Likelihood → Naïve Bayes

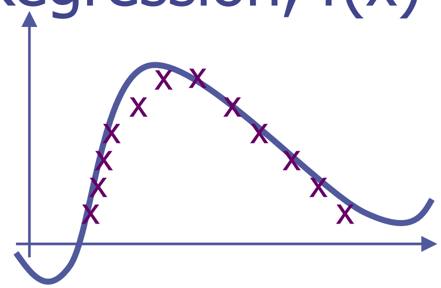
Tony Jebara, Columbia University

Unsupervised Learning

Classification

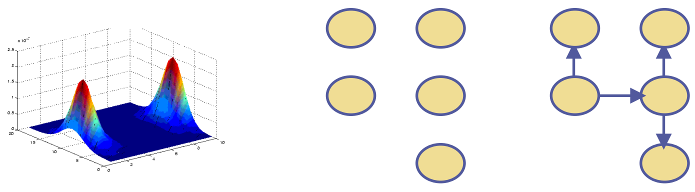


Regression, f(x)=y

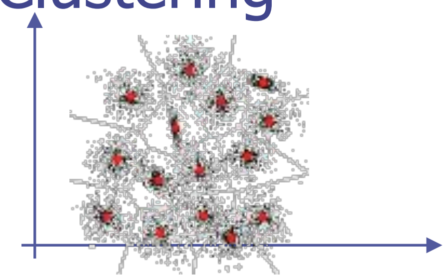


Supervised

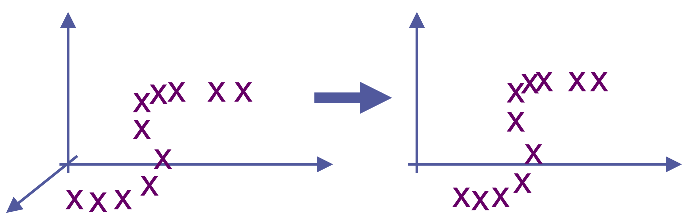
Density/Structure Estimation



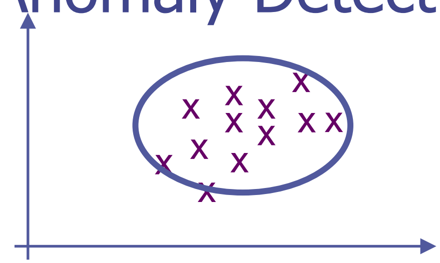
Clustering



Feature Selection



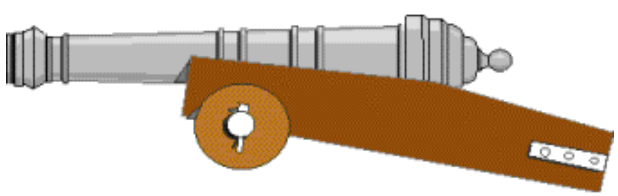
Anomaly Detection



Unsupervised
(can help
supervised)

Statistical Perspective

- Several problems with framework so far:
  - Only have input-output approaches (SVM, Neural Net)
  - Pulled non-linear squashing functions out of a hat
  - Pulled loss functions (squared error, etc.) out of a hat
- Better approach for classification?
- What if we have multi-class classification?
- What if other problems, i.e. unobserved values of x,y,etc…
- Also, what if we don't have a true function?
- Example of Projectile Cannon (c.f. Distal Learning)

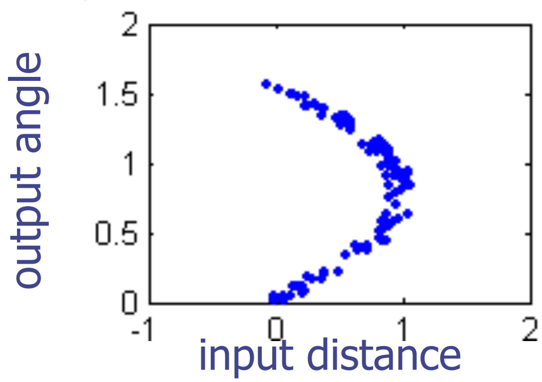


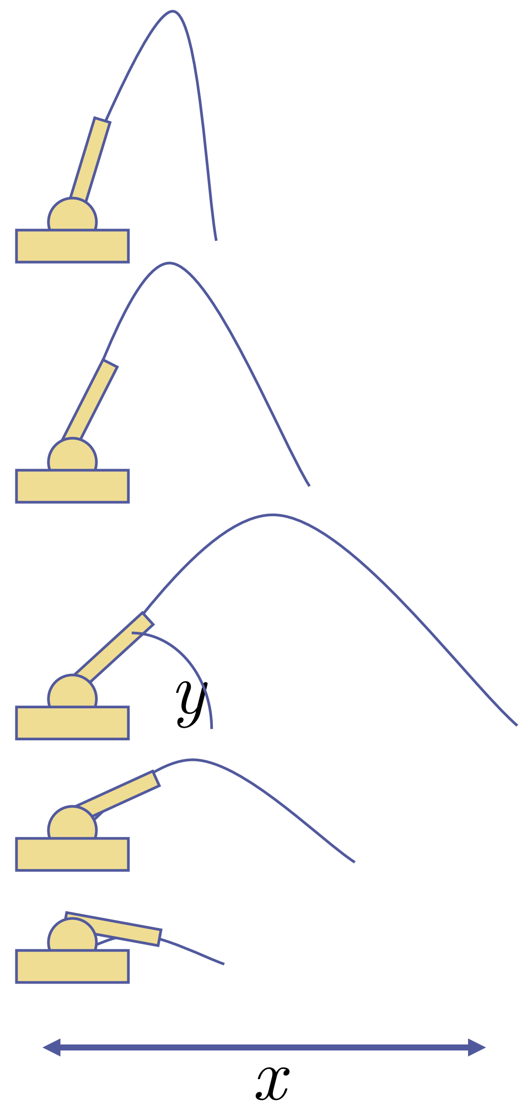
- Would like to train a regression function to control a cannon's angle of fire (y) given target distance (x)

Statistical Perspective

- Example of Projectile Cannon
  (45 degree problem)
  x = input target distance
  y = output cannon angle

$$x = \frac{v(0)^2}{g}\sin(2y) + noise$$



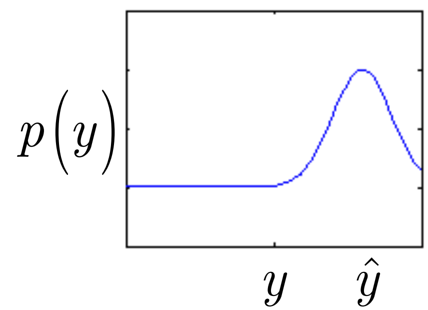


- What does least squares do?
- Conditional statistical models
  address this problem…

Probability Models

- Instead of deterministic functions, output is a probability
- Previously: our output was a scalar $\hat{y} = f\left(x\right) = \theta^T x + b$
- Now: our output is a probability $p\left(y\right)$

  e.g. a probability bump:

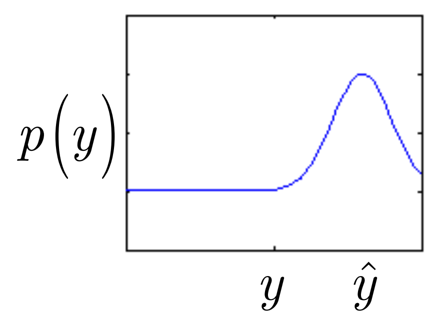


- $p\left(y\right)$ subsumes or is a superset of $\hat{y}$
- Why is this representation for our answer more general?

Probability Models

- Instead of deterministic functions, output is a probability
- Previously: our output was a scalar $\hat{y} = f(x) = \theta^T x + b$
- Now: our output is a probability $p(y)$

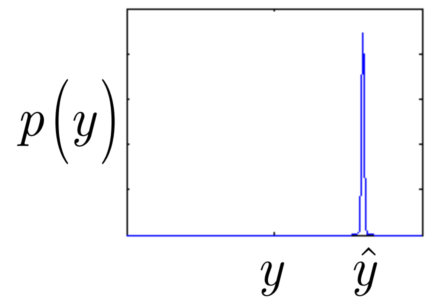
  e.g. a probability bump:



- $p(y)$ subsumes or is a superset of $\hat{y}$
- Why is this representation for our answer more general?

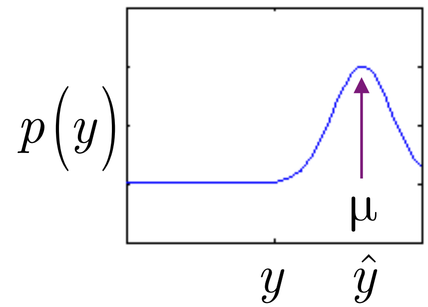
  → A deterministic answer $\hat{y}$ with complete confidence is like putting a probability $p(y)$ where all the mass is at $\hat{y}$!

$$\hat{y} \Leftrightarrow p(y) = \delta(y - \hat{y})$$



Probability Models

- Now: our output is a probability density function (pdf) $p(y)$
- Probability Model: a family of pdf's with adjustable parameters which lets us select one of many $p(y) \to p(y \mid \Theta)$
- E.g.: 1-dim Gaussian distribution 'given' 'mean' parameter μ:

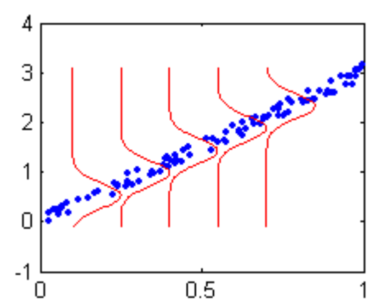


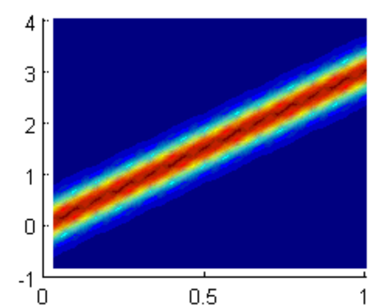
$$p(y \mid \mu) = N(y \mid \mu) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y-\mu)^2}$$

- Want mean centered on f(x)'s value $p(y) = N(y \mid f(x))$
- Now, linear regression is:

$$N(y \mid f(x)) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y-f(x))^2}$$

$$= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y-\theta^T x-b)^2}$$





Probability Models

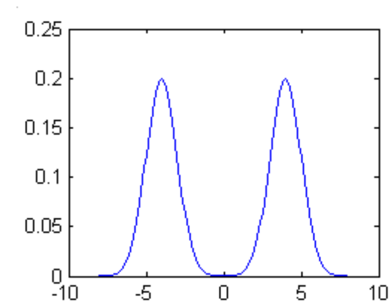
- To fit to data, we typically "maximize likelihood" of the probability model
- Log-likelihood = objective function (i.e. negative of cost) for probability models which we want to maximize
- Define (conditional) likelihood as $L(\Theta) = \prod_{i=1}^{N} p(y_i \mid x_i)$

  or log-Likelihood as $l(\Theta) = \log\left(L(\Theta)\right) = \sum_{i=1}^{N} \log p(y_i \mid x_i)$
- For Gaussian p(y|x), maximum likelihood is least squares!

$$\sum_{i=1}^{N} \log p(y_i \mid x_i) = \sum_{i=1}^{N} \log N\left(y_i \mid f(x_i)\right) = \sum_{i=1}^{N} \log \tfrac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(y_i - f(x_i)\right)^2}$$

$$= -N \log\left(\sqrt{2\pi}\right) - \sum_{i=1}^{N} \tfrac{1}{2}\left(y_i - f(x_i)\right)^2$$

Probability Models



- Can extend probability model to 2 bumps:

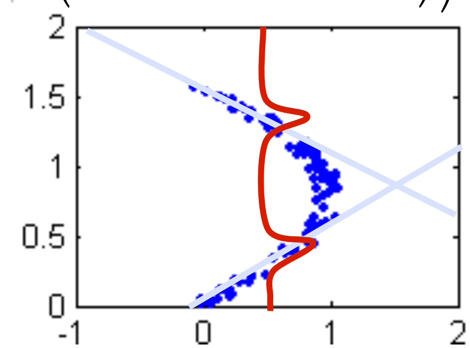
$$p\left(y \mid \Theta\right) = \tfrac{1}{2} N\left(y \mid \mu_1\right) + \tfrac{1}{2} N\left(y \mid \mu_2\right)$$

- Each mean can be a linear regression fn.

$$\begin{aligned} p\left(y \mid x, \Theta\right) &= \tfrac{1}{2} N\left(y \mid f_1\left(x\right)\right) + \tfrac{1}{2} N\left(y \mid f_2\left(x\right)\right) \\ &= \tfrac{1}{2} N\left(y \mid \theta_1^T x + b_1\right) + \tfrac{1}{2} N\left(y \mid \theta_2^T x + b_2\right) \end{aligned}$$

- Therefore the (conditional) log-likelihood to maximize is:

$$l\left(\Theta\right) = \sum_{i=1}^{N} \log\left(\tfrac{1}{2} N\left(y_i \mid \theta_1^T x_i + b_1\right) + \tfrac{1}{2} N\left(y_i \mid \theta_2^T x_i + b_2\right)\right)$$



- Maximize $l(\theta)$ using gradient ascent
- Nicely handles the "cannon firing" data

Probability Models

- Now classification: can also go beyond deterministic!
- Previously: wanted output to be binary $\hat{y} = \{0,1\}$
- Now: our output is a probability $p(y)$

  e.g. a probability table:

y=0	y=1
0.73	0.27

α

- This subsumes or is a superset again…
- Consider probability over binary events (coin flips!):

  e.g. Bernoulli distribution (i.e 1x2 probability table)
  with parameter α

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

- Linear classification can be done by setting α equal to f(x):

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

Probability Models

- Now linear classification is:

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

- Log-likelihood is (negative of cost function):

$$\begin{aligned}\sum_{i=1}^{N} \log p\left(y_i \mid x_i\right) &= \sum_{i=1}^{N} \log f\left(x_i\right)^{y_i}\left(1 - f\left(x_i\right)\right)^{1-y_i} \\ &= \sum_{i=1}^{N} y_i \log f\left(x_i\right) + \left(1 - y_i\right)\log\left(1 - f\left(x_i\right)\right) \\ &= \sum_{i \in class1} \log f\left(x_i\right) + \sum_{i \in class0} \log\left(1 - f\left(x_i\right)\right)\end{aligned}$$

- But, need a squashing function since f(x) in [0,1]
- Use sigmoid or logistic again…

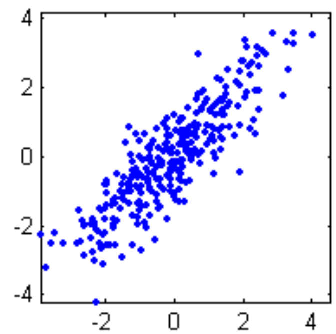
$$f\left(x\right) = sigmoid\left(\theta^T x + b\right) \in [0,1]$$

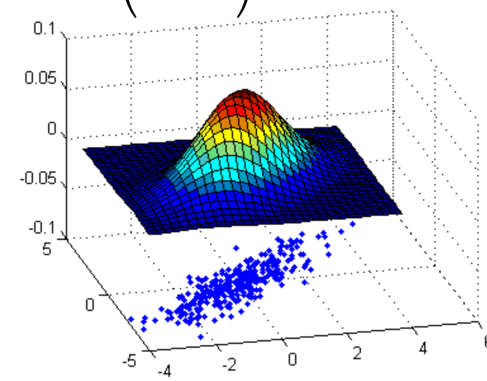
- Called logistic regression → *new loss function*
- Do gradient descent, similar to logistic output neural net!
- Can also handle multi-layer f(x) and do backprop again!

Generative Probability Models

- Idea: Extend probability to describe *both* X and Y
- Find probability density function over both: $p\left(x, y\right)$

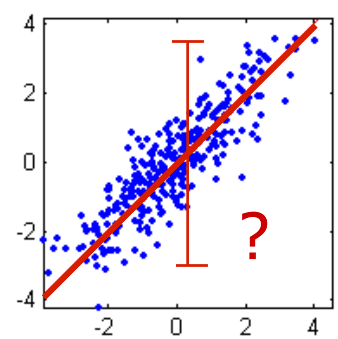
E.g. *describe* data with Multi-Dim. Gaussian (later…)





- Called a 'Generative Model' because we can use it to synthesize or re-generate data similar to the training data we learned from

- Regression models & classification boundaries
  are not as flexible
  don't keep info about X
  don't model noise/uncertainty



Properties of PDFs

- Let's review some basics of probability theory
- First, pdf is a function, multiple inputs, one output:

$$p\left(x_1,\ldots,x_n\right) \qquad p\left(x_1 = 0.3,\ldots,x_n = 1\right) = 0.2$$

- Function's output is always non-negative:

$$p\left(x_1,\ldots,x_n\right) \geq 0$$

- Can have discrete or continuous or both inputs:

$$p\left(x_1 = 1, x_2 = 0, x_3 = 0, x_4 = 3.1415\right)$$

- Summing over the domain of all inputs gives unity:

$$\int_{y=-\infty}^{\infty}\int_{x=-\infty}^{\infty} p\left(x,y\right)dxdy = 1 \qquad \sum_y \sum_x p\left(x,y\right) = 1$$

0.4	0.1
0.3	0.2

Continuous→integral, Discrete→sum

Properties of PDFs

- Marginalizing: integrate/sum out a variable leaves a marginal distribution over the remaining ones…

$$\sum_y p(x,y) = p(x)$$

- Conditioning: if a variable 'y' is 'given' we get a conditional distribution over the remaining ones…

$$p(x \mid y) = \frac{p(x,y)}{p(y)}$$

- Bayes Rule: mathematically just redo conditioning but has a deeper meaning (1764)… if we have X being data and θ being a model

$$\underbrace{p(\theta \mid \mathcal{X})}_{\textbf{posterior}} = \frac{\overbrace{p(\mathcal{X} \mid \theta)}^{\textbf{likelihood}}\,\overbrace{p(\theta)}^{\textbf{prior}}}{\underbrace{p(\mathcal{X})}_{\textbf{evidence}}}$$

Properties of PDFs

- **Expectation:** can use pdf p(x) to compute averages and expected values for quantities, denoted by:

$$E_{p(x)}\left\{f\left(x\right)\right\} = \int_x p\left(x\right) f\left(x\right) dx \quad or \quad = \sum_x p\left(x\right) f\left(x\right)$$

- **Properties:**

$$E\left\{cf\left(x\right)\right\} = cE\left\{f\left(x\right)\right\}$$

$$E\left\{f\left(x\right)+c\right\} = E\left\{f\left(x\right)\right\}+c$$

$$E\left\{E\left\{f\left(x\right)\right\}\right\} = E\left\{f\left(x\right)\right\}$$

example: speeding ticket

Fine=0$	Fine=20$
0.8	0.2

expected cost of speeding?
f(x=0)=0, f(x=1)=20
p(x=0)=0.8, p(x=1)=0.2

- **Mean:** expected value for x

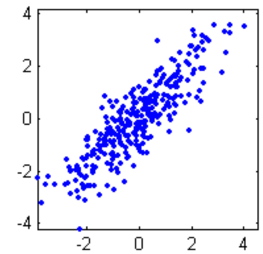
$$E_{p(x)}\left\{x\right\} = \int_{-\infty}^{\infty} p\left(x\right) x\, dx$$

- **Variance:** expected value of (x-mean)^2, how much x varies

$$Var\left\{x\right\} = E\left\{\left(x - E\left\{x\right\}\right)^2\right\} = E\left\{x^2 - 2xE\left\{x\right\} + E\left\{x\right\}^2\right\}$$

$$= E\left\{x^2\right\} - 2E\left\{x\right\}E\left\{x\right\} + E\left\{x\right\}^2 = E\left\{x^2\right\} - E\left\{x\right\}^2$$

Properties of PDFs



- Covariance: how strongly x and y vary together

$$Cov\left\{x,y\right\} = E\left\{\left(x - E\left\{x\right\}\right)\left(y - E\left\{y\right\}\right)\right\} = E\left\{xy\right\} - E\left\{x\right\}E\left\{y\right\}$$

- Conditional Expectation: $E\left\{y \mid x\right\} = \int_y p\left(y \mid x\right) y \, dy$

$$E\left\{E\left\{y \mid x\right\}\right\} = \int_x p\left(x\right) \int_y p\left(y \mid x\right) y \, dy \, dx = E\left\{y\right\}$$

- Sample Expectation: If we don't have pdf p(x,y) can approximate expectations using samples of data

$$E_{p\left(x\right)}\left\{f\left(x\right)\right\} \simeq \frac{1}{N}\sum_{i=1}^{N} f\left(x_i\right)$$

- Sample Mean: $E\left\{x\right\} \simeq \overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$
- Sample Var: $E\left\{\left(x - E\left(x\right)\right)^2\right\} \simeq \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \overline{x}\right)^2$
- Sample Cov: $E\left\{\left(x - E\left(x\right)\right)\left(y - E\left(y\right)\right)\right\} \simeq \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)$

More Properties of PDFs

- **Independence:** probabilities of independent variables multiply. Denote with the following notation:

$$x \perp\!\!\!\perp y \quad \rightarrow \quad p(x,y) = p(x)\,p(y)$$
$$x \perp\!\!\!\perp y \quad \rightarrow \quad p(x \mid y) = p(x)$$

also note in this case:

$$E_{p(x,y)}\{xy\} = \int_x \int_y p(x)\,p(y)\,xy\,dx\,dy$$
$$= \int_x p(x)\,x\,dx \int_y p(y)\,y\,dy = E_{p(x)}\{x\}\,E_{p(y)}\{y\}$$

- **Conditional independence:** when two variables become independent only if another is observed

$$x \perp\!\!\!\perp y \mid z \rightarrow \quad p(x \mid y,z) = p(x \mid z)$$
$$x \perp\!\!\!\perp y \mid z \rightarrow \quad p(x \mid y) \neq p(x)$$

The IID Assumption

- Most of the time, we will assume that a dataset
  independent and identically distributed (IID)

- In many real situations, data is generated by some
  black box phenomenon in an arbitrary order.
- Assume we are given a dataset:

$$\mathcal{X} = \left\{ x_1, \ldots, x_N \right\}$$

"Independent" means that (given the model θ) the probability of our data multiplies:

$$p\left(x_1, \ldots, x_N \mid \Theta\right) = \prod_{i=1}^{N} p_i\left(x_i \mid \Theta\right)$$

"Identically distributed" means that each marginal probability is the same for each data point

$$p\left(x_1, \ldots, x_N \mid \Theta\right) = \prod_{i=1}^{N} p_i\left(x_i \mid \Theta\right) = \prod_{i=1}^{N} p\left(x_i \mid \Theta\right)$$

The IID Assumption

- Bayes rule says likelihood is probability of data given model

$$\underbrace{p\left(\theta \mid \mathcal{X}\right)}_{\text{posterior}} = \frac{\overbrace{p\left(\mathcal{X} \mid \theta\right)}^{\text{likelihood}} \overbrace{p\left(\theta\right)}^{\text{prior}}}{\underbrace{p\left(\mathcal{X}\right)}_{\text{evidence}}}$$

- The likelihood of $\mathcal{X} = \left\{x_1,\ldots,x_N\right\}$ under IID assumptions is:

$$p\left(\mathcal{X} \mid \Theta\right) = p\left(x_1,\ldots,x_N \mid \Theta\right) = \prod_{i=1}^{N} p_i\left(x_i \mid \Theta\right) = \prod_{i=1}^{N} p\left(x_i \mid \Theta\right)$$

- Learn joint distribution $p\left(x \mid \Theta\right)$ by maximum likelihood:

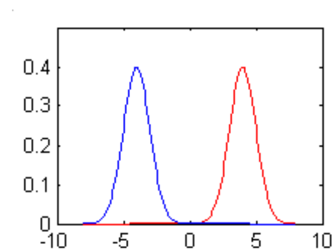
$$\Theta^* = \arg\max_\Theta \prod_{i=1}^{N} p\left(x_i \mid \Theta\right) = \arg\max_\Theta \sum_{i=1}^{N} \log p\left(x_i \mid \Theta\right)$$

- Learn conditional $p\left(y \mid x,\Theta\right)$ by max conditional likelihood:

$$\Theta^* = \arg\max_\Theta \prod_{i=1}^{N} p\left(y_i \mid x_i,\Theta\right) = \arg\max_\Theta \sum_{i=1}^{N} \log p\left(y_i \mid x_i,\Theta\right)$$

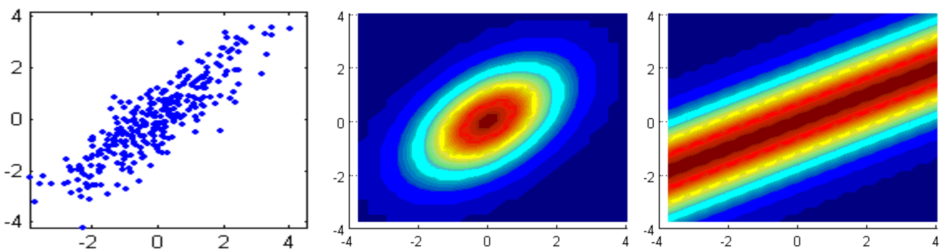
Uses of PDFs

- Classification: have p(x,y) and given x. Asked for discrete y output, give most probable one



$$p(x,y) \rightarrow p(y \mid x) \rightarrow \hat{y} = \arg\max_m p(y = m \mid x)$$

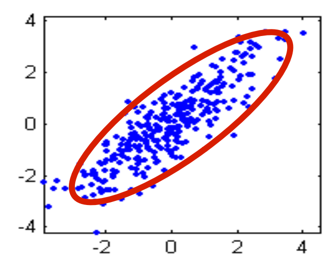
- Regression: have p(x,y) and given x. Asked for a scalar y output, give most probable or expected one



$$\hat{y} = \begin{cases} \arg\max_y p(y \mid x) \\ E_{p(y \mid x)}\{y\} \end{cases}$$

$$\{(x_i, y_i)\} \rightarrow p(x,y) \rightarrow p(y \mid x)$$

- Anomaly Detection: if have p(x,y) and given both x,y. Asked if it is similar → threshold



$$p(x,y) \geq threshold \rightarrow \{normal, anomaly\}$$

Machine Learning

4771

Instructor: Tony Jebara

Topic 8

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

Bernoulli Probability Models

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

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

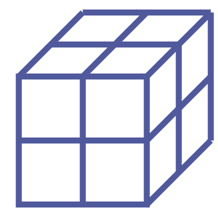
$x=0$	$x=1$
0.73	0.27

- 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?

Bernoulli Probability Models

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

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

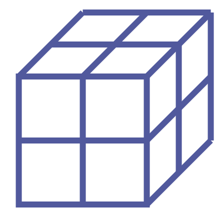
x=0	x=1
0.73	0.27

- 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

Bernoulli Probability Models

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

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

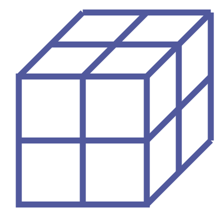
x=0	x=1
0.73	0.27

- 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 Maximum Likelihood

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

$$\frac{\partial}{\partial\alpha} \sum_{i=1}^{N} \log \alpha^{x_i}(1-\alpha)^{1-x_i} = 0$$

$$\frac{\partial}{\partial\alpha} \sum_{i=1}^{N} x_i \log \alpha + (1-x_i)\log(1-\alpha) = 0$$

$$\frac{\partial}{\partial\alpha} \sum_{i \in class1} \log \alpha + \sum_{i \in class0} \log(1-\alpha) = 0$$

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

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

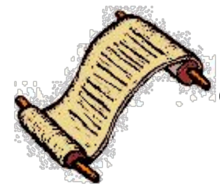
$$N_1(1-\alpha) - N_0\alpha = 0$$

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

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

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

Text Modeling via Naïve Bayes



$$\in \left\{ \begin{array}{c} email \\ spam \end{array} \right.$$

- Naïve Bayes: the simplest model of text
- 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 Modeling via 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 $\vec{\alpha}\left(d\right) = \frac{N_d}{N}$
- To classify a new document x, build two models α_{+1} α_{-1} & compare $prediction = \arg\max_{y \in \{\pm 1\}} p\left(\vec{x} \mid \vec{\alpha}_y\right)$

Categorical Probability Models

- Categorical: a distribution over a single multi-category event

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

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

- Encode events as binary indicator vectors

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

- Related to the more general *multinomial* distribution
- Find α using Maximum Likelihood (with IID assumption):

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

- Can't just take gradient over α, use sum= 1 constraint:
- Insert constraint using Lagrange multipliers

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

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

Categorical Maximum Likelihood

- 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

Multinomial Probability Model

- The multinomial is a categorical over *counts* of events

  Dice: 1,3,1,4,6,1,1 Word Dice: the, dog, jumped, the

- Say document i has W_i=2000 words, each an IID dice roll

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

- Get count of each time an event occurred

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

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

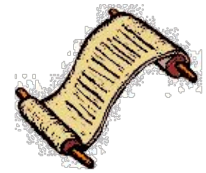
$$\begin{pmatrix} W_i \\ \vec{X}_i(1), \ldots, \vec{X}_i(D) \end{pmatrix} = \frac{W_i!}{\prod_{d=1}^{D} \vec{X}_i(d)!} = \frac{\left(\sum_{d=1}^{D} \vec{X}_i(d)\right)!}{\prod_{d=1}^{D} \vec{X}_i(d)!}$$

- Multinomial: over discrete integer vectors X summing to W

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

Text Modeling via Multinomial

- Also known as the bag-of-words model



$$\in \left\{ \begin{array}{c} email \\ spam \end{array} \right.$$

- 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

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

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

$$= \sum_{i=1}^{N} \sum_{d=1}^{D} \vec{X}_i\left(d\right) \log \vec{\alpha}\left(d\right) + const$$

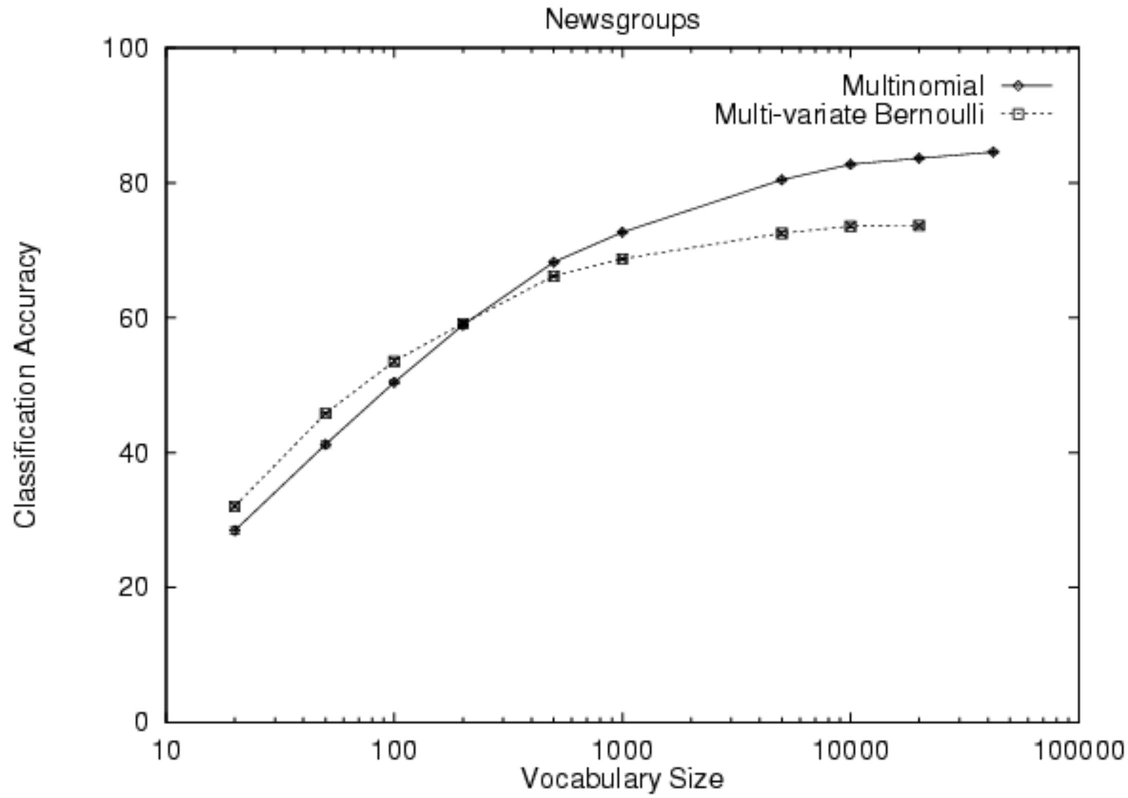
- Find α just like the multinomial maximum likelihood formula!

Text Modeling Experiments

•For text modeling (McCallum & Nigam '98)

Bernoulli better for small vocabulary

Multinomial better for large vocabulary



Machine Learning

4771

Instructor: Tony Jebara

Topic 9

- Continuous Probability Models
- Gaussian Distribution
- Maximum Likelihood Gaussian
- Sampling from a Gaussian

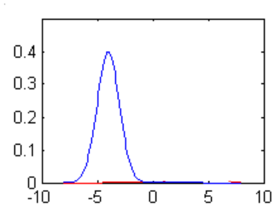
Continuous Probability Models

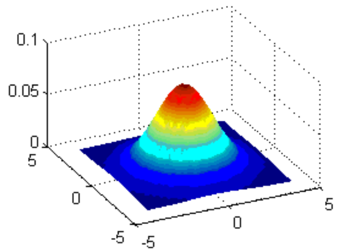
- Probabilities can have both discrete & continuous variables
- We will discuss:
  1) discrete probability tables

x=T	x=H
0.4	0.6

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

  2) continuous probability distributions



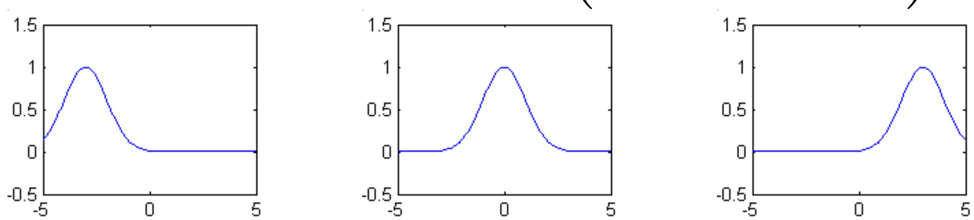


- Most popular continuous distribution = Gaussian

Gaussian Distribution

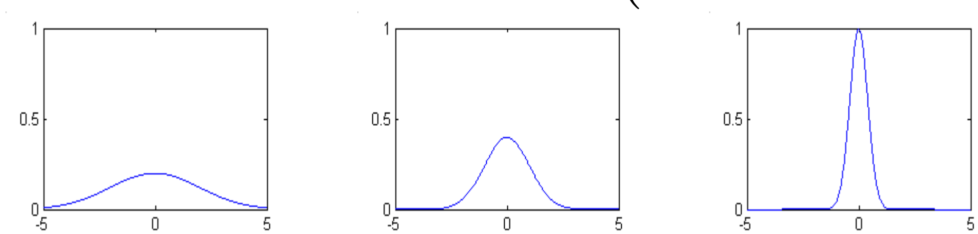
- Recall 1-dimensional Gaussian with mean parameter μ translates Gaussian left & right

$$p\left(x \mid \mu\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(x-\mu\right)^2\right)$$



- Can also have variance parameter σ^2 widens or narrows the Gaussian

$$p\left(x \mid \mu, \sigma\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(x-\mu\right)^2\right)$$



Note: $\int\limits_{x=-\infty}^{\infty} p\left(x\right) dx = 1$

Multivariate Gaussian

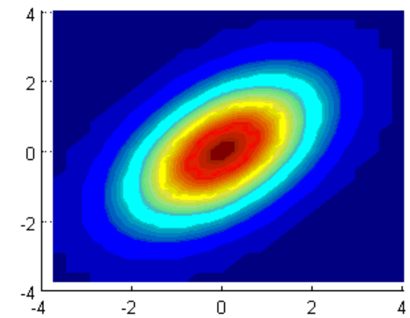
•Gaussian can extend to D-dimensions

•Gaussian mean parameter μ vector, it translates the bump
•Covariance matrix Σ stretches and rotates bump

$$p\left(\vec{x} \mid \vec{\mu}, \Sigma\right) = \frac{1}{\left(2\pi\right)^{D/2}\sqrt{\left|\Sigma\right|}} \exp\left(-\frac{1}{2}\left(\vec{x}-\vec{\mu}\right)^T \Sigma^{-1}\left(\vec{x}-\vec{\mu}\right)\right)$$

$$\vec{x} \in \mathbb{R}^D, \vec{\mu} \in \mathbb{R}^D, \Sigma \in \mathbb{R}^{D\times D}$$

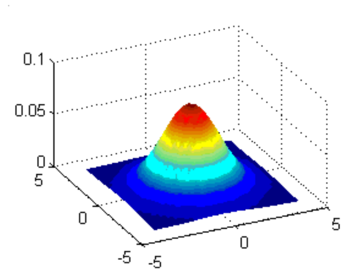
•Mean is any real vector
•Max and expectation = μ
•Variance parameter is now Σ matrix
•Covariance matrix is positive definite
•Covariance matrix is symmetric
•Need matrix inverse (inv)
•Need matrix determinant (det)
•Need matrix trace operator (trace)

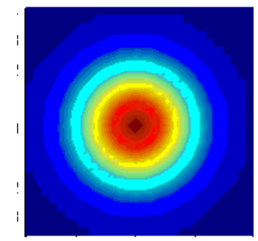


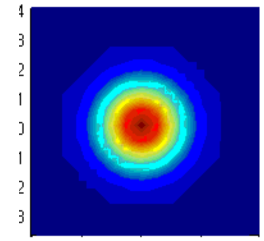
Multivariate Gaussian

- Spherical:

$$\Sigma = \sigma^2 I = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$$

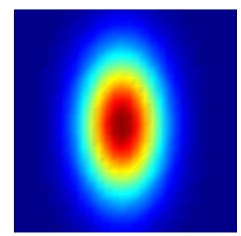


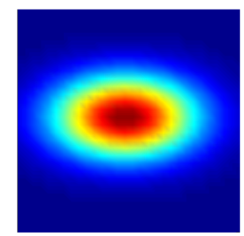




- Diagonal Covariance:
  dimensions of x are independent
  product of multiple 1d Gaussians

$$p\left(\vec{x} \mid \vec{\mu}, \Sigma\right) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi}\vec{\sigma}\left(d\right)} \exp\left(-\frac{\left(\vec{x}\left(d\right)-\vec{\mu}\left(d\right)\right)^2}{2\vec{\sigma}\left(d\right)^2}\right)$$





$$\Sigma = \begin{bmatrix} \vec{\sigma}\left(1\right)^2 & 0 & 0 & 0 \\ 0 & \vec{\sigma}\left(2\right)^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \vec{\sigma}\left(D\right)^2 \end{bmatrix}$$

Max Likelihood Gaussian

- Have IID samples as vectors i=1..N: $\mathcal{X} = \left\{ \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N \right\}$
- How do we recover the mean and covariance parameters?
- Standard approach: Maximum Likelihood (IID)
- Maximize probability of data given model (likelihood)

$$\begin{aligned} p\left(\mathcal{X} \mid \theta\right) &= p\left(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N \mid \theta\right) \\ &= \prod_{i=1}^{N} p\left(\vec{x}_i \mid \vec{\mu}_i, \Sigma_i\right) \quad \textit{independent Gaussian samples} \\ &= \prod_{i=1}^{N} p\left(\vec{x}_i \mid \vec{\mu}, \Sigma\right) \quad \textit{identically distributed} \end{aligned}$$

- Instead, work with maximum of log-likelihood

$$\sum_{i=1}^{N} \log p\left(\vec{x}_i \mid \vec{\mu}, \Sigma\right) = \sum_{i=1}^{N} \log \frac{1}{\left(2\pi\right)^{D/2} \sqrt{\left|\Sigma\right|}} \exp\left(-\frac{1}{2}\left(\vec{x}_i - \vec{\mu}\right)^T \Sigma^{-1} \left(\vec{x}_i - \vec{\mu}\right)\right)$$

Max Likelihood Gaussian

- Max over μ $$\frac{\partial}{\partial \mu}\left(\sum_{i=1}^{N} \log \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}\left(\vec{x}_i - \vec{\mu}\right)^T \Sigma^{-1}\left(\vec{x}_i - \vec{\mu}\right)\right)\right) = 0$$

$$\frac{\partial}{\partial \mu}\left(\sum_{i=1}^{N} -\frac{D}{2}\log 2\pi - \frac{1}{2}\log\left|\Sigma\right| - \frac{1}{2}\left(\vec{x}_i - \vec{\mu}\right)^T \Sigma^{-1}\left(\vec{x}_i - \vec{\mu}\right)\right) = 0$$

$$\boxed{\frac{\partial \vec{x}^T\vec{x}}{\partial \vec{x}} = 2\vec{x}^T}$$

$$\sum_{i=1}^{N}\left(\vec{x}_i - \vec{\mu}\right)^T \Sigma^{-1} = \vec{0}$$

see Jordan Ch. 12, get sample mean... $$\vec{\mu} = \frac{1}{N}\sum_{i=1}^{N}\vec{x}_i$$

- For Σ need Trace operator: $$tr\left(A\right) = tr\left(A^T\right) = \sum_{d=1}^{D} A\left(d,d\right)$$

and several properties:

$$tr\left(AB\right) = tr\left(BA\right)$$

$$tr\left(BAB^{-1}\right) = tr\left(A\right)$$

$$tr\left(\vec{x}\vec{x}^T A\right) = tr\left(\vec{x}^T A\vec{x}\right) = \vec{x}^T A\vec{x}$$

Max Likelihood Gaussian

- Likelihood rewritten in trace notation:

$$l = \sum_{i=1}^{N} -\tfrac{D}{2}\log 2\pi - \tfrac{1}{2}\log\left|\Sigma\right| - \tfrac{1}{2}\left(\vec{x}_i - \vec{\mu}\right)^T \Sigma^{-1}\left(\vec{x}_i - \vec{\mu}\right)$$

$$= -\tfrac{ND}{2}\log 2\pi + \tfrac{N}{2}\log\left|\Sigma^{-1}\right| - \tfrac{1}{2}\sum_{i=1}^{N} tr\left[\left(\vec{x}_i - \vec{\mu}\right)^T \Sigma^{-1}\left(\vec{x}_i - \vec{\mu}\right)\right]$$

$$= -\tfrac{ND}{2}\log 2\pi + \tfrac{N}{2}\log\left|\Sigma^{-1}\right| - \tfrac{1}{2}\sum_{i=1}^{N} tr\left[\left(\vec{x}_i - \vec{\mu}\right)\left(\vec{x}_i - \vec{\mu}\right)^T \Sigma^{-1}\right]$$

$$= -\tfrac{ND}{2}\log 2\pi + \tfrac{N}{2}\log\left|A\right| - \tfrac{1}{2}\sum_{i=1}^{N} tr\left[\left(\vec{x}_i - \vec{\mu}\right)\left(\vec{x}_i - \vec{\mu}\right)^T A\right]$$

$$\frac{\partial \log\left|A\right|}{\partial A} = \left(A^{-1}\right)^T \qquad \frac{\partial tr\left[BA\right]}{\partial A} = B^T$$

- Max over $A=\Sigma^{-1}$ use properties:

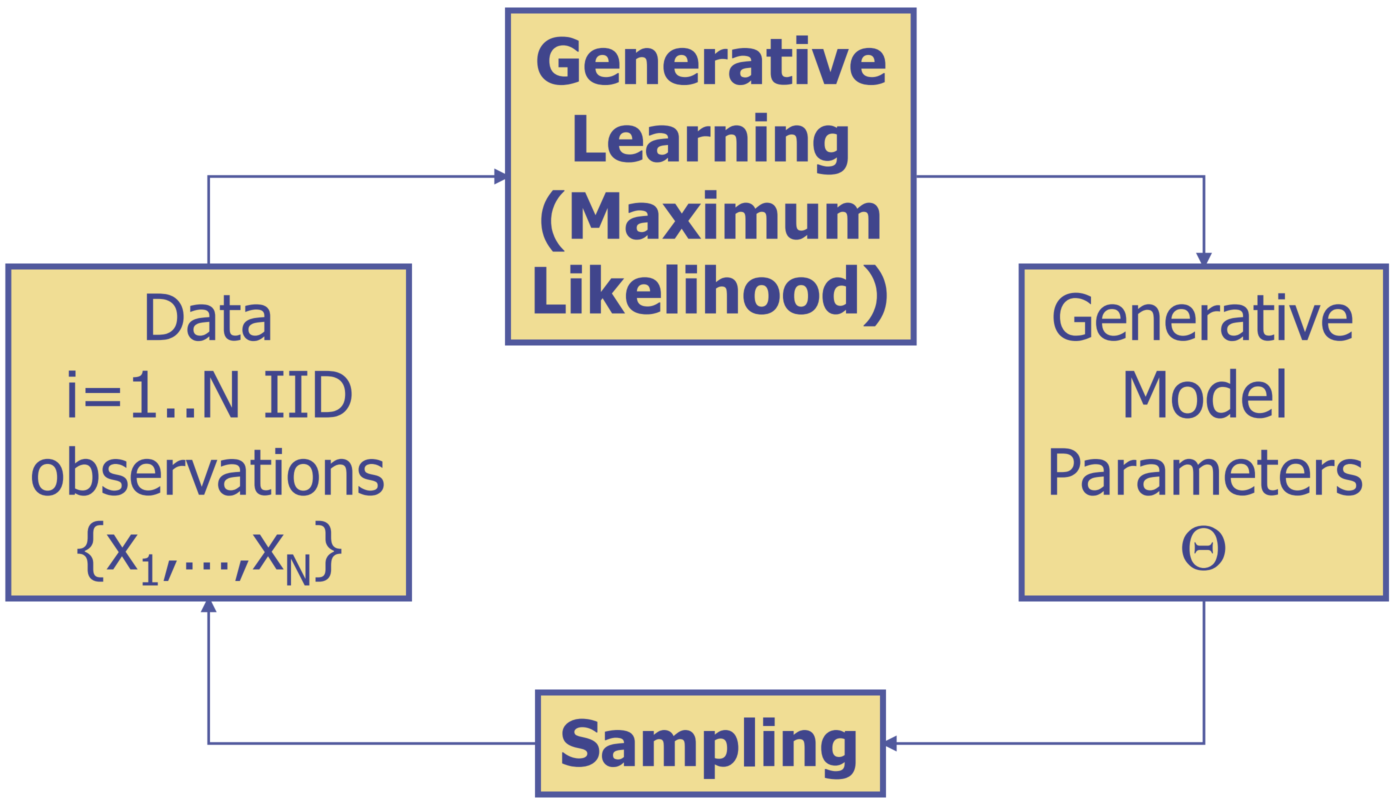
$$\frac{\partial l}{\partial A} = -0 + \tfrac{N}{2}\left(A^{-1}\right)^T - \tfrac{1}{2}\sum_{i=1}^{N}\left[\left(\vec{x}_i - \vec{\mu}\right)\left(\vec{x}_i - \vec{\mu}\right)^T\right]^T$$

$$= \tfrac{N}{2}\Sigma - \tfrac{1}{2}\sum_{i=1}^{N}\left(\vec{x}_i - \vec{\mu}\right)\left(\vec{x}_i - \vec{\mu}\right)^T$$

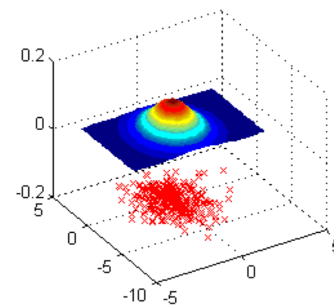
- Get sample covariance:

$$\frac{\partial l}{\partial A} = 0 \;\rightarrow\; \Sigma = \tfrac{1}{N}\sum_{i=1}^{N}\left(\vec{x}_i - \vec{\mu}\right)\left(\vec{x}_i - \vec{\mu}\right)^T$$

Sampling & Max Likelihood

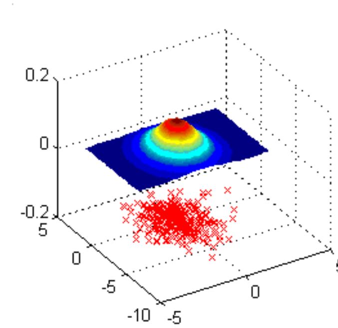


Sampling from a Gaussian



•Fit Gaussian to data, how is this Generative?

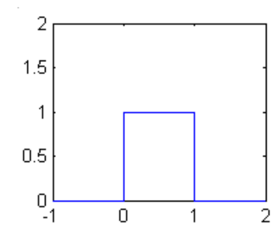
Sampling from a Gaussian



- Fit Gaussian to data, how is this Generative?
- Sampling! Generating discrete data easy:

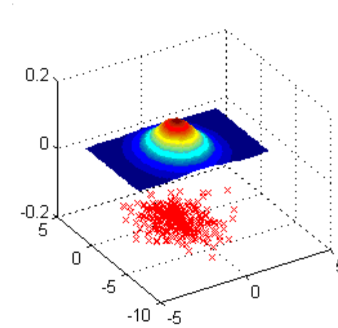
0.73	0.1	0.17

- Assume we can do uniform sampling:
  i.e. rand between (0,1)
  if 0.00 <= rand < 0.73 get A
  if 0.73 <= rand < 0.83 get B
  if 0.83 <= rand < 1.00 get C



- What are we doing?

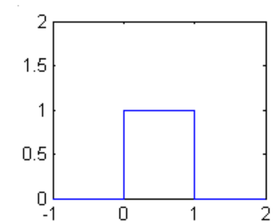
Sampling from a Gaussian



- Fit Gaussian to data, how is this Generative?
- Sampling! Generating discrete data easy:

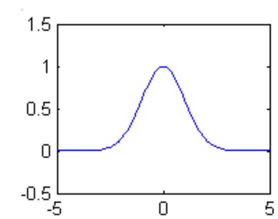
0.73	0.1	0.17

- Assume we can do uniform sampling:
  i.e. rand between (0,1)
  if 0.00 <= rand < 0.73 get A
  if 0.73 <= rand < 0.83 get B
  if 0.83 <= rand < 1.00 get C

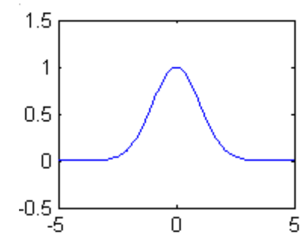


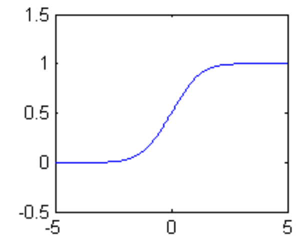
0.73	0.83	1.00

- What are we doing?
  Sum up the Probability Density Function (PDF)
  to get Cumulative Density Function (CDF)
- For 1d Gaussian, Integrate Probability Density Function get Cumulative Density Function Integral is like summing many discrete bars



Sampling from a Gaussian





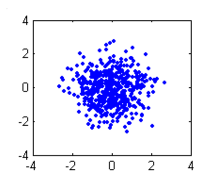
- Integrate 1d Gaussian to get CDF:

$$p(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\tfrac{1}{2}x^2\right)$$

$$F(x) = \int_{-\infty}^{x} p(t)\,dt = \tfrac{1}{2} erf\left(\tfrac{1}{\sqrt{2}}x\right) + \tfrac{1}{2}$$

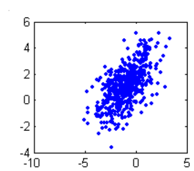
- If sample from uniform, get: $u \sim uniform(0,1)$
- Compute mapping: $x = F^{-1}(u) = \sqrt{2}erfinv(2u-1)$
- This is a Gaussian sample: $x \sim N(x \mid 0,1)$
- For D-dimensional Gaussian N(z|0,I) concatenate samples:

$$\vec{x} = \left[\vec{x}(1) \ldots \vec{x}(D)\right]^T \sim p(\vec{x} \mid 0, I) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi}} \exp\left(-\tfrac{1}{2}\vec{x}(d)^2\right)$$



- For N(z|μ,Σ), add mean & multiply by root cov

$$\vec{z} = \Sigma^{1/2}\vec{x} + \vec{\mu} \sim p(\vec{z} \mid \vec{\mu}, \Sigma)$$



- Example code: gendata.m