http://www.cs.washington.edu/education/courses/cse446/13sp/

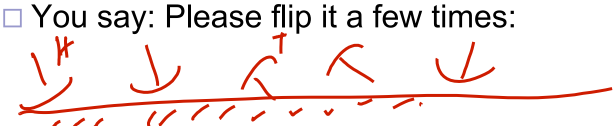
Point Estimation

Machine Learning – CSE446
Carlos Guestrin
University of Washington

April 3, 2013

Your first consulting job

- A billionaire from the suburbs of Seattle asks you a question:
  - He says: I have thumbtack, if I flip it, what's the probability it will fall with the nail up?
  - You say: Please flip it a few times:



$P(H) = 3/5$

  - You say: The probability is:
  - **He says: Why???**
  - You say: Because…

Thumbtack – Binomial Distribution

Model

- P(Heads) = θ, P(Tails) = 1-θ

observed HHTTH

$$P(D|\theta) = P(HHTTH) = \theta\ \theta\ (1-\theta)\ (1-\theta)\ \theta$$
$$= \theta^3\ (1-\theta)^2$$

learning task
choose $\hat{\theta}$
pick θ that maximizes the likelihood of seeing these observations according to my model

- Flips are i.i.d.:
  - Independent events
  - Identically distributed according to Binomial distribution iid
- Sequence $\underline{D}$ of α_H Heads and α_T Tails

$$P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$$

MLE

Maximum Likelihood Estimation

- **Data:** Observed set D of α_H Heads and α_T Tails
- **Hypothesis:** Binomial distribution
- Learning θ is an optimization problem
  - What's the objective function?
- MLE: Choose θ that maximizes the probability of observed data:

$$\begin{aligned}\widehat{\theta} &= \arg\max_{\theta} \quad P(\mathcal{D} \mid \theta) \\ &= \arg\max_{\theta} \quad \ln P(\mathcal{D} \mid \theta)\end{aligned}$$

Your first learning algorithm

$$\widehat{\theta} = \arg\max_{\theta} \ \ln P(\mathcal{D} \mid \theta)$$
$$= \arg\max_{\theta} \ \ln \theta^{\alpha_H}(1-\theta)^{\alpha_T}$$

- Set derivative to zero: $\boxed{\frac{d}{d\theta} \ln P(\mathcal{D} \mid \theta) = 0}$

How many flips do I need?

$$\widehat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T}$$

- Billionaire says: I flipped 3 heads and 2 tails.
- You say: θ = 3/5, I can prove it!
- He says: What if I flipped 30 heads and 20 tails?
- You say: Same answer, I can prove it!
- **He says: What's better?**
- You say: Humm… The more the merrier???
- He says: Is this why I am paying you the big bucks???

Simple bound (based on Hoeffding's inequality)

- For $N = \alpha_H + \alpha_T$, and $\widehat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T}$
- Let θ^* be the true parameter, for any $\varepsilon>0$:

$$P(\mid \widehat{\theta} - \theta^* \mid \geq \epsilon) \leq 2e^{-2N\epsilon^2}$$

PAC Learning

- PAC: Probably Approximate Correct
- Billionaire says: I want to know the thumbtack parameter θ, within $\varepsilon = 0.1$, with probability at least $1-\delta = 0.95$. How many flips?

$$P(\mid \widehat{\theta} - \theta^* \mid \geq \epsilon) \leq 2e^{-2N\epsilon^2}$$

What about continuous variables?

- Billionaire says: If I am measuring a continuous variable, what can you do for me?
- **You say: Let me tell you about Gaussians…**

$$P(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Some properties of Gaussians

- affine transformation (multiplying by scalar and adding a constant)
  - X ~ $N(\mu,\sigma^2)$
  - Y = aX + b ➔ Y ~ $N(a\mu+b,a^2\sigma^2)$
- Sum of Gaussians
  - X ~ $N(\mu_X,\sigma^2_X)$
  - Y ~ $N(\mu_Y,\sigma^2_Y)$
  - Z = X+Y ➔ Z ~ $N(\mu_X+\mu_Y, \sigma^2_X+\sigma^2_Y)$

Learning a Gaussian

- Collect a bunch of data
  - Hopefully, i.i.d. samples
  - e.g., exam scores
- Learn parameters
  - Mean
  - Variance

$$P(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

MLE for Gaussian

- Prob. of i.i.d. samples $D=\{x_1,\ldots,x_N\}$:

$$P(\mathcal{D} \mid \mu, \sigma) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \prod_{i=1}^{N} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}}$$

- Log-likelihood of data:

$$\begin{aligned} \ln P(\mathcal{D} \mid \mu, \sigma) &= \ln \left[\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \prod_{i=1}^{N} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}}\right] \\ &= -N \ln \sigma\sqrt{2\pi} - \sum_{i=1}^{N} \frac{(x_i-\mu)^2}{2\sigma^2} \end{aligned}$$

Your second learning algorithm: MLE for mean of a Gaussian

- What's MLE for mean?

$$\frac{d}{d\mu}\ln P(\mathcal{D}\mid\mu,\sigma) = \frac{d}{d\mu}\left[-N\ln\sigma\sqrt{2\pi}-\sum_{i=1}^{N}\frac{(x_i-\mu)^2}{2\sigma^2}\right]$$

MLE for variance

- Again, set derivative to zero:

$$\begin{aligned}\frac{d}{d\sigma}\ln P(\mathcal{D}\mid\mu,\sigma) &= \frac{d}{d\sigma}\left[-N\ln\sigma\sqrt{2\pi}-\sum_{i=1}^{N}\frac{(x_i-\mu)^2}{2\sigma^2}\right]\\ &= \frac{d}{d\sigma}\left[-N\ln\sigma\sqrt{2\pi}\right]-\sum_{i=1}^{N}\frac{d}{d\sigma}\left[\frac{(x_i-\mu)^2}{2\sigma^2}\right]\end{aligned}$$

Learning Gaussian parameters

- MLE:

$$\widehat{\mu}_{MLE} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\widehat{\sigma}^2_{MLE} = \frac{1}{N}\sum_{i=1}^{N} (x_i - \widehat{\mu})^2$$

- BTW. MLE for the variance of a Gaussian is **biased**
  - Expected result of estimation is **not** true parameter!
  - Unbiased variance estimator:

$$\widehat{\sigma}^2_{unbiased} = \frac{1}{N-1}\sum_{i=1}^{N} (x_i - \widehat{\mu})^2$$

What you need to know…

- Learning is…
  - Collect some data
    - E.g., thumbtack flips
  - Choose a hypothesis class or model
    - E.g., binomial
  - Choose a loss function
    - E.g., data likelihood
  - Choose an optimization procedure
    - E.g., set derivative to zero to obtain MLE
  - Collect the big bucks

- Like everything in life, there is a lot more to learn…
  - Many more facets… Many more nuances…
  - The fun will continue…

Announcement: R Tutorial

- R: open-source scripting language for stats & ML
- A lot of resources online

- Tutorial:
  - When: Thursday April 4th at 6:00pm
  - Where: EEB 125

- Before attending please download and install R:
  - http://www.r-project.org/
- We recommend using an R environment such as:
  - R studio: http://www.rstudio.com/
  - Tinn-R: http://www.sciviews.org/Tinn-R/

Linear Regression
Bias-Variance Tradeoff

Machine Learning – CSE446
Carlos Guestrin
University of Washington

April 3, 2013

Prediction of continuous variables

- Billionaire sayz: Wait, that's not what I meant!
- You sayz: Chill out, dude.
- He sayz: I want to predict a continuous variable for continuous inputs: I want to predict salaries from GPA.
- You sayz: **I can regress that…**

The regression problem

- **Instances:** <$\mathbf{x}_j$, t_j>
- **Learn:** Mapping from x to t($\mathbf{x}$)
- **Hypothesis space:**
  - Given, basis functions
  - Find coeffs $\mathbf{w}=\{w_1,\ldots,w_k\}$

$$H = \{h_1, \ldots, h_K\}$$

$$\underbrace{t(\mathbf{x})}_{\text{data}} \approx \widehat{f}(\mathbf{x}) = \sum_i w_i h_i(\mathbf{x})$$

  - Why is this called linear regression???
    - model is linear in the parameters

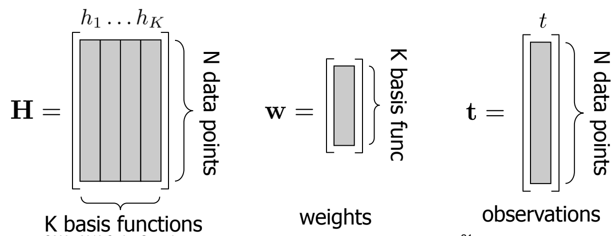
- Precisely, minimize the residual squared error:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

The regression problem in matrix notation

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \underbrace{(\mathbf{H}\mathbf{w} - \mathbf{t})^T (\mathbf{H}\mathbf{w} - \mathbf{t})}_{\text{residual error}}$$



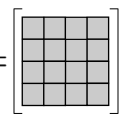
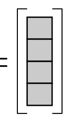
Minimizing the Residual

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \underbrace{(\mathbf{H}\mathbf{w} - \mathbf{t})^T (\mathbf{H}\mathbf{w} - \mathbf{t})}_{\text{residual error}}$$

Regression solution = simple matrix operations

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \underbrace{(\mathbf{H}\mathbf{w} - \mathbf{t})^T(\mathbf{H}\mathbf{w} - \mathbf{t})}_{\text{residual error}}$$

$$\text{solution: } \mathbf{w}^* = \underbrace{(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}}_{\mathbf{A}^{-1}} \underbrace{\mathbf{H}^{\mathrm{T}}\mathbf{t}}_{\mathbf{b}} = \mathbf{A}^{-1}\mathbf{b}$$

where $\mathbf{A} = \mathbf{H}^{\mathrm{T}}\mathbf{H} =$  $\mathbf{b} = \mathbf{H}^{\mathrm{T}}\mathbf{t} =$ 

k×k matrix
for k basis functions

k×1 vector

But, why?

- Billionaire (again) says: Why sum squared error???
- You say: Gaussians, Dr. Gateson, Gaussians…
- Model: prediction is linear function plus Gaussian noise
  - $t(\mathbf{x}) = \sum_i w_i h_i(\mathbf{x}) + \varepsilon_{\mathbf{x}}$
- Learn **w** using MLE

$$P(t \mid \mathbf{x}, \mathbf{w}, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-[t - \sum_i w_i h_i(\mathbf{x})]^2}{2\sigma^2}}$$

Maximizing log-likelihood

Maximize:

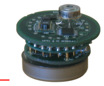
$$\ln P(\mathcal{D} \mid \mathbf{w}, \sigma) = \ln \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \prod_{j=1}^{N} e^{\frac{-\left[t_j - \sum_i w_i h_i(\mathbf{x}_j)\right]^2}{2\sigma^2}}$$

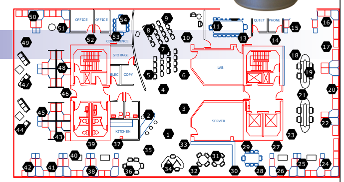
Least-squares Linear Regression is MLE for Gaussians!!!

Applications Corner 1

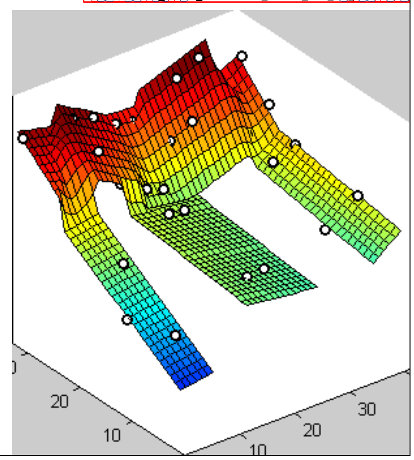
- Predict stock value over time from
  - past values
  - other relevant vars
    - e.g., weather, demands, etc.

Applications Corner 2





- Measure temperatures at some locations
- Predict temperatures throughout the environment



[Guestrin et al. '04]

Applications Corner 3

- Predict when a sensor will fail
  - based several variables
    - age, chemical exposure, number of hours used,…

Bias-Variance tradeoff – Intuition

- Model too "simple" → does not fit the data well
  - A biased solution

- Model too complex → small changes to the data, solution changes a lot
  - A high-variance solution

(Squared) Bias of learner

- Given dataset *D* with *N* samples, learn function $h_D(x)$
- If you sample a different dataset *D'* with *N* samples, you will learn different $h_D'(x)$
- **Expected hypothesis**: $E_D[h_D(x)]$

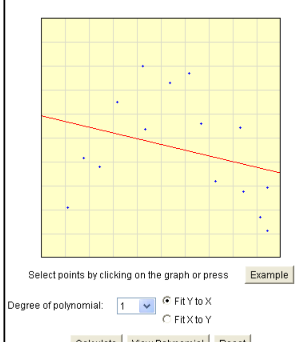
- **Bias:** difference between what you expect to learn and truth
  - Measures how well you expect to represent true solution
  - Decreases with more complex model
  - Bias^2 at one point *x*:
  - Average Bias^2:

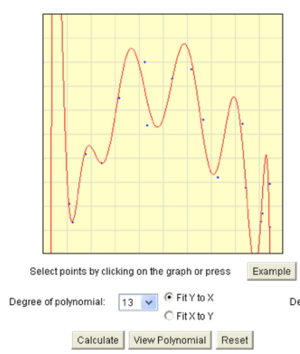
Variance of learner

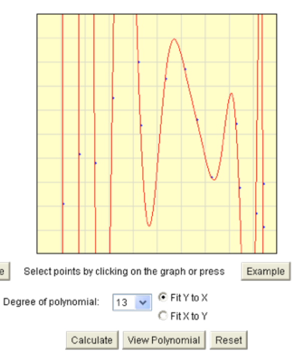
- Given dataset *D* with *N* samples, learn function $h_D(x)$
- If you sample a different dataset *D'* with *N* samples, you will learn different $h_D'(x)$
- **Variance:** difference between what you expect to learn and what you learn from a particular dataset
  - Measures how sensitive learner is to specific dataset
  - Decreases with simpler model
  - Variance at one point *x*:
  - Average pariance:

Bias-Variance Tradeoff

- Choice of hypothesis class introduces learning bias
  - More complex class → less bias
  - More complex class → more variance







Bias-Variance Decomposition of Error

$\bar{h}_N(x) = E_D[h_D(x)]$

- Expected mean squared error: $\text{MSE} = E_D\left[E_x\left[(t(x) - h_D(x))^2\right]\right]$
- To simplify derivation, drop x:
- Expanding the square:

Moral of the Story: Bias-Variance Tradeoff Key in ML

- Error can be decomposed:

$$\text{MSE} = E_D\left[E_x\left[(t(x) - h_D(x))^2\right]\right]$$
$$= E_x\left[\left(t(x) - \bar{h}_N(x)\right)^2\right] + E_D\left[E_x\left[\left(\bar{h}(x) - h_D(x)\right)^2\right]\right]$$

- Choice of hypothesis class introduces learning bias
  - More complex class → less bias
  - More complex class → more variance

What you need to know

- Regression
  - Basis function = features
  - Optimizing sum squared error
  - Relationship between regression and Gaussians
- Bias-variance trade-off
- Play with Applet