http://courses.cs.washington.edu/courses/csep546/14wi/

What's learning? Point Estimation

Machine Learning – CSEP546
Carlos Guestrin
University of Washington

December 6, 2014

What is Machine Learning ?

Machine Learning

Study of algorithms that

- improve their performance
- at some task
- with experience

Classification

from data to discrete classes

Spam filtering

data

prediction

Osman Khan to Carlos show details Jan 7 (6 days ago) Reply

sounds good
+ok

Carlos Guestrin wrote:
Let's try to chat on Friday a little to coordinate and more on Sunday in person?

Carlos

Welcome to New Media Installation: Art that Learns

Carlos Guestrin to 10615-announce, Osman, Miche show details 3:15 PM (8 hours ago) Reply

Hi everyone,

Welcome to New Media Installation:Art that Learns

The class will start tomorrow.
Make sure you attend the first class, even if you are on the Wait List.
The classes are held in Doherty Hall C316, and will be Tue, Thu 01:30-4:20 PM.

By now, you should be subscribed to our course mailing list: 10615-announce@cs.cmu.edu.
You can contact the instructors by emailing: 10615-instructors@cs.cmu.edu

Natural _LoseWeight SuperFood Endorsed by Oprah Winfrey, Free Trial 1 bottle, pay only $5.95 for shipping mfw rlk Spam | x

Jaquelyn Halley to nherrlein, bcc: thehorney, bcc: anç show details 9:52 PM (1 hour ago) Reply

=== Natural WeightL0SS Solution ===

Vital Acai is a natural WeightL0SS product that Enables people to lose wieght and cleansing their bodies faster than most other products on the market.

Here are some of the benefits of Vital Acai that You might not be aware of. These benefits have helped people who have been using Vital Acai daily to Achieve goals and reach new heights in there dieting that they never thought they could.

* Rapid WeightL0SS
* Increased metabolism - BurnFat & calories easily!
* Better Mood and Attitude
* More Self Confidence
* Cleanse and Detoxify Your Body
* Much More Energy
* BetterSexLife
* A Natural Colon Cleanse

5

Text classification

→ Company home page

vs

Personal home page

vs

Univeristy home page

vs

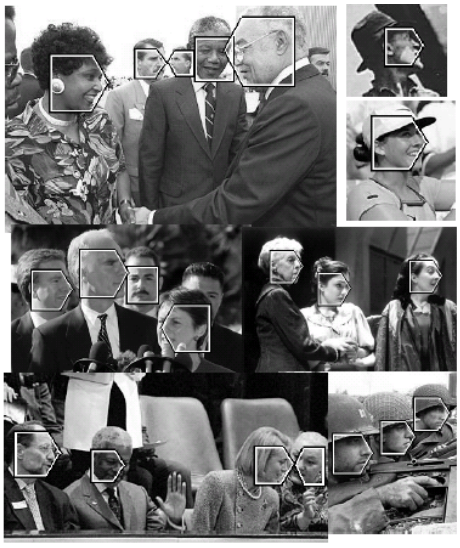
…

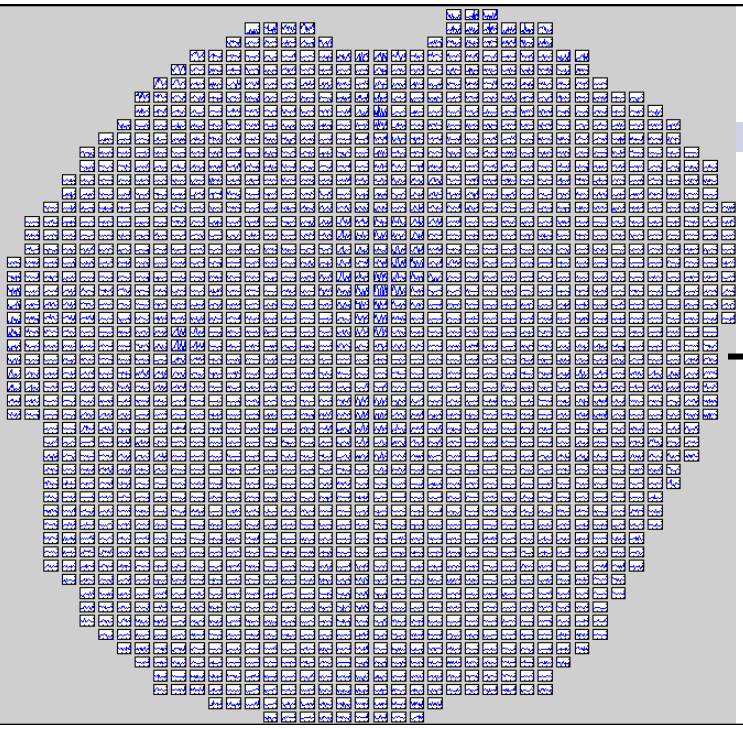
6

Object detection

(Prof. H. Schneiderman)

Example training images for each orientation

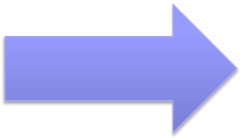




Reading a noun (vs verb)

[Rustandi et al., 2005]

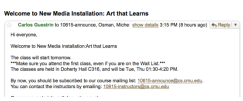
Weather prediction



The classification pipeline

Training

Testing



Regression

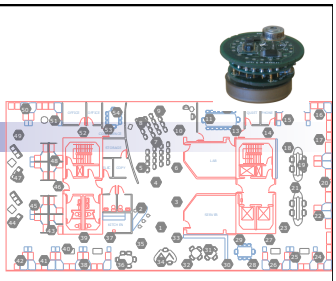
predicting a numeric value

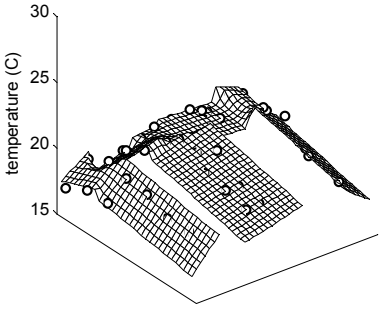
Stock market

Weather prediction revisted

Temperature

Modeling sensor data





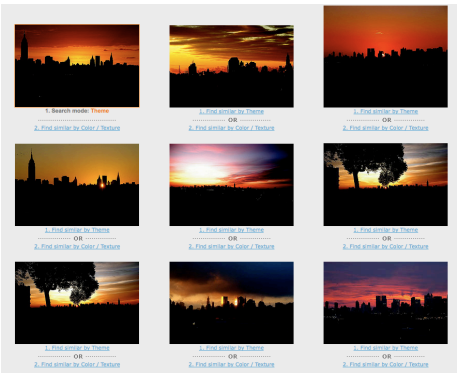
- Measure temperatures at some locations
- Predict temperatures throughout the environment

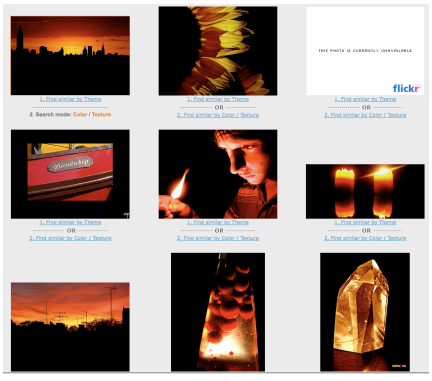
[Guestrin et al. '04]

Similarity

finding data

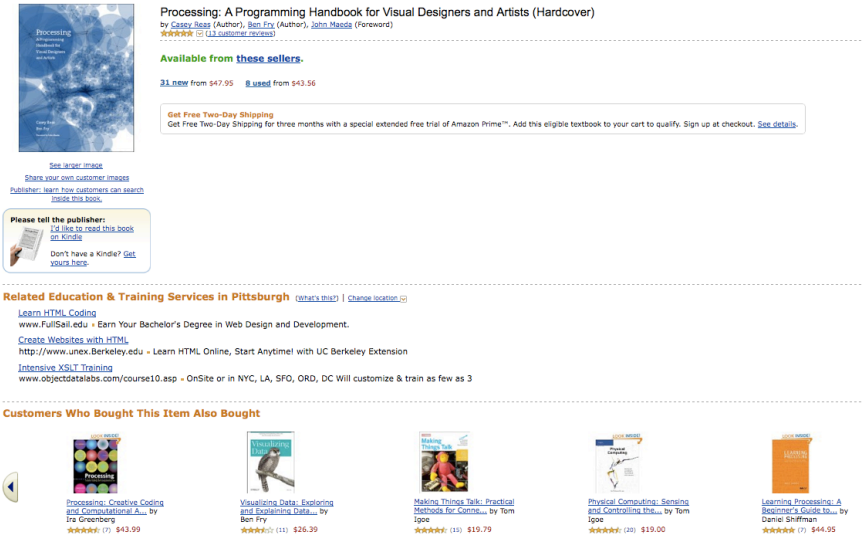
Given image, find similar images





http://www.tiltomo.com/

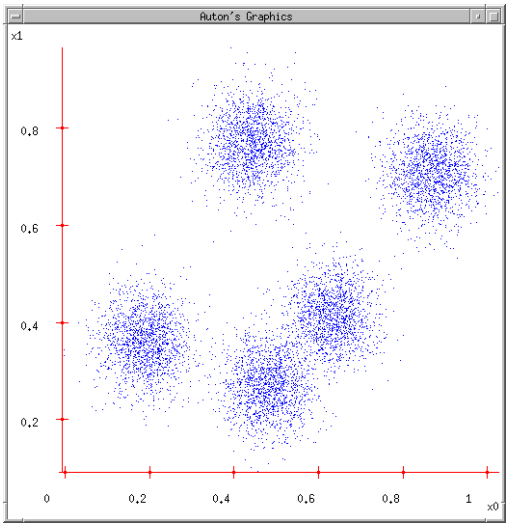
Similar products



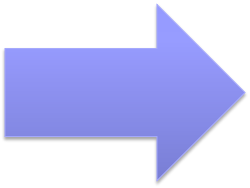
Clustering

discovering structure in data

Clustering Data: Group similar things

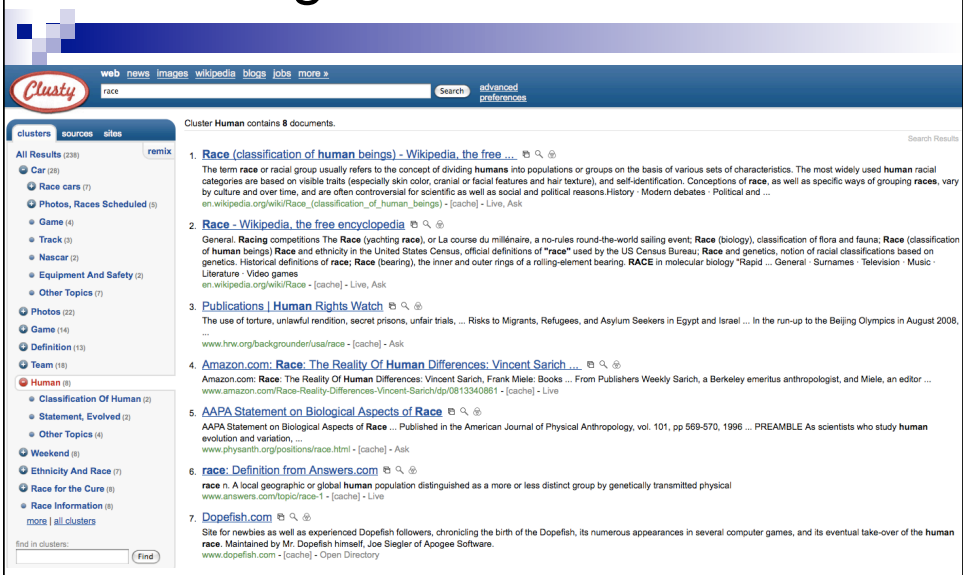


Clustering images



[Goldberger et al.]

Clustering web search results

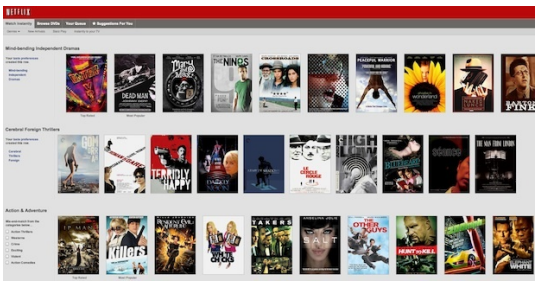


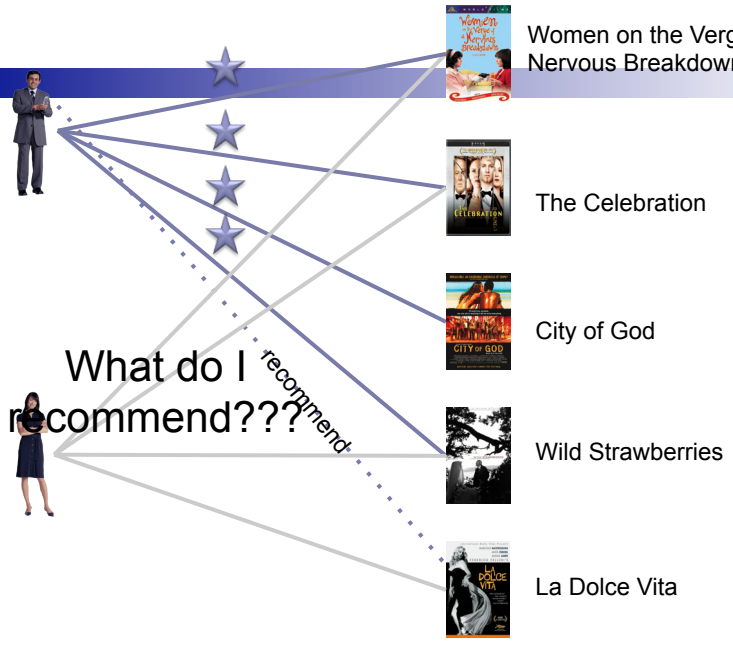
Recommender Systems

figuring out what your customers want

Collaborative Filtering

- **Goal:** Find movies of interest to a user based on movies watched by the user and others
- **Methods:** matrix factorization





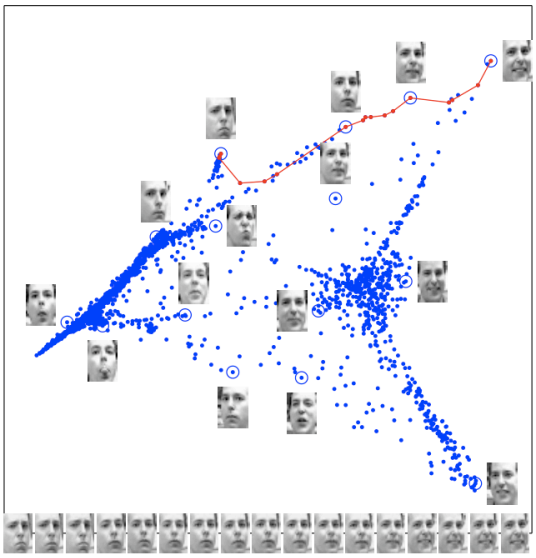
Embedding

visualizing data

Embedding images

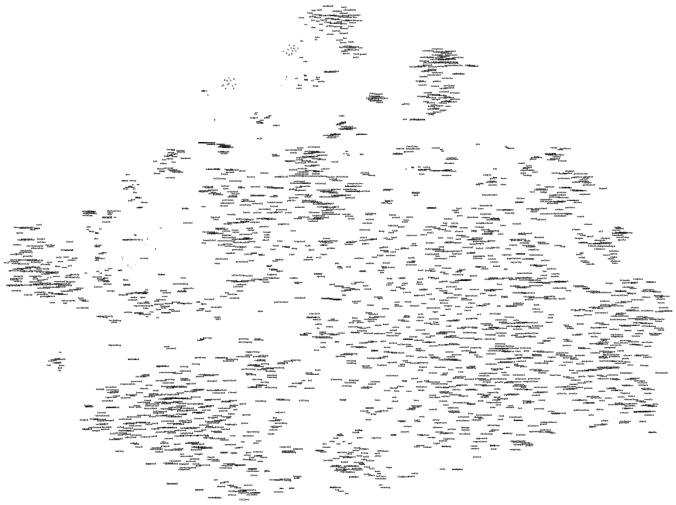
Images have thousands or millions of pixels.

Can we give each image a coordinate, such that similar images are near each other?



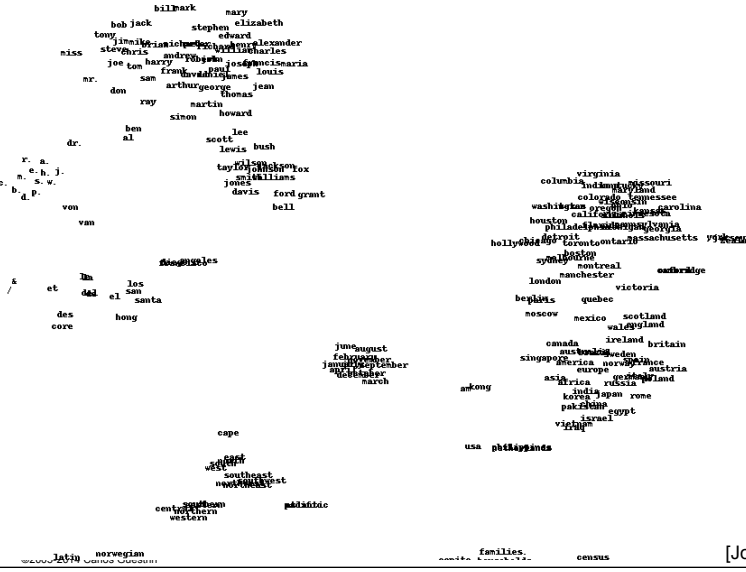
[Saul & Roweis '03]

Embedding words



[Joseph Turian] 27

Embedding words (zoom in)



[Joseph Turian] 28

Reinforcement Learning

training by feedback

Learning to act

- Reinforcement learning
- An agent
  - Makes sensor observations
  - Must select action
  - Receives rewards
    - positive for “good” states
    - negative for “bad” states

[Ng et al. ’05]

Bringing it all together…

Combining video, text and audio

HURLEY: Uh ... the Chinese people have water.
(Sayid and Kate go to check it out.)

[EXT. BEACH - CRASH SITE]

(Sayid holds the empty bottle in his hand and questions Sun.)

SAYID: (quietly) Where did you get this?
(He looks at her.)

[EXT. JUNGLE]

(Sawyer is walking through the jungle. He reaches a spot. He kneels down and looks back to check that no one's followed him.

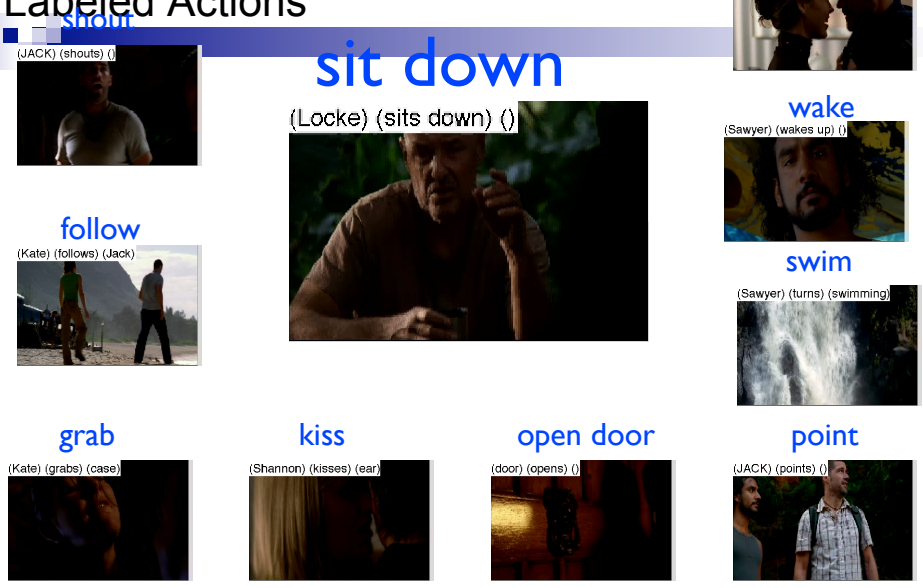
SAYID

SUN

HOLDING

Taskar et al.

Automatically Discovered and Labeled Actions



Growth of Machine Learning

One of the most sought for specialties in industry today!!!!

- Machine learning is preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - Computational biology
  - Sensor networks
  - …
- This trend is accelerating, especially with **Big Data**
  - Improved machine learning algorithms
  - Improved data capture, networking, faster computers
  - Software too complex to write by hand
  - New sensors / IO devices
  - Demand for self-customization to user, environment

Syllabus

- Covers a wide range of Machine Learning techniques – from basic to state-of-the-art
- You will learn about the methods you heard about:
  - Point estimation, regression, naïve Bayes, logistic regression, nearest-neighbor, decision trees, boosting, perceptron, overfitting, regularization, dimensionality reduction, PCA, recommender systems, matrix factorization, SVMs, kernels, margin bounds, K-means, EM, mixture models, semi-supervised learning, neural networks, reinforcement learning…
- Covers algorithms, theory and applications
- **It's going to be fun and hard work ☺**

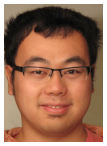
Prerequisites

- Formally:
  - STAT 341, STAT 391, or equivalent
- Probabilities
  - Distributions, densities, marginalization…
- Basic statistics
  - Moments, typical distributions, regression…
- Algorithms
  - Dynamic programming, basic data structures, complexity…
- Programming
  - Python will be very useful
- We provide some background, but the class will be fast paced
- Ability to deal with "abstract mathematical concepts"

Staff

- Two Great TAs: Great resource for learning, interact with them!
  - **Akshay Srinivasan**
    Office hours: Wednesdays 4:30-6:30pm

  - **TianyiZhou**
    Office hours: Tuesdays 4:30-6:30pm



  - Prof: **Carlos Guestrin**
    Office hours: Mondays 5:30-6:30pm

Communication Channels

- Only channel for announcements, questions, etc. – Catalyst Group:
  - https://catalyst.uw.edu/gopost/board/tianyizh/35317/
  - Subscribe!
  - All non-personal questions should go here
  - Answering your question will help others
  - Feel free to chime in
- For e-mailing instructors about personal issues, use:
  - csep546-instructors@cs.washington.edu

Text Books

- Required Textbook:
  - Machine Learning: a Probabilistic Perspective; Kevin Murphy
- Optional Books:
  - Pattern Recognition and Machine Learning; Chris Bishop
  - The Elements of Statistical Learning: Data Mining, Inference, and Prediction; Trevor Hastie, Robert Tibshirani, Jerome Friedman
  - Machine Learning; Tom Mitchell
  - Information Theory, Inference, and Learning Algorithms; David MacKay

Grading

- 4 homeworks (70%)
  - First one goes out this week
    - Start early, Start early
- Final project (30%)
  - Full details out around next week
  - Projects done individually, or groups of two students

Homeworks

- Homeworks are hard, start early ☺
- Due in the beginning of class
- 33% subtracted per late day
- All homeworks **must be handed in**, even for zero credit
- Use Catalyst to submit homeworks

- Collaboration
  - You may **discuss** the questions
  - Each student writes their own answers
  - Write on your homework anyone with whom you collaborate
  - Each student must write their own code for the programming part
  - **Please don't search for answers on the web, Google, previous years' homeworks, etc.**
    - please ask us if you are not sure if you can use a particular reference

Projects

- An opportunity to exercise what you learned and to learn new things
- Individually or groups of two
- Must involve real data
  - Must be data that you have available to you by the time of the project proposals
- Must involve machine learning
- It's encouraged to be related to your research, but must be something new you did this quarter
  - Not a project you worked on during the summer, last year, etc.

- Full details in a week or so

- Mon., January 27 by 6:30pm: **Project Proposals**
- Mon., February 24 by 6:30pm: **Project Milestone**
- Mon., March 17 by 6:30pm: **Poster Session**
- Mon., March 19 by 6:30pm: **Project Report**

Enjoy!

- ML is becoming ubiquitous in science, engineering and beyond
- It's one of the hottest topics in industry today
- This class should give you the basic foundation for applying ML and developing new methods
- The fun begins…

Point Estimation MLE

Machine Learning – CSEP546

Carlos Guestrin
University of Washington

January 6, 2014

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

Thumbtack – Binomial Distribution

- P(Heads) = θ, P(Tails) = 1-θ
- Flips are i.i.d.:
  - Independent events
  - Identically distributed according to Binomial distribution
- Sequence *D* of α_H Heads and α_T Tails

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

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:

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

Your first learning algorithm

$$\begin{aligned}\hat{\theta} &= \arg\max_{\theta} \ln P(\mathcal{D} \mid \theta) \\ &= \arg\max_{\theta} \ln \theta^{\alpha_H}(1-\theta)^{\alpha_T}\end{aligned}$$

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

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:

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

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

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

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

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…

Linear Regression

Machine Learning – CSEP546
Carlos Guestrin
University of Washington

January 6, 2014

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(**x**)
- **Hypothesis space:**
  - Given, basis functions
  - Find coeffs **w**={w_1,…,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

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

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

$$\mathbf{H} = \begin{bmatrix} h_1 \ldots h_K \\ \vdots \\ \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} \\ \vdots \\ \end{bmatrix} \quad \mathbf{t} = \begin{bmatrix} t \\ \vdots \\ \end{bmatrix}$$

N data points; K basis functions; K basis func; N data points

weights; observations

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

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

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

$$\text{where } \mathbf{A} = \mathbf{H}^{\mathrm{T}}\mathbf{H} = \underbrace{\left[\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline\end{array}\right]}_{\text{k×k matrix for k basis functions}} \quad \mathbf{b} = \mathbf{H}^{\mathrm{T}}\mathbf{t} = \underbrace{\left[\begin{array}{|c|}\hline \\ \hline \\ \hline \\ \hline\end{array}\right]}_{\text{k×1 vector}}$$

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

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

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!!!

Bias-Variance Tradeoff

Machine Learning – CSEP546
Carlos Guestrin
University of Washington

January 6, 2014

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

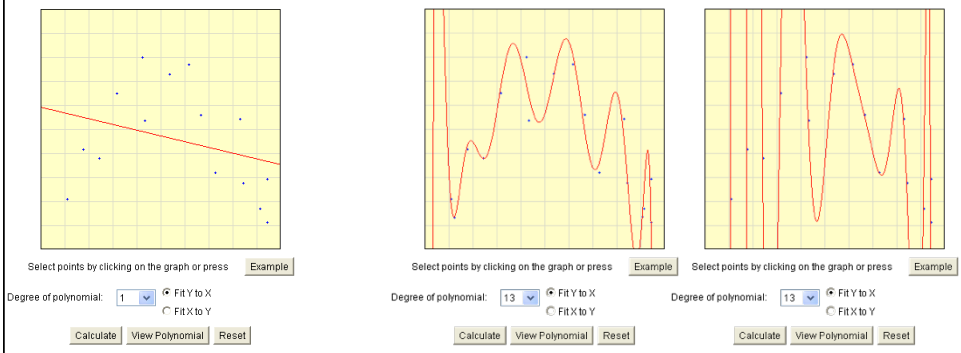
- Given dataset *D* with *N* samples, learn function $h_D(x)$
- If you sample a different dataset *D'* with *N* samples, you will learn different $h_D'(x)$
- **Expected hypothesis**: $E_D[h_D(x)]$
- **Bias:** difference between what you expect to learn and truth
  - Measures how well you expect to represent 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 variance:

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

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

- Error can be decomposed:

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

- 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

Overfitting

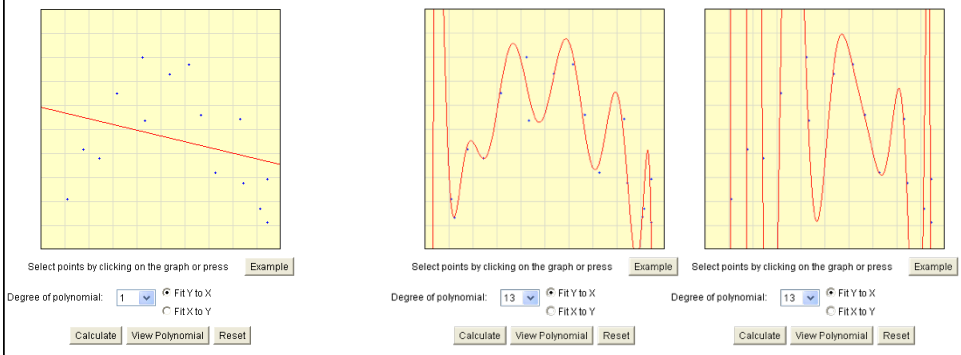
Machine Learning – CSE446

Carlos Guestrin
University of Washington

January 6, 2014

Bias-Variance Tradeoff

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



Training set error

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

- Given a dataset (Training data)
- Choose a loss function
  - e.g., squared error (L_2) for regression
- **Training set error:** For a particular set of parameters, loss function on training data:

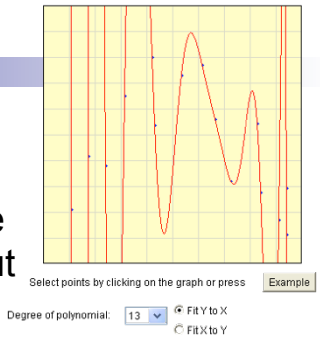
$$error_{train}(\mathbf{w}) = \frac{1}{N_{train}} \sum_{j=1}^{N_{train}} \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

Training set error as a function of model complexity

$$error_{train}(\mathbf{w}) = \frac{1}{N_{train}} \sum_{j=1}^{N_{train}} \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

Prediction error

- Training set error can be poor measure of "quality" of solution
- **Prediction error:** We really care about error over all possible input points, not just training data:

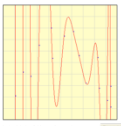


$$error_{true}(\mathbf{w}) = E_{\mathbf{x}}\left[\left(t(\mathbf{x}) - \sum_i w_i h_i(\mathbf{x})\right)^2\right]$$

$$= \int_{\mathbf{x}} \left(t(\mathbf{x}) - \sum_i w_i h_i(\mathbf{x})\right)^2 p(\mathbf{x}) d\mathbf{x}$$

Prediction error as a function of model complexity: Bias/Variance tradeoff

$$error_{true}(\mathbf{w}) = \int_{\mathbf{x}} \left(t(\mathbf{x}) - \sum_i w_i h_i(\mathbf{x})\right)^2 p(\mathbf{x}) d\mathbf{x}$$



Prediction error as a function of model complexity: train v. true error

$$error_{train}(\mathbf{w}) = \frac{1}{N_{train}} \sum_{j=1}^{N_{train}} \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

$$error_{true}(\mathbf{w}) = \int_{\mathbf{x}} \left(t(\mathbf{x}) - \sum_i w_i h_i(\mathbf{x}) \right)^2 p(\mathbf{x}) d\mathbf{x}$$

Computing prediction error

- Computing prediction
  - Hard integral
  - May not know t(**x**) for every **x**

$$error_{true}(\mathbf{w}) = \int_{\mathbf{x}} \left(t(\mathbf{x}) - \sum_i w_i h_i(\mathbf{x}) \right)^2 p(\mathbf{x}) d\mathbf{x}$$

- Monte Carlo integration (sampling approximation)
  - Sample a set of i.i.d. points $\{\mathbf{x}_1,\ldots,\mathbf{x}_M\}$ from p(**x**)
  - Approximate integral with sample average

$$error_{true}(\mathbf{w}) \approx \frac{1}{M} \sum_{j=1}^{M} \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

Why training set error doesn't approximate prediction error?

- Sampling approximation of prediction error:

$$error_{true}(\mathbf{w}) \approx \frac{1}{M}\sum_{j=1}^{M}\left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j)\right)^2$$

- Training error :

$$error_{train}(\mathbf{w}) = \frac{1}{N_{train}}\sum_{j=1}^{N_{train}}\left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j)\right)^2$$

- Very similar equations!!!
  - Why is training set a bad measure of prediction error???

Why training set error doesn't approximate prediction error?

-
-

Because you cheated!!!

Training error good estimate for a single **w**,
But you optimized **w** with respect to the training error,
and found **w** that is good for this set of samples

Training error is a (optimistically) biased estimate of prediction error

- Very similar equations!!!
  - Why is training set a bad measure of prediction error???

Test set error

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

- Given a dataset, **randomly** split it into two parts:
  - Training data – $\{\mathbf{x}_1, \ldots, \mathbf{x}_{Ntrain}\}$
  - Test data – $\{\mathbf{x}_1, \ldots, \mathbf{x}_{Ntest}\}$
- Use training data to optimize parameters **w**
- **Test set error:** For the ***final output*** $\hat{\mathbf{w}}$, evaluate the error using:

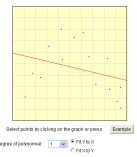
$$error_{test}(\mathbf{w}) = \frac{1}{N_{test}} \sum_{j=1}^{N_{test}} \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

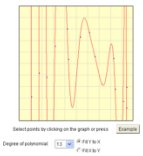
Test set error as a function of model complexity

$$error_{train}(\mathbf{w}) = \frac{1}{N_{train}} \sum_{j=1}^{N_{train}} \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

$$error_{true}(\mathbf{w}) = \int_{\mathbf{x}} \left(t(\mathbf{x}) - \sum_i w_i h_i(\mathbf{x}) \right)^2 p(\mathbf{x}) d\mathbf{x}$$

$$error_{test}(\mathbf{w}) = \frac{1}{N_{test}} \sum_{j=1}^{N_{test}} \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$





Overfitting

- **Overfitting:** a learning algorithm overfits the training data if it outputs a solution **w** when there exists another solution **w'** such that:

$$[error_{train}(\mathbf{w}) < error_{train}(\mathbf{w}')] \wedge [error_{true}(\mathbf{w}') < error_{true}(\mathbf{w})]$$

How many points to I use for training/testing?

- Very hard question to answer!
  - Too few training points, learned **w** is bad
  - Too few test points, you never know if you reached a good solution
  - Some theoretical bounds can be useful in theory… ☺
- More on this later this quarter, but still hard to answer
- Typically:
  - If you have a reasonable amount of data, pick test set "large enough" for a "reasonable" estimate of error, and use the rest for learning
  - If you have little data, then you need to pull out the big guns…
    - e.g., bootstrapping

Error estimators

$$error_{true}(\mathbf{w}) = \int_{\mathbf{x}} \left(t(\mathbf{x}) - \sum_i w_i h_i(\mathbf{x}) \right)^2 p(\mathbf{x}) d\mathbf{x}$$

$$error_{train}(\mathbf{w}) = \frac{1}{N_{train}} \sum_{j=1}^{N_{train}} \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

$$error_{test}(\mathbf{w}) = \frac{1}{N_{test}} \sum_{j=1}^{N_{test}} \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

Error as a function of number of training examples for a fixed model complexity

$$error_{train}(\mathbf{w}) = \frac{1}{N_{train}} \sum_{j=1}^{N_{train}} \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

$$error_{true}(\mathbf{w}) = \int_{\mathbf{x}} \left(t(\mathbf{x}) - \sum_i w_i h_i(\mathbf{x}) \right)^2 p(\mathbf{x}) d\mathbf{x}$$

little data → infinite data

Error estimators

Be careful!!!

Test set only unbiased if you never never ever ever do any any any any learning on the test data

For example, if you use the test set to select the degree of the polynomial… no longer unbiased!!! (We will address this problem later in the quarter)

$$error_{test}(\mathbf{w}) = \frac{1}{N_{test}} \sum_{j=1}^{N_{test}} \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

What you need to know

- True error, training error, test error
  - Never learn on the test data
  - Never learn on the test data
  - Never learn on the test data
  - Never learn on the test data
  - Never learn on the test data
- Overfitting