Introduction to

Artificial Intelligence (AI)

Computer Science cpsc502, Lecture 16

Nov, 3, 2011

Slide credit: C. Conati, S. Thrun, P. Norvig, Wikipedia
Peter Mark Pollefeys, Dan Klein, Chris Manning

Supervised ML: Formal Specification

ONE EXAMPLE

$$
\begin{array}{ccccc}
X_{11} & X_{12} & \cdots & X_{1N} & \rightarrow Y_1 \\
X_{21} & & \cdots & X_{2N} & \rightarrow Y_2 \\
\vdots & & & & \\
X_{M1} & & \cdots & X_{MN} & \rightarrow Y_M
\end{array}
$$

unsupervised

N features

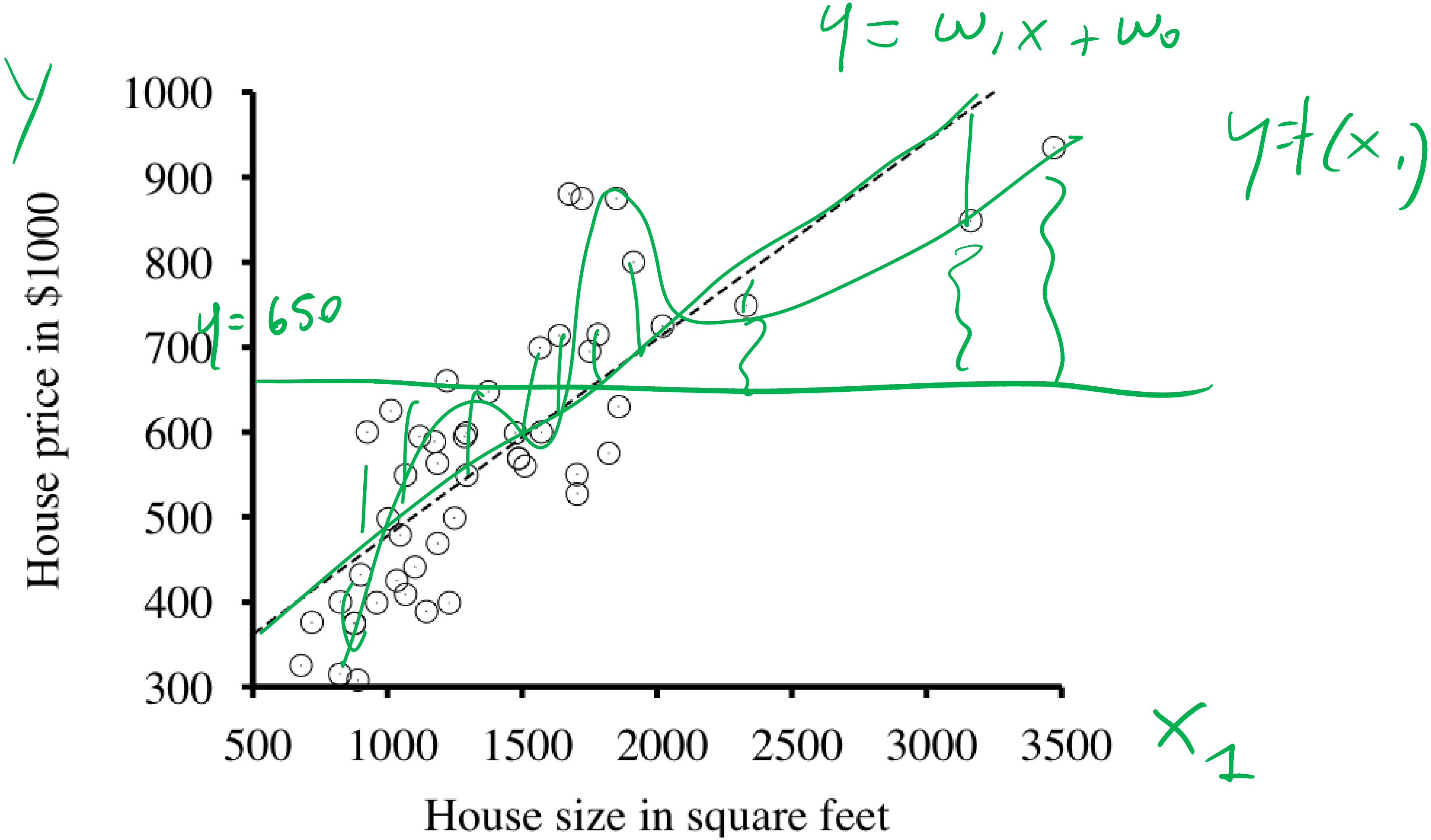
M examples

$$f(\bar{X}) \rightarrow Y$$

Today Nov 3

- **Supervised Machine Learning**
  - Naïve Bayes
  - Markov-Chains
  - Decision Trees — Classification y discrete
  - Regression y continuous
  - Logistic Regression $y \rightsquigarrow z \in [0,1]$
- **Unsupervised Machine Learning**
  - K-means
  - Intro to EM

Regression: Example



Regression

- Only very simple examples
  - Linear regression
- Linear model
  - $y = m\,x + b$
  - $h_{\mathbf{w}}(x) = y = w_1\,x + w_0$
- Find best values for parameters
  - "maximize goodness of fit"
  - "maximize probability" or "minimize loss"

error

$\arg\max_{w} P(D|\bar{w})$

Regression: Minimizing Loss

- Assume true function *f* is given by

$$y = f(x) = m\,x + b + noise$$

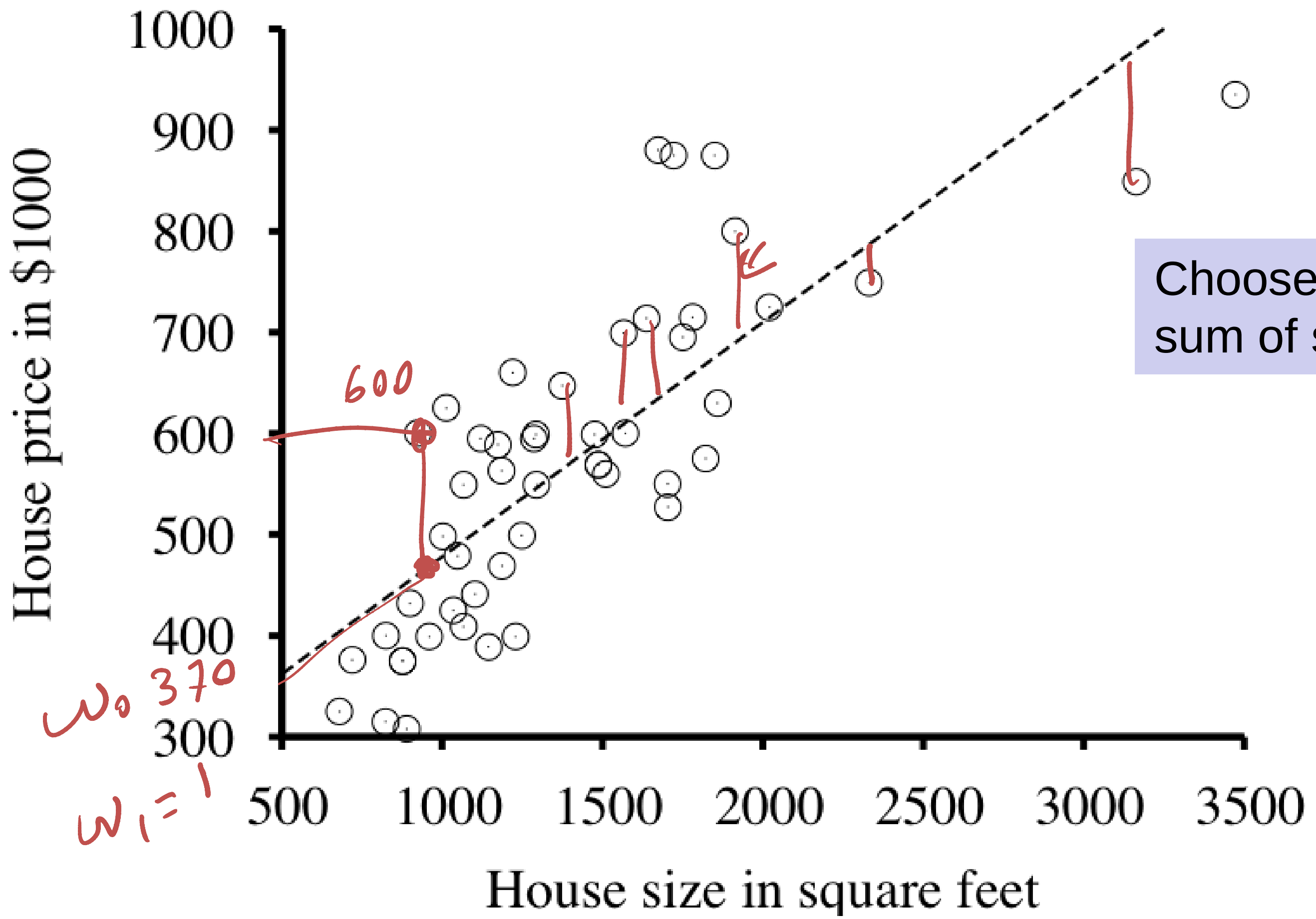
where noise is normally distributed

- Then **most probable values of parameters** found by **minimizing squared-error loss**:

$$Loss(h_{\mathbf{w}}) = \Sigma_j \, (\boxed{y_j} - h_{\mathbf{w}}(x_j))^2$$

training data

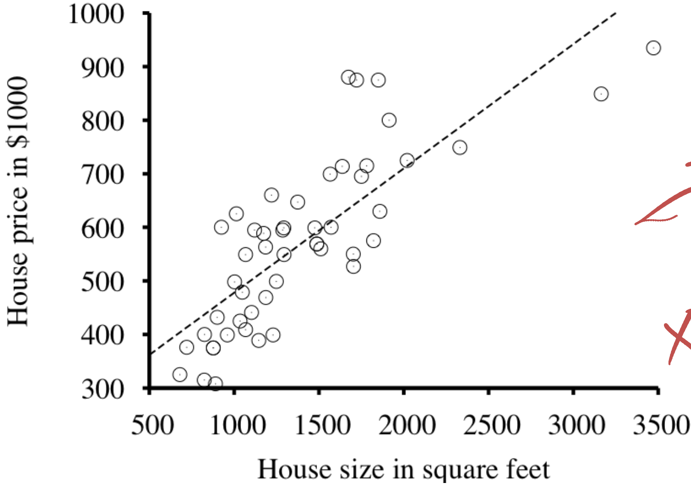
Regression: Minimizing Loss



Choose weights to minimize sum of squared errors

Regression: Minimizing Loss

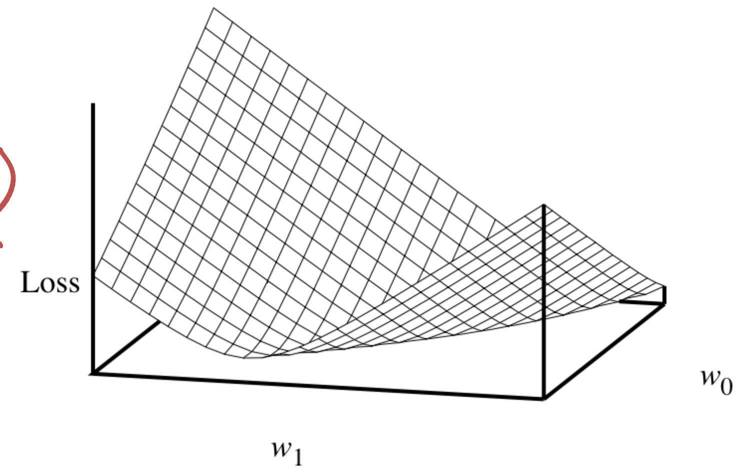
$$y = w_1\, x + w_0$$



$$\frac{\partial Loss(w)}{\partial w_1}$$

x_1

$w_1 x + w_0$



$$Loss(h_{\mathbf{w}}) = \Sigma_j \; (y_j - h_{\mathbf{w}}(x_j))^2$$

Algebra gives an exact solution to the minimization problem

$$\frac{\partial Loss(w)}{\partial w_0} = \sum_j (y_j - w_1 x_j - w_0)^2 = \sum_j -2(y_j - w_1 x_j - w_0) = 0$$

Compute w_0 and w_1 in closed form

$$\frac{\partial}{\partial w_0} \sum_i^n (y_i - w_1 x_i - w_0)^2 = 0$$

$$\sum_i -2(y_i - w_1 x_i - w_0) = 0$$

$$-\sum_i^n y_i + \sum_i^n w_1 x_i + \sum_i^n w_0 = 0$$

$$-\sum_i y_i + w_1 \sum_i x_i + n w_0 = 0$$

$$w_0 = \frac{\sum y_i}{n} - w_1 \frac{\sum x_i}{n}$$

$\Rightarrow$ next page

$$\frac{\partial}{\partial w_1} \sum (y_i - w_1 x_i - w_0)^2 = 0$$

$$\cancel{2} \sum -x_i \, (y_i - w_1 x_i - w_0) = 0$$

expressed as a function of w_1

$$\sum -x_i y_i + w_1 x_i^2 + x_i \boxed{w_0}$$

plug in w_0 from previous derivation

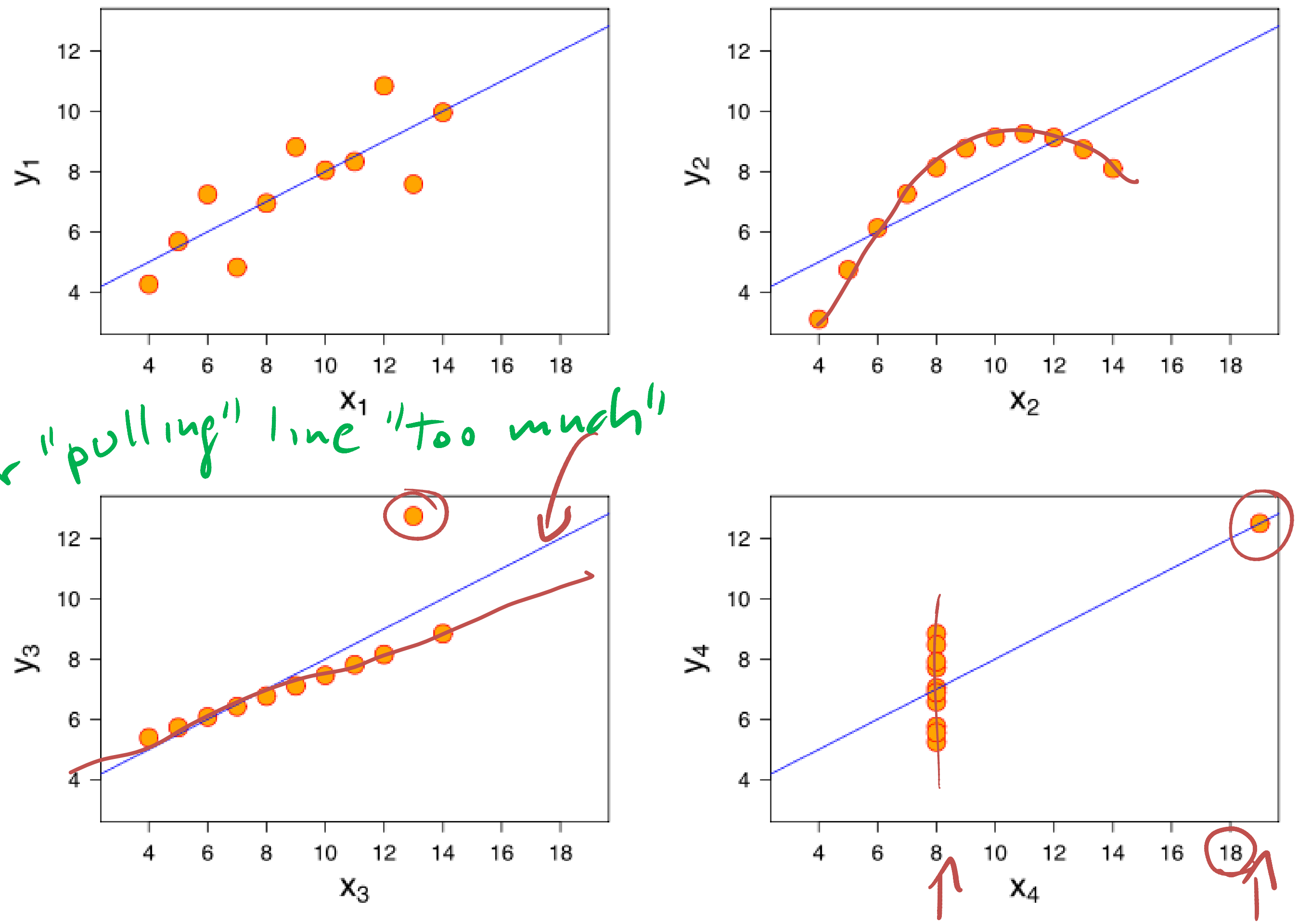
$$\vdots \qquad \vdots$$

$$w_1 = \frac{\sum x_i y_i - \frac{1}{n} \sum x_i \sum y_i}{\sum x_i^2 - \frac{1}{n} \left(\sum x_i\right)^2}$$

COMPUTE THIS FROM DATA

THEN USE THE VALUE TO COMPUTE w_0

Don’t Always Trust Linear Models



outlier "pulling" line "too much"

- Anscombe's quartet

Four datasets with “identical” statistical properties according to regression

Multivariate Regression

- $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$
  - $h_{\mathbf{w}}(\mathbf{x}) = \boxed{\mathbf{w}} \cdot \mathbf{x} = \mathbf{w}\,\mathbf{x}^{\mathsf{T}} = \Sigma_i \; w_i \, x_i$

$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$

$w_1 \ldots w_n$

$x_1 \ldots x_n$

- The most probable set of weights, **w*** (minimizing squared error):
  - $(\mathbf{y} - \mathbf{X}\mathbf{w})^{\mathsf{T}} (\mathbf{y} - \mathbf{X}\mathbf{w})$
  - $\mathbf{w}^* = (\mathbf{X}^{\mathsf{T}} \mathbf{X})^{-1} \mathbf{X}^{\mathsf{T}} \mathbf{y}$

Derivative of the Squared error

$X^T(y - Xw) = 0$

check text

Overfitting

- To avoid overfitting, don't just minimize loss
- Also minimize complexity of the model
- Can be stated as minimization: squared error
  - $\boxed{\text{Cost}(h)} = \boxed{\text{EmpiricalLoss}(h)} + \boxed{\lambda\ \text{Complexity}(h)}$
- For linear models, consider

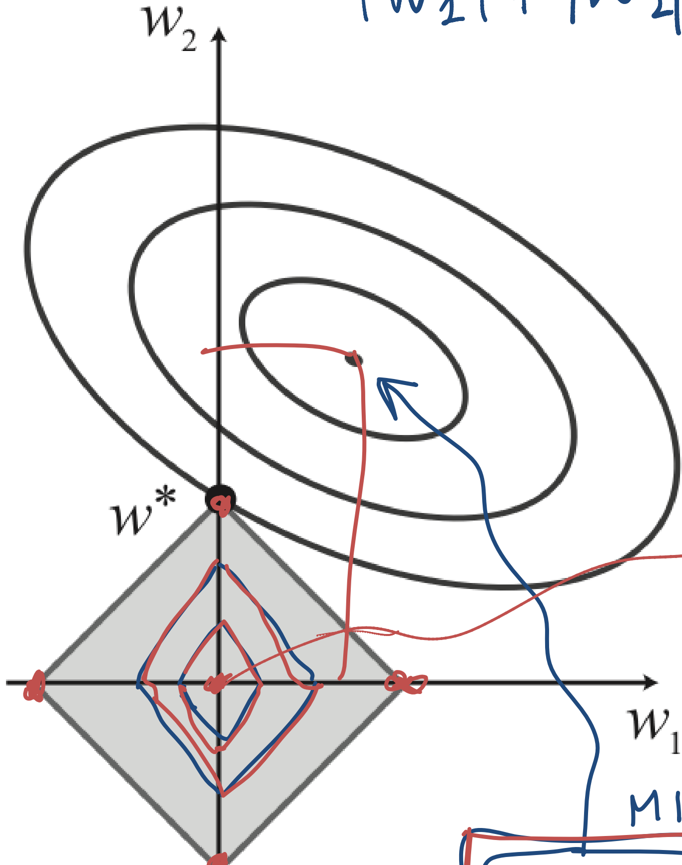
$$\text{Complexity}(h_{\mathbf{w}}) = L_q(\mathbf{w}) = \sum_i |w_i|^q$$

  - L_1 regularization minimizes sum of abs. values
  - L_2 regularization minimizes sum of squares

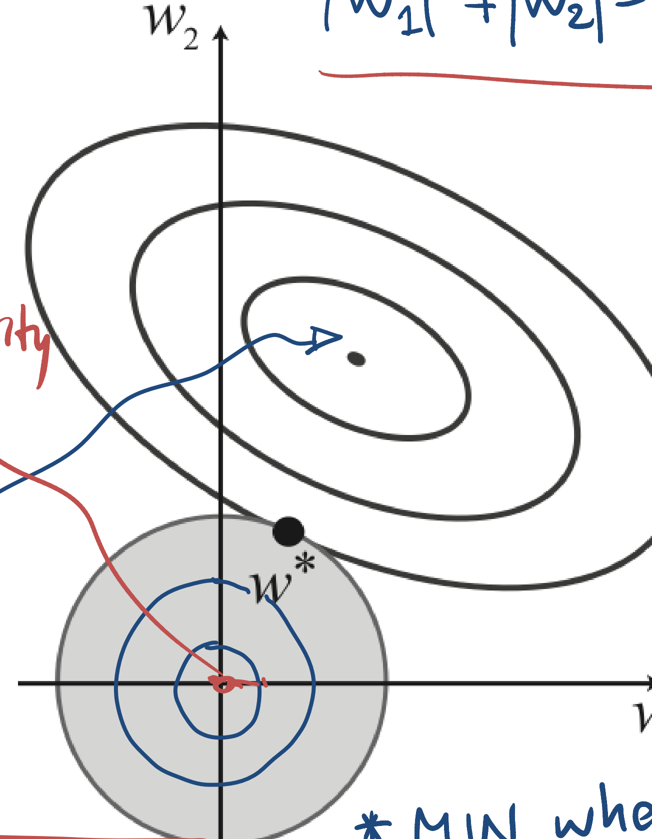
Regularization and Sparsity

$|w_1| + |w_2| = K$

$|w_1|^2 + |w_2|^2 = K$



MINs when considering only complexity



MIN

w^* MIN when combined

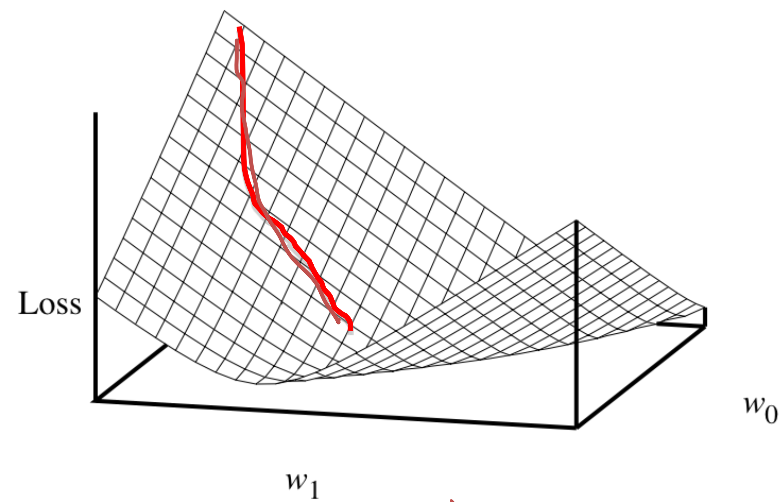
Cost(h) = EmpiricalLoss(h) + λ Complexity(h)

L_1 regularization

L_2 regularization

Regression by Gradient Descent

- No closed-form solution for complex loss functions



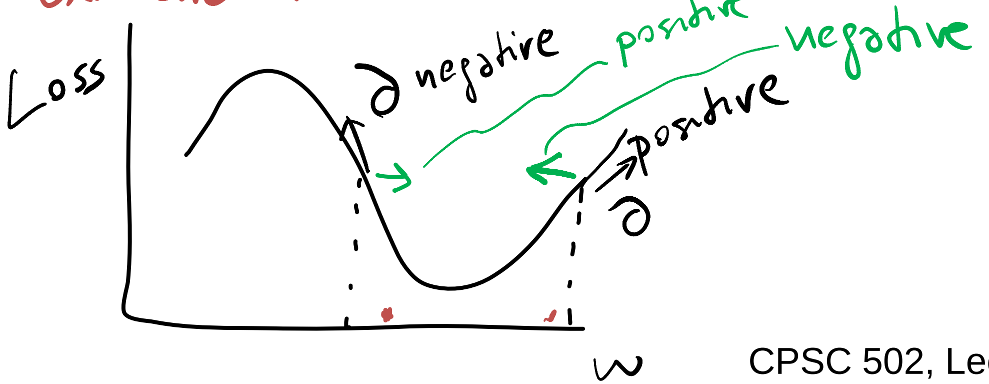
$w_1 \dots w_n$

w = any point
loop until convergence do:
for each w_i in **w** do:

$$w_i^m = w_i^{m-1} - \alpha \frac{\partial}{\partial w_i} Loss(w_i^{m-1})$$

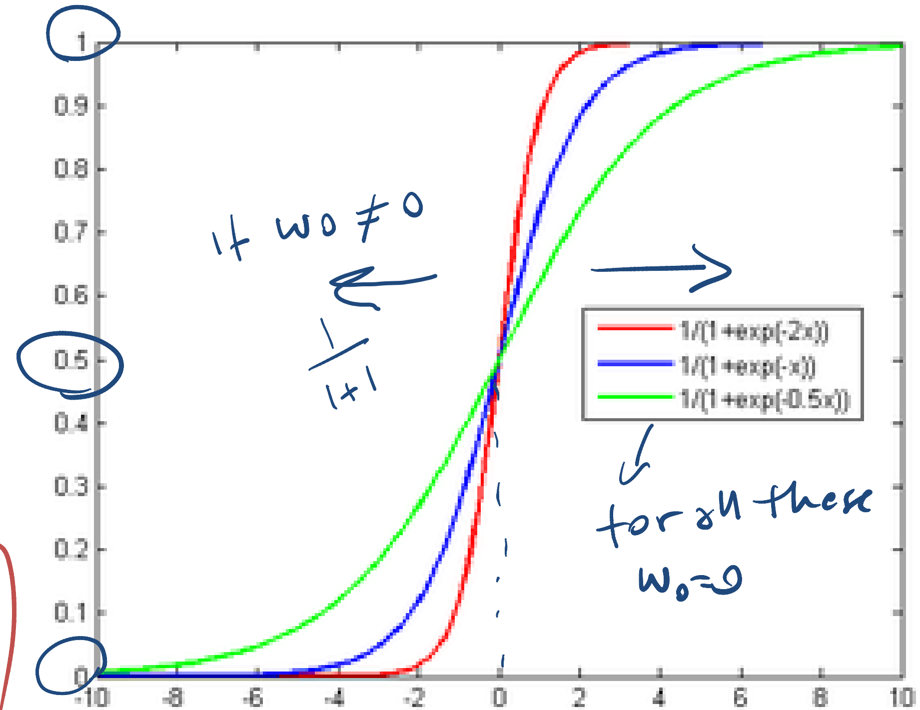
complex function

ex. one dimension



is it convex for L1 L2?

Logistic Regression



- Predicts prob. of y given z
- Fitting data to a logistic function

$$P(y) = \boxed{\frac{1}{1+e^{-w_0 - w_1 z}}}$$

generalized

$$\frac{1}{1+e^{-f(x)}}$$

$$y = f(x) = w_0 + w_1 x_1 + \cdots + w_n x_n$$

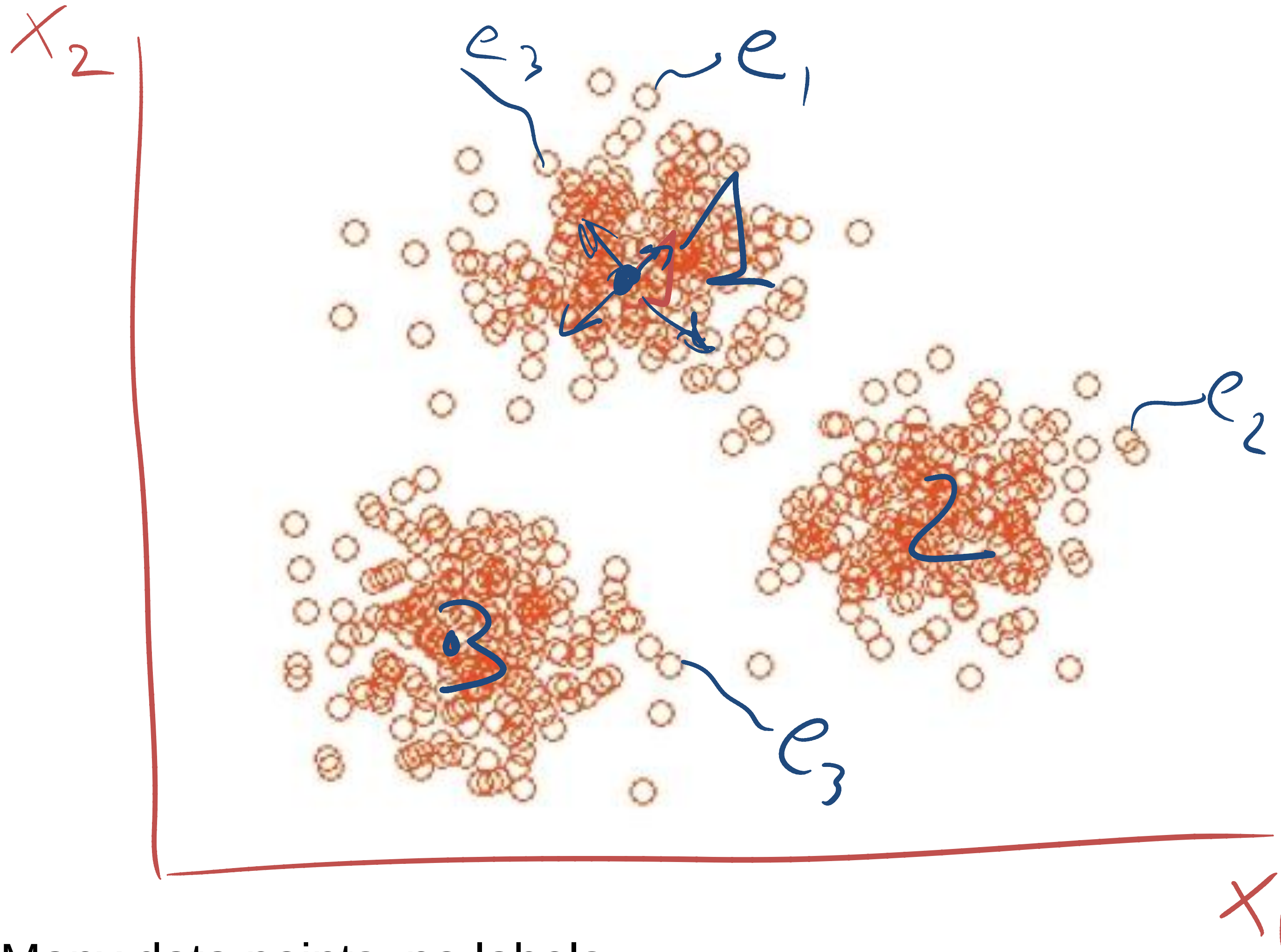
- Learning: iterative optimization (gradient)
- *Used in one of the NLP papers we will read*

Today Nov 3

- **Supervised Machine Learning**
  - Naïve Bayes
  - Markov-Chains
  - Decision Trees

    Classification y discrete
  - Regression y continuous
  - Logistic Regression $y \rightsquigarrow z \in [0,1]$
- **Unsupervised Machine Learning**
  - K-means
  - Intro to EM (very general technique!)

The unsupervised learning problem



Many data points, no labels

K-Means

Choose a fixed number of clusters

Ideally…

Choose cluster centers and point-cluster allocations to minimize error

can't do this by exhaustive search, because there are too many possible allocations.

MIN

$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of i'th cluster}} \left\| x_j - \mu_i \right\|^2 \right\}$$

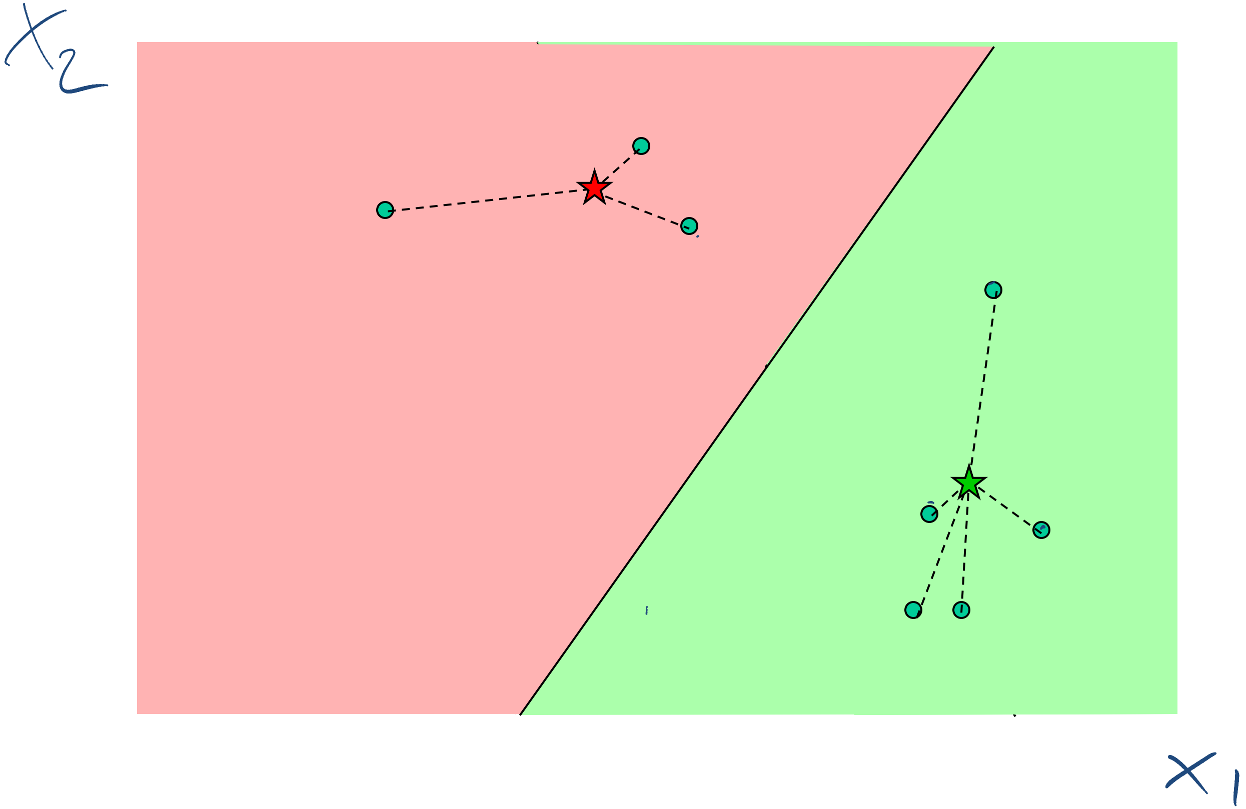
Algorithm

2 clusters

- Fix cluster centers;
- Allocate points to closest cluster
- With fixed allocation; compute best cluster centers
- Until nothing changes

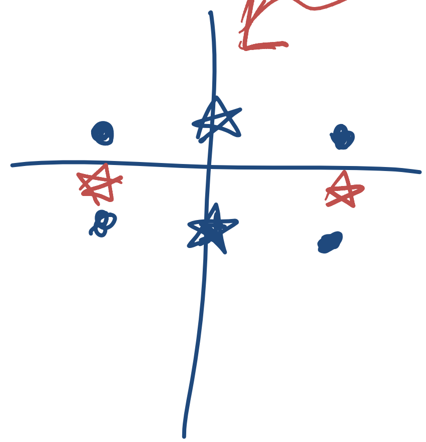
* From Marc Pollefeys COMP 256 2003

K-Means



Problems with K-means

better clusters



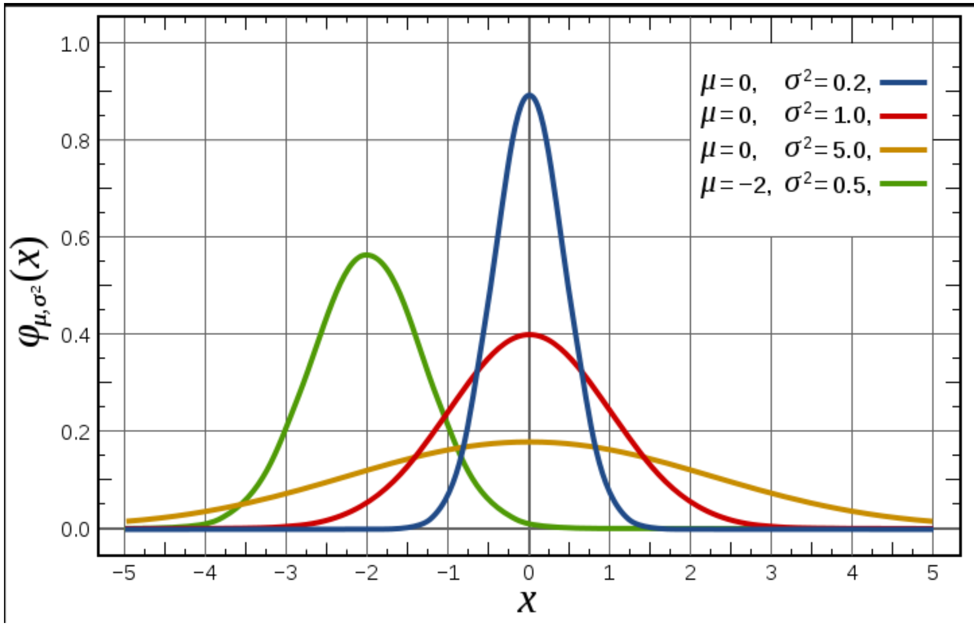
- Need to know *k*
- Local minima
- High dimensionality
- Lack of mathematical basis

Today Nov 3

- **Supervised Machine Learning**
  - Naïve Bayes
  - Markov-Chains
  - Decision Trees — Classification y discrete
  - Regression y continuous
  - Logistic Regression $y \rightsquigarrow z \in [0,1]$
- **Unsupervised Machine Learning**
  - K-means
  - Intro to EM (very general technique!)

Gaussian Distribution

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



- Models a large number of phenomena encountered in practice
- Under mild conditions the sum of a large number of random variables is distributed approximately normally

Gaussian Learning: Parameters

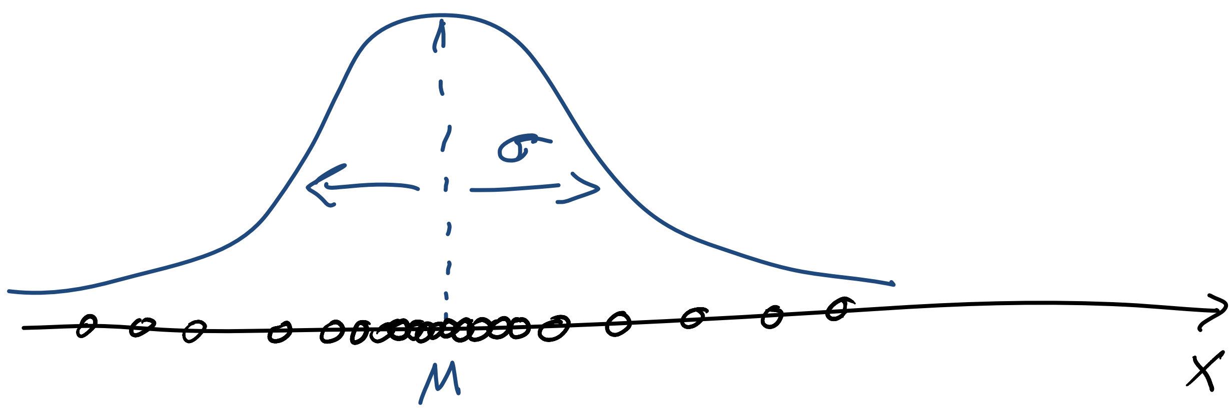
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$

average

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2$$

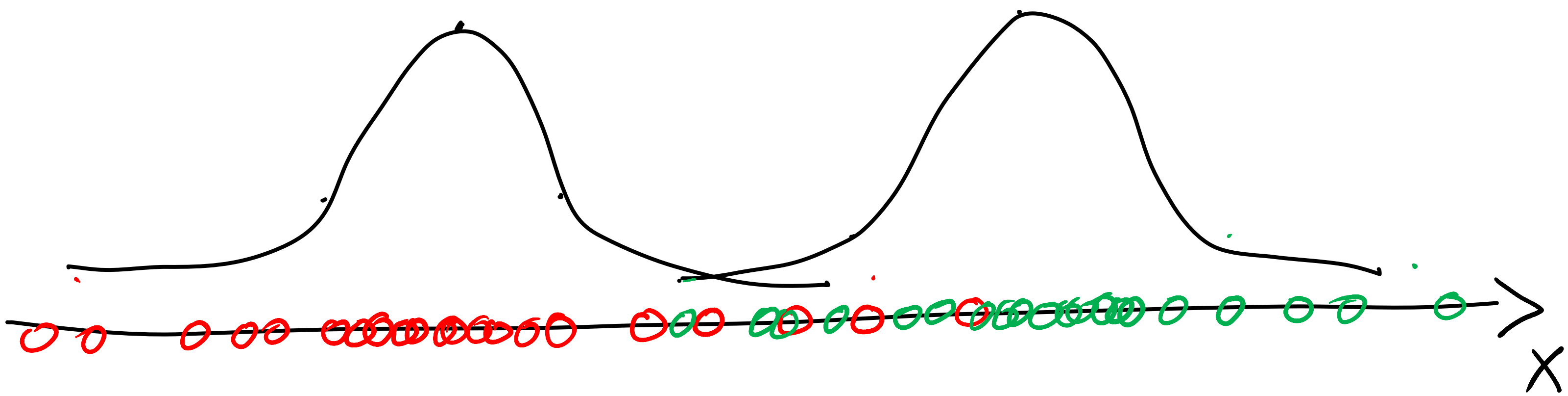
average deviation



- n data points

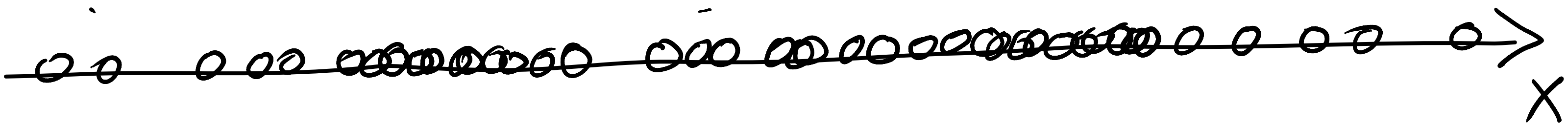
Expectation Maximzation for Clustering: Idea

- **Lets assume:** that our Data were generated from several Gaussians (a mixture, technically)
- **For simplicity** – one dimensional data – only two Gaussians (with same variance, but possibly different ……….)
- Generation Process
  - Gaussian/Cluster is selected
  - Data point is sampled from that cluster



But this is what we start from

- n data points without labels! And we have to cluster them into two (soft) clusters.



- "Identify the two Gaussians that best explain the data"
- Since we assume they have the same variance, we "just" need to find their priors and their means

- *In K-means we assume we know the center of the clusters and iterate…..*

Here we assume that we know

- Prior for clusters and the two means

$\theta_1 \quad \theta_2 \qquad \mu_1 \quad \mu_2$

- We can compute the probability that data point x_i corresponds to the cluster N_j $\quad P(N_J \mid X_1) = \dfrac{P(N_J, X_i)}{P(X_1)}$

$$z_{ij} = \frac{\theta_j * N(x_i \mid \mu_j, \sigma)}{\sum_{m=1}^{2} \theta_m * N(x_i \mid \mu_m, \sigma)}$$

$$N(x_i \mid \mu_j, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}$$

	N_1	N_2
X_1	.9	.1
X_2	.82	.18
X_3	. .	. .
⋮		
X_n	.01	.99

We can now recompute

- Prior for clusters

$$\theta_j = \frac{\sum_{i=1}^{n} z_{ij}}{n} \qquad \theta_1 = \frac{\sum_{i=1}^{n} z_{i1}}{n}$$

	N_1	N_2
x_1	.9	.1
x_2	.82	.18
x_3	.	.
⋮		
x_n	.01	.99

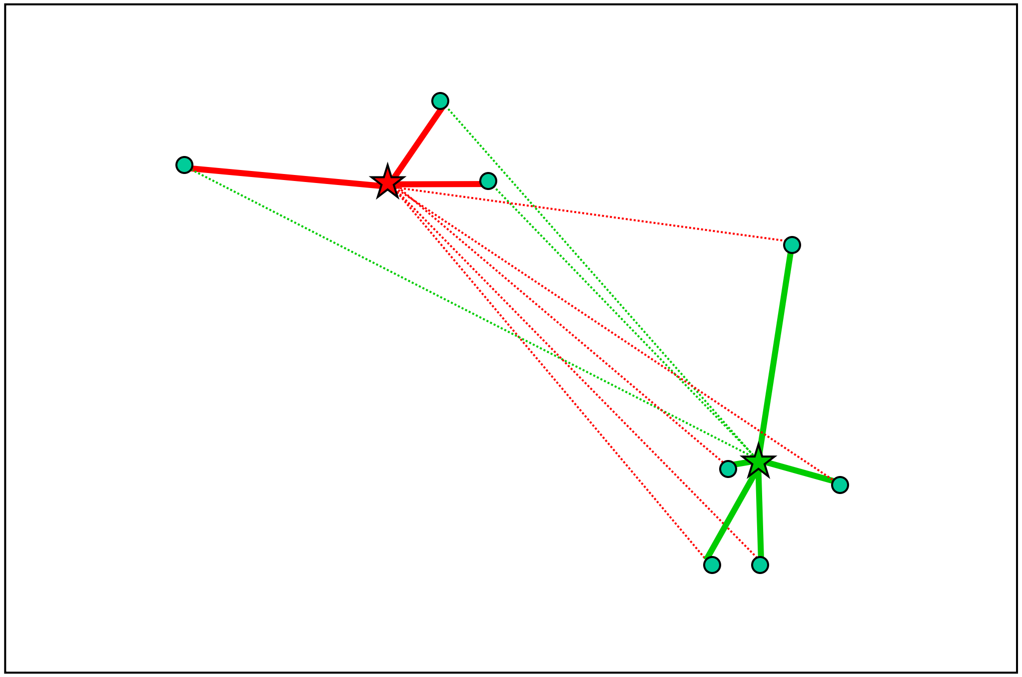
- The means

$$\mu_j = \frac{\sum_{i=1}^{n} z_{ij} x_i}{\sum_{i=1}^{n} z_{ij}} \qquad \mu_1 = \frac{\sum_{i=1}^{n} z_{i1} x_i}{\sum_{i=1}^{n} z_{i1}}$$

Hard cluster

$$\mu_1 = \frac{\sum \text{values of points in } N_1}{\#\text{ points in } N_1}$$

Intuition for EM in two dim.
(as a generalization of k-means)



Expectation Maximization

Converges! ☺

Proof [Neal/Hinton, McLachlan/Krishnan]:

- E/M step does not decrease data likelihood

But does not assure optimal solution ☹

Practical EM

Number of Clusters unknown

Algorithm:

- Guess initial # of clusters
- Run EM
  - ✓ Kill cluster center that doesn't contribute (two clusters with the same data)
  - ✓ Start new cluster center if many points "unexplained" (uniform cluster distribution for lots of data points)

EM is a very general method!

- **Baum-Welch algorithm** (also known as *forward-backward*): Learn HMMs from unlabeled data

- **Inside-Outside algorithm:** unsupervised induction of probabilistic context-free grammars.

- More generally, learn parameters for hidden variables in any Bnets (see textbook example 11.1.3 to learn parameters of NB classifier)

Machine Learning: Where are we?

Supervised Learning

- Examples of correct answers are given
  - Discrete answers: **Classification**
  - Continuous answers: **Regression**

Unsupervised Learning

- No feedback from teacher; detect patterns

Next Week: Reinforcement Learning

- Feedback consists of rewards/punishment (Robotics, Interactive Systems)

TODO for next Tue

- Read 11.3: Reinforcement Learning
- Assignment 3-Part2 out soon