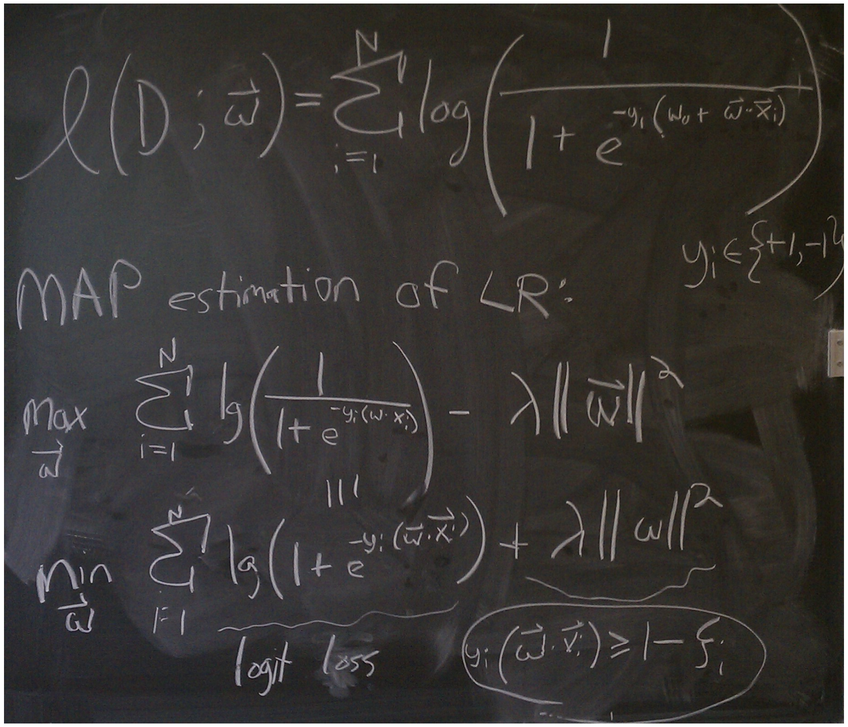
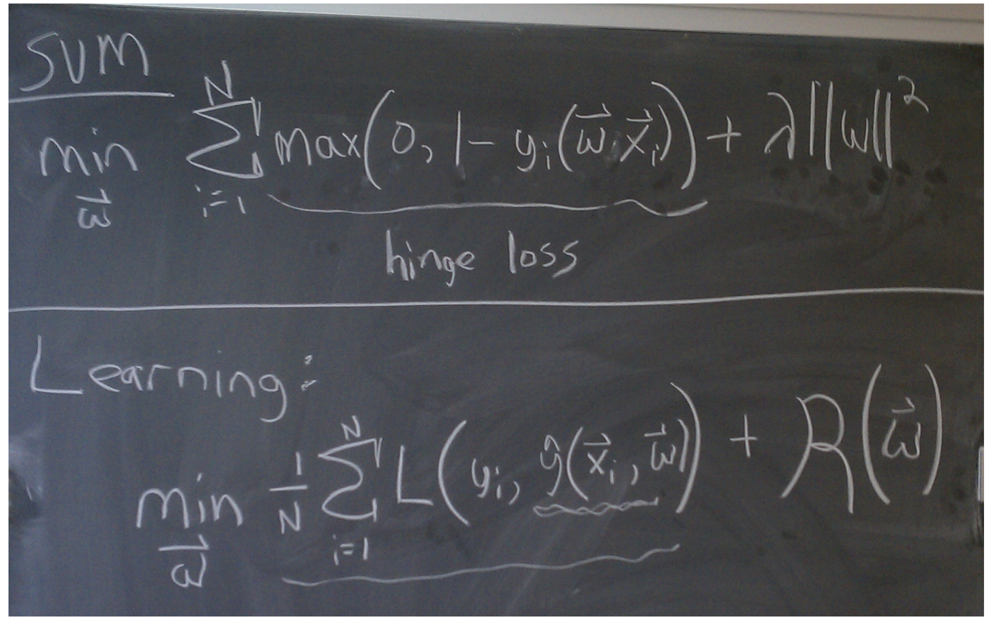
Regression
Lecture 20

David Sontag
New York University

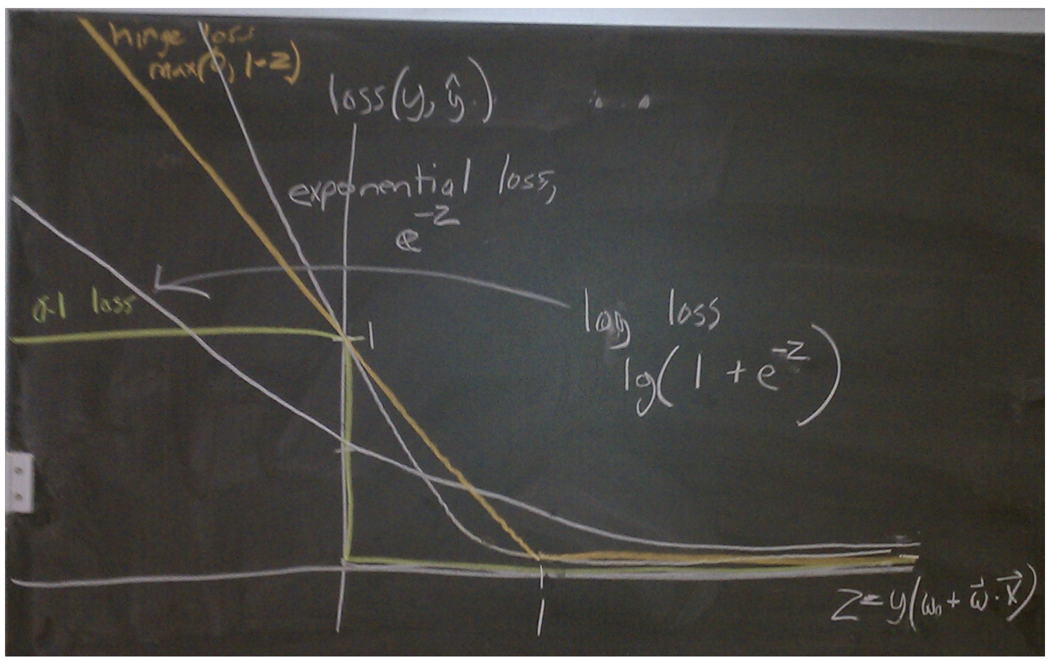
Slides adapted from Vibhav Gogate, Luke Zettlemoyer, Carlos Guestrin, and Dan Weld



$\ell(D;\vec{w}) = \sum_{i=1}^{N} \log\left(\frac{1}{1+e^{-y_i(w_0+\vec{w}\cdot\vec{x}_i)}}\right)$
$y_i \in \{+1,-1\}$
MAP estimation of LR:
$\max_{\vec{w}} \sum_{i=1}^{N} \lg\left(\frac{1}{1+e^{-y_i(w\cdot x_i)}}\right) - \lambda\|\vec{w}\|^2$
$\min_{\vec{w}} \sum_{i=1}^{N} \lg\left(1+e^{-y_i(\vec{w}\cdot\vec{x}_i)}\right) + \lambda\|w\|^2$
logit loss
$y_i(\vec{w}\cdot\vec{x}_i) \geq 1-\xi_i$



SVM
$\min_{\vec{w}} \sum_{i=1}^{N} \max(0, 1-y_i(\vec{w}\cdot\vec{x}_i)) + \lambda\|w\|^2$
hinge loss
Learning:
$\min_{\vec{w}} \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{y}(\vec{x}_i, \vec{w})) + R(\vec{w})$



hinge loss
max(0, 1-z)
loss(y, ŷ)
exponential loss,
e^{-z}
0-1 loss
log loss
$\lg(1+e^{-z})$
1
1
$z = y(w_0 + \vec{w}\cdot\vec{x})$

Naïve Bayes *vs.* Logistic Regression

Learning: $h: \mathbf{X} \mapsto Y$

X – features

Y – target classes

Generative

- Assume functional form for
  - P(X|Y) **assume cond indep**
  - P(Y)
  - Est params from train data
- Gaussian NB for cont features
- Bayes rule to calc. P(Y|X= x)
  - $P(Y \mid \mathbf{X}) \propto P(\mathbf{X} \mid Y)\, P(Y)$
- **Indirect** computation
  - Can also generate a sample of the data

Discriminative

- Assume functional form for
  - P(Y|X) **no assumptions**
  - Est params from training data
- Handles discrete & cont features
- Directly calculate P(Y|X=x)
  - Can't generate data sample

Naïve Bayes vs. Logistic Regression

[Ng & Jordan, 2002]

- Generative vs. Discriminative classifiers
- Asymptotic comparison

  (# training examples → infinity)

  - when model correct
    - NB and LDA (with class independent variances) and Logistic Regression produce identical classifiers
  - when model incorrect
    - LR is less biased – does not assume conditional independence
      - **therefore LR expected to outperform NB**

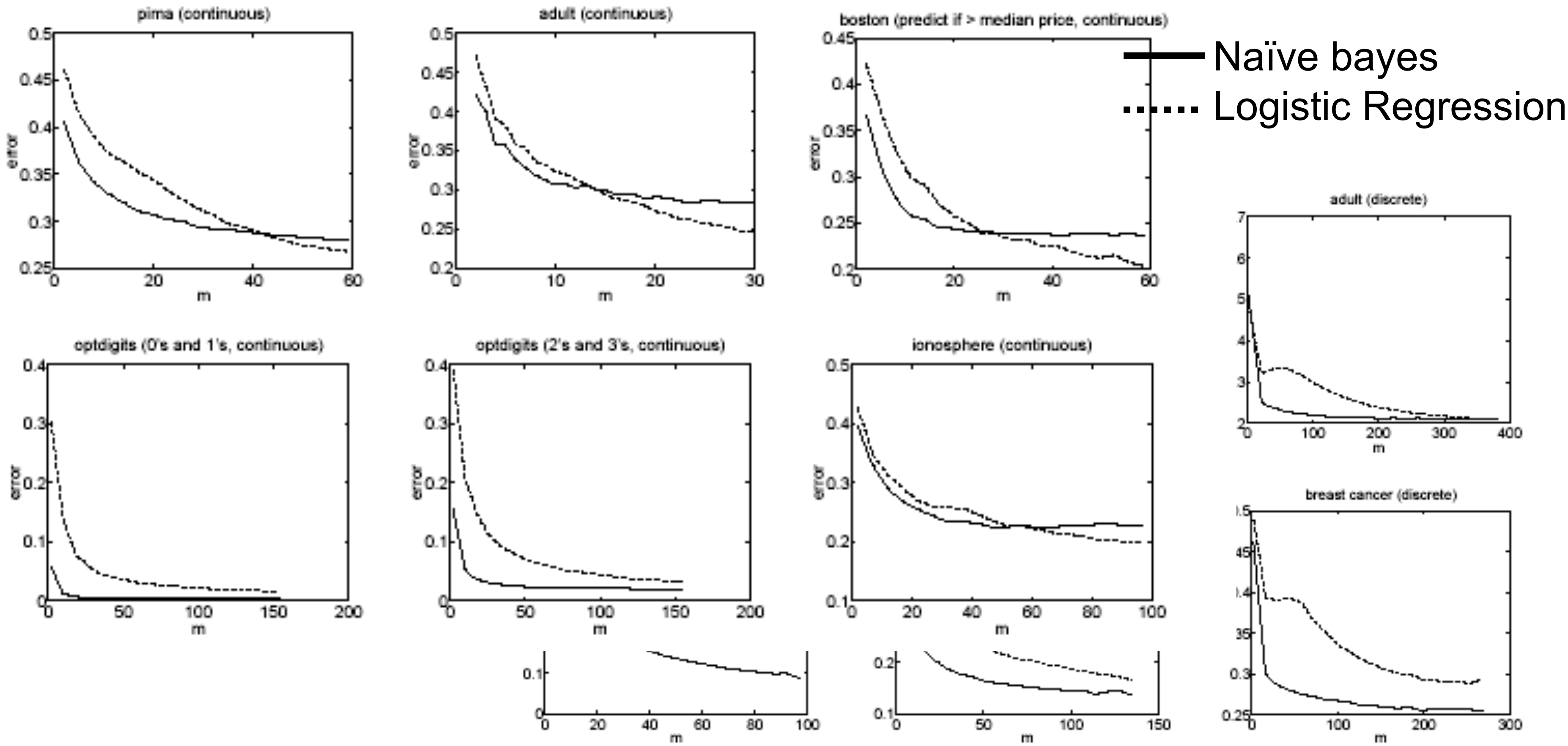
Naïve Bayes vs. Logistic Regression

[Ng & Jordan, 2002]

- Generative vs. Discriminative classifiers
- Non-asymptotic analysis
  - convergence rate of parameter estimates,

    (n = # of attributes in X)
    - Size of training data to get close to infinite data solution
    - Naïve Bayes needs *O*(log n) samples
    - Logistic Regression needs *O*(n) samples

  - Naïve Bayes converges more quickly to its ***(perhaps less helpful)*** asymptotic estimates



Some experiments from UCI data sets

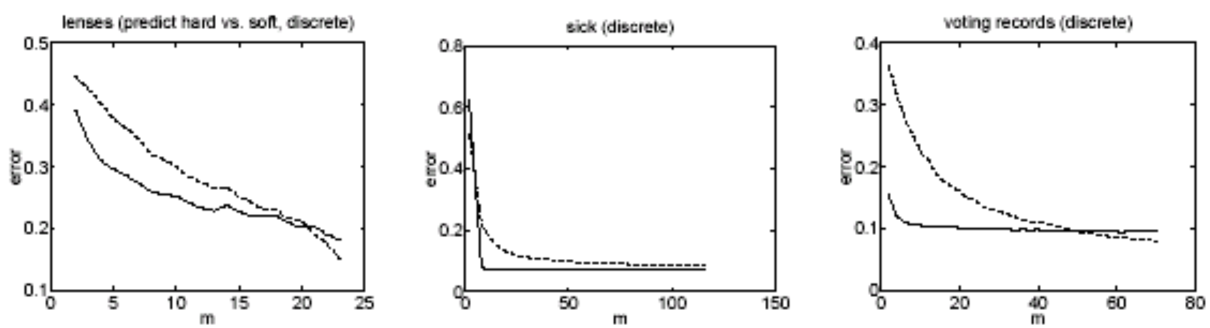


Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning repository. Plots are of generalization error vs. m (averaged over 1000 random train/test splits). Dashed line is logistic regression; solid line is naive Bayes.

Logistic regression for discrete classification

Logistic regression in more general case, where set of possible *Y* is $\{y_1,\ldots,y_R\}$

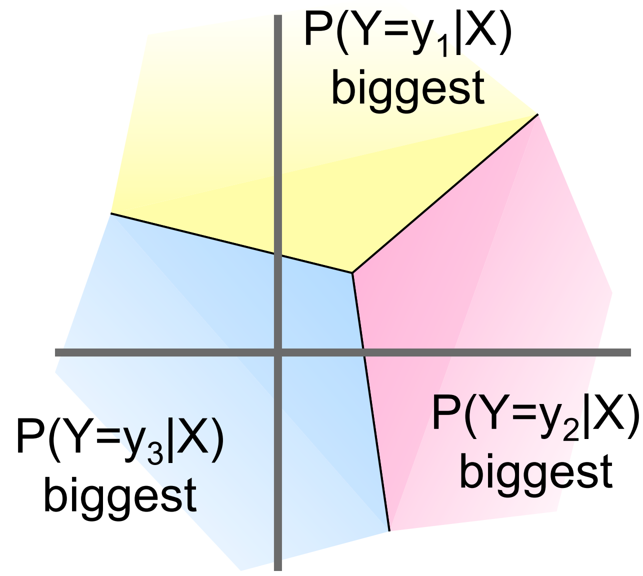
- Define a weight vector w_i for each y_i, i=1,…,R-1

$$P(Y=1|X) \propto \exp(w_{10} + \sum_i w_{1i} X_i)$$

$$P(Y=2|X) \propto \exp(w_{20} + \sum_i w_{2i} X_i)$$

$$\ldots$$

$$P(Y=r|X) = 1 - \sum_{j=1}^{r-1} P(Y=j|X)$$



Logistic regression for discrete classification

- Logistic regression in more general case, where
  Y is in the set $\{y_1,\ldots,y_R\}$

  for $k<R$

$$P(Y=y_k|X)=\frac{\exp(w_{k0}+\sum_{i=1}^{n} w_{ki}X_i)}{1+\sum_{j=1}^{R-1}\exp(w_{j0}+\sum_{i=1}^{n} w_{ji}X_i)}$$

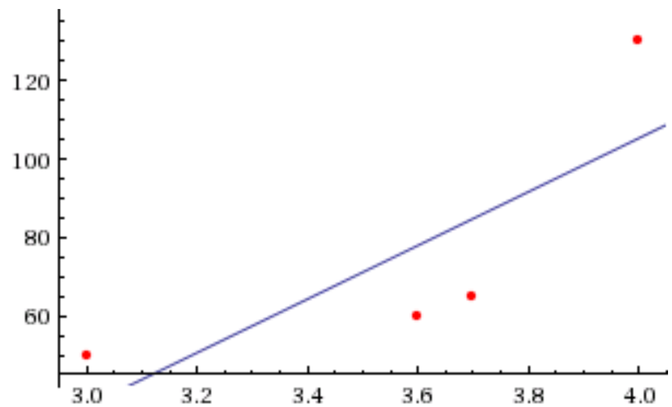
  for $k=R$ (normalization, so no weights for this class)

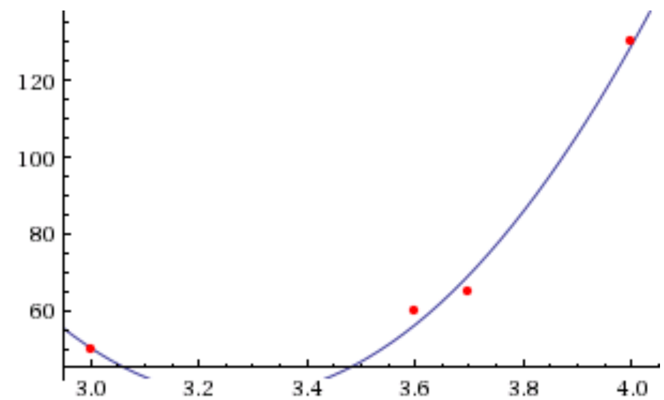
$$P(Y=y_R|X)=\frac{1}{1+\sum_{j=1}^{R-1}\exp(w_{j0}+\sum_{i=1}^{n} w_{ji}X_i)}$$

Features can be discrete or continuous!

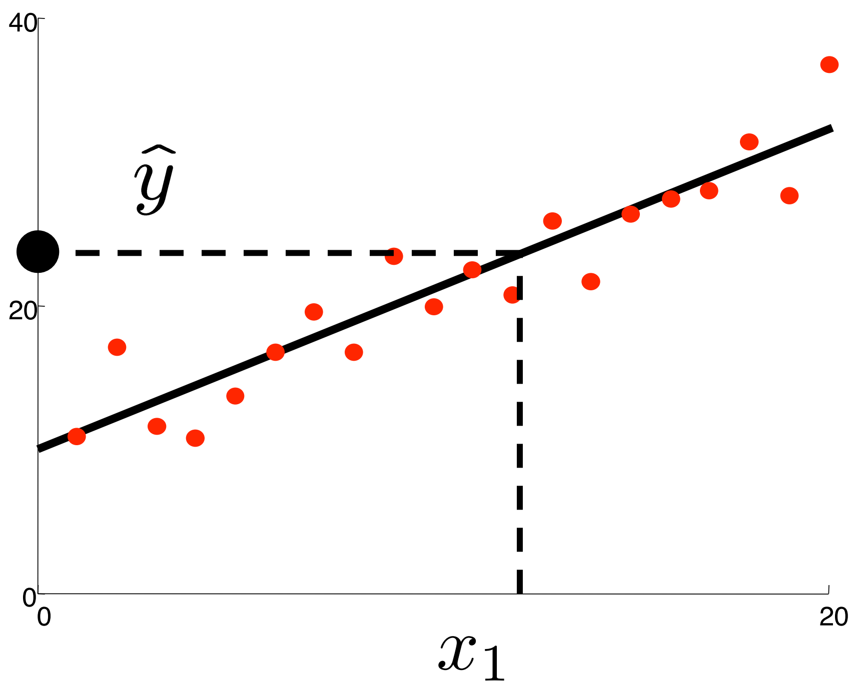
Prediction of continuous variables

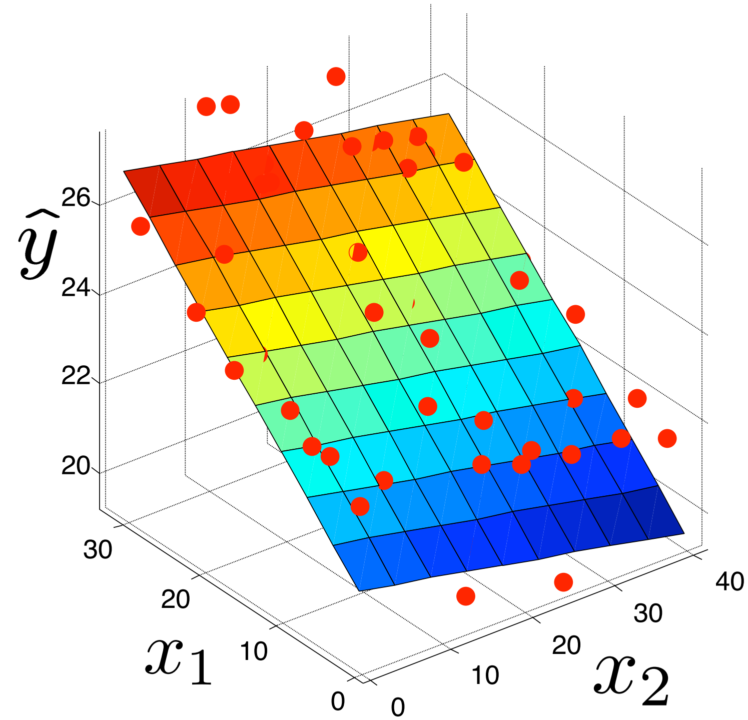
- Billionaire says: Wait, that's not what I meant!
- You say: Chill out, dude.
- He says: I want to predict a continuous variable for continuous inputs: I want to predict salaries from GPA.
- You say: **I can regress that…**





Linear Regression





Prediction

$$\hat{y} = w_0 + w_1 x_1$$

Prediction

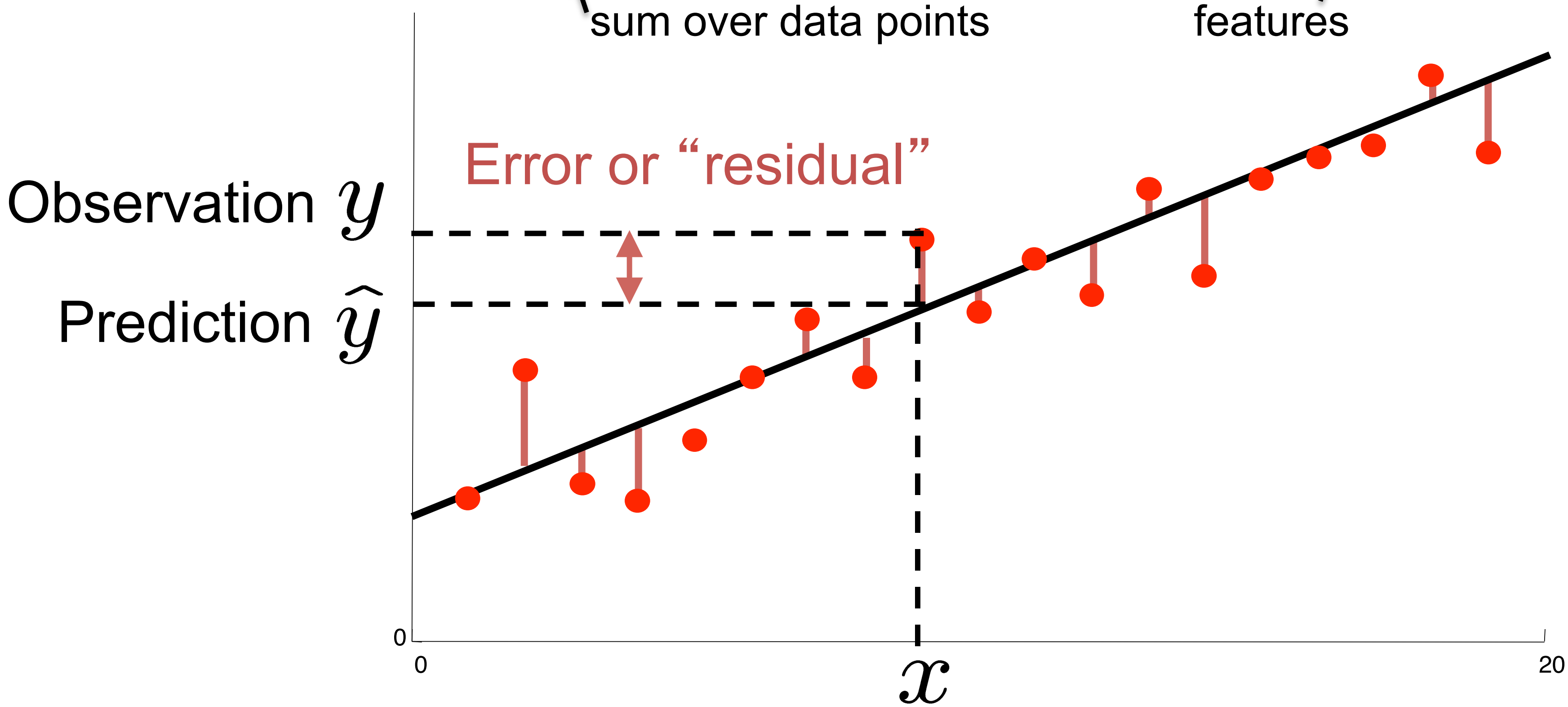
$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2$$

Ordinary Least Squares (OLS)

$$\text{total error} = \sum_i (y_i - \hat{y}_i)^2 = \sum_i \left(y_i - \sum_k w_k x_k^{(i)} \right)^2$$

sum over data points

features



The regression problem

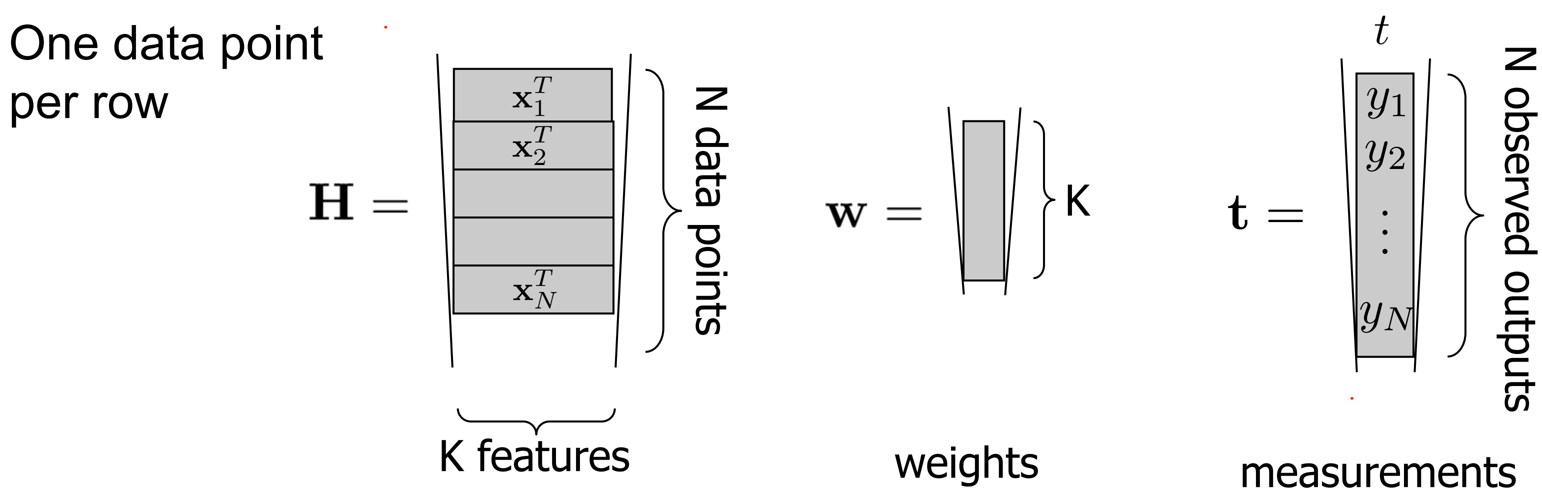
- Precisely, minimize the residual squared error:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_i \left(y_i - \sum_k w_k x_i^k \right)^2$$

Regression: matrix notation

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_i \left(y_i - \sum_k w_k x_i^k \right)^2 = \sum_i \left(\mathbf{x}_i^T \mathbf{w} - y_i \right)^2$$

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

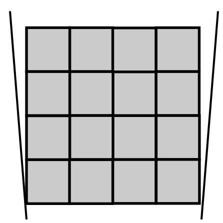


Regression solution: simple matrix math

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

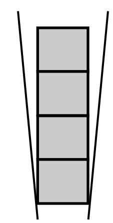
$$\text{solution:}\ \ \mathbf{w}^* = \underbrace{(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}}_{\mathbf{A}^{-1}} \underbrace{\mathbf{H}^{\mathrm{T}}\mathbf{t}}_{\mathbf{b}} = \mathbf{A}^{-1}\mathbf{b}$$

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



K×K matrix of feature correlations

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



K×1 vector

But, why?

- Billionaire (again) says: Why sum squared error???
- You say: Gaussians, Dr. Gateson, Gaussians…
- Model: prediction is **deterministic** linear function **plus Gaussian noise**:

$$y_{\text{observed}} = \sum_k w_k x_k + \epsilon \qquad\qquad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

- Learn **w** using MLE:

$$\Pr(y_{\text{observed}} \mid \mathbf{x}, \mathbf{w}, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(y_{\text{observed}} - \sum_k w_k x_k)^2}{2\sigma^2}}$$

Maximizing log-likelihood

Maximize wrt w:

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

$$\arg\max_w \ln \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N + \sum_{j=1}^{N} \frac{-[t_j - \sum_i w_i h_i(x_j)]^2}{2\sigma^2}$$

$$= \arg\max_w \sum_{j=1}^{N} \frac{-[t_j - \sum_i w_i h_i(x_j)]^2}{2\sigma^2}$$

$$= \arg\min_w \sum_{j=1}^{N} [t_j - \sum_i w_i h_i(x_j)]^2$$

(note that the notation here is slightly different)

Least-squares Linear Regression is MLE for Gaussian noise!!!