Regression

Practical Machine Learning

Fabian Wauthier

09/10/2009

Adapted from slides by Kurt Miller and Romain Thibaux

Outline

- Ordinary Least Squares Regression
  - Online version
  - Normal equations
  - Probabilistic interpretation
- Overfitting and Regularization
- Overview of additional topics
  - L_1 Regression
  - Quantile Regression
  - Generalized linear models
  - Kernel Regression and Locally Weighted Regression

Outline

- Ordinary Least Squares Regression
  - Online version
  - Normal equations
  - Probabilistic interpretation
- Overfitting and Regularization
- Overview of additional topics
  - L_1 Regression
  - Quantile Regression
  - Generalized linear models
  - Kernel Regression and Locally Weighted Regression

Regression vs. Classification:

Classification

$$X \Rrightarrow Y$$

Anything:

- continuous ($\Re$, $\Re^d$, …)
- discrete ({0,1}, {1,…k}, …)
- structured (tree, string, …)
- …

- Discrete:
  - {0,1} *binary*
  - {1,…k} *multi-class*
  - tree, etc. *structured*

Regression vs. Classification:

Classification

$$X \Rightarrow Y$$

Anything:

- continuous ($\Re$, $\Re^d$, …)
- discrete ({0,1}, {1,…k}, …)
- structured (tree, string, …)
- …

Kernel trick

Perceptron
Logistic Regression
Support Vector Machine

Decision Tree
Random Forest

Regression vs. Classification:

Regression

$$X \Rrightarrow Y$$

Anything:

- continuous ($\Re$, $\Re^d$, …)
- discrete ({0,1}, {1,…k}, …)
- structured (tree, string, …)
- …

- continuous:
  - $\Re$, $\Re^d$

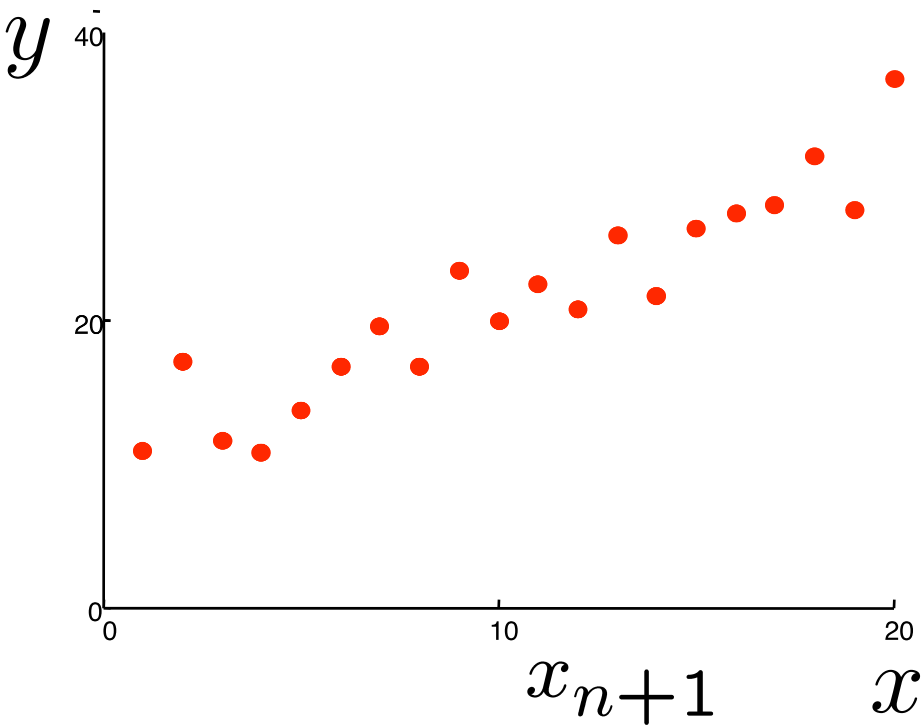
Examples

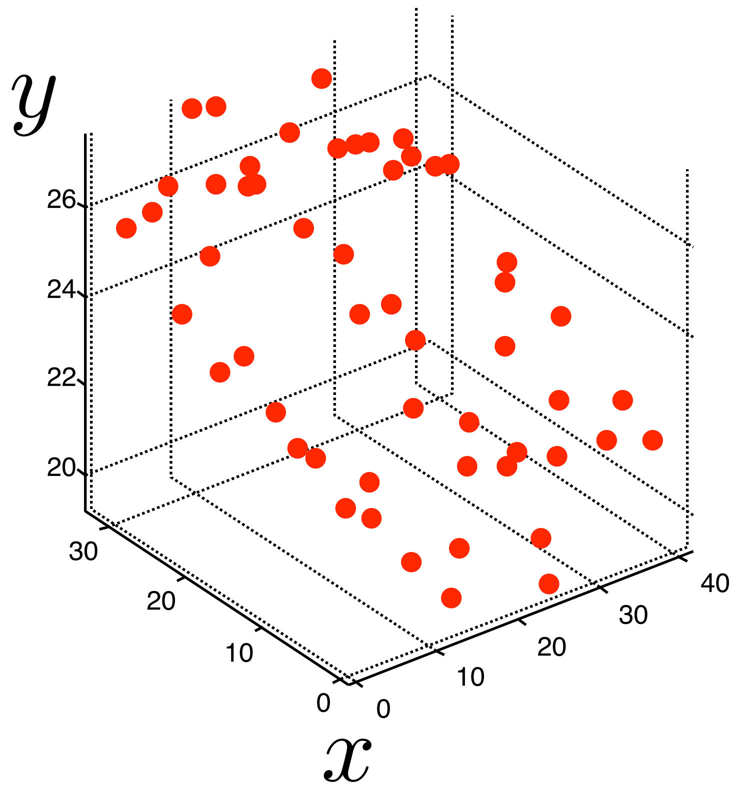
- Voltage $\Rightarrow$ Temperature
- Processes, memory $\Rightarrow$ Power consumption
- Protein structure $\Rightarrow$ Energy
- Robot arm controls $\Rightarrow$ Torque at effector
- Location, industry, past losses $\Rightarrow$ Premium

Linear regression

Given examples $(x_i, y_i)_{i=1...n}$

Predict y_{n+1} given a new point x_{n+1}



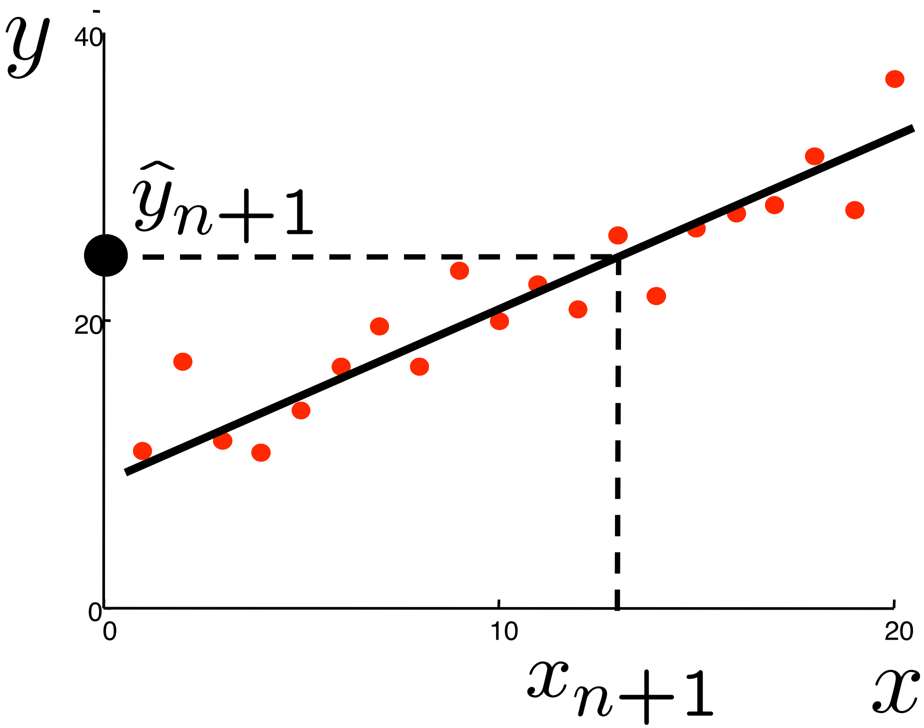


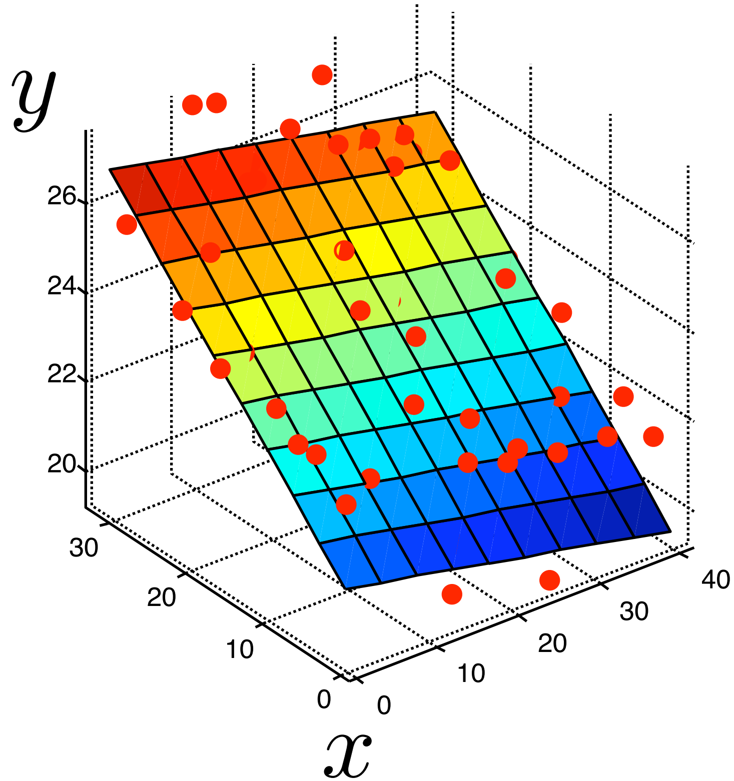
Linear regression

We wish to estimate $\hat{y}$ by a linear function of our data x :

$$\begin{aligned} \hat{y}_{n+1} &= w_0 + w_1 x_{n+1,1} + w_2 x_{n+1,2} \\ &= w^\top x_{n+1} \end{aligned}$$

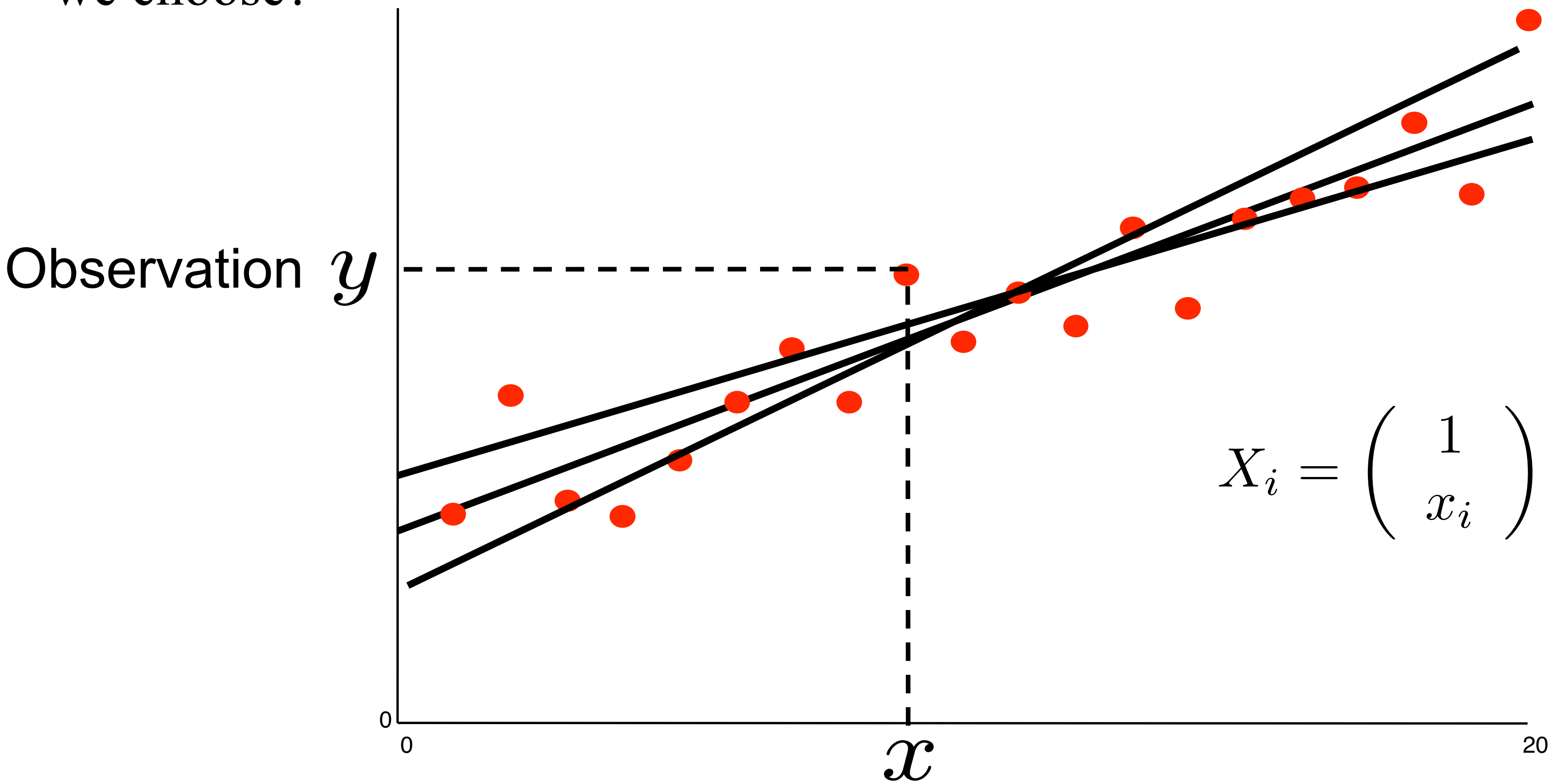
where w is a parameter to be estimated and we have used the standard convention of letting the first component of x be 1.





Choosing the regressor

Of the many regression fits that approximate the data, which should we choose?

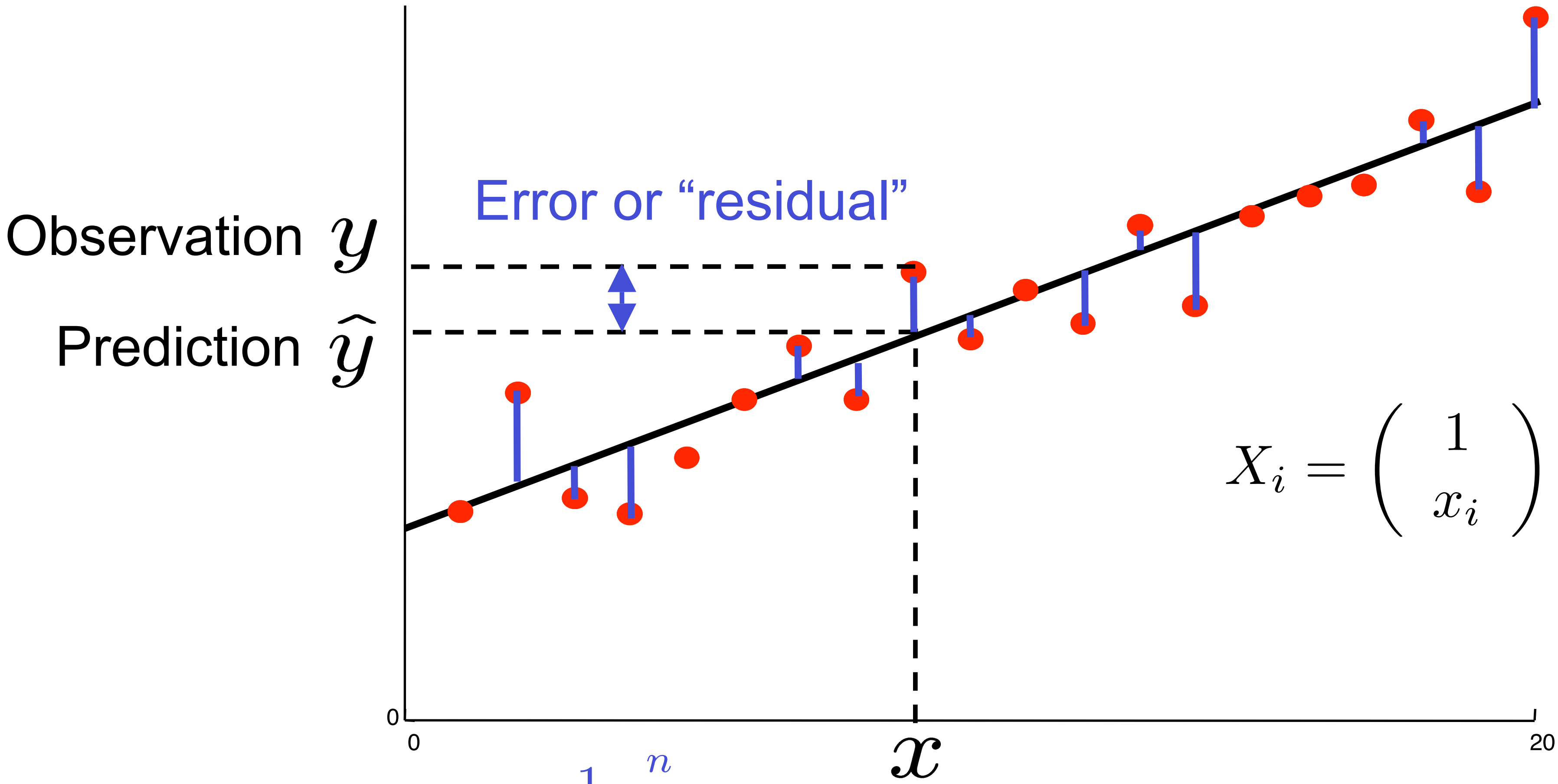


10

LMS Algorithm

(Least Mean Squares)

In order to clarify what we mean by a good choice of w, we will define a cost function for how well we are doing on the training data:



$$\text{Cost} = \frac{1}{2}\sum_{i=1}^{n}(w^\top x_i - y_i)^2$$

LMS Algorithm
(Least Mean Squares)

The best choice of w is the one that minimizes our cost function

$$E = \frac{1}{2}\sum_{i=1}^{n}(w^\top x_i - y_i)^2 = \sum_{i=1}^{n} E_i$$

In order to optimize this equation, we use standard gradient descent

$$w^{t+1} := w^t - \alpha \frac{\partial}{\partial w} E$$

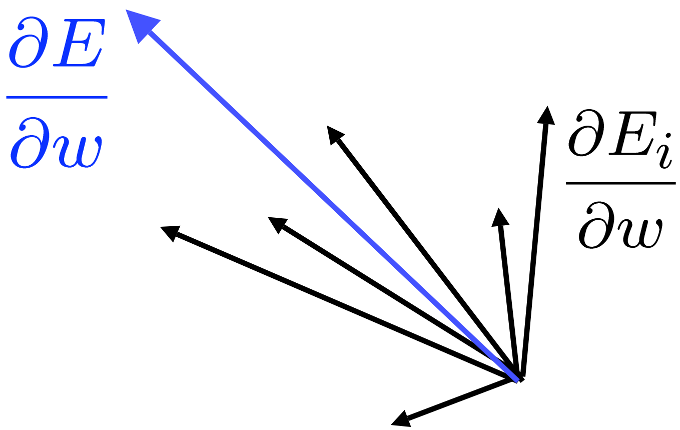
where

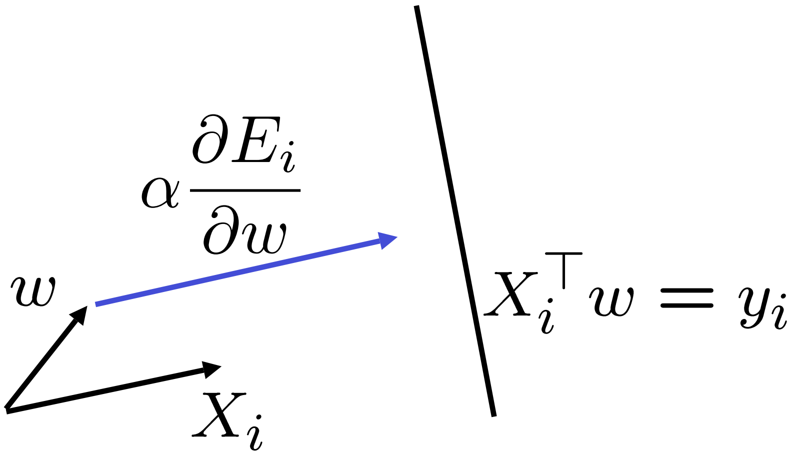
$$\frac{\partial}{\partial w} E = \sum_{i=1}^{n} \frac{\partial}{\partial w} E_i \qquad \text{and} \qquad \begin{aligned} \frac{\partial}{\partial w} E_i &= \frac{1}{2}\frac{\partial}{\partial w}(w^\top x_i - y_i)^2 \\ &= (w^\top x_i - y_i)x_i \end{aligned}$$

LMS Algorithm
(Least Mean Squares)

The LMS algorithm is an online method that performs the following update for each new data point

$$
\begin{aligned}
w^{t+1} &:= w^t - \alpha \frac{\partial}{\partial w} E_i \\
&= w^t + \alpha (y_i - x_i^\top w) x_i
\end{aligned}
$$





LMS, Logistic regression, and Perceptron updates

- LMS

$$w^{t+1} \quad := \quad w^t + \alpha(y_i - x_i^\top w)x_i$$

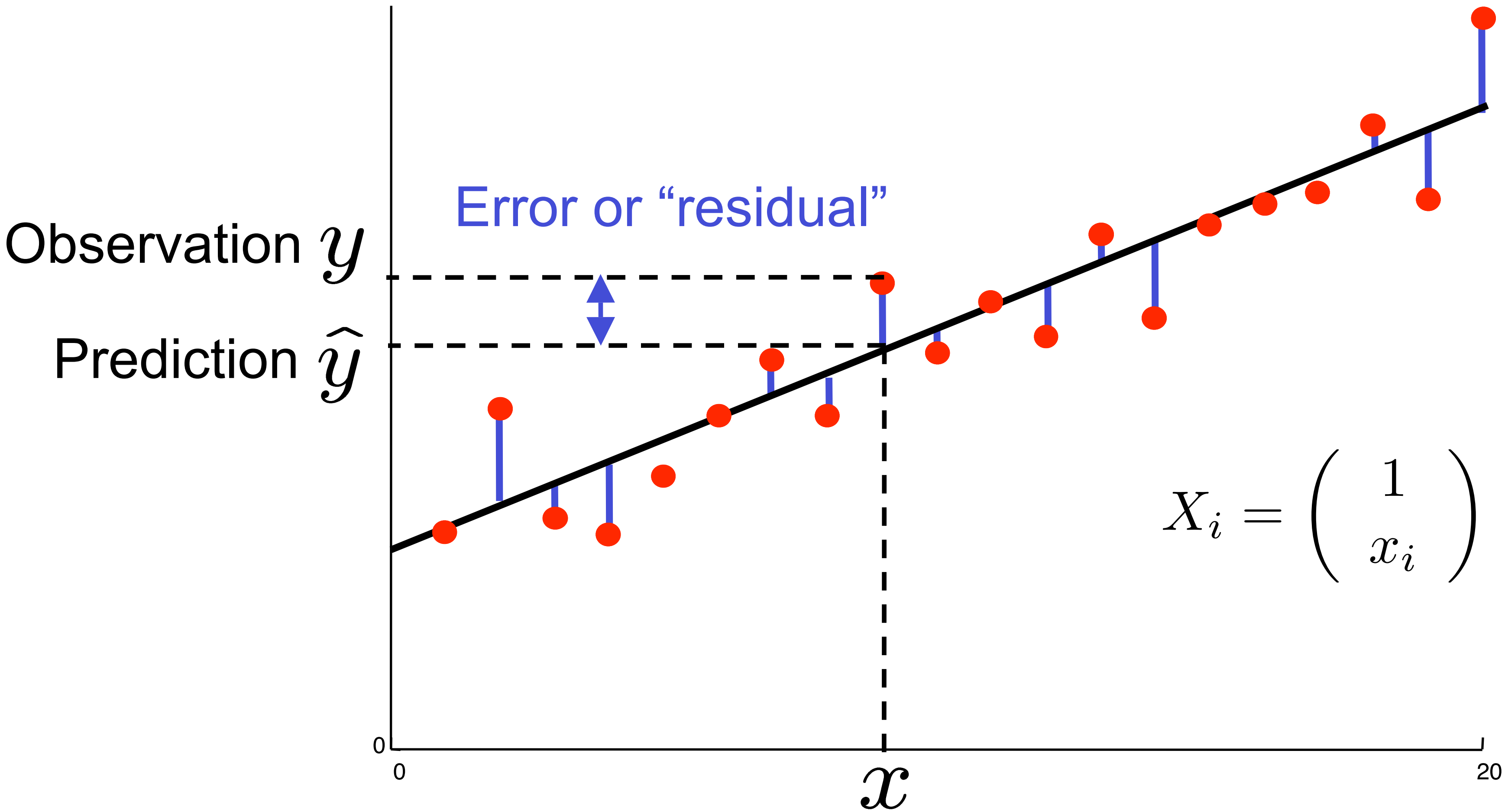
- Logistic Regression

$$w^{t+1} \quad := \quad w^t + \alpha(y_i - f_w(x_i))x_i$$

- Perceptron

$$w^{t+1} \quad := \quad w^t + \alpha(y_i - f_w(x_i))x_i$$

Ordinary Least Squares (OLS)



$$\text{Cost} = \frac{1}{2}\sum_{i=1}^{n}(w^\top x_i - y_i)^2$$

Minimize the sum squared error

$$
\begin{aligned}
E &= \frac{1}{2}\sum_{i=1}^{n}(w^\top x_i - y_i)^2 \\
&= \frac{1}{2}(Xw - y)^\top (Xw - y) \\
&= \frac{1}{2}(w^\top X^\top X w - 2y^\top X w + y^\top y)
\end{aligned}
$$

$$X = \begin{pmatrix} -x_1^\top - \\ -x_2^\top - \\ \dots \end{pmatrix}$$

n

d

$$\frac{\partial}{\partial w}E = X^\top X w - X^\top y$$

Setting the derivative equal to zero gives us the *Normal Equations*

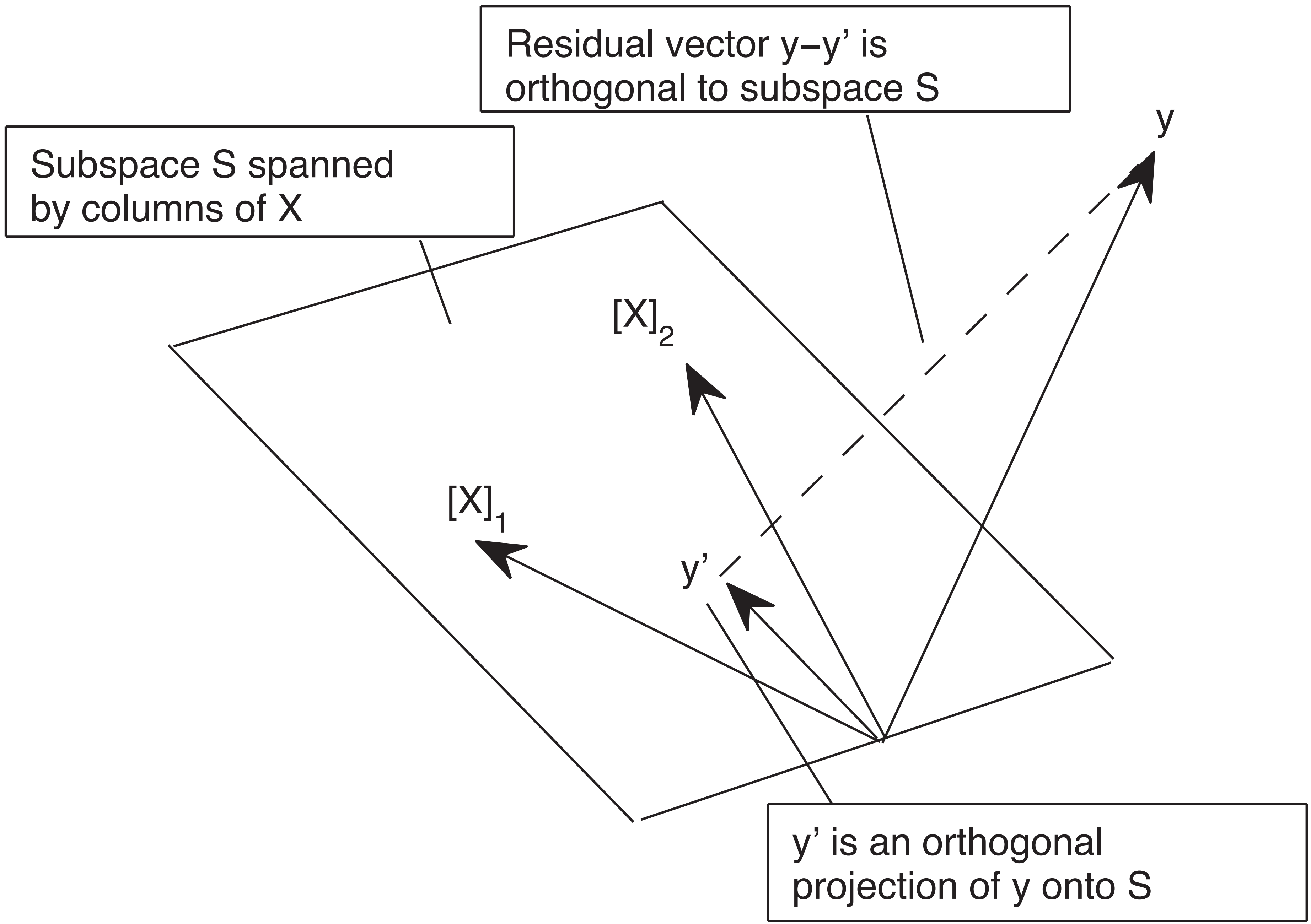
$$
\begin{aligned}
X^\top X w &= X^\top y \\
w &= (X^\top X)^{-1} X^\top y
\end{aligned}
$$

A geometric interpretation

We solved $\frac{\partial}{\partial w} E = X^\top (Xw - y) = 0$

$\Rightarrow$ Residuals are orthogonal to columns of X

$\Rightarrow$ $\hat{y} = Xw$ gives the best reconstruction of y in the range of X



Residual vector y–y’ is orthogonal to subspace S
y
Subspace S spanned by columns of X
[X]2
[X]1
y’
y’ is an orthogonal projection of y onto S

Computing the solution

We compute $w = (X^\top X)^{-1} X^\top y$.

If $X^\top X$ is invertible, then $(X^\top X)^{-1} X^\top$ coincides with the pseudoinverse X^+ of X and the solution is unique.

If $X^\top X$ is not invertible, there is no unique solution w.

In that case $w = X^+ y$ chooses the solution with smallest Euclidean norm.

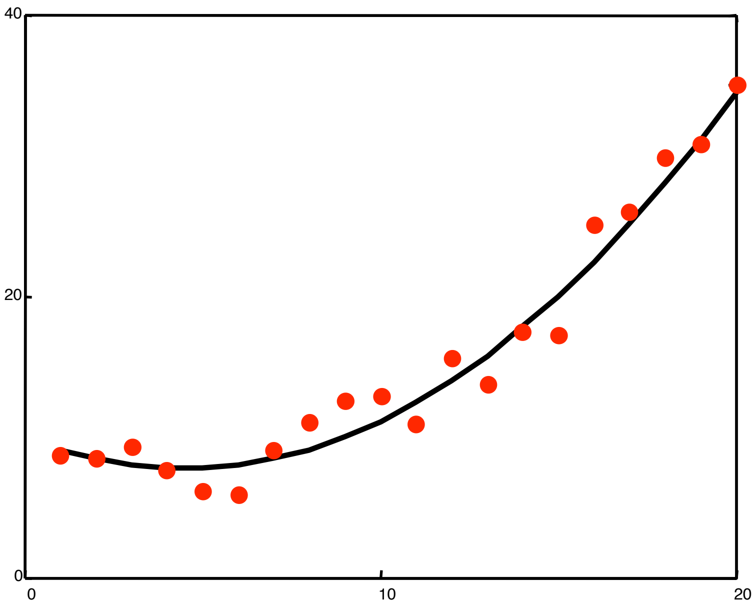
An alternative way to deal with non-invertible $X^\top X$ is to add a small portion of the identity matrix (= Ridge regression).

19

Beyond lines and planes

Linear models become powerful function approximators when we consider non-linear feature transformations.

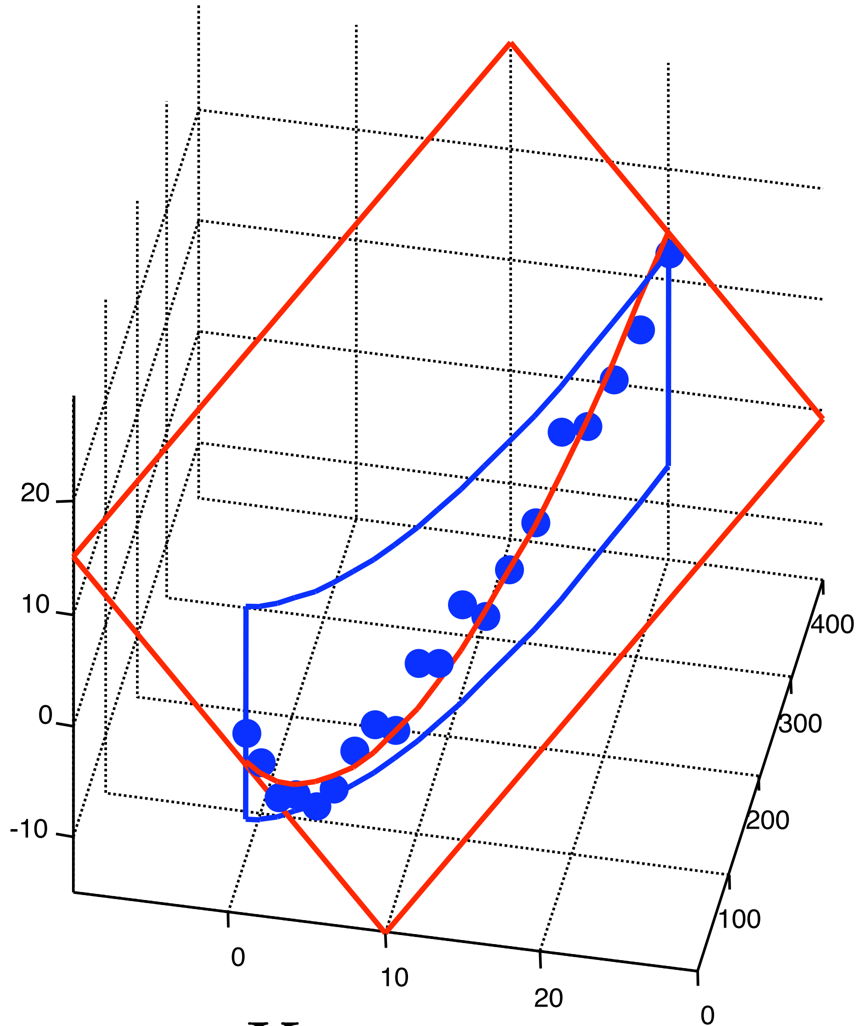
$$X_i = \begin{pmatrix} 1 \\ x_i \\ x_i^2 \end{pmatrix} \Longrightarrow \hat{y}_i = w_0 + w_1 x_i + w_2 x_i^2$$



Predictions are still linear in X !

All the math is the same!

Geometric interpretation



$$\hat{y} = w_0 + w_1 x + w_2 x^2$$

$X_2 = x^2$

$X_1 = x$

[Matlab demo]

Ordinary Least Squares [summary]

Given examples $(x_i, y_i)_{i=1\ldots n}$

Let $X_i^\top = \begin{pmatrix} f_1(x_i) & f_2(x_i) & \ldots & f_d(x_i) \end{pmatrix}$

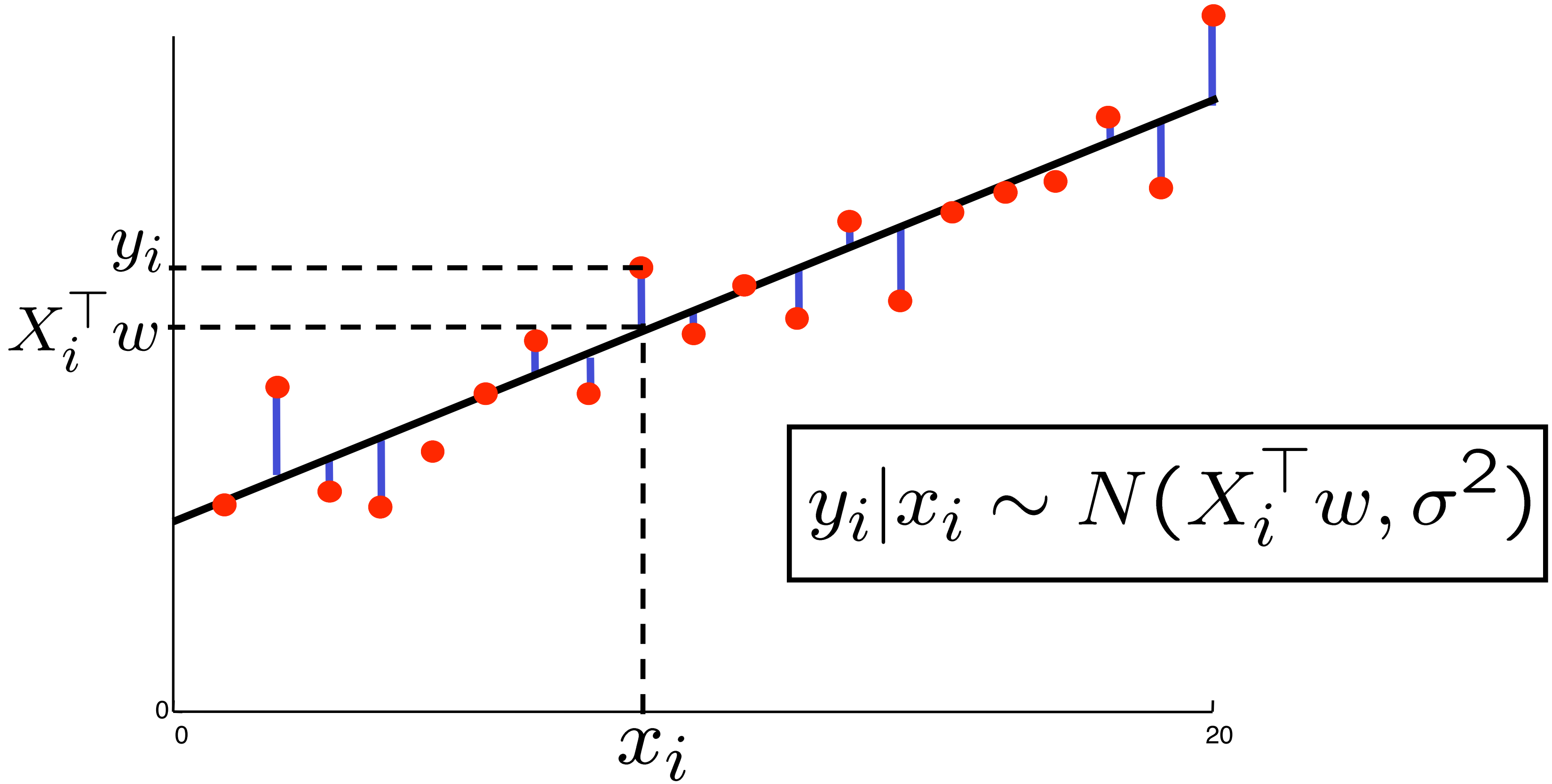
For example $X_i^\top = \begin{pmatrix} 1 & x_{i,1} & x_{i,2} & x_{i,1}^2 & x_{i,2}^2 & x_{i,1}x_{i,2} \end{pmatrix}$

Let $X = \underbrace{\begin{pmatrix} -X_1^\top- \\ -X_2^\top- \\ \ldots \end{pmatrix}}_{d} \quad n \qquad y = \begin{pmatrix} y_1 \\ y_2 \\ \ldots \end{pmatrix}$

Minimize $\|Xw - y\|_2^2$ by solving $\left(X^\top X\right) w = X^\top y$

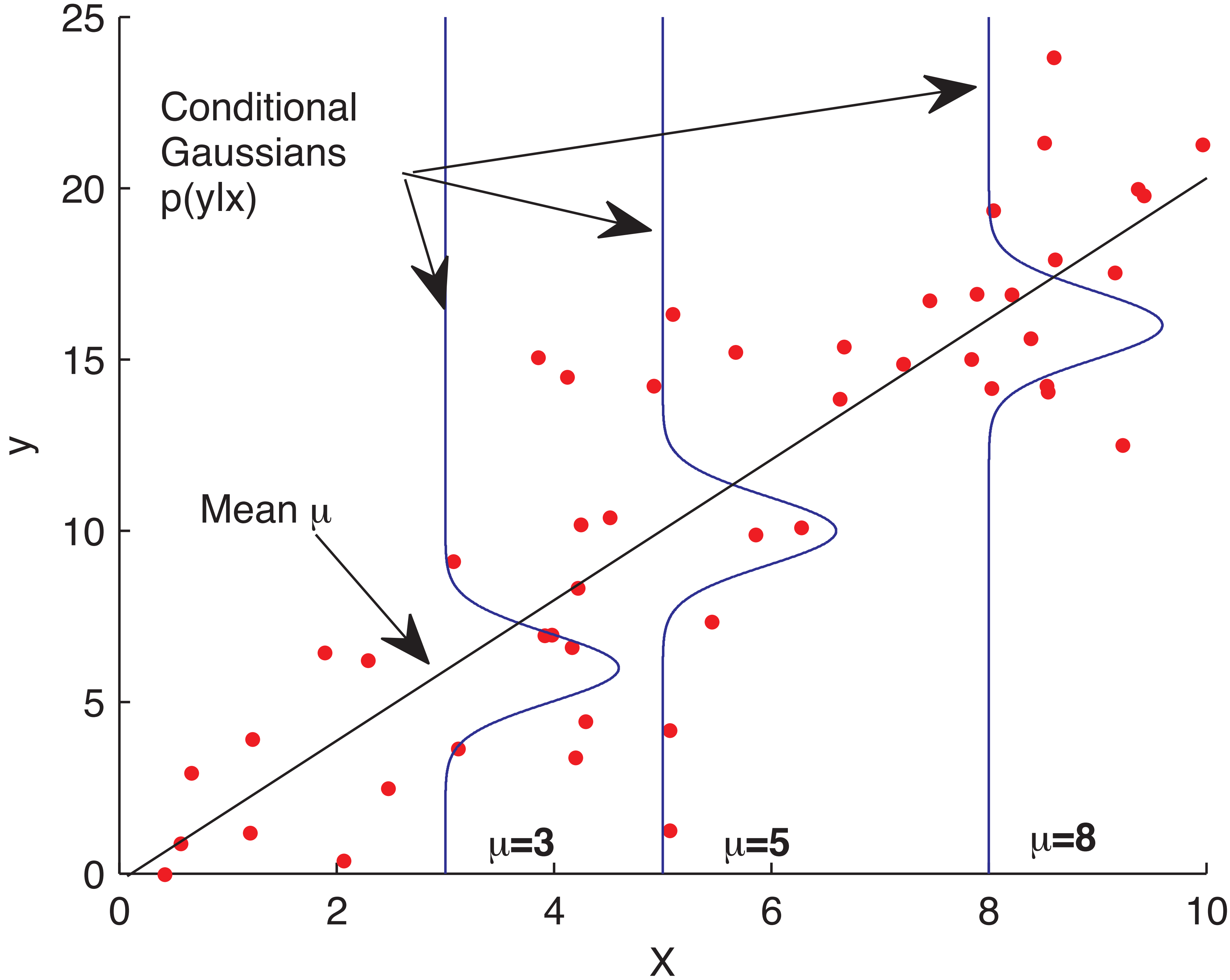
Predict $\widehat{y}_{n+1} = X_{n+1}^\top w$

Probabilistic interpretation



Likelihood $L = \prod_i \exp -\frac{1}{2\sigma^2}(X_i^\top w - y_i)^2 = \exp -\frac{1}{2\sigma^2} \sum_i (X_i^\top w - y_i)^2$

$$\underset{w}{\operatorname{argmax}} L = \underset{w}{\operatorname{argmin}} E$$


Conditional
Gaussians
p(y|x)
Mean μ
μ=3
μ=5
μ=8
y
x
0
5
10
15
20
25
0
2
4
6
8
10

BREAK

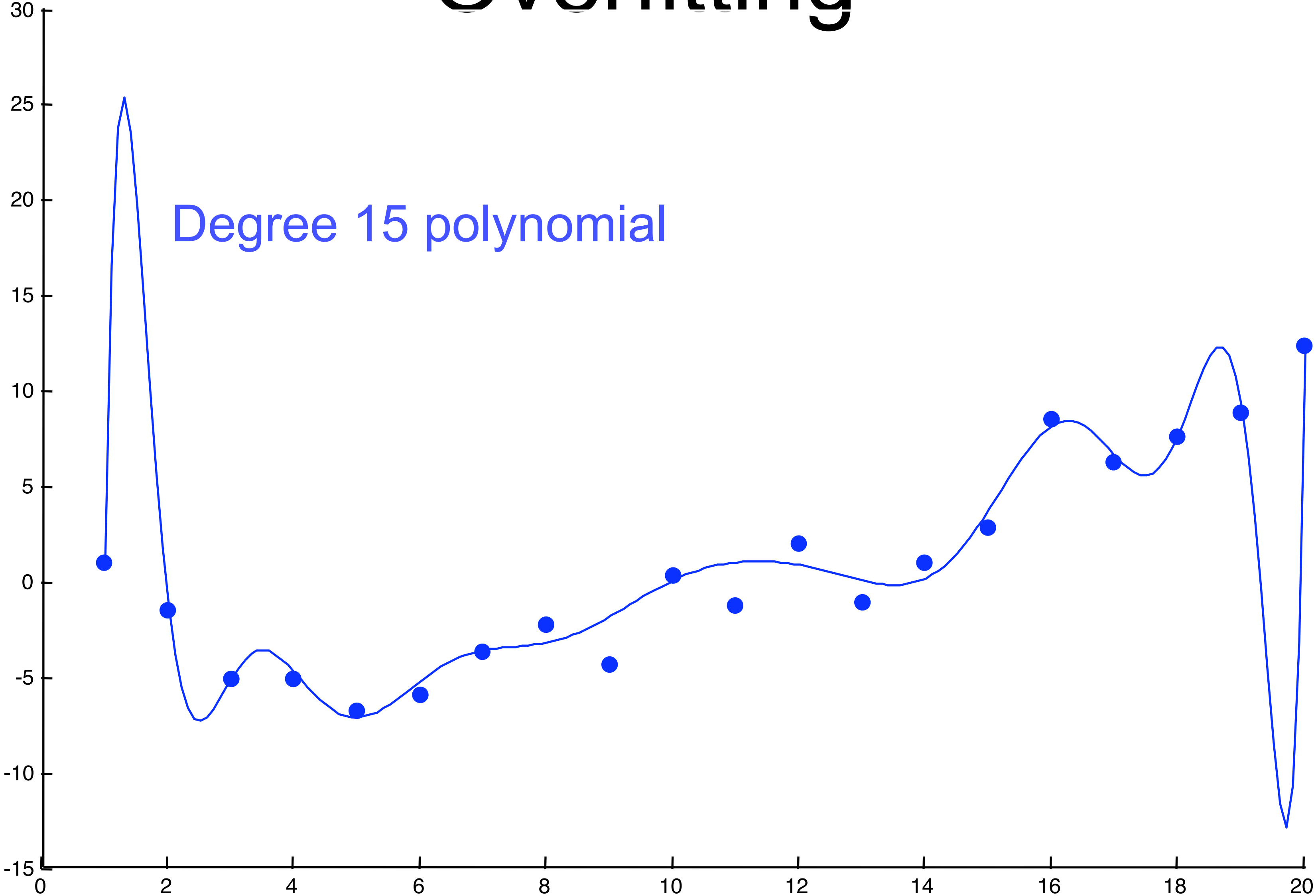
Outline

- Ordinary Least Squares Regression
  - Online version
  - Normal equations
  - Probabilistic interpretation
- Overfitting and Regularization
- Overview of additional topics
  - L_1 Regression
  - Quantile Regression
  - Generalized linear models
  - Kernel Regression and Locally Weighted Regression

Overfitting

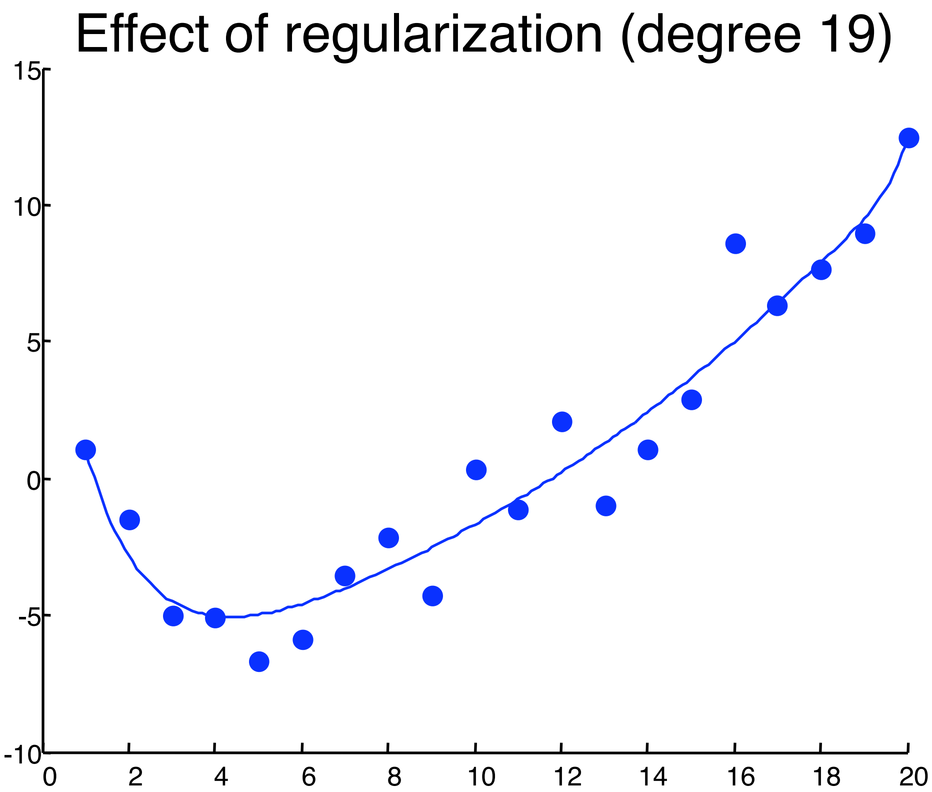
- So the more features the better? NO!
- Carefully selected features can improve model accuracy.
- But adding too many can lead to overfitting.
- Feature selection will be discussed in a separate lecture.

Overfitting



[Matlab demo]

Ridge Regression (Regularization)



Minimize

$$\frac{1}{2}\|Xw - y\|_2^2 + \epsilon\|w\|_2^2$$

with ϵ "small" by solving

$$(X^\top X + \epsilon I)w = X^\top y$$

[Continue Matlab demo]

Probabilistic interpretation

Likelihood $y_i|x_i \sim N(X_i^\top w, \sigma^2)$

Prior $w \sim N\left(0, \frac{\sigma^2}{\epsilon}\right)$

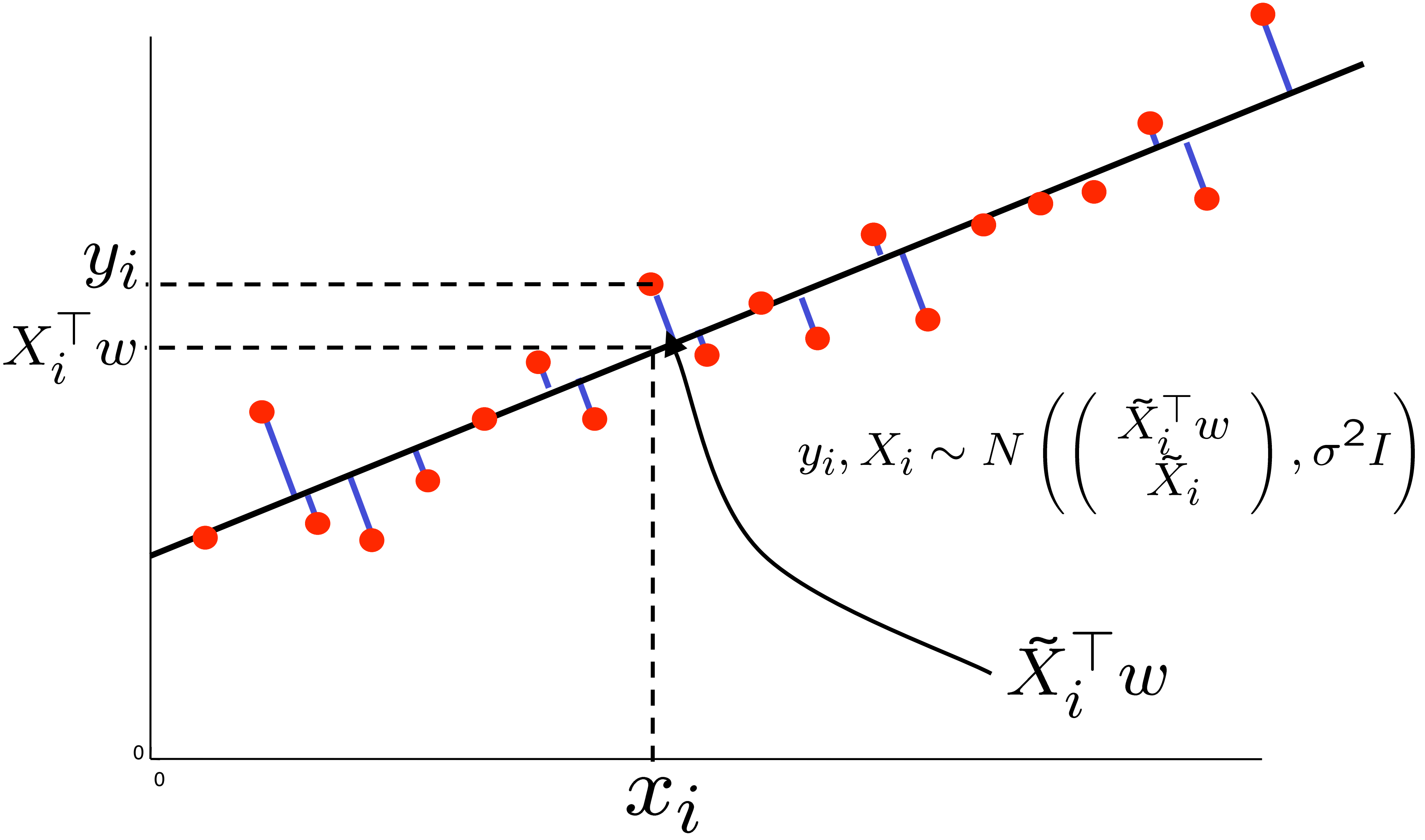
Posterior

$$
\begin{aligned}
P(w|X,y) &= \frac{P(w, x_1, \ldots, x_n, y_1, \ldots, y_n)}{P(x_1, \ldots, x_n, y_1, \ldots, y_n)} \\
&\propto P(w.x_1, \ldots, x_1, y_1, \ldots, y_n) \\
&\propto \exp\left\{-\frac{\epsilon}{2\sigma^2}||w||_2^2\right\} \prod_i \exp\left\{-\frac{1}{2\sigma^2}\left(X_i^\top w - y_i\right)^2\right\} \\
&= \exp\left\{-\frac{1}{2\sigma^2}\left[\epsilon||w||_2^2 + \sum_i (X_i^\top w - y_i)^2\right]\right\}
\end{aligned}
$$

Outline

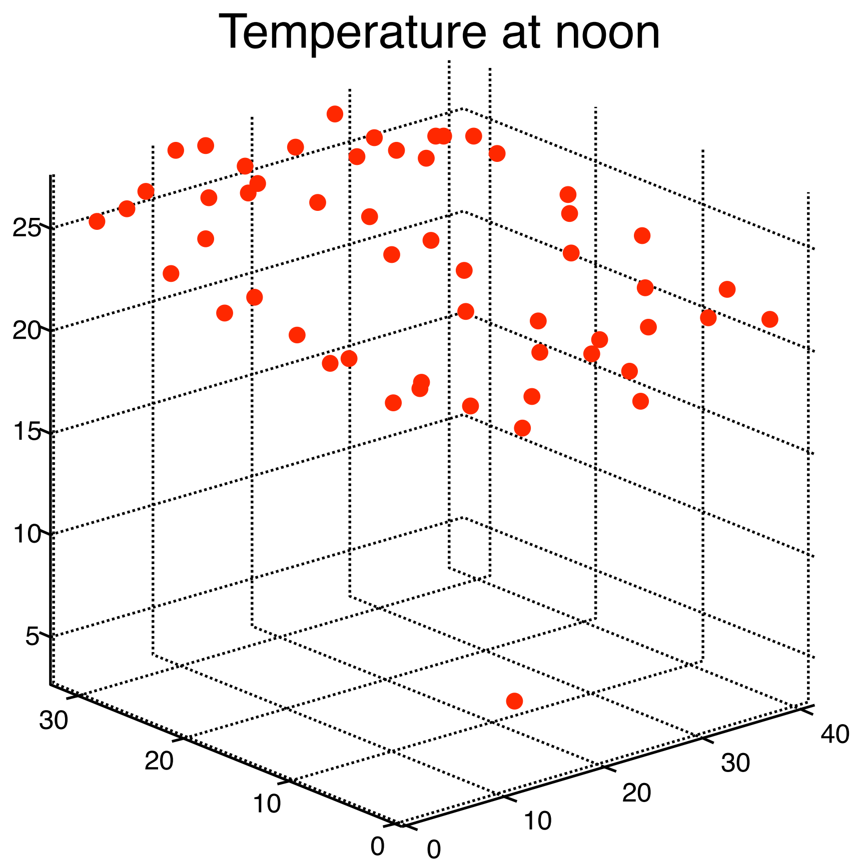
- Ordinary Least Squares Regression
  - Online version
  - Normal equations
  - Probabilistic interpretation
- Overfitting and Regularization
- Overview of additional topics
  - L_1 Regression
  - Quantile Regression
  - Generalized linear models
  - Kernel Regression and Locally Weighted Regression

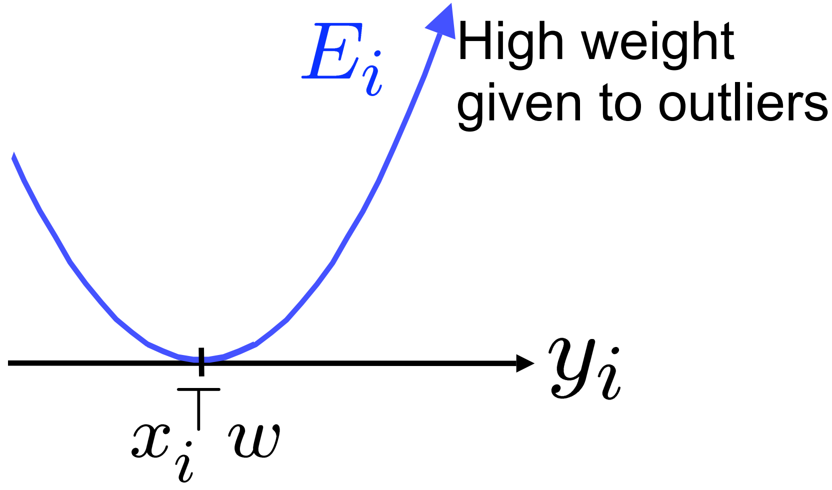
Errors in Variables (Total Least Squares)

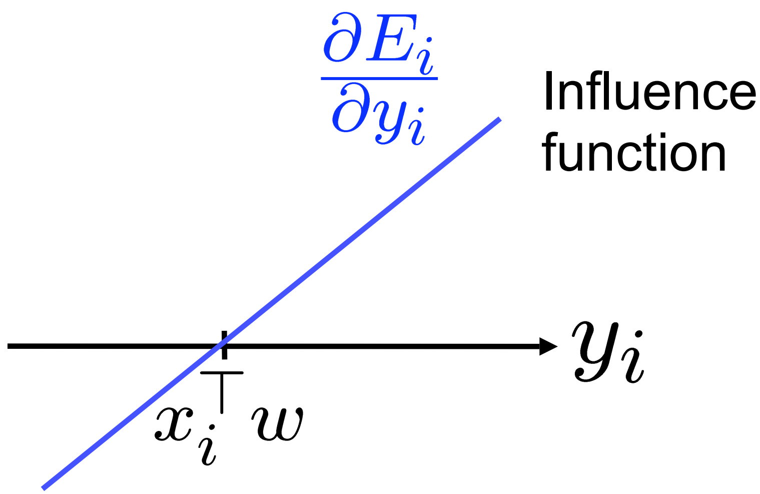


Sensitivity to outliers

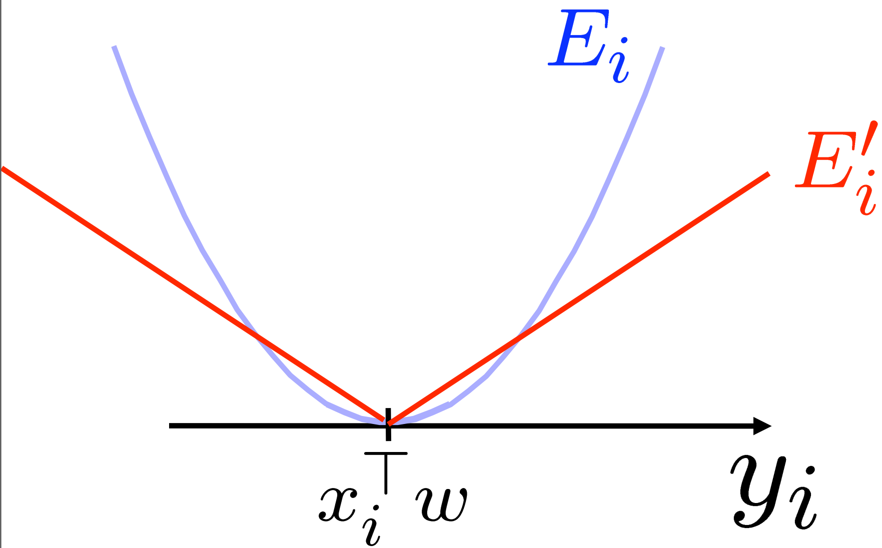
$$E = \sum_i (x_i^\top w - y_i)^2 = \sum_i E_i$$



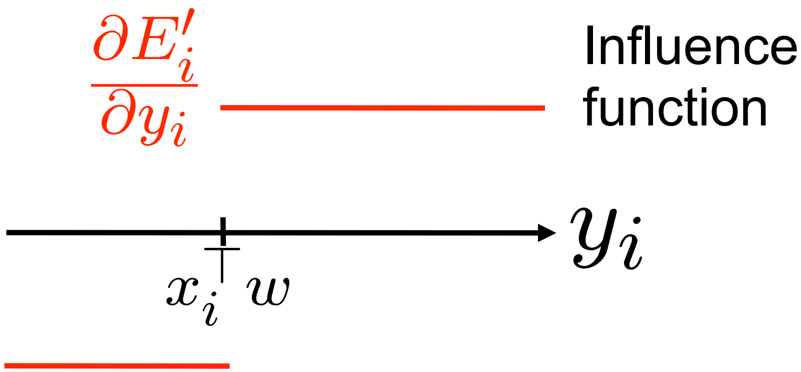




L_1 Regression



$$E' = \sum_i |x_i^\top w - y_i|$$
$$= \sum_i E'_i$$



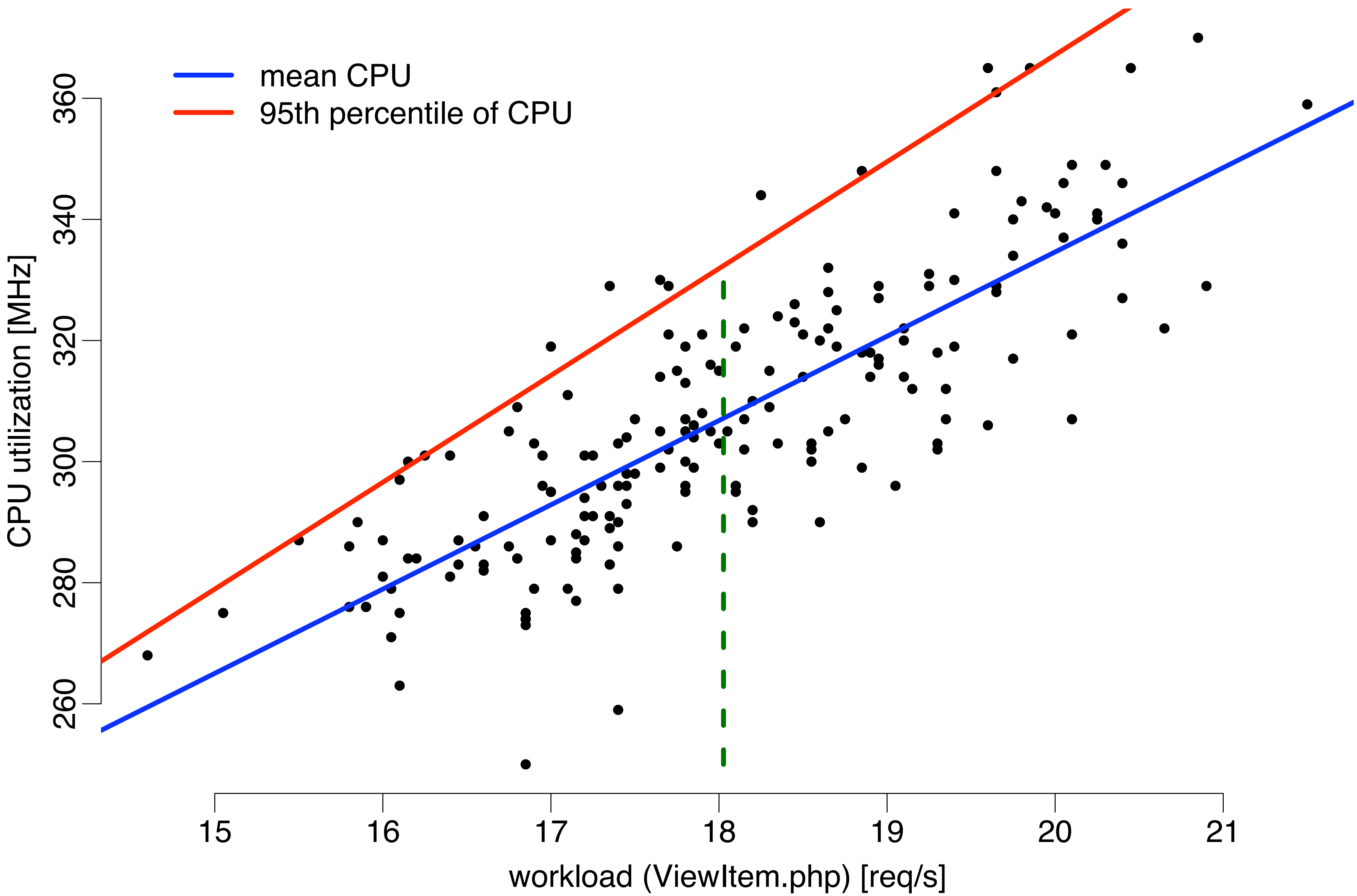
Linear program

$$\min_{w,c} \sum_i c_i$$

s.t.
$$x_i^\top w - y_i \leq c_i \quad \forall i$$
$$y_i - x_i^\top w \leq c_i \quad \forall i$$

[Matlab demo]

Quantile Regression



Slide courtesy of Peter Bodik

Generalized Linear Models

Probabilistic interpretation of OLS

Mean is linear in X_i

$$y_i | x_i \sim N(X_i^\top w, \sigma^2)$$

OLS: linearly predict the mean of a Gaussian conditional.

GLM: predict the mean of some other conditional density.

$$y_i | x_i \sim p\left(f(X_i^\top w)\right)$$

May need to transform linear prediction by $f(\cdot)$ to produce a valid parameter.

36

Example: "Poisson regression"

Suppose data y are event counts: $y \in \mathbb{N}_0$

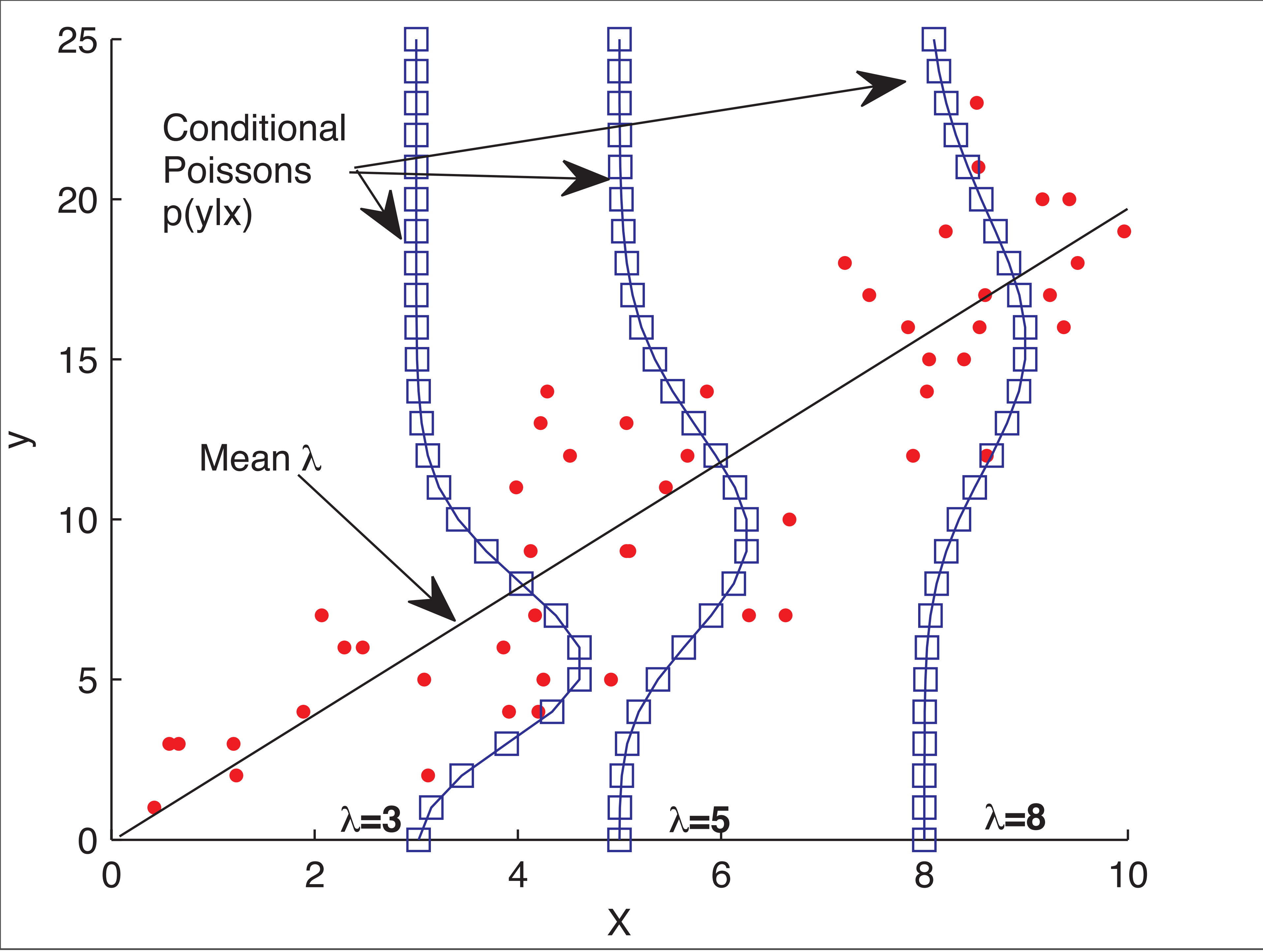
Typical distribution for count data: Poisson

$$\text{Poisson}(y|\lambda) = \frac{e^{-\lambda}\lambda^y}{y!}$$

Mean parameter is $\lambda > 0$

Say we predict $\lambda = f(x^\top w) = \exp\left\{x^\top w\right\}$

GLM: $y_i | x_i \sim \text{Poisson}\left(f(X_i^\top w)\right)$


Conditional
Poissons
p(y|x)
Mean λ
λ=3
λ=5
λ=8
y
x
0
5
10
15
20
25
0
2
4
6
8
10

Poisson regression: learning

As for OLS: optimize w by maximizing the likelihood of data.

Equivalently: maximize log likelihood.

Likelihood $L = \prod_i \text{Poisson}\left(y_i | f(X_i^\top w)\right)$

Log likelihood $l = \sum_i \left(X_i^\top w y_i - \exp\left\{X_i^\top w\right\}\right) + \text{const.}$

Batch gradient:

$$\frac{\partial l}{\partial w} = \sum_i \left(y_i - \exp\left\{X_i^\top w\right\}\right) X_i$$

$$= \sum_i \underbrace{\left(y_i - f\left(X_i^\top w\right)\right)}_{\text{"residual"}} X_i$$

39

LMS, Logistic regression, Perceptron and GLM updates

- GLM (online)

$$w^{t+1} \quad := \quad w^t + \alpha(y_i - f_w(x_i))x_i$$

- LMS

$$w^{t+1} \quad := \quad w^t + \alpha(y_i - x_i^\top w)x_i$$

- Logistic Regression

$$w^{t+1} \quad := \quad w^t + \alpha(y_i - f_w(x_i))x_i$$

- Perceptron

$$w^{t+1} \quad := \quad w^t + \alpha(y_i - f_w(x_i))x_i$$

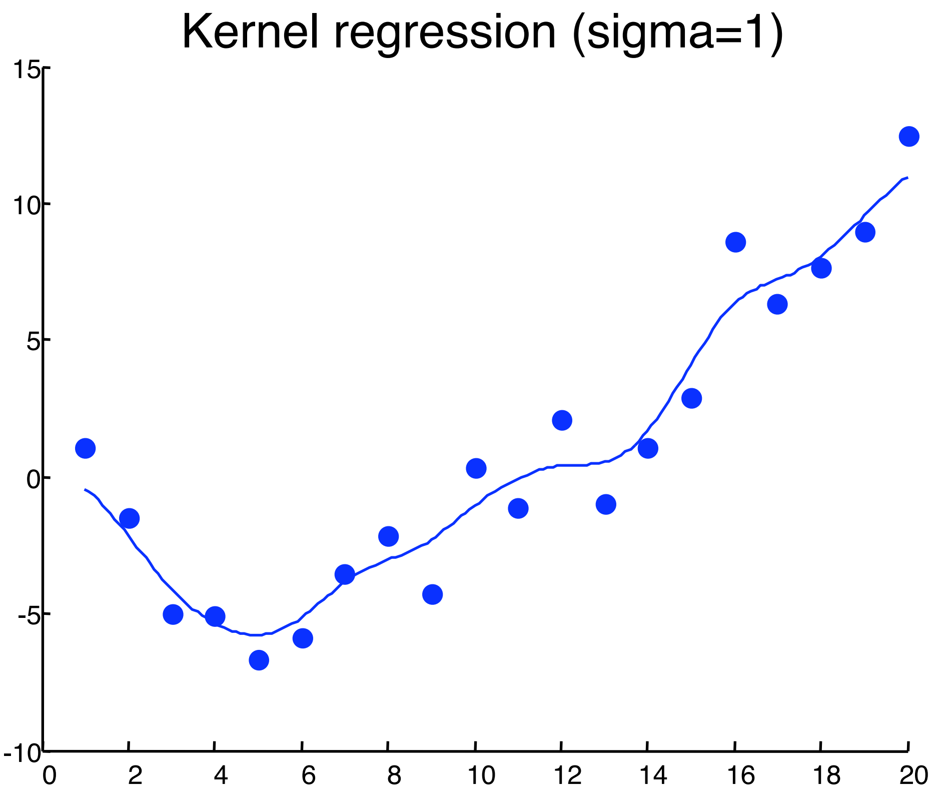
Kernel Regression and Locally Weighted Linear Regression

- **Kernel Regression:**
  Take a very very conservative function approximator called AVERAGING. Locally weight it.

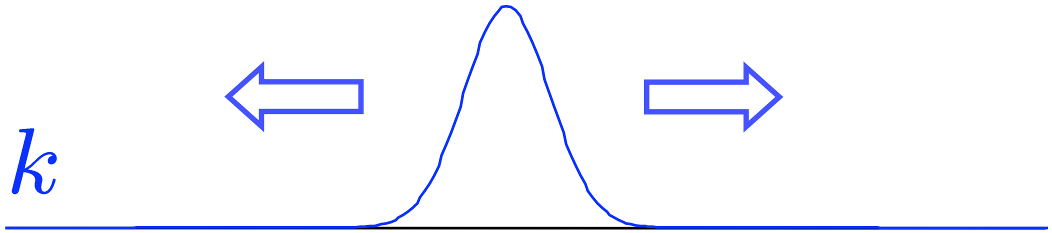
- **Locally Weighted Linear Regression:**
  Take a conservative function approximator called LINEAR REGRESSION. Locally weight it.

Slide from Paul Viola 2003

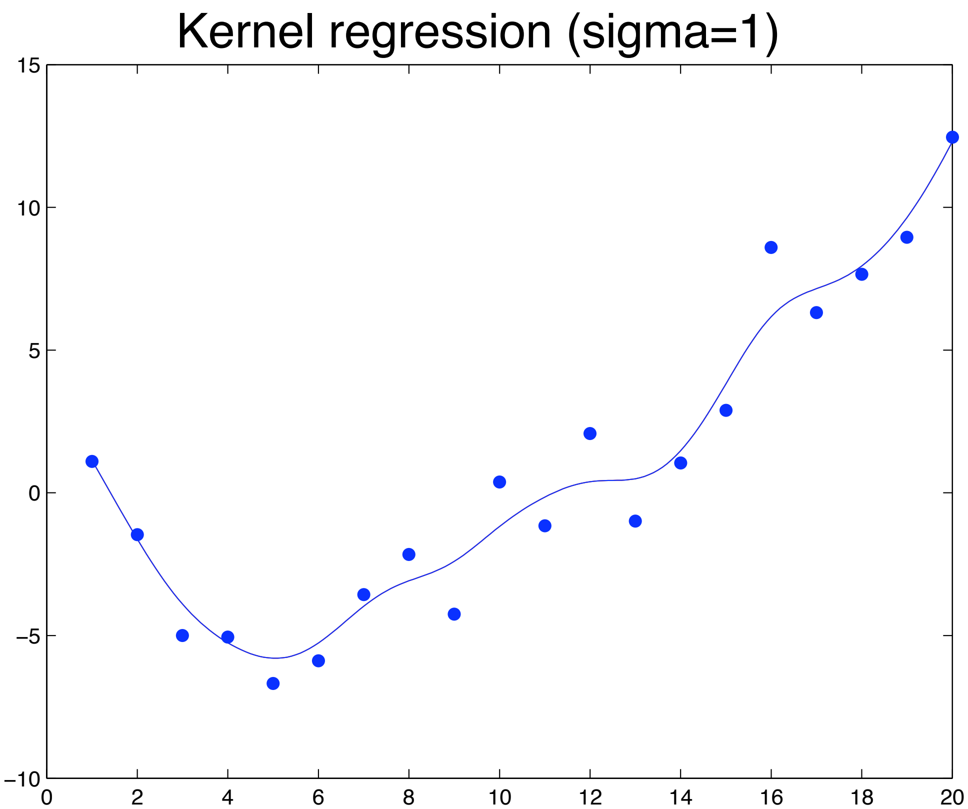
Kernel Regression



$$\hat{y}(x) = \frac{\sum_i y_i k(x_i - x)}{\sum_i k(x_i - x)}$$



Locally Weighted Linear Regression (LWR)

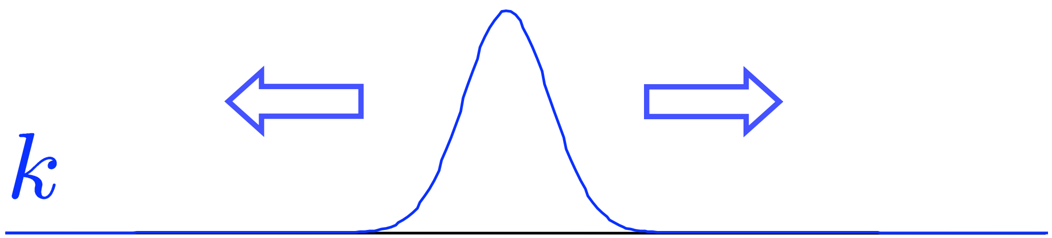


OLS cost function:

$$E = \frac{1}{2}\sum_{i=1}^{n}(w^\top x_i - y_i)^2$$

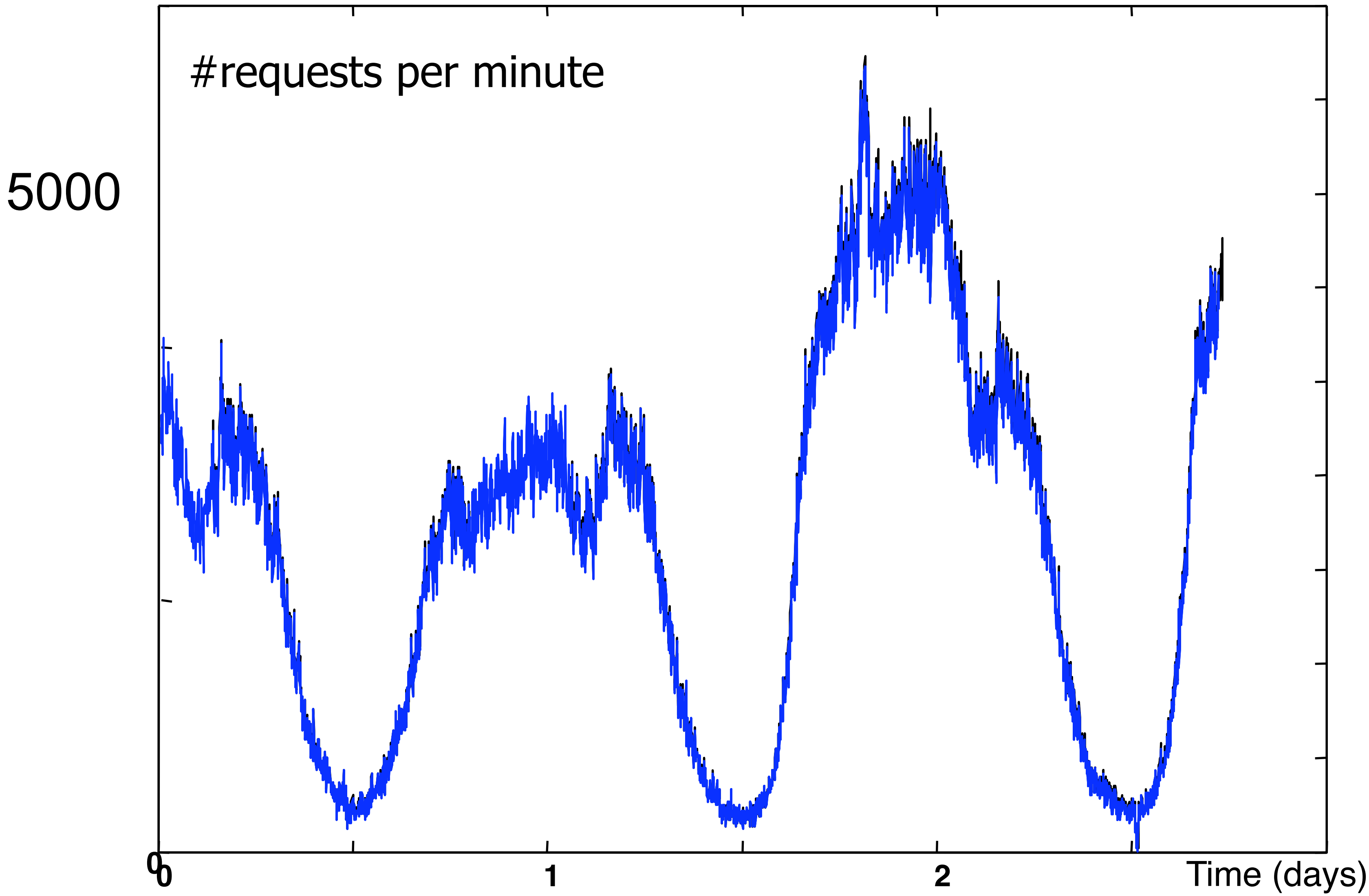
LWR cost function:

$$E' = \sum_{i=1}^{n} k(x_i - x)(w^\top x_i - y_i)^2$$



[Matlab demo]

Heteroscedasticity



What we covered

- Ordinary Least Squares Regression
  - Online version
  - Normal equations
  - Probabilistic interpretation
- Overfitting and Regularization
- Overview of additional topics
  - L_1 Regression
  - Quantile Regression
  - Generalized linear models
  - Kernel Regression and Locally Weighted Regression