Regression

Machine Learning
CSx824/ECEx242
Bert Huang
Virginia Tech

Outline

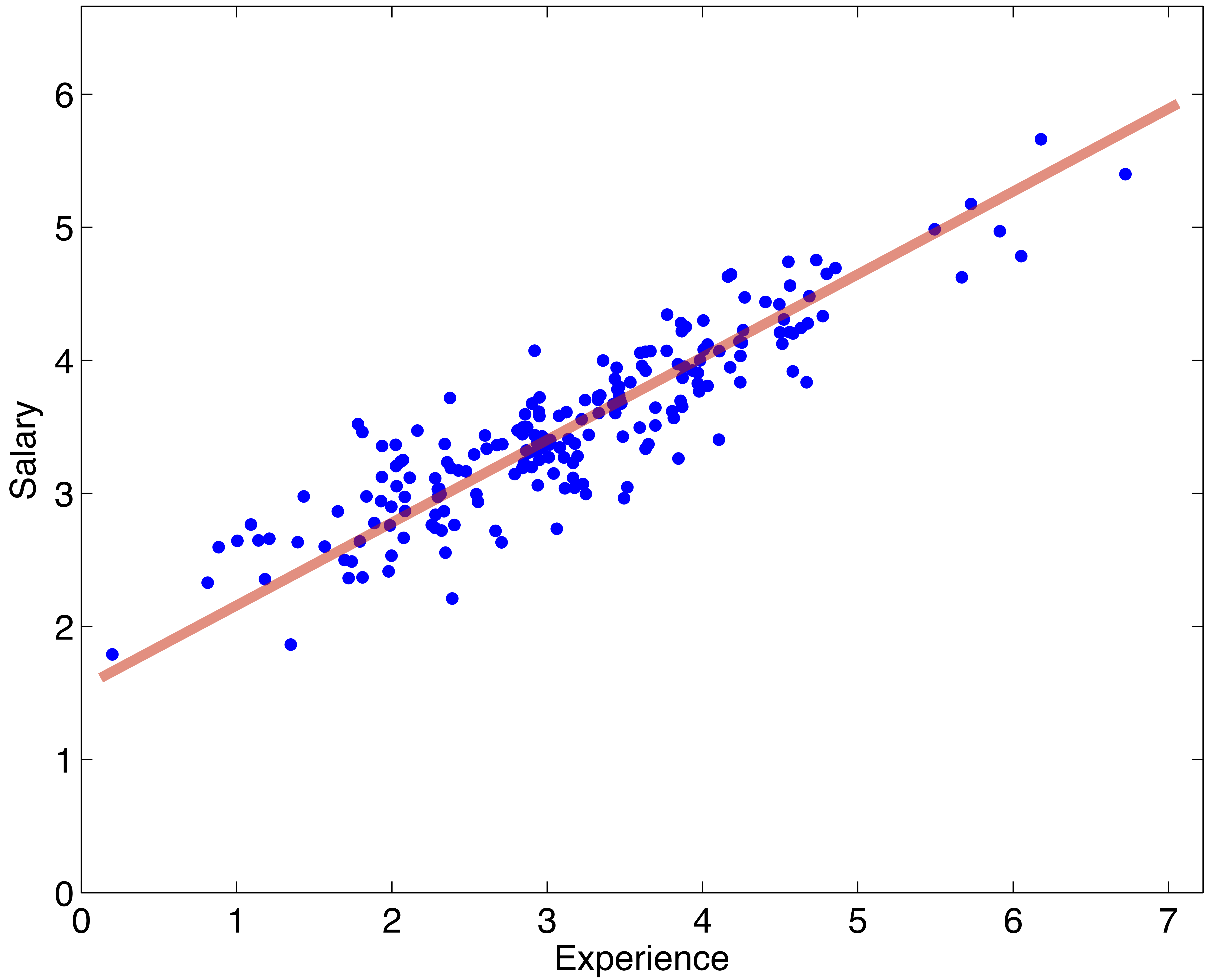
- Regression vs classification
- Least squares linear regression
- Non-linear regression
  - Neural networks for regression
  - Basis functions
  - SVM regression

Regression vs. Classification

- Regression: a measure of the relation between the mean value of one variable (e.g., output) and corresponding values of other variables (e.g., time and cost).

- Technically more general than classification

- Colloquially: regression = continuous output



Salary
Experience
0
1
2
3
4
5
6
7

Least Squares Linear Regression

$$f(x) := w^\top x \qquad w, x \in \mathbb{R}^d$$

$$y, f(x) \in \mathbb{R}$$

Training:

$$\min_w \; \frac{1}{2} \sum_{i=1}^n (f(x_i) - y_i)^2 + \frac{\lambda}{2} w^\top w$$

squared loss regularizer

Matrix form:

$$f(X) = X^\top w$$

$$J(w) = \frac{1}{2}(X^\top w - y)^\top (X^\top w - y) + \frac{\lambda}{2} w^\top w$$

Training: $f(x) := w^\top x$ $\min_w \frac{1}{2}\sum_{i=1}^{n}(f(x_i) - y_i)^2 + \frac{\lambda}{2} w^\top w$

squared loss regularizer

Matrix form: $f(X) = X^\top w$

$$J(w) = \frac{1}{2}(X^\top w - y)^\top (X^\top w - y) + \frac{\lambda}{2} w^\top w$$

$$= \frac{1}{2} w^\top X X^\top w - y^\top X^\top w + \frac{1}{2} y^\top y + \frac{\lambda}{2} w^\top w$$

$$\nabla_w J = X X^\top w - Xy + \lambda w = 0$$

(btw, this is not a kernel matrix)

$$(XX^\top + \lambda \mathbf{I})w = Xy \qquad w = (XX^\top + \lambda \mathbf{I})^{-1} Xy$$

$$f(x) := w^\top x \qquad \nabla_w f(x) = x$$

$$\ell(z, y) = \frac{1}{2}(z - y)^2 \qquad \ell'(z) = (z - y)$$

$$\min_w \sum_{i=1}^{n} \ell(f(x_i), y_i) + \frac{\lambda}{2} w^\top w$$

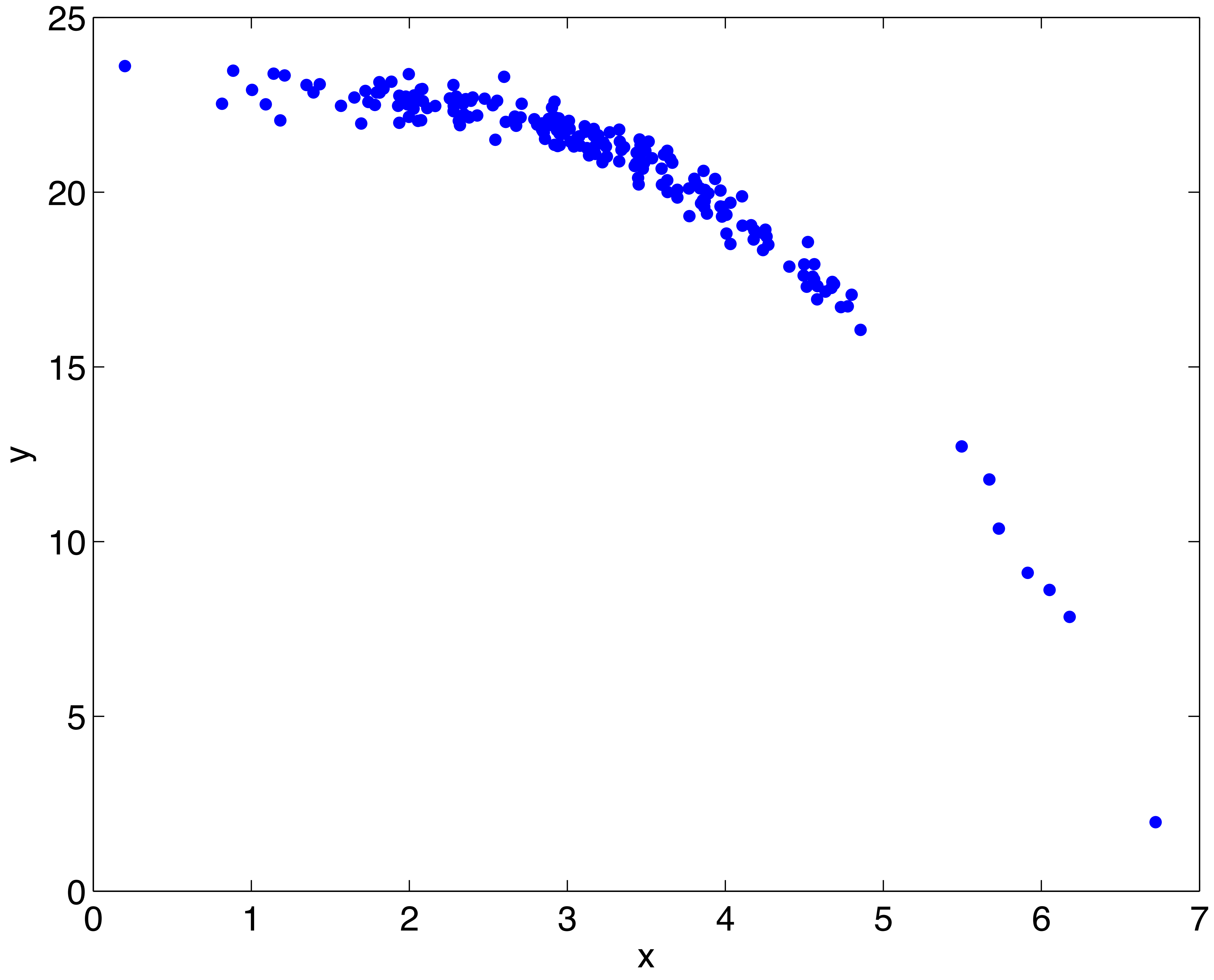
$$\nabla_w \sum_{i=1}^{n} \ell(f(x_i; w), y_i) + \nabla_w \frac{\lambda}{2} w^\top w$$

general form: $$\sum_{i=1}^{n} \ell'(f(x_i; w)) \nabla_w f(x_i; w) + \nabla_w \frac{\lambda}{2} w^\top w$$

We can use any loss or ***f***…

Nonlinear Regression

- Pet peeve: linear regression on obviously nonlinear data
- We'll see two approaches for nonlinear regression



Neural Networks for Regression

general form gradient: $$\sum_{i=1}^{n} \ell'(f(x_i; w)) \nabla_w f(x_i; w) + \nabla_w \frac{\lambda}{2} w^\top w$$

use neural network for ***f*** and use back propagation

error is gradient of loss function:

$$\ell(z, y) = \frac{1}{2}(z - y)^2 \qquad \ell'(z) = (z - y)$$

Basis Functions

- Do linear regression on transformed features

$$f(x) = w^\top \Phi(x)$$

E.g., $\Phi(x) = [x, x^2, x^3, x^4, ...]^\top$

$$w = (\Phi(X)\Phi(X)^\top + \lambda \mathbf{I})^{-1}\Phi(X)y$$

$d \times d$

Need to explicitly compute feature functions

SVM Regression

$$\min_w \frac{1}{2} w^\top w$$

$$\text{s.t. } w^\top x_i - y_i \leq \epsilon, \; \forall i$$

$$y_i - w^\top x_i \leq \epsilon, \; \forall i$$

- Add slack variables
- Take KKT dual
- Kernelize
- Kernel SVM regression: linear in mapped feature space

Summary

- Regression vs classification
- Least squares linear regression
- Non-linear regression
  - Neural networks for regression
  - Basis functions
  - SVM regression