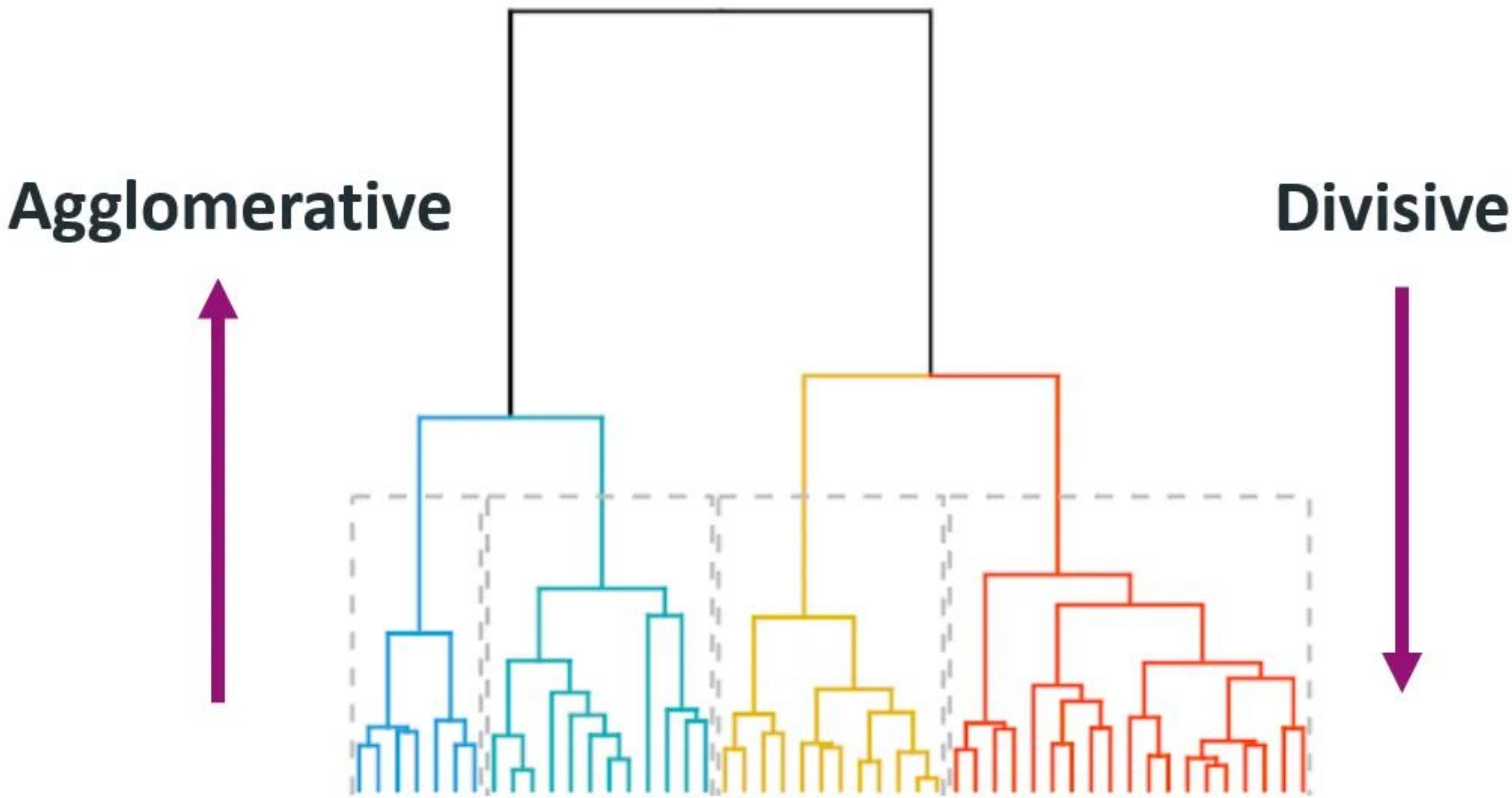
Announcements

- PSet-2 announced, due on Tues, March 4th

Last time: Clustering

- Agglomerative Clustering
- Divisive Clustering
- K-means
- Vector Quantization with K-Means
- Mixtures of Gaussians
- Expectation Maximization

Last time: Clustering



K-Means in the neural nets era

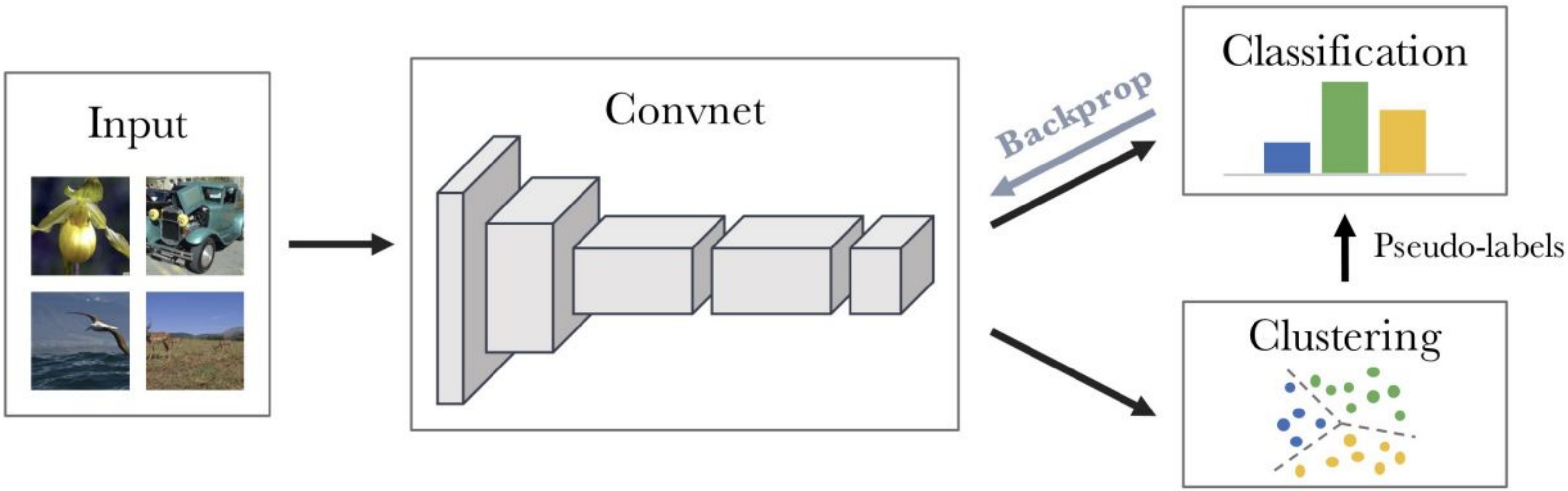
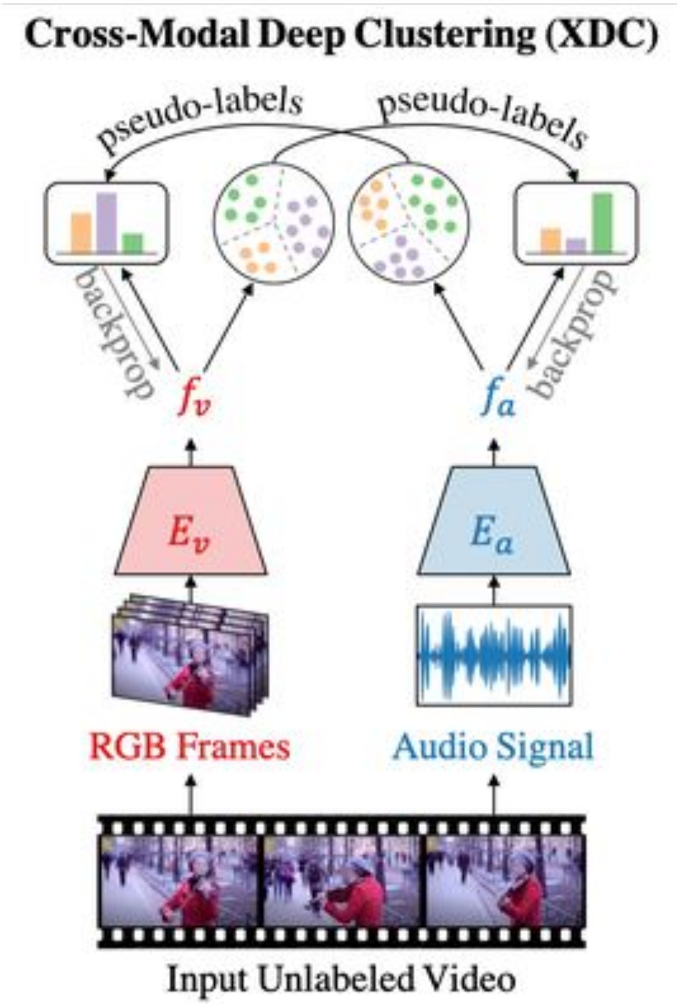


Fig. 1: Illustration of the proposed method: we iteratively cluster deep features and use the cluster assignments as pseudo-labels to learn the parameters of the convnet.

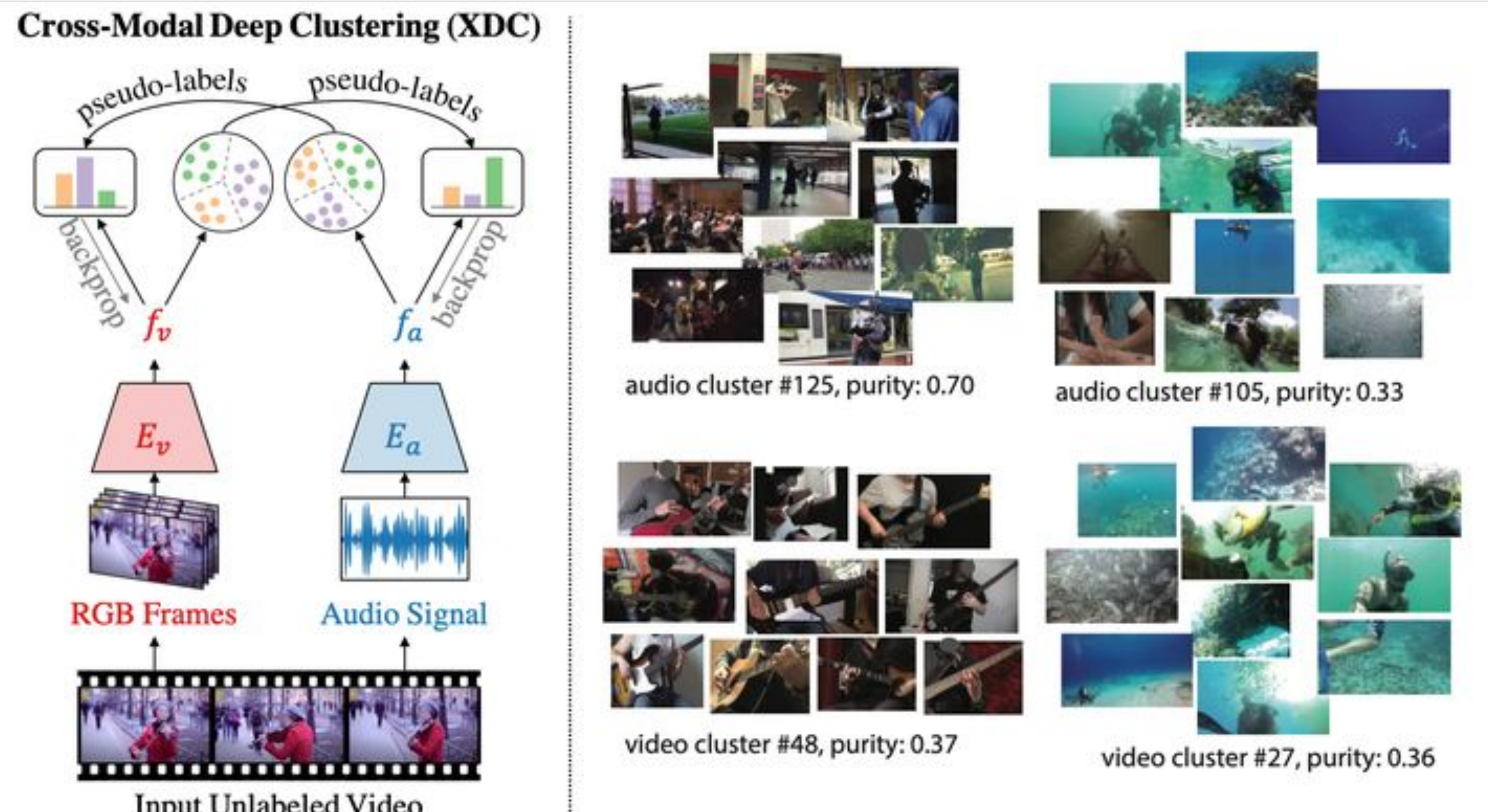
Deep cluster: A self-supervised learning algorithm

M. Caron,et al. "Deep clustering for unsupervised learning of visual features." *ECCV,* 2018

Self-supervised + cross-modal learning



Self-supervised + cross-modal learning



H. Alwassel, et al. "Self-supervised learning by cross-modal audio-video clustering." *NeurIPS'20*

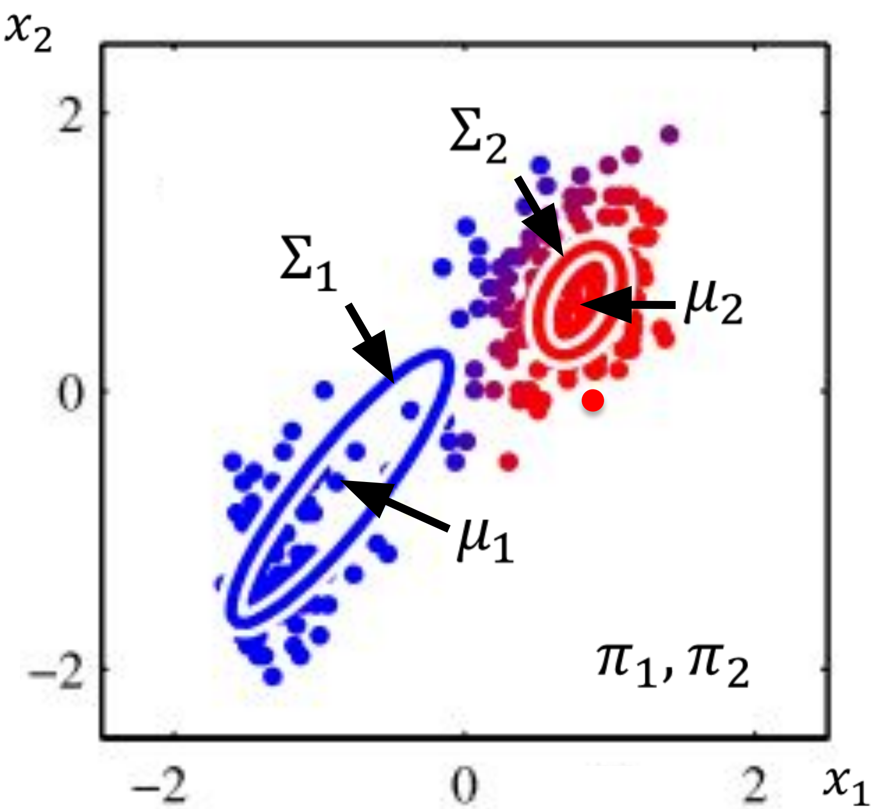
Today

- Expectation Maximization
- Linear Regression
- Analyzing your model
- Regularizing Linear Regression

Today

- **Expectation Maximization**
- Linear Regression
- Analyzing your model
- Regularizing Linear Regression

Mixture of Gaussians



- Each component represents a cluster.
- K-means → mixture of K Gaussians.
- K-th component Gaussian has parameters μ_k, Σ_k

Maximum Likelihood Solution for Mixture of Gaussians

- This distribution is known as a Mixture of Gaussians

$$p(x) = \sum_{k=1}^{\boxed{K}} \boxed{\pi_k} \mathcal{N}(x \mid \boxed{\mu_k}, \boxed{\Sigma_k})$$

- We can estimate these parameters via Expectation Maximization (EM)

- **Solution:** Use coordinate descent

Recall: Parameters

Variable	Role
K	Number of clusters / mixture models
μ_k	Mean of Gaussian distribution (k)
Σ_k	Variance of Gaussian distribution (k)
δ_k	Cluster membership indicator
$p(\delta)$	Marginal distribution of mixture of Gaussian membership
$p(x)$	Distribution of the Mixture of Gaussians

Expectation Maximization Algorithm

- A general technique for finding maximum likelihood estimators in latent variable models
- Initialize and iterate until convergence:

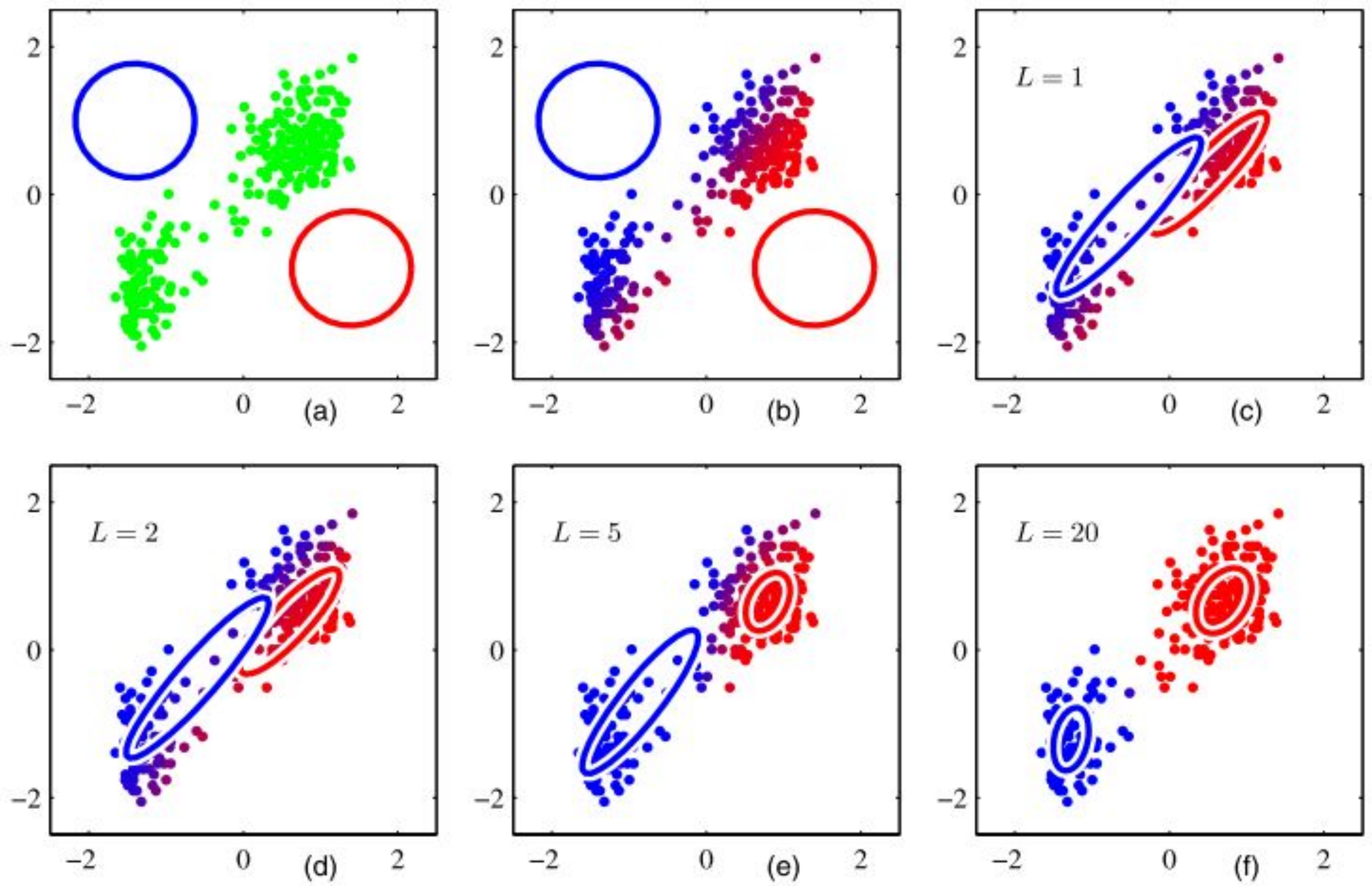
E-Step: estimate posterior probability of the latent variables $p(\delta_k|x)$, holding parameters fixed

Expectation Maximization Algorithm

- A general technique for finding maximum likelihood estimators in latent variable models
- Initialize and iterate until convergence:

M-Step: maximize likelihood w.r.t parameters (here $\mu_k, \Sigma_\mathrm{k}, \pi_k$) using latent probabilities from E-step

EM for Gaussian Mixtures Example



Bishop **Figure 9.8** Illustration of the EM algorithm using the Old Faithful set as used for the illustration of the K-means algorithm in Figure 9.1. See the text for details.

EM for Gaussian Mixtures

1. Initialize parameters θ with means μ_k, covariances Σ_k, and mixing coefficients π_k, and evaluate the initial value of the log likelihood
2. **E step.** Evaluate x_i using the current parameter values $\theta^{(n)}$
3. **M step.** Re-estimate the parameters using current x_i

$$\mu_k^{(n+1)} = \frac{\sum_i x_i w_{ij}}{\sum_i w_{ij}} \qquad w_{ij} = p\left(\delta_{ij} = 1 \middle| \theta^{(n)}, x\right)$$

$$\Sigma_k^{(n+1)} = \frac{\sum_i \left(x_i - \mu_k^{(n+1)}\right)\left(x_i - \mu_k^{(n+1)}\right)^T}{\sum_i w_{ij}} \qquad \pi_j^{(n+1)} = \frac{\sum_i w_{ij}}{N}$$

Today

- Expectation Maximization
- **Linear Regression**
- Analyzing your model
- Regularizing Linear Regression

What is the difference between classification and regression? Select all that apply.

Regression

- Given a training set consisting of **inputs** and **outputs**, learn to map novel, unseen inputs to outputs

- The novel inputs are called a test set

Output

Categorical → **Classification**

Continuous → **Regression**

Multidimensional inputs

Task- *Predicting price*

Size (feet^2)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

x (features) y (label)

Goal: $f(x) = y$

Why do we want to learn f?

Goal: $f(x) = y$

- Helps *estimate* the cost of homes given new x.

Multidimensional inputs

Task- *Predicting price*

Size ($feet^2$)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

$$\underbrace{\text{Size, Number of bedrooms, Number of floors, Age of home}}_{\text{x (features)}} \qquad \underbrace{\text{Price}}_{\text{y (label)}}$$

x (features) y (label)

$$\boxed{\text{Goal: } f(x) = y}$$

- **f** is also referred to as hypothesis.

Linear Regression with one variable

Task- *Predicting price*

Size ($feet^2$)	Price ($1000)
2104	460
1416	232
1534	315
852	178
…	…

x (feature) y (label)

Goal: $f(x) = y$

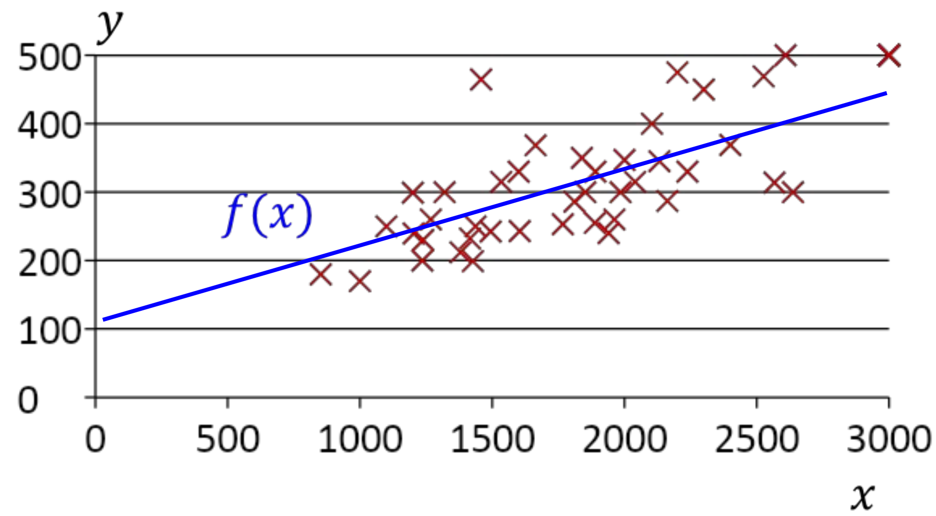
- **f** is also referred to as hypothesis.

Linear Regression with one variable

Hypothesis: $\mathrm{f(x)} = \mathrm{y}$

$$y = x^T\beta + \xi$$

β, ξ: Parameters



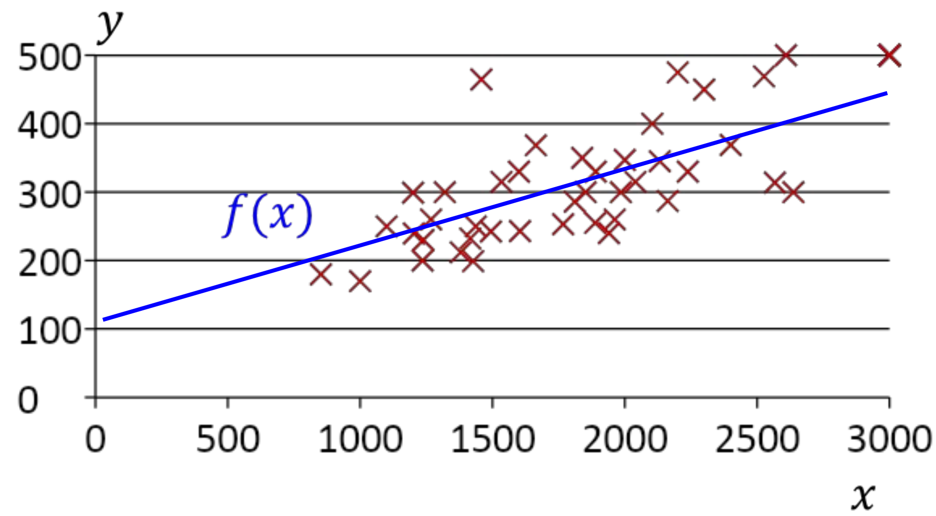
- **Goal:** Choose parameter values such that f(x) is close to y for the red crosses (your training data).

Linear Regression with one variable

Hypothesis:

$$y = x^T\beta + \xi$$

β, ξ: Parameters



Goal: Minimize

$$L(\beta) = \frac{1}{N}\sum_{i=1}^{M}((f(x^i) - y^i)^2$$

SSD = sum of squared differences, also known as
SSE = sum of squared errors

Two potential solutions

$$\min_{\beta} \; \mathcal{L}(\beta)$$

Gradient descent (or other iterative algorithm)

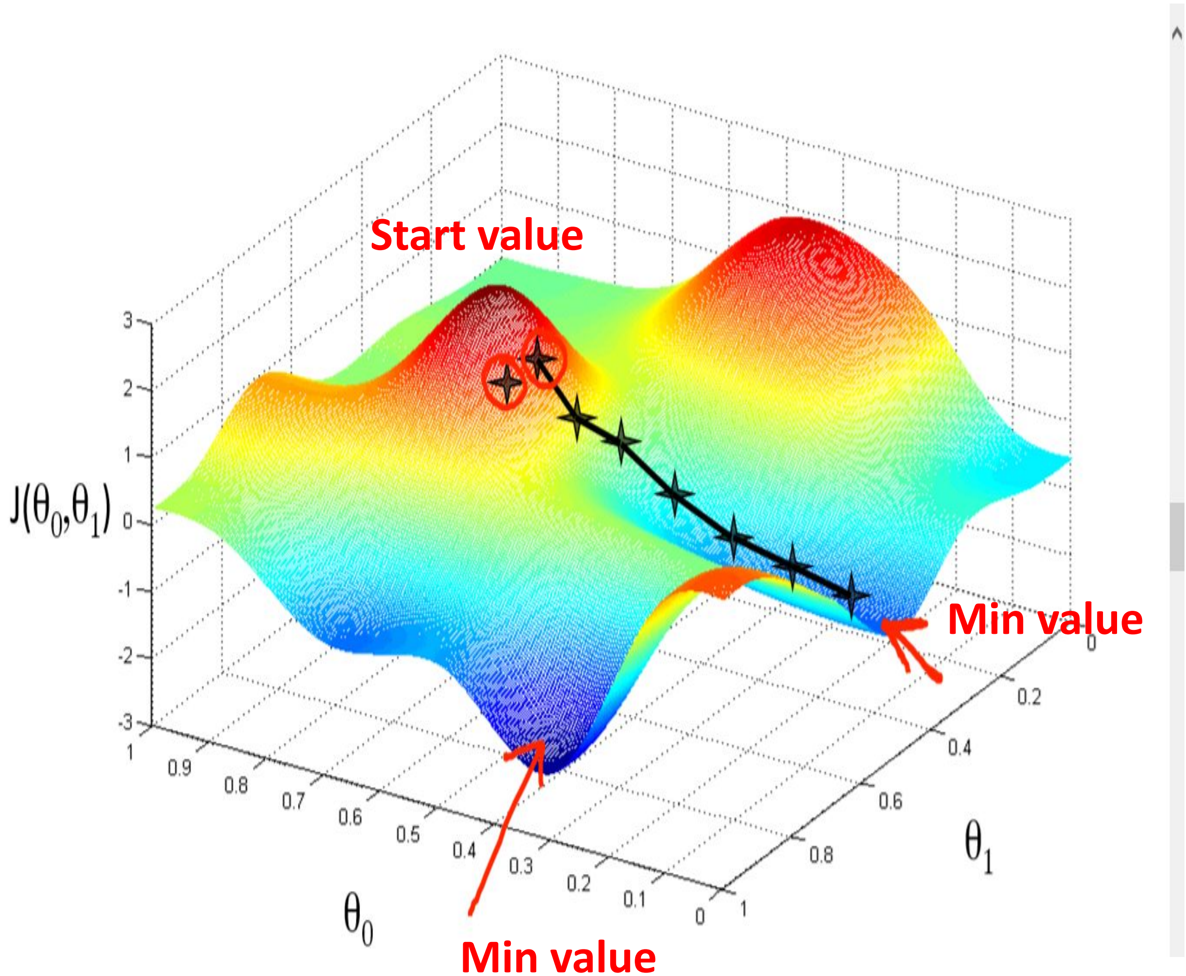
- Start with a guess for β
- Change β to decrease $\mathcal{L}(\beta)$
- Until reach minimum

Recall: Intuitive visualization of the optimization landscape

We use iterative algorithms to find our way to the bottom of the landscape

Slide credit: Justin Johnson

Visualizing gradient descent



Slide credit: Andrew Ng

Two potential solutions

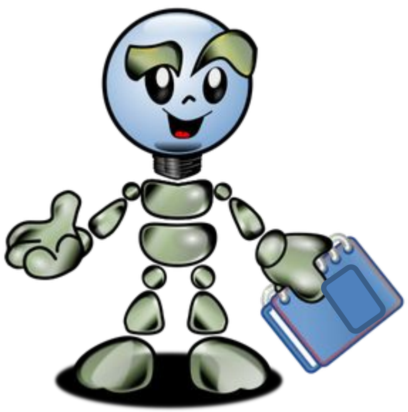
$$\min_{\beta} \ \mathcal{L}(\beta)$$

Gradient descent (or other iterative algorithm)

- Start with a guess for β
- Change β to decrease $\mathcal{L}(\beta)$
- Until reach minimum

Direct minimization

- Take derivative, set to zero
- Sufficient condition for minima
- Not possible for most "interesting" cost functions

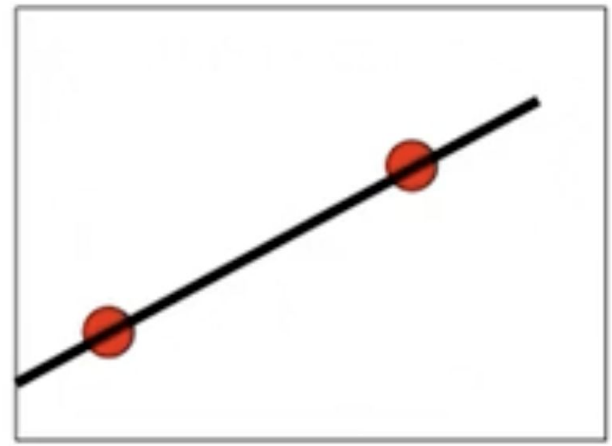


Solving Linear Regression

Direct Solution

Simple case: 2 points

Hypothesis: $y = x^T \boxed{\beta} + \boxed{\xi}$



$$y^1 = x^1\beta + \xi$$

$$y^2 = x^2\beta + \xi$$

Two unknowns, two points, two equations

Rearrange and find the right values for β, ξ

Generalizing direct solution

- More generally, N > d
  - Hard to find a linear function that fits all the data.

Size (feet^2)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

N

d

Solution: Solve for means squared error function

Direct solution

Hypothesis: $y = x^T\beta + \xi$

Want to minimize Sum of Squared distances (SSD):

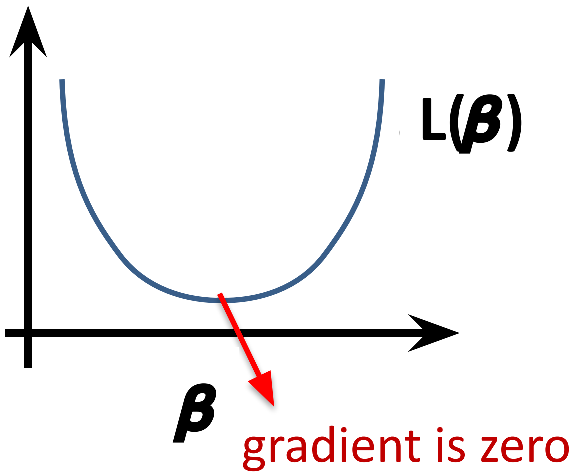
$$\xi_i = \frac{1}{N}\sum_i \left(y_i - x_i^T\beta\right)^2$$

Find minima where the gradient is zero

$\boldsymbol{\beta} \in \mathbb{R}^d$

$\min_{\beta_j} \mathcal{L}(\beta_j)$ (for every j)

Solve for $\beta_1, \beta_2, \dots, \beta_d$



Direct solution

$$\xi_i = \frac{1}{N} \sum_i \left(y_i - x_i^T \beta\right)^2$$

Rewrite SSD using vector-matrix notation:

$$\xi = \frac{1}{N} (y - \mathcal{X}\beta)^T (y - \mathcal{X}\beta)$$

Where:

$$\mathcal{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ \cdots \\ x_n^T \end{pmatrix} \qquad y = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}$$

Derivation of Normal Equations

- SSE in matrix form:
  - $\xi = \frac{1}{N}(y - \mathcal{X}\beta)^T(y - \mathcal{X}\beta)$

$$= \frac{1}{N}(\beta^T(\mathcal{X}^T\mathcal{X})\,\beta - 2(\mathcal{X}^T y)^T\beta + const)$$

$y^T y$

Derivation of Normal Equations

- SSE in matrix form:
  - $$\xi = \frac{1}{N}(y - \mathcal{X}\beta)^T(y - \mathcal{X}\beta)$$
    $$= \frac{1}{N}(\beta^T(\mathcal{X}^T\mathcal{X})\,\beta - 2(\mathcal{X}^T y)^T\beta + const)$$
- Take derivative with respect to β (vector), set to 0
  - $$\frac{\partial\xi}{\partial\beta} \propto \mathcal{X}^T\mathcal{X}\beta - \mathcal{X}^T y = 0$$ ignore constant multiplier
  - $$\beta = (\mathcal{X}^T\mathcal{X})^{-1}\ \mathcal{X}^T y$$

Derivation of Normal Equations

- SSE in matrix form:
  - $$\xi = \frac{1}{N}(y - \mathcal{X}\beta)^T(y - \mathcal{X}\beta)$$
    $$= \frac{1}{N}\left(\beta^T(\mathcal{X}^T\mathcal{X})\,\beta - 2(\mathcal{X}^T y)^T\beta + const\right)$$
- Take derivative with respect to β (vector), set to 0
  - $$\frac{\partial \xi}{\partial \beta} \propto \mathcal{X}^T\mathcal{X}\beta - \mathcal{X}^T y = 0$$ ignore constant multiplier
  - $$\boxed{\beta = (\mathcal{X}^T\mathcal{X})^{-1}\ \mathcal{X}^T y}$$
- Also known as the least mean squares, or least squares solution

Example: N $= 4, d = 4$

x_0	Size (feet[2]) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

N

d

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}$$

a.k.a Design Matrix

$$y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

To find parameters, evaluate the Normal Equations

$$\beta = (\mathcal{X}^T\mathcal{X})^{-1}\ \mathcal{X}^T y$$

Trade-offs

N training examples, d features.

Gradient Descent

- Need to choose learning rate (η)
- Requires multiple iterations

- For a given N, performs well even if d is large

Normal Equations

- No need to choose η
- Don't need to iterate

- Need to compute

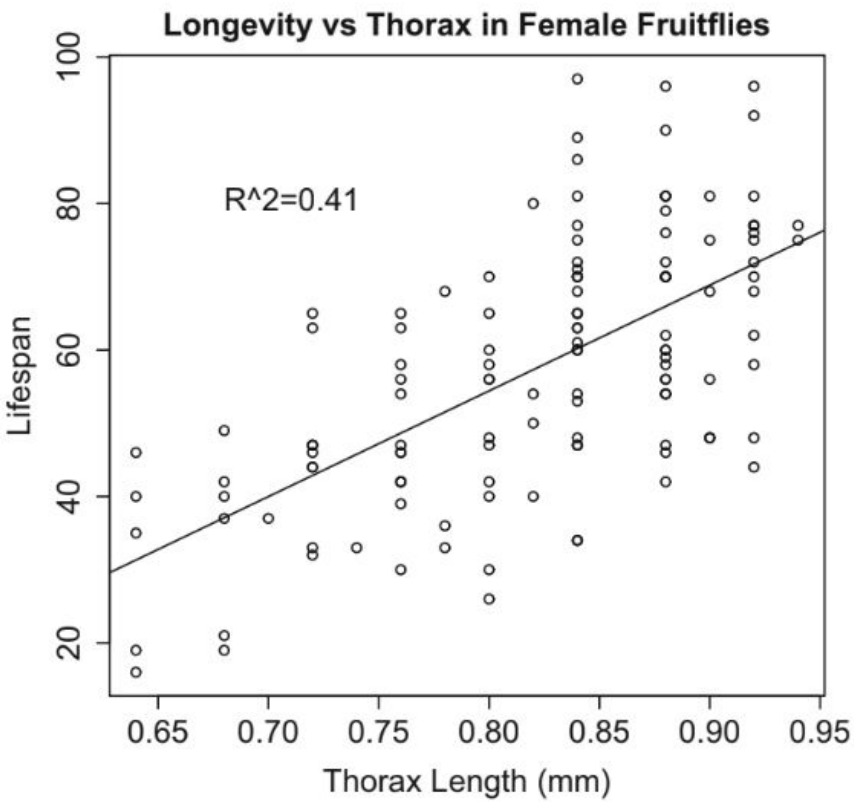
$$(\mathcal{X}^T\mathcal{X})^{-1}$$

- For a given N, slow if d is large

Today

- Expectation Maximization
- Linear Regression
- **Analyzing your model**
- Regularizing Linear Regression

Poorly performing regression models



How do we determine if we have a good model?

Metric: Mean-squared error

Assume have found an estimate our parameters $\hat{\beta}$ by solving:

$$\mathcal{X}^T\mathcal{X}\,\hat{\beta} - \mathcal{X}^T y = 0$$

We can compute the error using the **residual vector**

$$e = y - \mathcal{X}\hat{\beta}$$

Which gives us **mean-squared error**:

$$\boxed{m = \frac{e^T e}{N}}$$

Lower is better!

Metric: R-squared

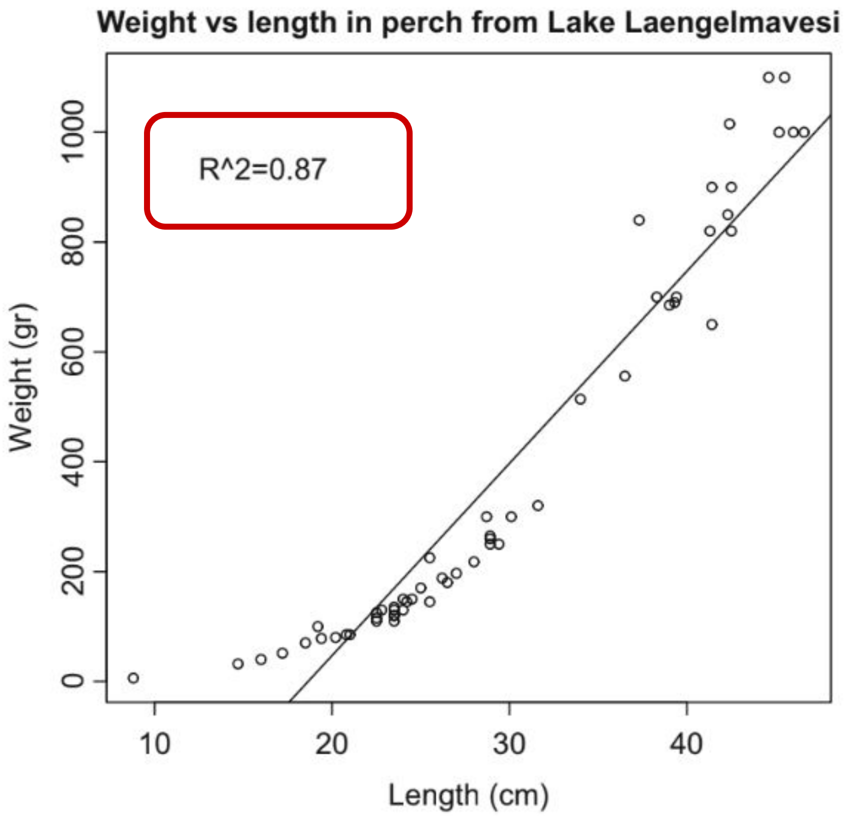
Model: $$y = \mathcal{X}\beta + e$$

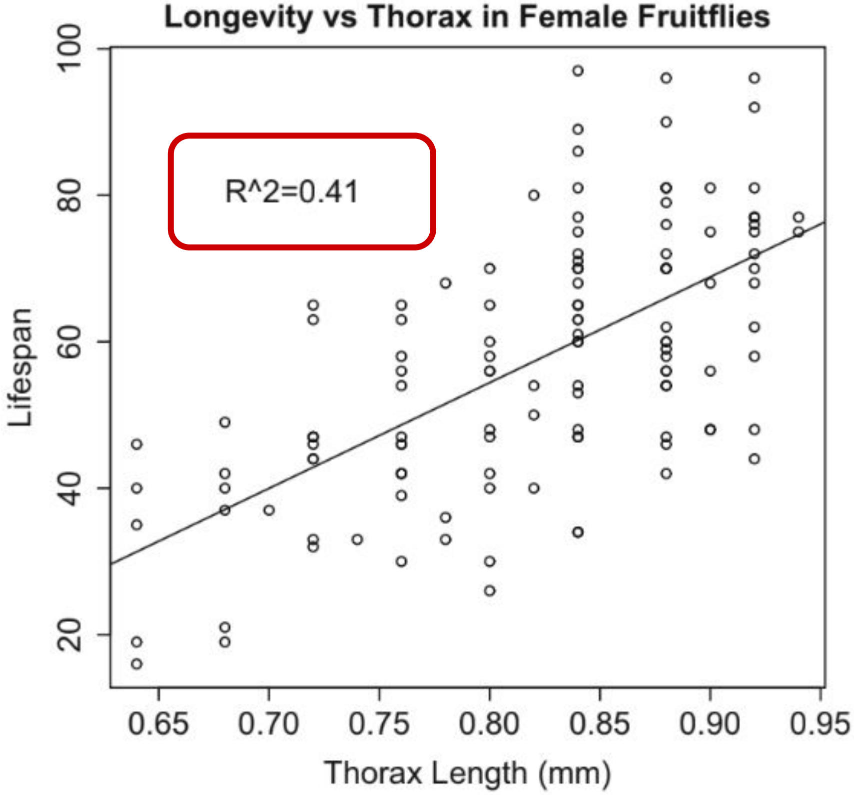
Quality Measure: $$\mathrm{var}[y] = \mathrm{var}[\mathcal{X}\beta] + \mathrm{var}[e]$$

$$R^2 = \frac{\mathrm{var}[\mathcal{X}\beta]}{\mathrm{var}[y]}$$

Higher is better!

Comparing R^2





R vs R-squared

- **R (correlation):** the strength of the relationship between an independent and a dependent variable.

- **R-squared:** the extent to which the variance of one variable explains the variance of the second variable.

Metric: Cook's Distance

Goal: check the effect of removing a data point on regression

Model: estimate linear regression parameters by omitting the ith data point

$$y_i^{(p)} = \mathcal{X}\beta_i$$

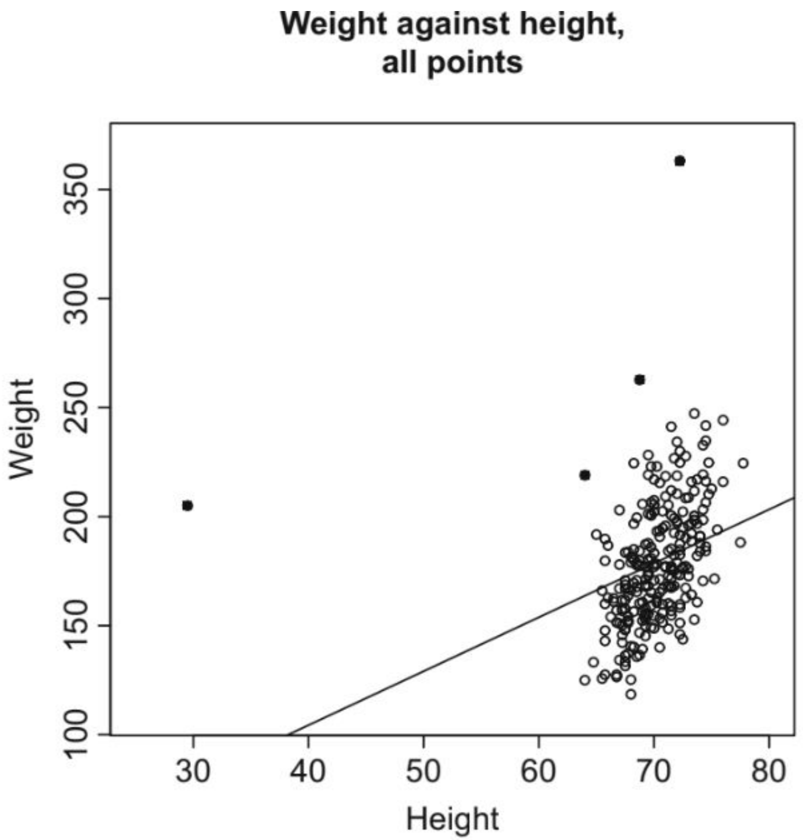
this would give us the Cook's distance for the ith data point

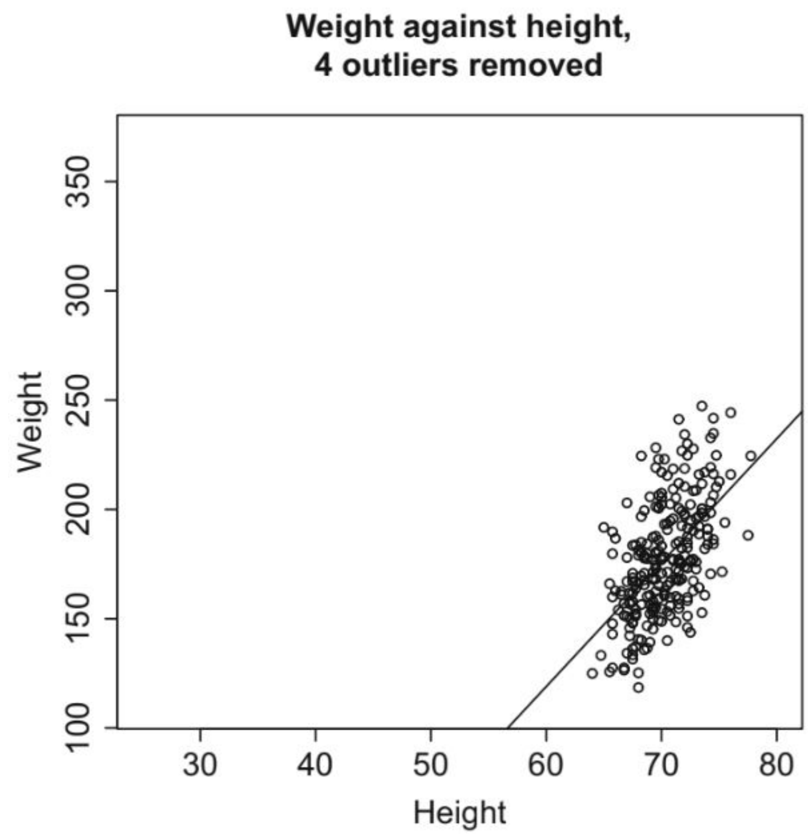
- Points with higher cook's distance - require closer inspection.

Will an outlier have a high cook's distance?

Why use Cook's distance?

Points with high cook's distance means other points cannot predict it well, e.g.,





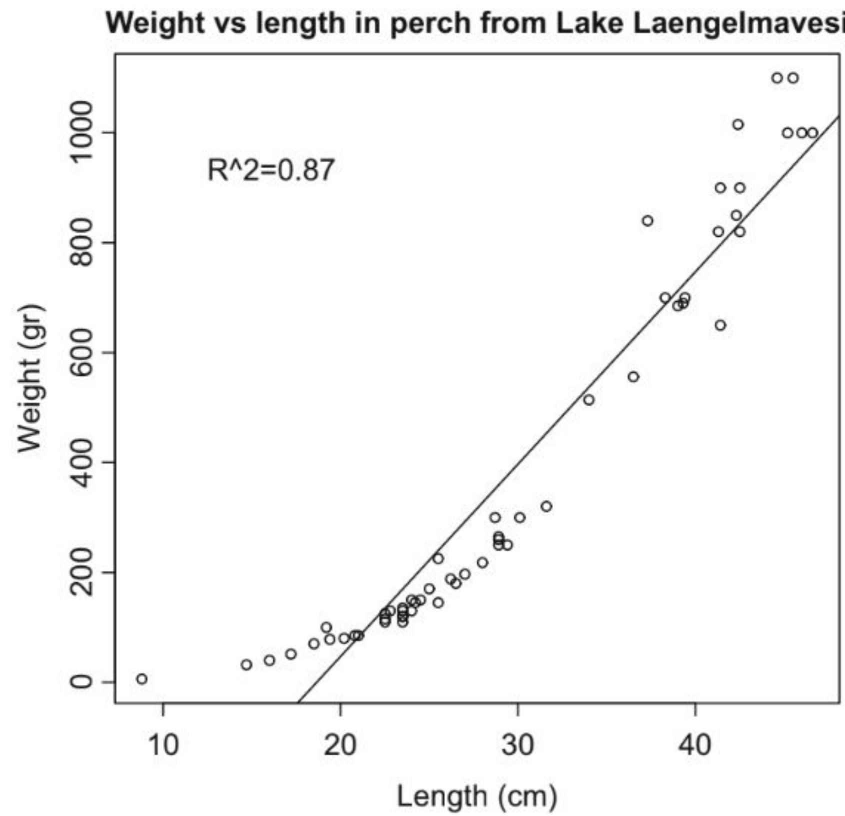
Summary:Metrics for model analysis

- Mean-squared error (lower is better)
- R-squared (higher is better)
- Cook's distance (lower is better)

- What does cook's distance offer differently from R-squared?

Tips for evaluating your regression

- Plot your data
- Check if it predicts a constant
- Check for a random residual



Box-Cox transformation

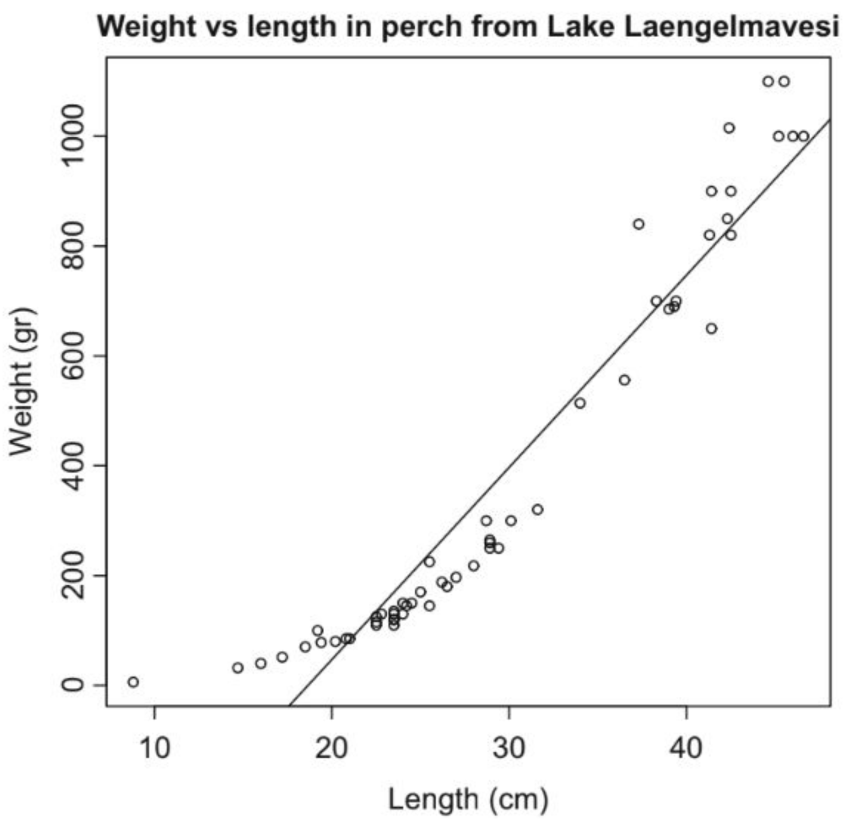
Definition: Transformation of the dependent variable that improves the regression

$$y_i^{(bc)} = \begin{cases} \dfrac{y_i^{\lambda} - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log y_i & \text{if } \lambda = 0 \end{cases}$$

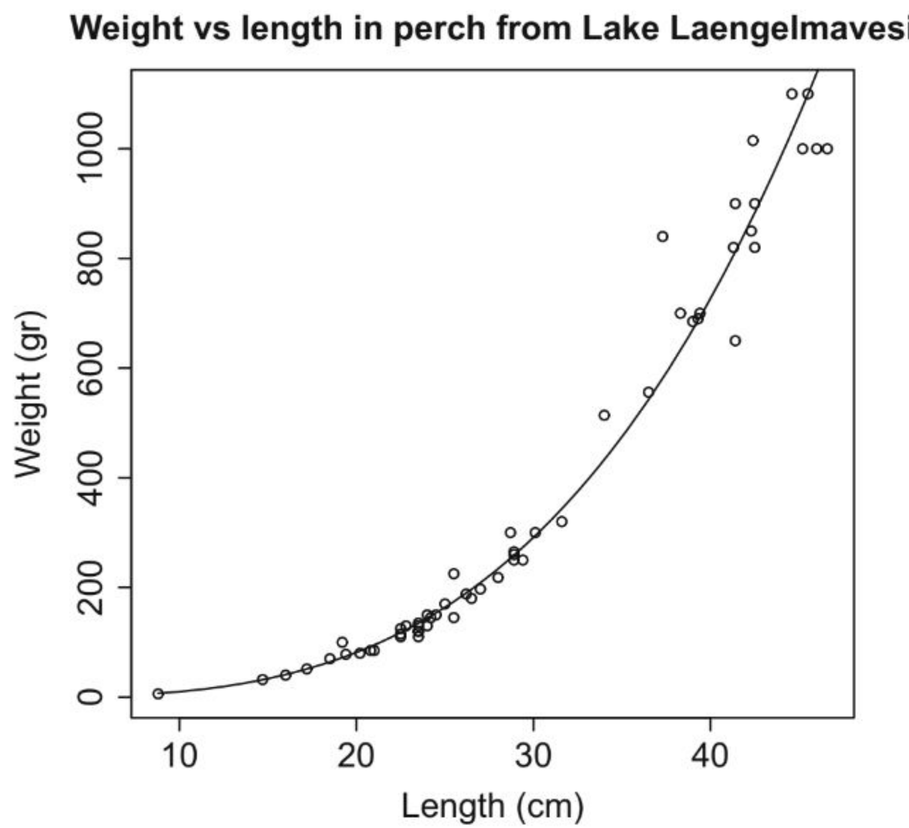
Estimate λ using maximum likelihood

Box-Cox Example

Without Transformation



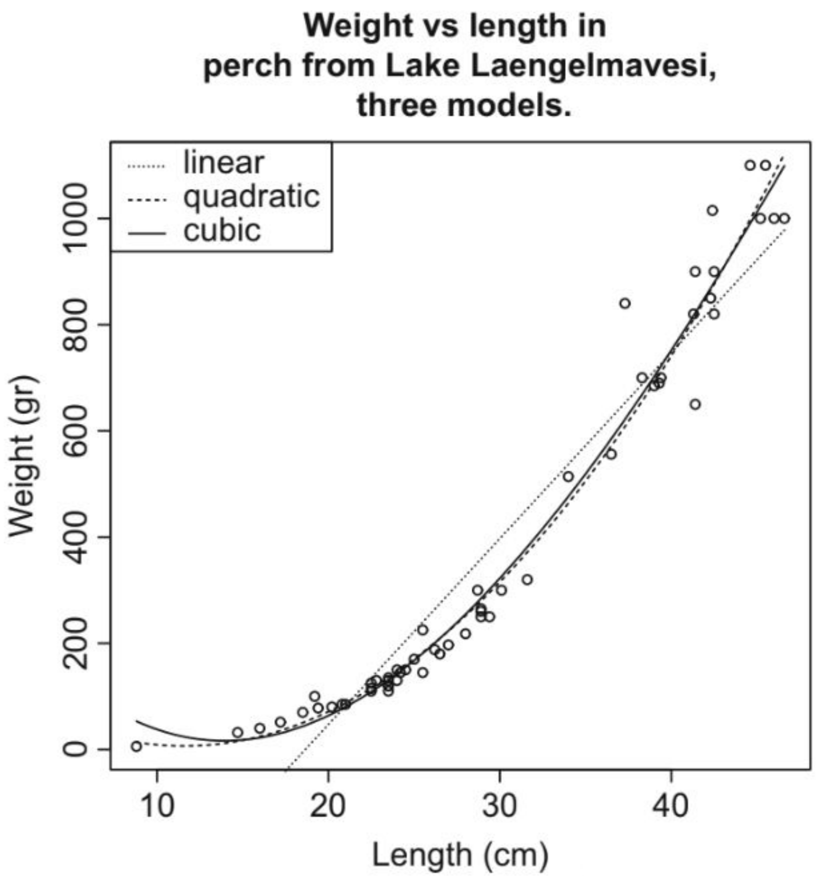
With Transformation

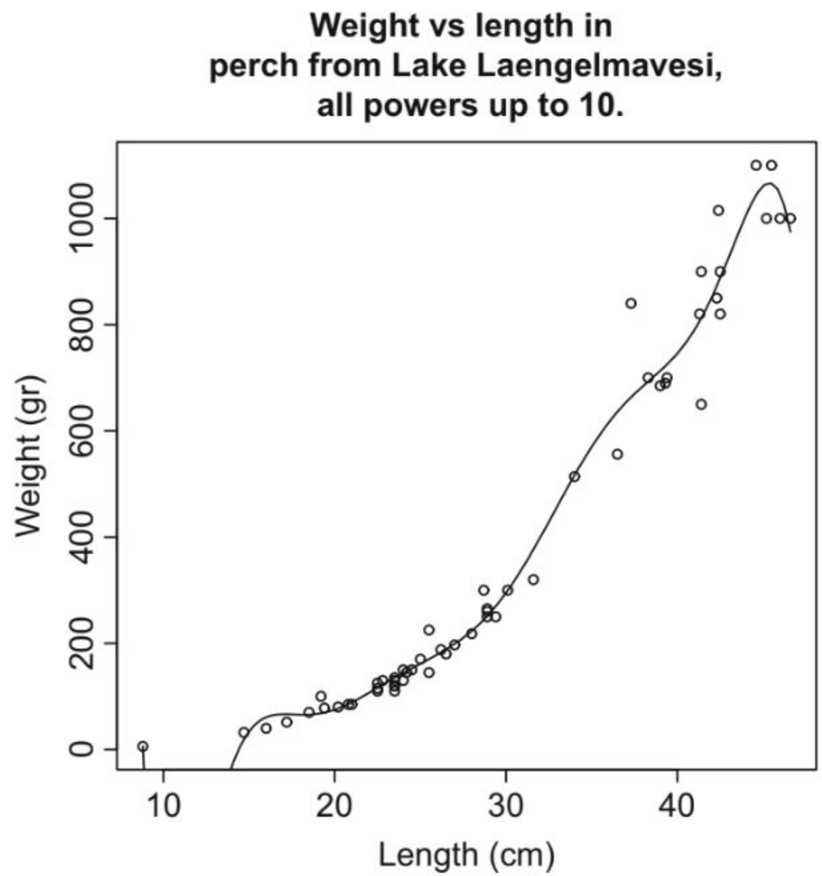


Today

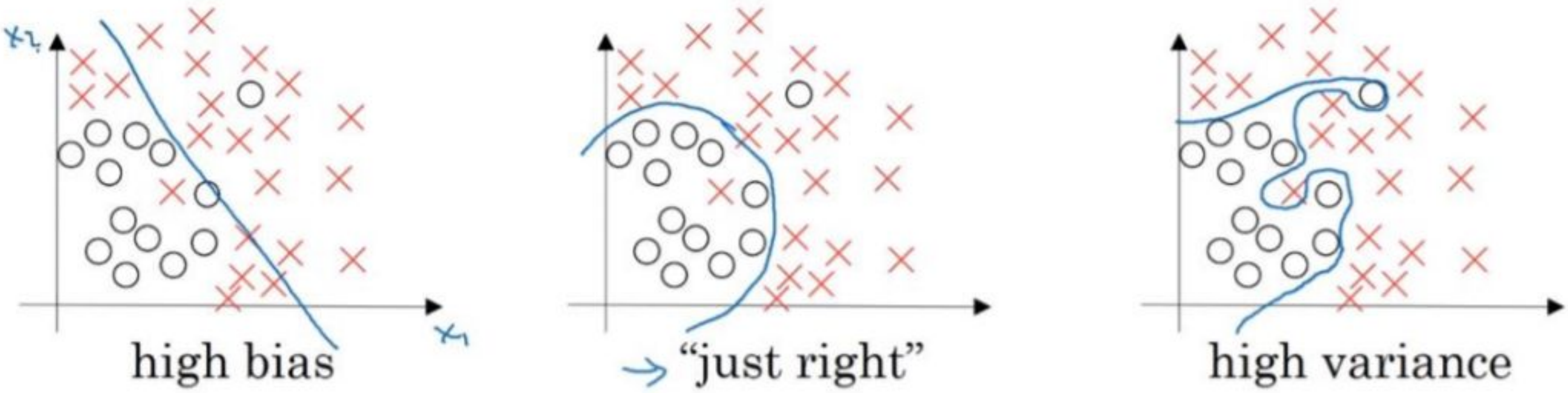
- Expectation Maximization
- Linear Regression
- Analyzing your model
- **Regularizing Linear Regression**

Recall: Overfitting





Bias vs variance



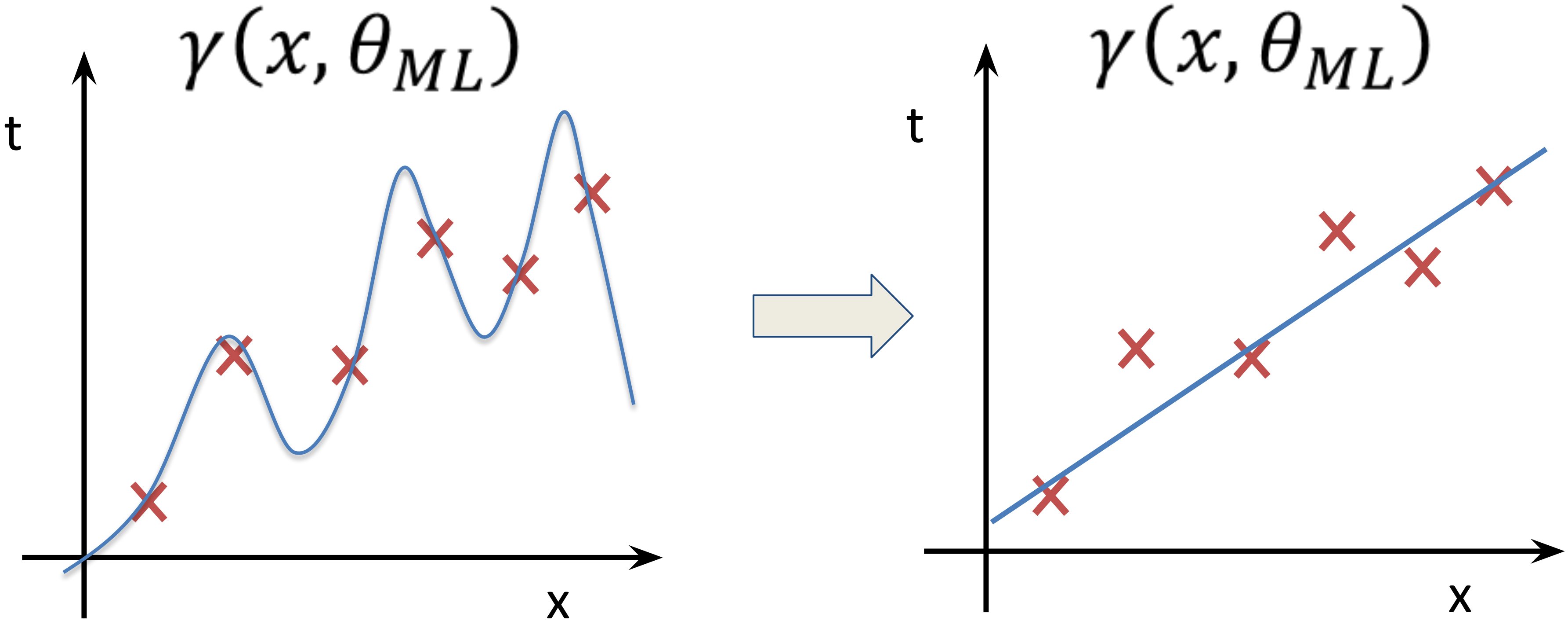
Not performing well on training data

(underfit)

Not generalizing well from training to unseen data

(overfit)

Bias and Variance

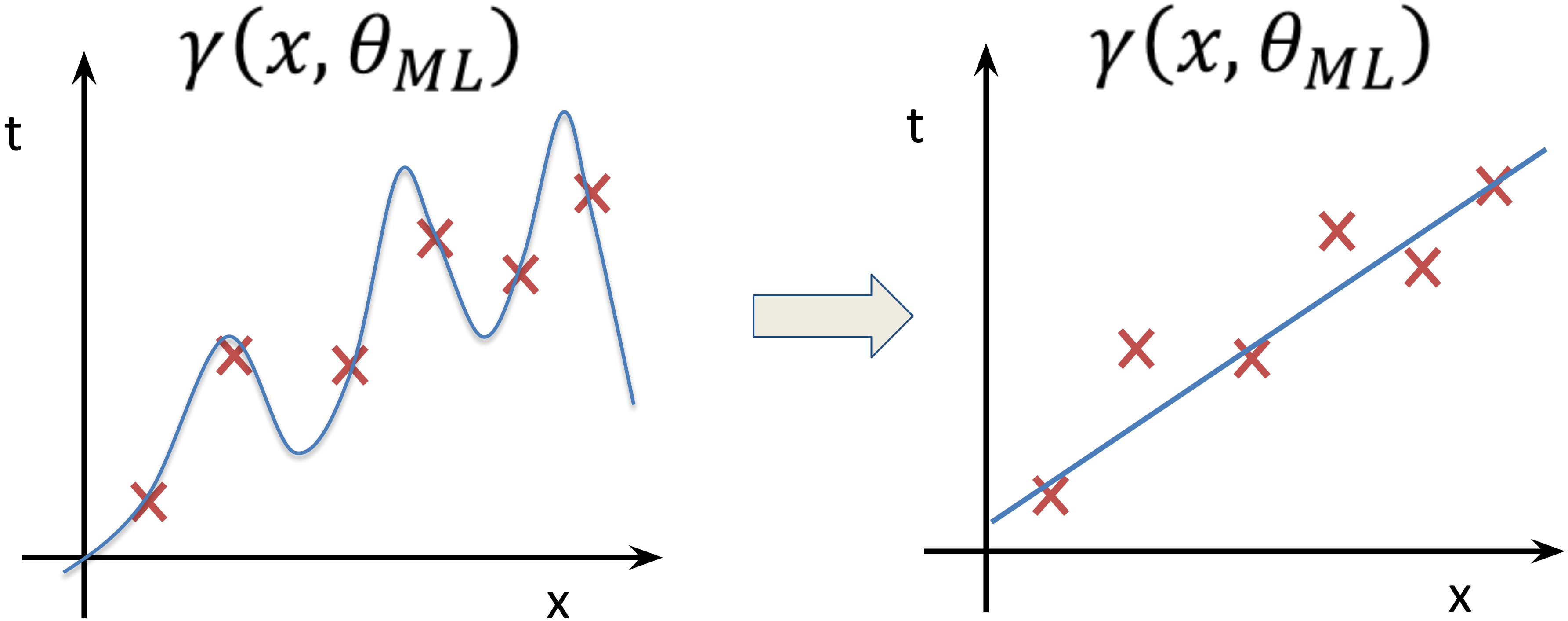


Low training error

High training error

Bias: Inability of the model to capture the true relationship between data and labels

Bias: Inability of the model to capture the true relationship between data and labels



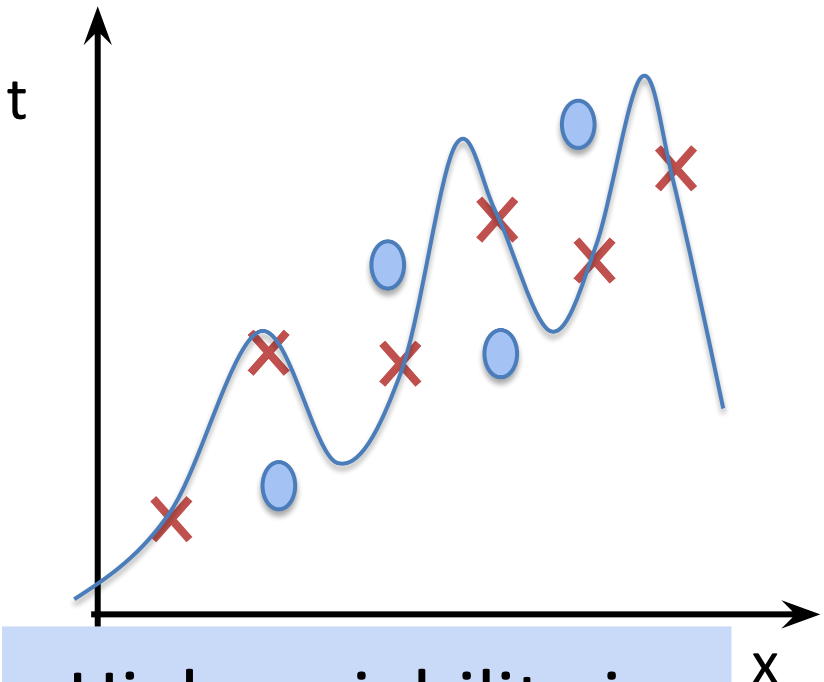
Low training error

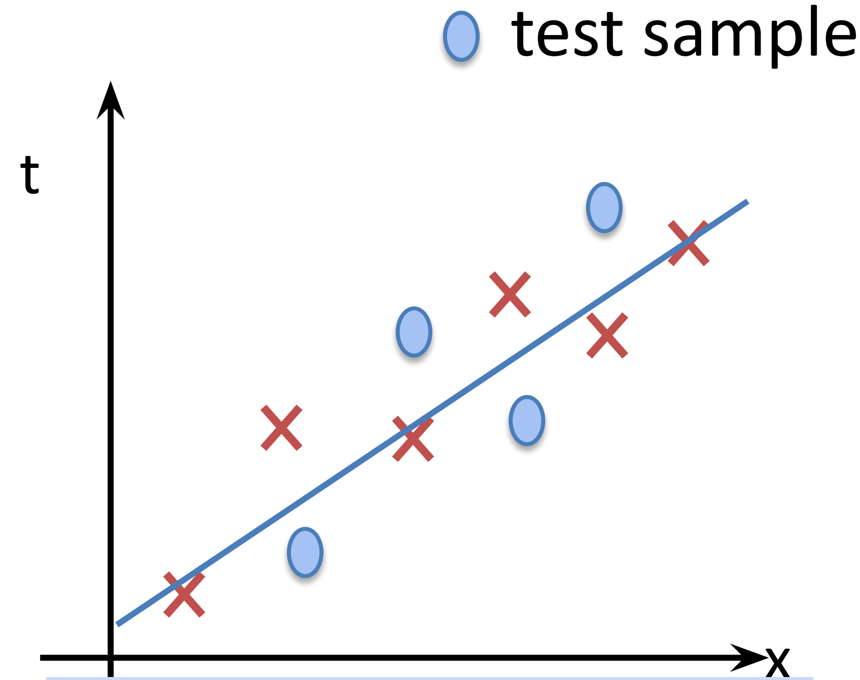
Low bias

High training error

High bias

Test data





High variability in loss on test data

High variance

Low variability in the loss on test data

Low variance

Bias vs Variance

- **Irreducible Error:** Unavoidable incorrect predictions

- **Error due to Bias**: Difference between the model prediction and the correct value.

- **Error due to Variance**: How much the predictions for a given point vary between different realizations of the model.

The Bias-Variance Trade-off

There is a trade-off between bias and variance:

- Less complex models (fewer parameters) have high bias and hence low variance
- More complex models (more parameters) have low bias and hence high variance
- Optimal model will have a balance

Which is worse between having a model with high bias or a model with high variance?

Recall: Reasons for overfitting

- **What is it:** Model overfits to the underlying training data.

Model too complex	Model not well-trained	Small training data	Train-test data distribution mismatch
⇩	⇩	⇩	⇩
1. Model pruning 2. Regularization	1. Early stopping 2. Ensemble models 3. Better loss functions	Data augmentation	Iterative training

Recall: Reasons for overfitting

- **What is it:** Model overfits to the underlying training data.

Model too complex	Model not well-trained	Small training data	Train-test data distribution mismatch
1. Model pruning **2. Regularization**	1. Early stopping 2. Ensemble models 3. Better loss functions	Data augmentation	Iterative training

Recall: Classification training objective

λ = regularization strength (hyperparameter)

$$\text{Loss} = \underbrace{\frac{1}{N}\sum_{i=1}^{N} C(y_i, \gamma_i)}_{} + \underbrace{\lambda\left(\frac{a^T a}{2}\right)}_{}$$

Data loss: Model predictions should align with ground truth

Regularizer: Prevent the model from doing too well.

Slide motivation: Justin Johnson

Regularized Regression Objective

Basic Idea:

$$\boxed{\text{Error}} + \boxed{\text{Regularizer}}$$

$$\boxed{\frac{1}{N}(y - X\beta)^T(y - X\beta)} + \boxed{\lambda\beta^T\beta}$$

where $\lambda > 0$

The Bias-Variance Trade-off

There is a trade-off between bias and variance:

- Less complex models (fewer parameters) have high bias and hence low variance
- More complex models (more parameters) have low bias and hence high variance
- Optimal model will have a balance

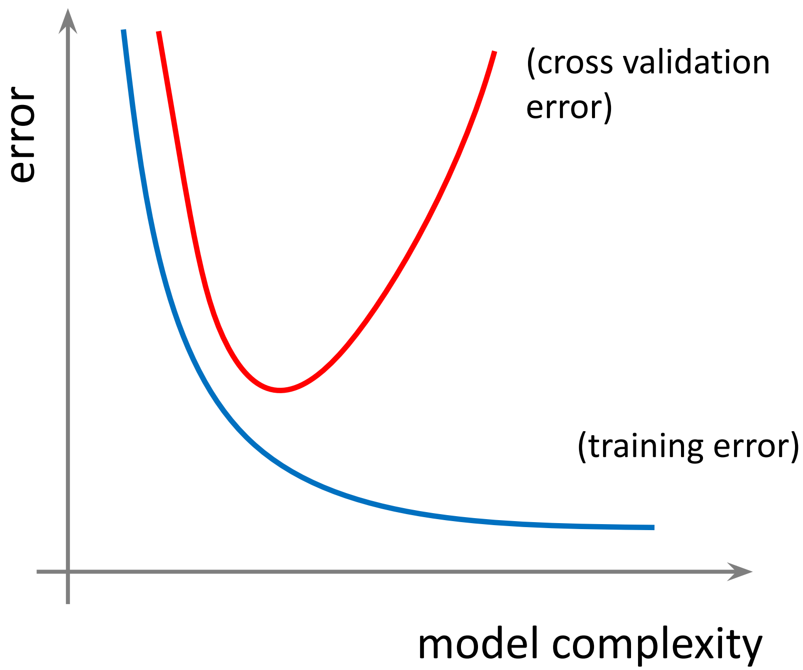
How do we search for the right model hyper parameters?

Diagnosing model's performance

Suppose your learning algorithm is performing less well than you were hoping. ($\mathbb{E}_{cv}$ or $\mathbb{E}_{test}$ is high.) Is it a bias problem or a variance problem?

Diagnosing model's performance

Suppose your learning algorithm is performing less well than you were hoping. ($\mathbb{E}_{cv}$ or $\mathbb{E}_{test}$ is high.) Is it a bias problem or a variance problem?



Bias (underfit):

$\mathbb{E}_{train}$ will be high,
$\mathbb{E}_{cv} \approx \mathbb{E}_{train}$

Variance (overfit):

$\mathbb{E}_{train}$ will be low,
$\mathbb{E}_{cv} \gg \mathbb{E}_{train}$

Next Class

Regression II: More model selection, regression trees

Reading: Forsyth Ch 11.1-11.2