Introduction

ECE 449, Machine Learning

Classification

- From data to discrete classes

Examples: Spam Filtering

Data

Welcome to New Media Installation: Art that Learns

Carlos Guestrin to 10615-announce, Osman, Michel show details 3:15 PM (8 hours ago) Reply

Hi everyone,

Welcome to New Media Installation:Art that Learns

The class will start tomorrow.
Make sure you attend the first class, even if you are on the Wait List.
The classes are held in Doherty Hall C316, and will be Tue, Thu 01:30-4:20 PM.

By now, you should be subscribed to our course mailing list: 10615-announce@cs.cmu.edu.
You can contact the instructors by emailing: 10615-instructors@cs.cmu.edu

Natural _LoseWeight SuperFood Endorsed by Oprah Winfrey, Free Trial 1 bottle, pay only $5.95 for shipping mfw rlk Spam | X

Jaquelyn Halley to nherrlein, bcc: thehorney, bcc: ang show details 9:52 PM (1 hour ago) Reply

=== Natural WeightL0SS Solution ===

Vital Acai is a natural WeightL0SS product that Enables people to lose wieght and cleansing their bodies faster than most other products on the market.

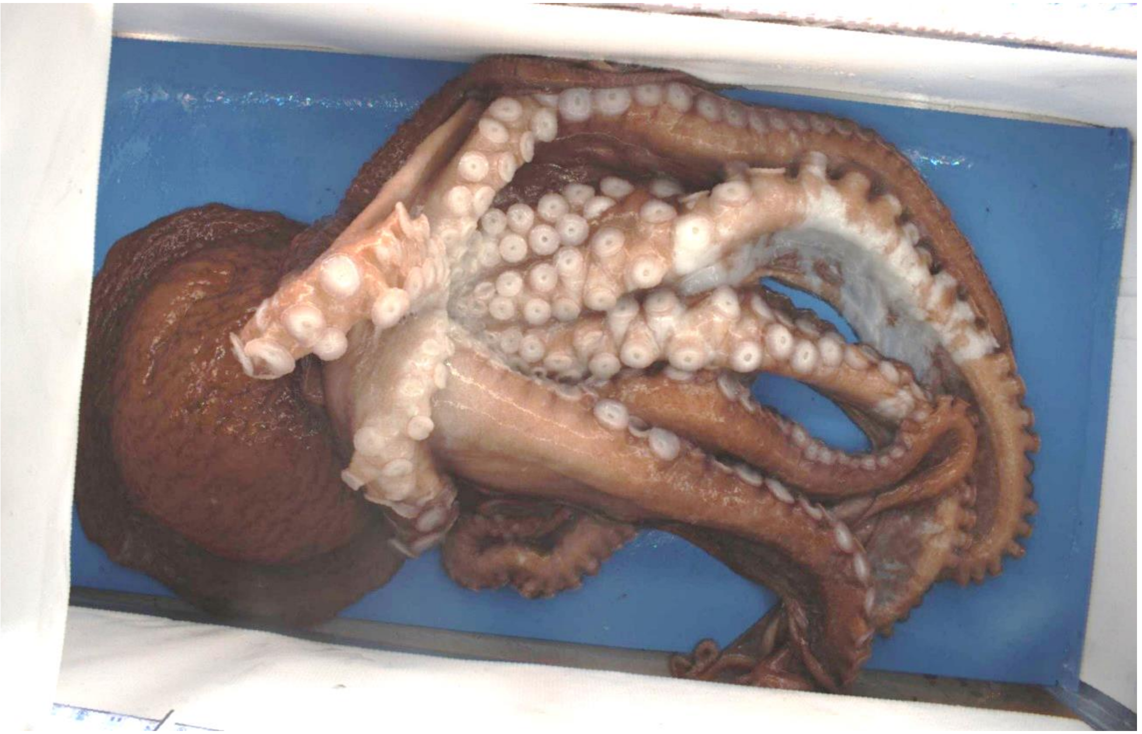
Here are some of the benefits of Vital Acai that You might not be aware of. These benefits have helped people who have been using Vital Acai daily to Achieve goals and reach new heights in there dieting that they never thought they could.

* Rapid WeightL0SS
* Increased metabolism - BurnFat & calories easily!
* Better Mood and Attitude
* More Self Confidence
* Cleanse and Detoxify Your Body
* Much More Energy
* BetterSexLife
* A Natural Colon Cleanse

Prediction

Spam/Not Spam

Examples: Image Classification



Examples: Weather Prediction

Regression

- Predicting a numeric value

Examples: Stock Market

- Predict stock prices given stock history and today's news

Examples: Weather Prediction Revisited

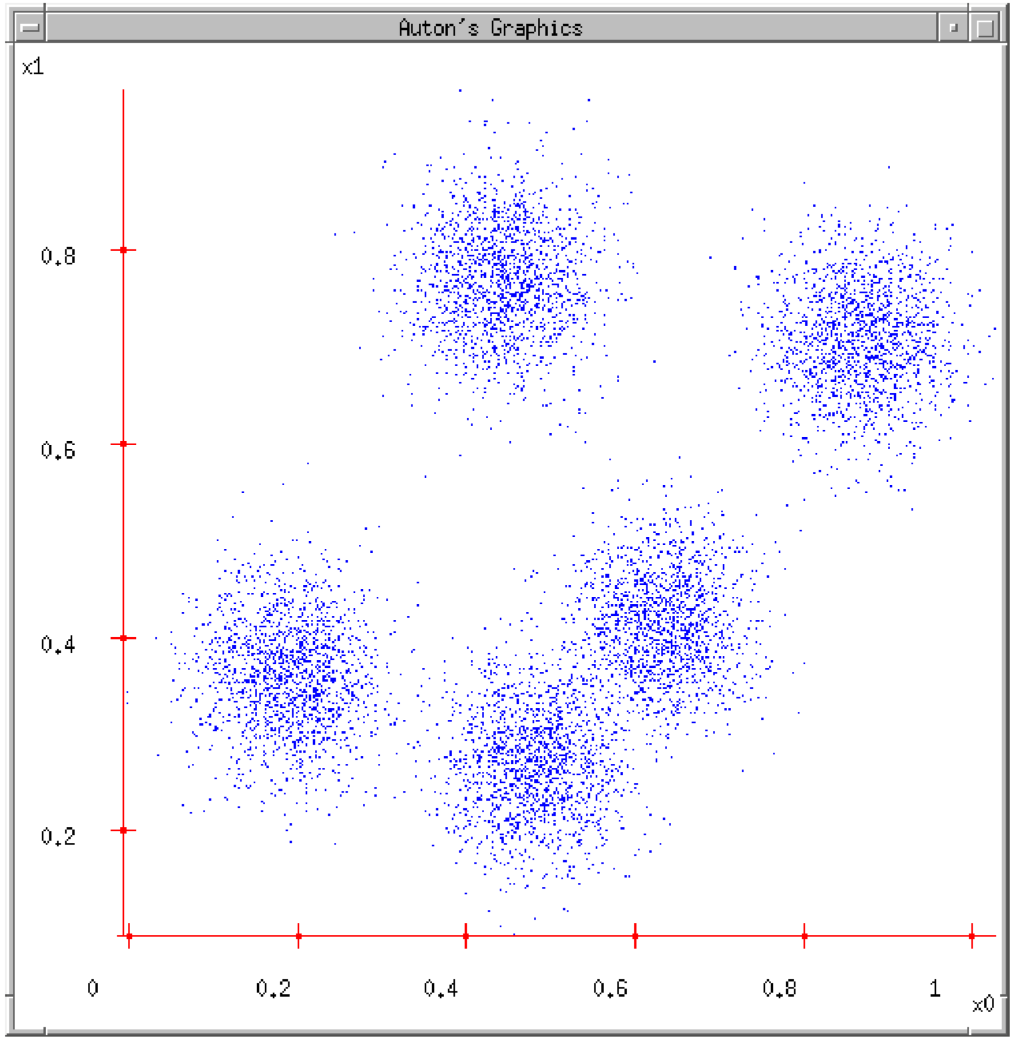
Temperature

72° F

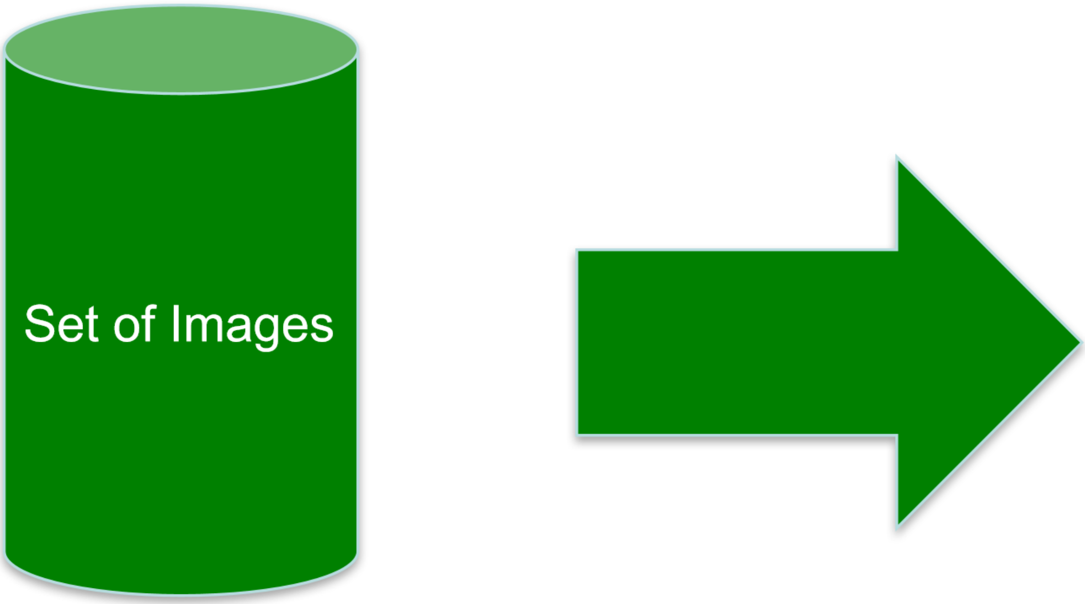
Clustering

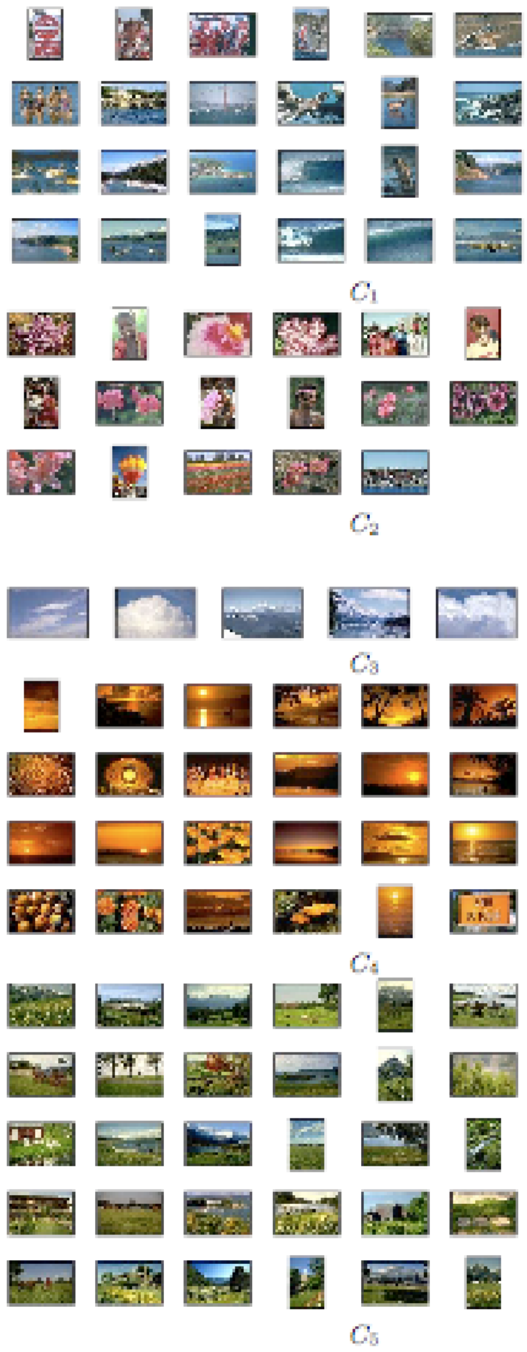
- Discovering structure in data

Examples: Clustering Data



Examples: Clustering Images



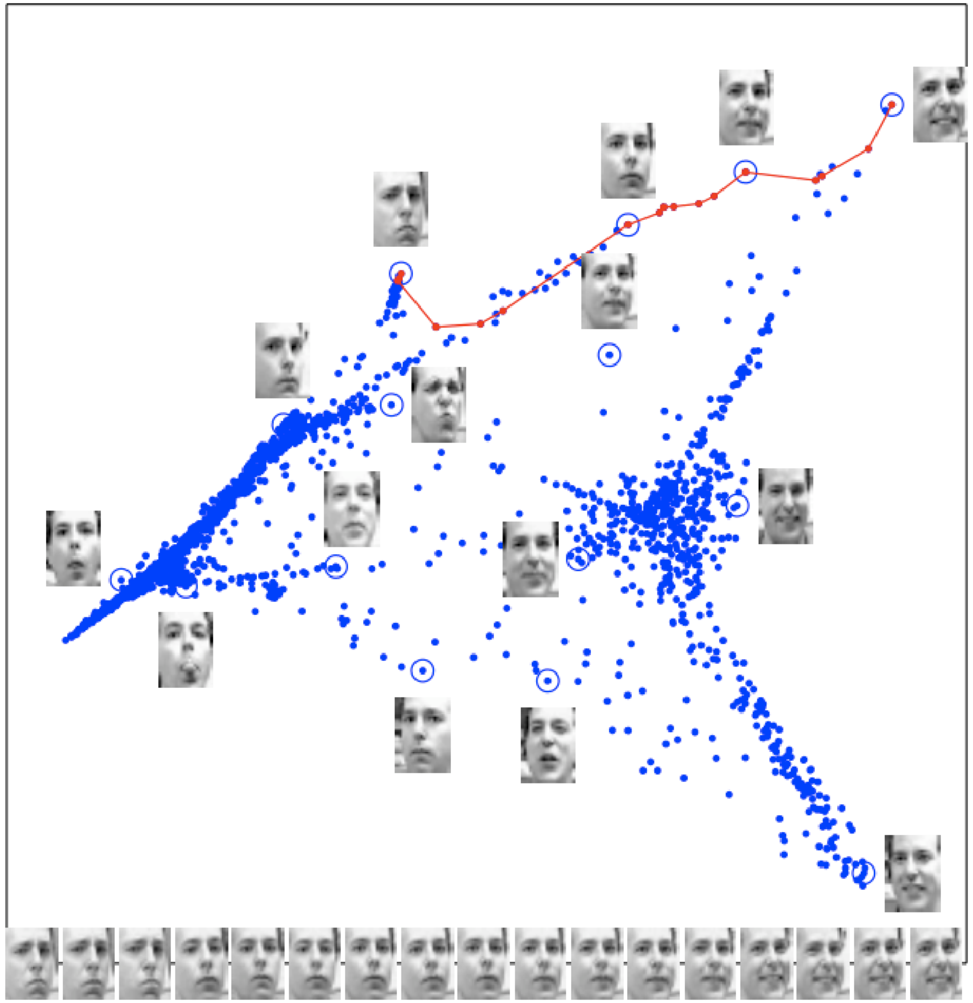


[Goldberger et al.]

Embedding

- Visualization and discriminative feature learning

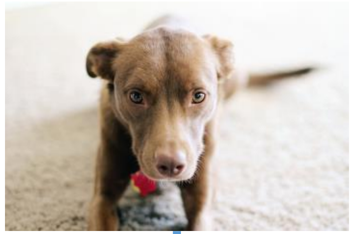
Examples: Face Embedding



[Saul & Roweis '03]

Classification

- Predict discrete classes



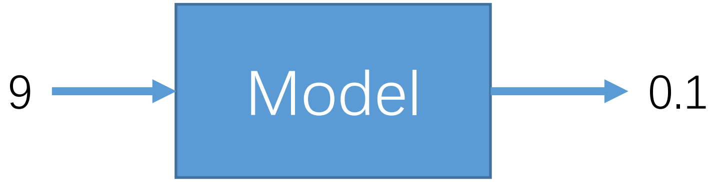
Model

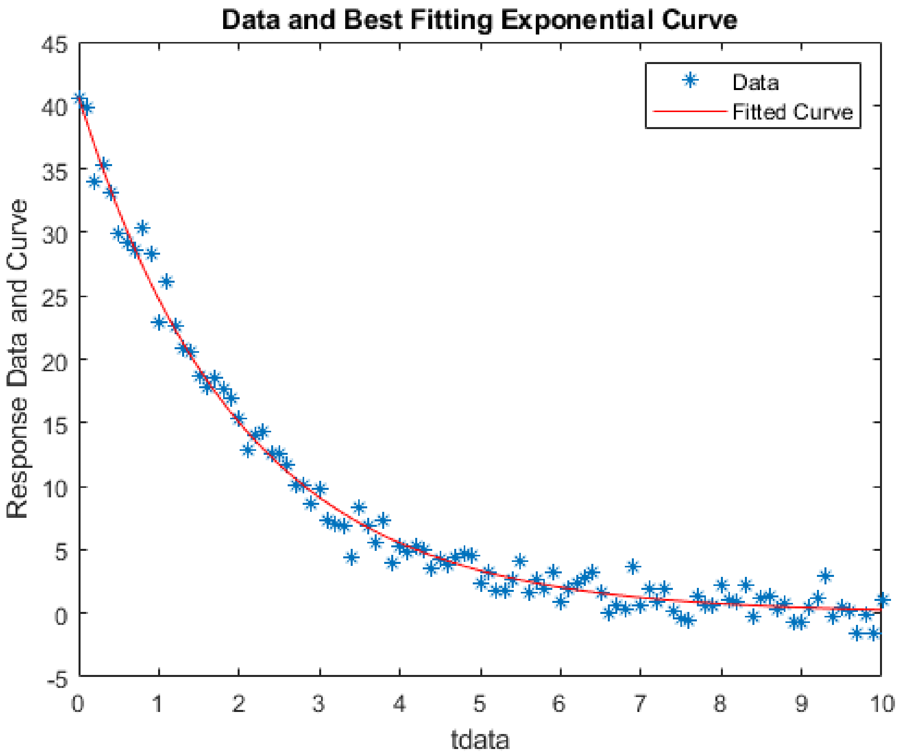
Prediction 0.8 0.15 0.05

Dog Cat Bird

Regression

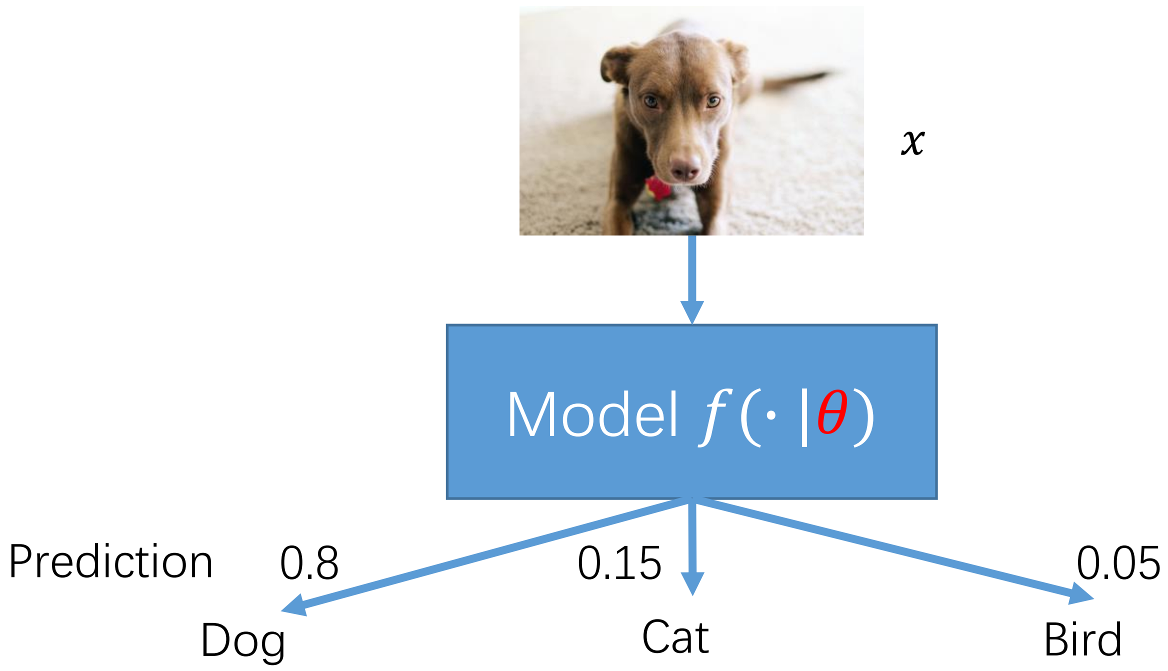
- Predicting a numeric value



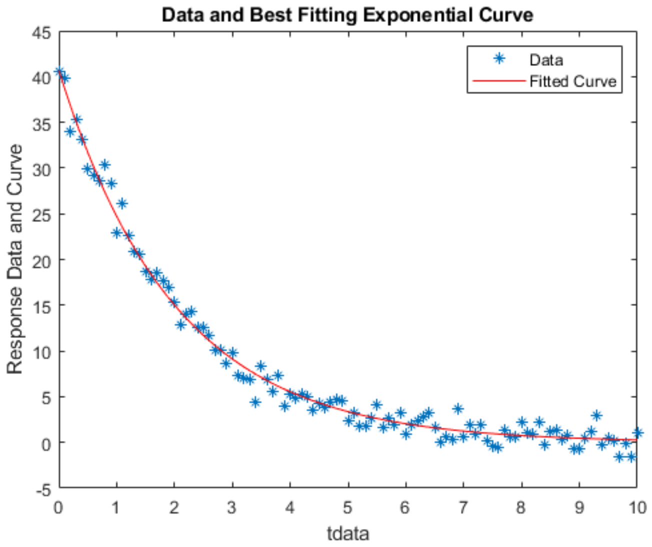


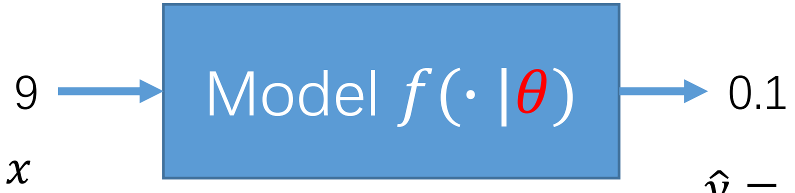
Model

- Treat model as a function (linear or nonlinear)



$$[\hat{p}_{dog}, \hat{p}_{cat}, \hat{p}_{bird}] = f(x|\theta)$$

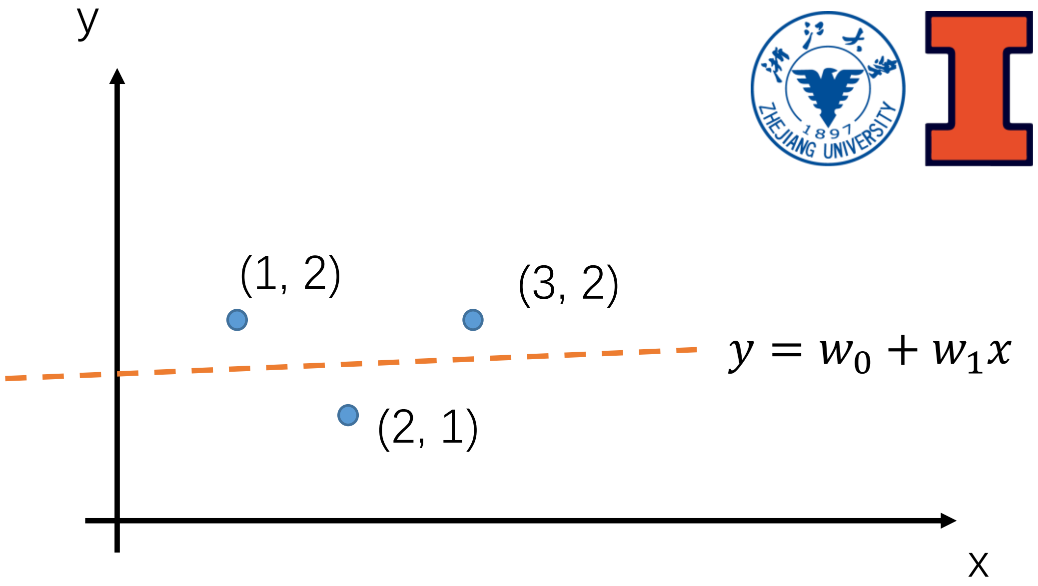




$$\hat{y} = f(x|\theta)$$

Training

- Given data, Learn θ
- For example, line fitting
  - We have three points (x,y), i.e., (1,2), (2,1), (3,2)
  - $f(x|\theta) = w_0 + w_1 x$, here $\theta = \{w_0, w_1\}$



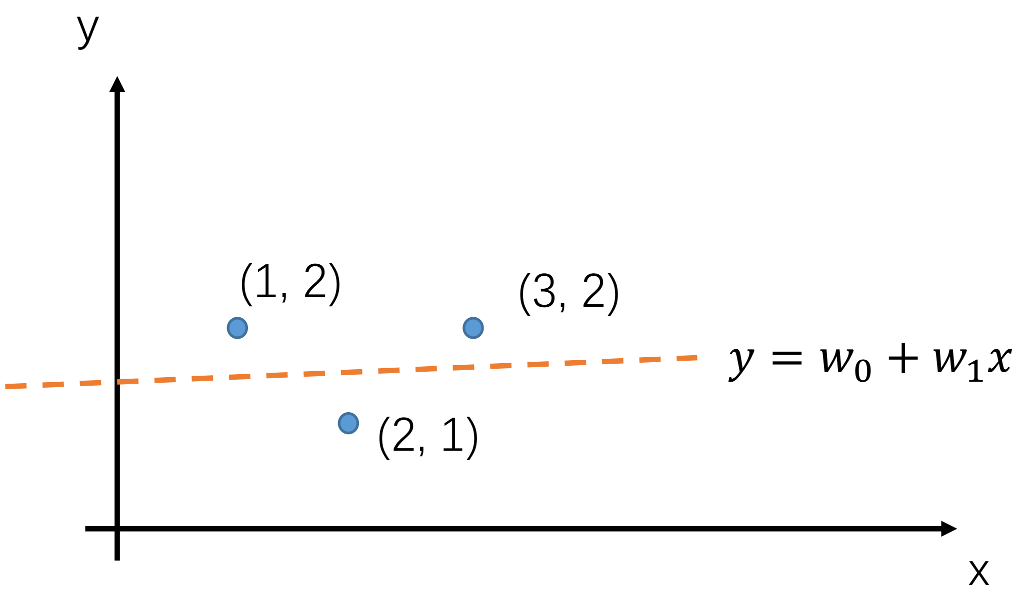
- Let's define the fitting error
  - $loss = (2 - f(1|\theta))^2 + (1 - f(2|\theta))^2 + (2 - f(3|\theta))^2$

Data (1,2) Data (2,1) Data (3,2)

  - Define prediction $\hat{y} = f(x|\theta)$
  - $loss = l(y, \hat{y}) = \sum_i (y - f(x|\theta))^2$

Training

- Estimate $\theta = \{w_0, w_1\}$
  - $l = \sum_i (y - f(x|\theta))^2 = \left(y_1 - (w_0 + w_1 x_1)\right)^2 + \left(y_2 - (w_0 + w_1 x_2)\right)^2 + \left(y_3 - (w_0 + w_1 x_3)\right)^2$



- Minimize the loss
  - $\hat{\theta} = arg \min_{\theta} l\,(y, f(x|\theta))$
- Solve θ by setting the gradient to 0
  - $\frac{\partial l}{\partial w_0} = 0, \frac{\partial l}{\partial w_1} = 0$

Training

- How about classification?
  - Set classifier $[\hat{p}_{dog}, \hat{p}_{cat}, \hat{p}_{bird}] = f(x|\theta)$
  - If we have many image-label pairs, we want to estimate θ as well.
- Convert class label to one-hot label
  - Set $y = [dog, cat, bird]$, then $y_1 = [1,0,0]$, $y_2 = [0,1,0]$, $y_3 = [0,0,1]$
- Define loss
  - $loss = -\left(y_1^T \log\left(f(x_1|\theta)\right) + y_2^T \log\left(f(x_2|\theta)\right) + y_3^T \log\left(f(x_3|\theta)\right)\right)$
  - $loss = -\sum_i y_i^T \log\left(f(x_i|\theta)\right)$
- Minimize the loss
  - $\hat{\theta} = arg \min_{\theta} l\,(y, f(x|\theta))$

x_1=

y_1=dog

x_2=

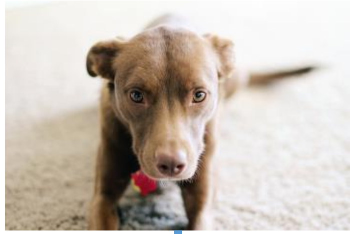
y_2=cat

x_3=

y_3=bird

Testing

- Given the learned model (function) $f(\cdot|\hat{\theta})$, we can input any testing data x to get the prediction $\hat{y} = f(x|\hat{\theta})$



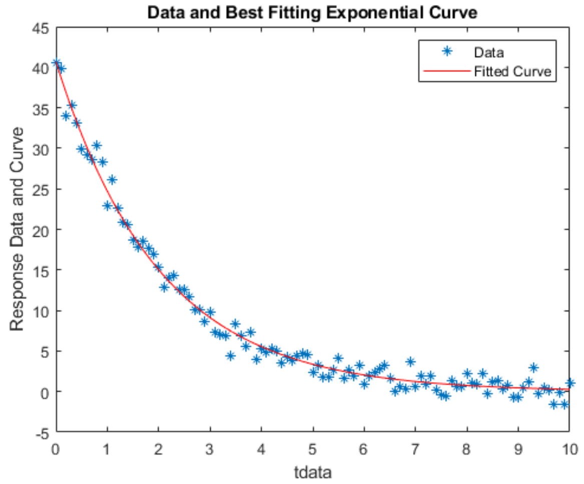
x

Model $f(\cdot|\hat{\theta})$

Prediction 0.8 0.15 0.05

Dog Cat Bird

$[\hat{p}_{dog}, \hat{p}_{cat}, \hat{p}_{bird}] = f(x|\hat{\theta})$



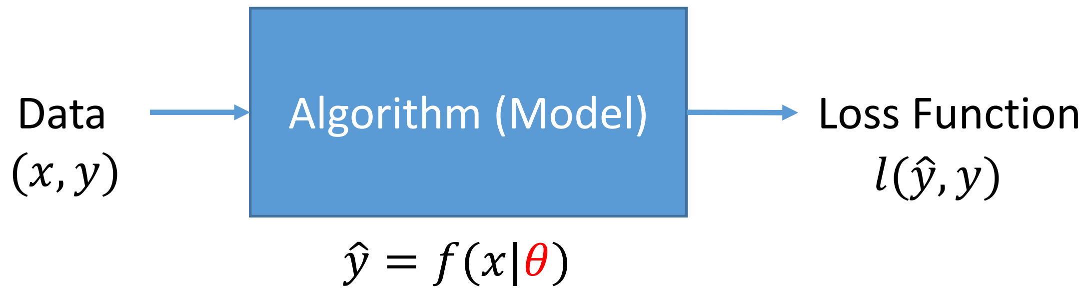
9 → Model $f(\cdot|\hat{\theta})$ → 0.1

x

$\hat{y} = f(x|\hat{\theta})$

Big Picture

- Algorithms that give computers the ability to learn from experience (data) to do specific tasks
  - Different tasks use different types of data, different learning algorithms
  - Performance driven learning: minimize loss function



x: Input data
y: Ground truth label or supervision signal
θ: Model parameters
$f(x|\theta)$: Mapping from input to the target output
l: Loss function

Types of Data

- Data can be
  - Binary, numerical or categorical (ordered or not) or a combination
  - A vector/matrix/graph
- Raw input data gets mapped to numerical or indicator form (feature extraction)
- Form of output data impacts loss function

Types of Learning Algorithms/Models

- We need to define the model (function) configuration $f(x|\theta)$ before training.
  - Linear model (with respect to θ)
    - $f(x|\theta) = w_0 + w_1 x$
    - $f(x|\theta) = w_0 + w_1 x + w_2 x^2$
    - $f(x|\theta) = w_0 + w_1 x_1 + w_2 x_1 x_2$
    - …
  - Non-linear function
    - $f(x|\theta) = w_0 + w_1 x + w_2 x^2 + w_2 \log(w_1) x^3$
    - …
    - Deep learning models

Types of Learning Algorithms/Models

- Supervised learning
  - Learning data includes examples with target output, goal is to find a decision function
- Unsupervised learning
  - Learning data has no target output, goal is to learn interesting structure
- Reinforcement learning
  - Sequential decision making in a scenario with changing state and occasional reward/penalty
- Semi-supervised learning, active learning, incremental learning, curriculum learning, federated learning …

Types of Loss Functions

- Mean squared error (usually for regression, i.e., the goal is to predict continuous numerical values)
  - $\frac{1}{N}\sum(y_i - \hat{y}_i)^2$
- Cross entropy (usually for classification, i.e., the goal is to predict discrete classes)
  - $-\sum_{i=1}^{C} y_i \log \hat{y}_i$

Training and Inference/Testing

- Training
  - Given data (x, y), algorithm, minimize the loss function to estimate the model parameters θ

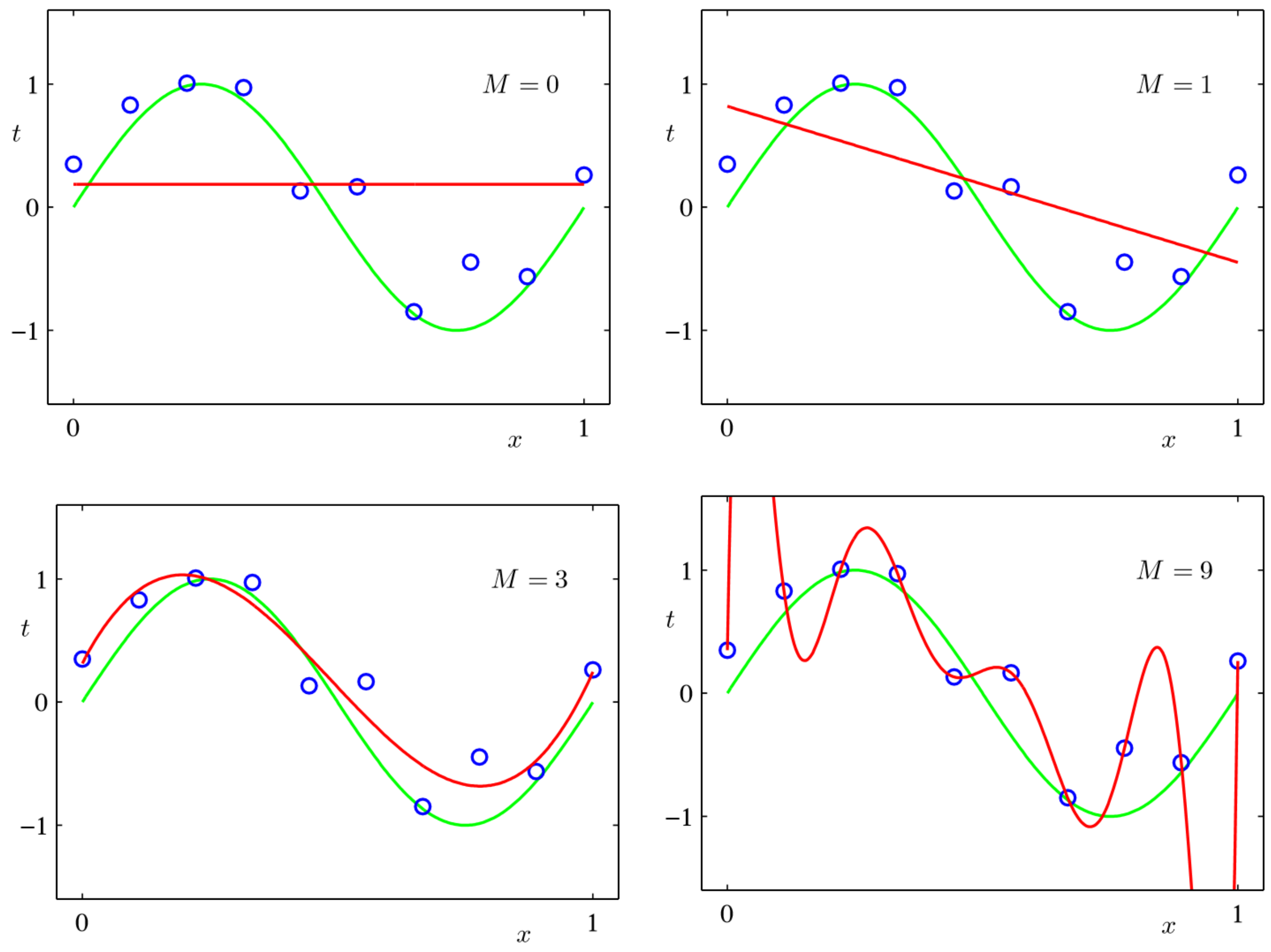
- Inference/Testing
  - Given data x (without y), algorithm and model parameters θ, get the prediction $\hat{y} = f(x|\theta)$

Machine Learning Options

- Non-parametric
  - use the data directly
  - Ex: nearest-neighbor
- Parametric
  - Assume a particular distribution
    - Ex: Gaussian → find mean & var from data
  - Assume a functional form
    - Ex: linear ($a^t x$)→ find coeffs a^t from data

Bias vs. Variance

- Example
  - N-th order regression



Bias vs. Variance

- Consider the regression problem
- Assume the perfect decision function exists: $y = f(x)$

$$\mathrm{MSE} = E_{\mathcal{T}}\left[\left(\hat{y}_0 - f(x_0)\right)^2\right]$$

$$= E_{\mathcal{T}}[(\hat{y}_0 - E_{\mathcal{T}}[\hat{y}_0])^2] + \left(E_{\mathcal{T}}[\hat{y}_0] - f(x_0)\right)^2$$

$$= \mathrm{Var}(\hat{y}_0) + \mathrm{Bias}^2(\hat{y}_0)$$

- where $\mathcal{T}$ is the training set (random samples)

Bias vs. Variance

$$MSE(x_0) = E_{\mathcal{T}}\left[\left(\hat{y}_0 - f(x_0)\right)^2\right]$$

$$= E_{\mathcal{T}}\left[\left(\hat{y}_0 - E_{\mathcal{T}}[\hat{y}_0] + E_{\mathcal{T}}[\hat{y}_0] - f(x_0)\right)^2\right]$$

$$= E_{\mathcal{T}}[(\hat{y}_0 - E_{\mathcal{T}}[\hat{y}_0])^2] + E_{\mathcal{T}}\left[\left(E_{\mathcal{T}}[\hat{y}_0] - f(x_0)\right)^2\right]$$
$$+ 2E_{\mathcal{T}}\left[(\hat{y}_0 - E_{\mathcal{T}}[\hat{y}_0])\left(E_{\mathcal{T}}[\hat{y}_0] - f(x_0)\right)\right]$$

$$= E_{\mathcal{T}}[(\hat{y}_0 - E_{\mathcal{T}}[\hat{y}_0])^2] + \left(E_{\mathcal{T}}[\hat{y}_0] - f(x_0)\right)^2$$

$$= Var(\hat{y}_0) + Bias^2(\hat{y}_0)$$

Bias vs. Variance

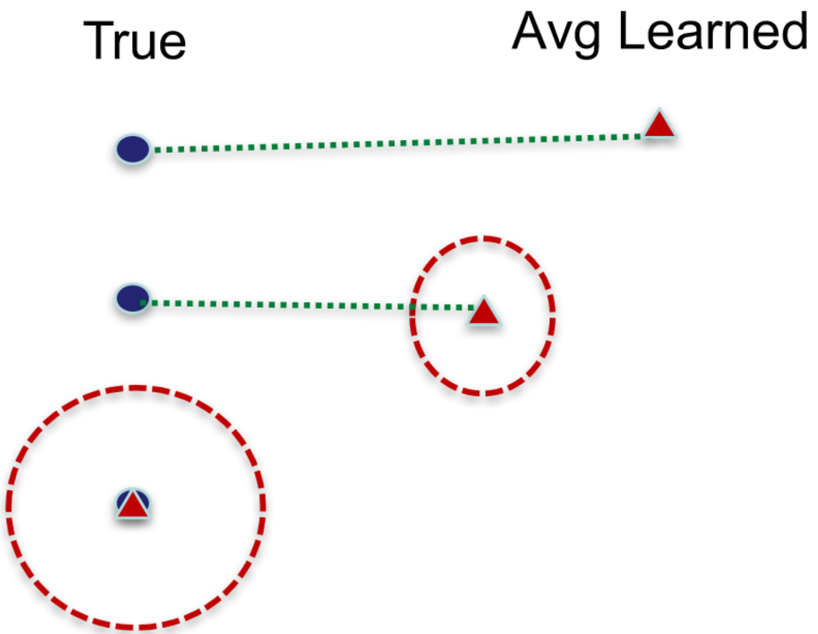
- Bias = distance between average model & theoretical best
- Variance = variability with different training samples

Deterministic classifier

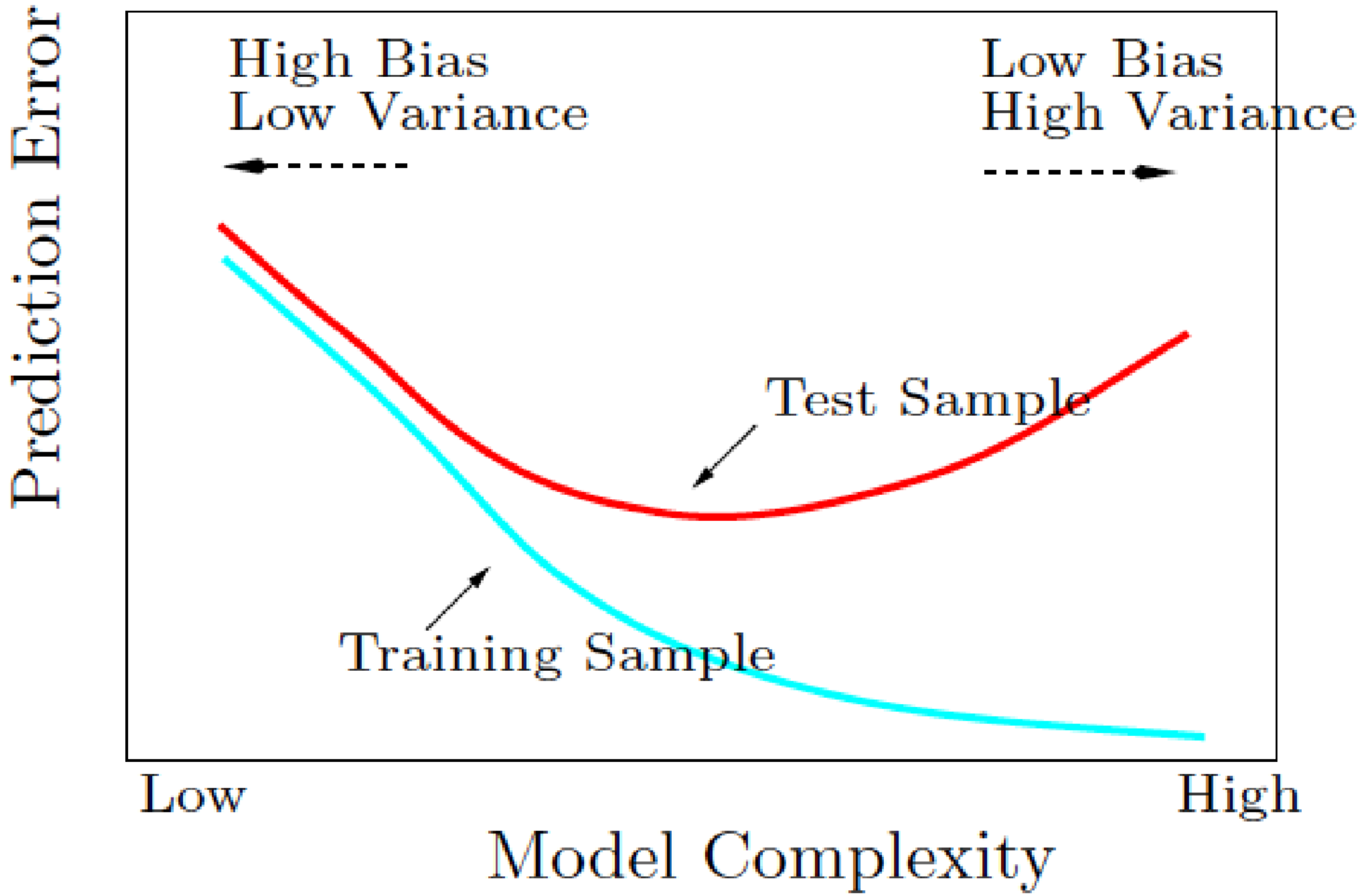
$$\alpha(x) = \omega_j \ \forall x$$

Linear classifier

True n-th order classifier

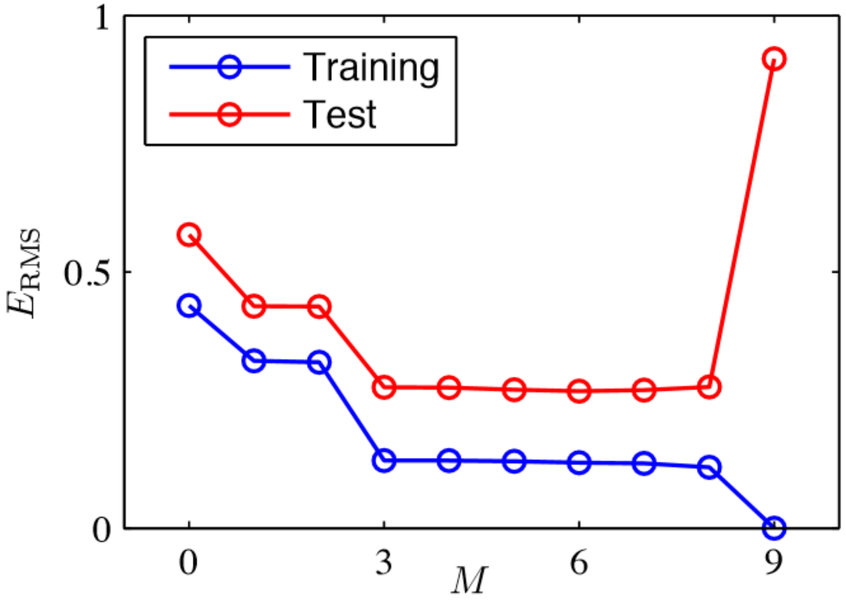


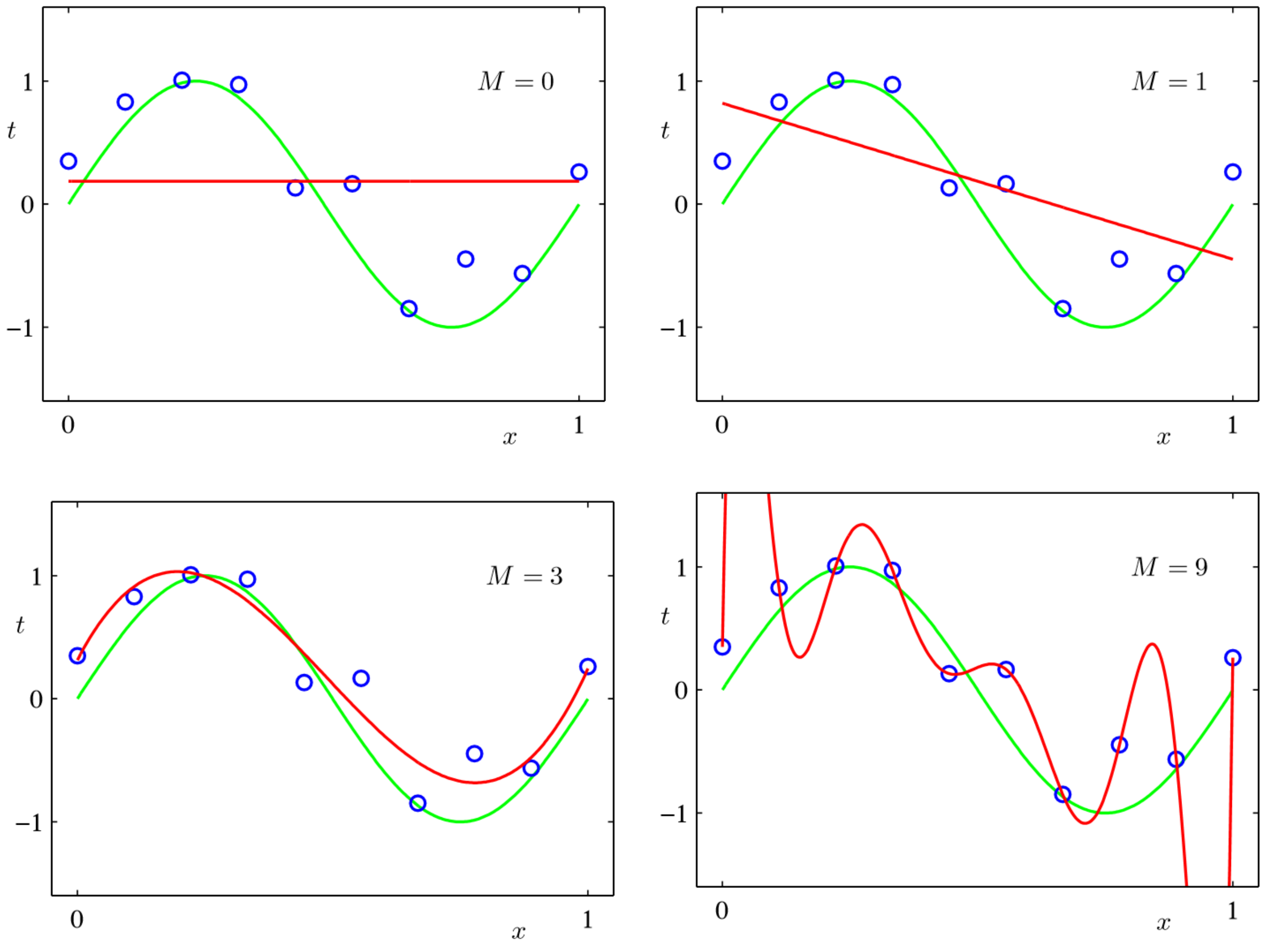
Given a Fixed Training Set



Examples

- N-th order regression





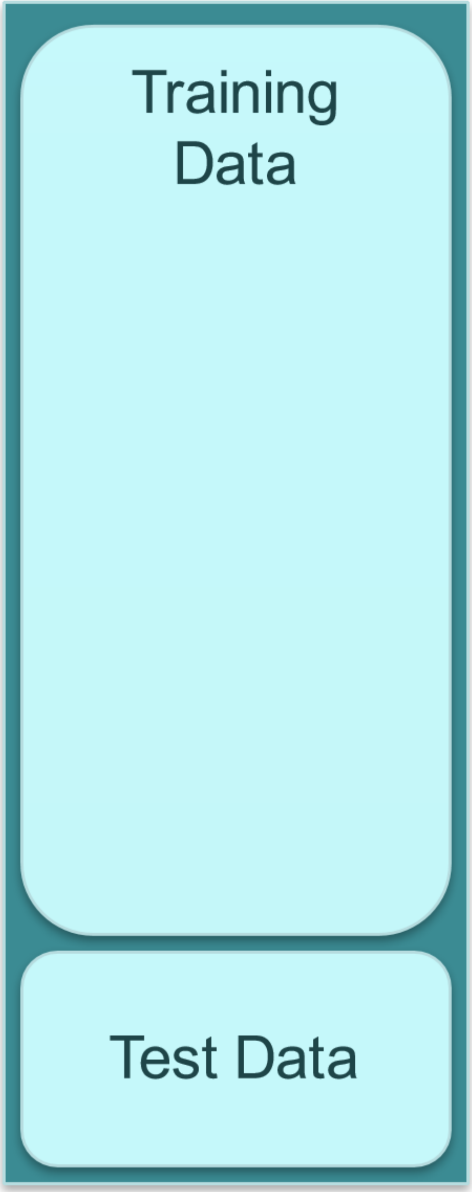
Set the Hyper-Parameters

- The model complexity ≫ #training samples → overfitting
- Whether overfitting → check the testing error

- What happens when # samples N grows?
  - For a specific model: variance ↓as N ↑
- Model complexity also depends on feature dimensionality d (higher d more params)
  - For y=Ax: scalar x,y → scalar A, vector x,y → $d_Y d_X$ params in A
  - Dimensionality reduction (feature selection or projection, supervised or unsupervised)
  - Feature extraction driven by domain knowledge

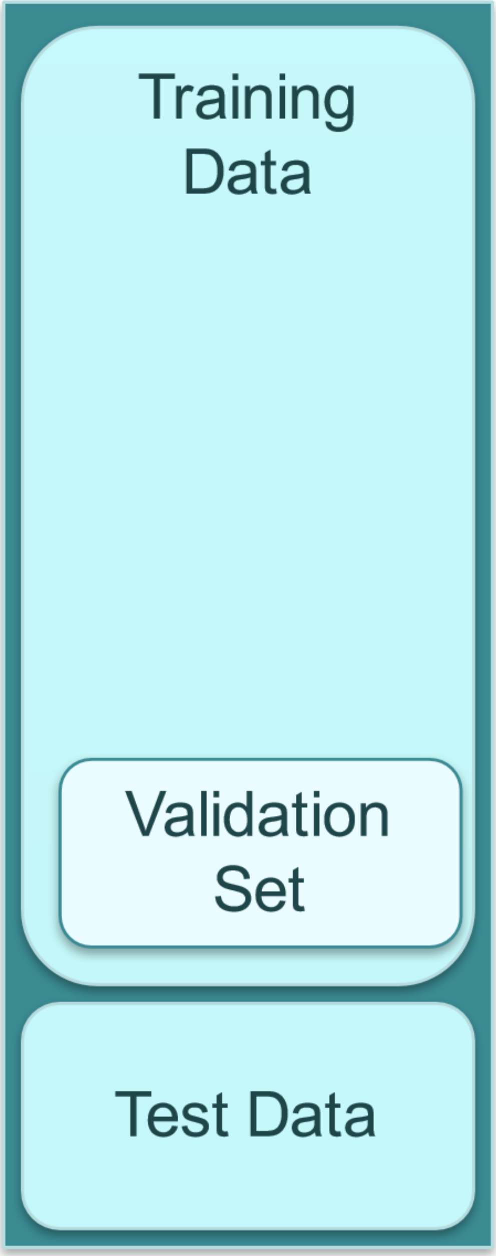
Practical Implications

- ALWAYS assess performance on data that you haven’t looked at in training or model selection (independent test set)
- What does it mean to be “independent”?
  - Two sentences in the same document are not independent
  - Two segments in the same image are not independent
- Use regularization to encourage some parameters to be small



Practical Implications

- Use a held-out validation set or cross-validation for model selection and parameter tuning
- Cross-validation (CV)
  - Partition data into N subsets
  - Train on N-1, validate on Nth
  - Rotate through all N options
  - Choose best configuration
  - Retrain on all the data with best config
- Trade-offs of CV vs. held-out
  - CV makes better use of small data sets
  - CV is more expensive



Other Wrong "Model" Problems

- Training data is not representative → impacts all ML approaches
- Could be due to
  - Sampling bias
  - Noisy observations
  - Samples are not independent