STA414/2104
Statistical Methods for Machine Learning II

Murat A. Erdogdu

Department of Computer Science
Department of Statistical Sciences

Lecture 1

UNIVERSITY OF TORONTO

Today's lecture

- This lecture is being recorded and will be shared on quercus!
- Ask questions anytime either on chat or raising your hand.

Topics:

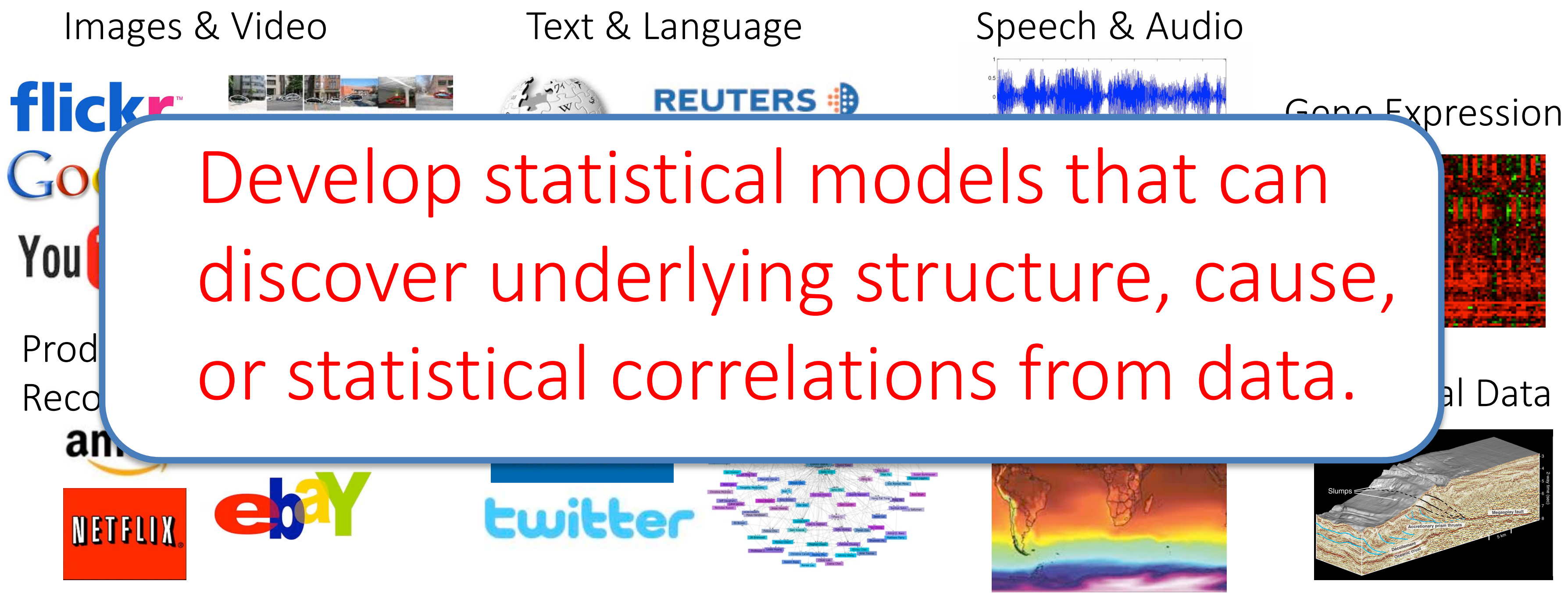
- Machine learning applications and challenges
- Course information and syllabus
- Types of learning methods
- Supervised learning: least squares
- Fundamentals: overfitting, generalization, regularization

Machine Learning's Successes

- Biostatistics / Computational Biology.
- Neuroscience.
- Medical Imaging:
  - computer-aided diagnosis, image-guided therapy.
  - image registration, image fusion.
- Information Retrieval / Natural Language Processing:
  - Text, audio, and image retrieval.
  - Parsing, machine translation, text analysis.
- Speech processing:
  - Speech recognition, voice identification.
- Robotics:
  - Autonomous car driving, planning, control.

Mining for Structure

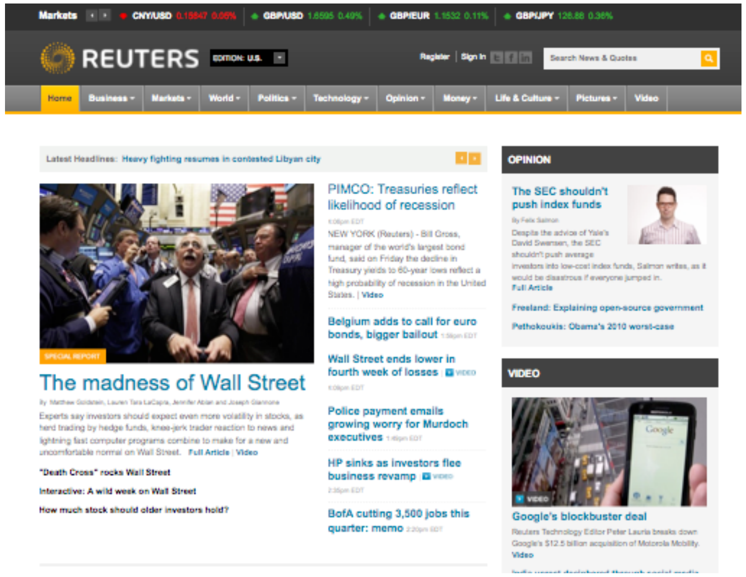
Massive increase in both computational power and the amount of data available from web, video cameras, laboratory measurements.



Develop statistical models that can discover underlying structure, cause, or statistical correlations from data.

Finding Structure in Data

Vector of word counts on a webpage



Latest Headlines: Heavy fighting resumes in contested Libyan city

The madness of Wall Street

By Matthew Goldstein, Lauren Tara LaCapra, Jennifer Ablan and Joseph Giannone

Experts say investors should expect even more volatility in stocks, as herd trading by hedge funds, knee-jerk trader reaction to news and lightning fast computer programs combine to make for a new and uncomfortable normal on Wall Street. Full Article | Video

"Death Cross" rocks Wall Street

Interactive: A wild week on Wall Street

How much stock should older investors hold?

PIMCO: Treasuries reflect likelihood of recession

4:06pm EDT

NEW YORK (Reuters) - Bill Gross, manager of the world's largest bond fund, said on Friday the decline in Treasury yields to 60-year lows reflect a high probability of recession in the United States. | Video

Belgium adds to call for euro bonds, bigger bailout 1:58pm EDT

Wall Street ends lower in fourth week of losses | VIDEO

4:06pm EDT

Police payment emails growing worry for Murdoch executives 1:46pm EDT

HP sinks as investors flee business revamp | VIDEO

2:36pm EDT

BofA cutting 3,500 jobs this quarter: memo 2:20pm EDT

OPINION

The SEC shouldn't push index funds

By Felix Salmon

Despite the advice of Yale's David Swensen, the SEC shouldn't push average investors into low-cost index funds, Salmon writes, as it would be disastrous if everyone jumped in.

Full Article

Freeland: Explaining open-source government

Pethokoukis: Obama's 2010 worst-case

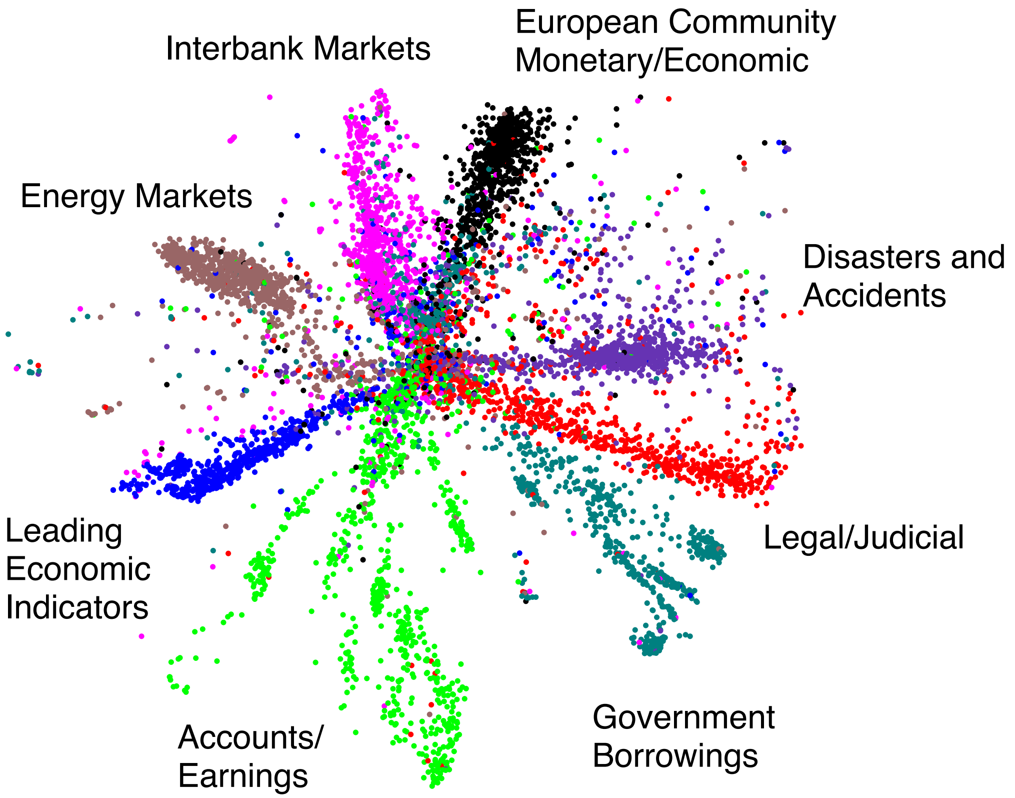
VIDEO

Google's blockbuster deal

Reuters Technology Editor Peter Lauria breaks down Google's $12.5 billion acquisition of Motorola Mobility.

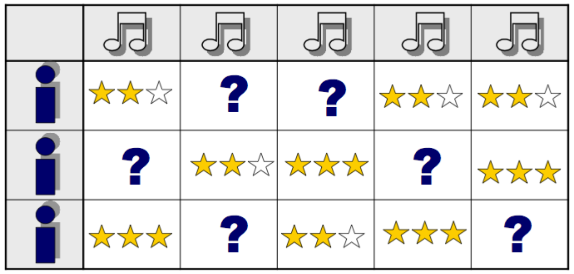
Video

804,414 newswire stories



Recommender systems

Collaborative Filtering/
Matrix Factorization/



	♫	♫	♫	♫	♫
👤	★★☆	?	?	★★☆	★★☆
👤	?	★★☆	★★★	?	★★★
👤	★★★	?	★★☆	★★★	?

Recommender systems

Collaborative Filtering/
Matrix Factorization/

	♫	♫	♫	♫	♫
👤	★★☆	?	?	★★☆	★★☆
👤	?	★★☆	★★★	?	★★★
👤	★★★	?	★★☆	★★★	?

Hierarchical Bayesian Model

Rating value of user i for item j

Latent user feature (preference) vector

Latent item feature vector

$$r_{ij} | \mathbf{u}_i, \mathbf{v}_j, \sigma \sim \mathcal{N}(\mathbf{u}_i^\top \mathbf{v}_j, \sigma^2),$$

$$\left.\begin{array}{lcll} \mathbf{u}_i | \sigma_u & \sim & \mathcal{N}(\mathbf{0}, \sigma_u^2 \mathbf{I}), & i = 1, ..., N. \\ \mathbf{v}_j | \sigma_v & \sim & \mathcal{N}(\mathbf{0}, \sigma_v^2 \mathbf{I}), & j = 1, ..., M. \end{array}\right\}$$

Latent variables that we infer from observed ratings.

Prediction: predict a rating r^*_{ij} for user i and query movie j.

$$P(r^*_{ij} | \mathbf{R}) = \iint P(r^*_{ij} | \mathbf{u}_i, \mathbf{v}_j) \underbrace{P(\mathbf{u}_i, \mathbf{v}_j | \mathbf{R})}_{\textbf{Posterior over Latent Variables}} d\mathbf{u}_i d\mathbf{v}_j$$

Infer latent variables and make predictions using Markov chain Monte Carlo.

Finding Structure in Data

Collaborative Filtering/
Matrix Factorization/
Product Recommendation

movielens
helping you find the *right* movies

amazon

	♫	♫	♫	♫	♫
●	★★☆	?	?	★★☆	★★☆
●	?	★★☆	★★★	?	★★★
●	★★★	?	★★☆	★★★	?

Learned ``genre''

Netflix dataset:
480,189 users
17,770 movies
Over 100 million ratings.

Fahrenheit 9/11
Bowling for Columbine
The People vs. Larry Flynt
Canadian Bacon
La Dolce Vita

Independence Day
The Day After Tomorrow
Con Air
Men in Black II
Men in Black

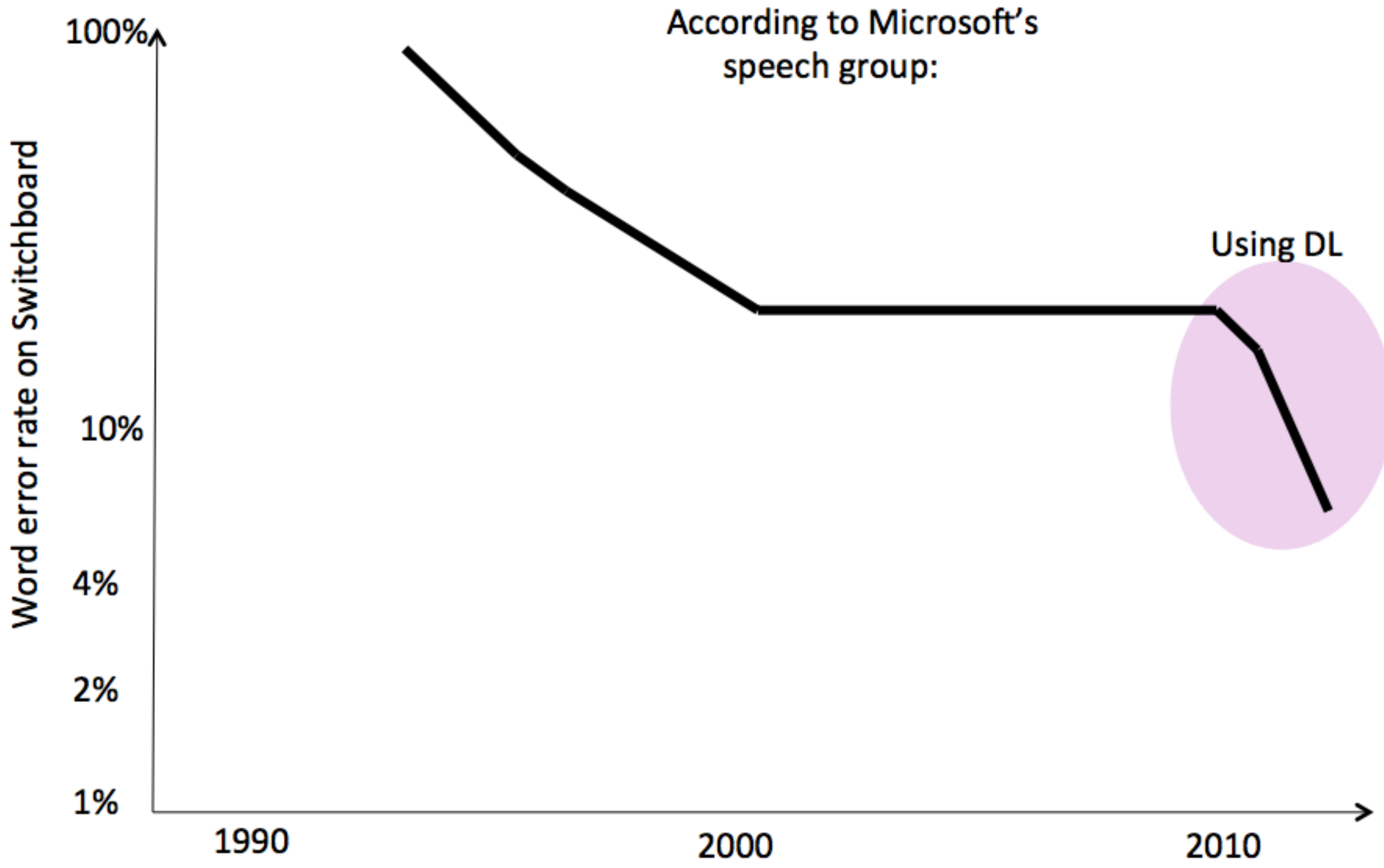
Friday the 13th
The Texas Chainsaw Massacre
Children of the Corn
Child's Play
The Return of Michael Myers

• Part of the wining solution in the Netflix contest (1 million-dollar prize).

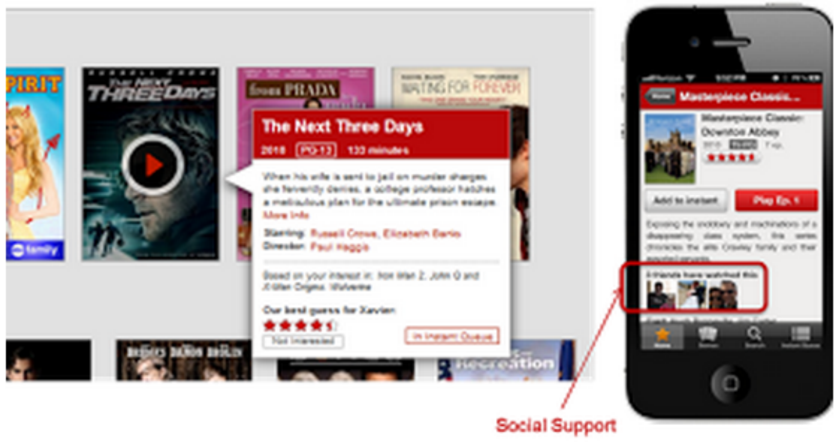
Impact of Neural Networks

- Speech Recognition
- Computer Vision
- Recommender Systems
- Language Understanding
- Drug Discovery and Medical Image Analysis

Speech Recognition



NETFLIX The Netflix Tech Blog



Netflix uses:

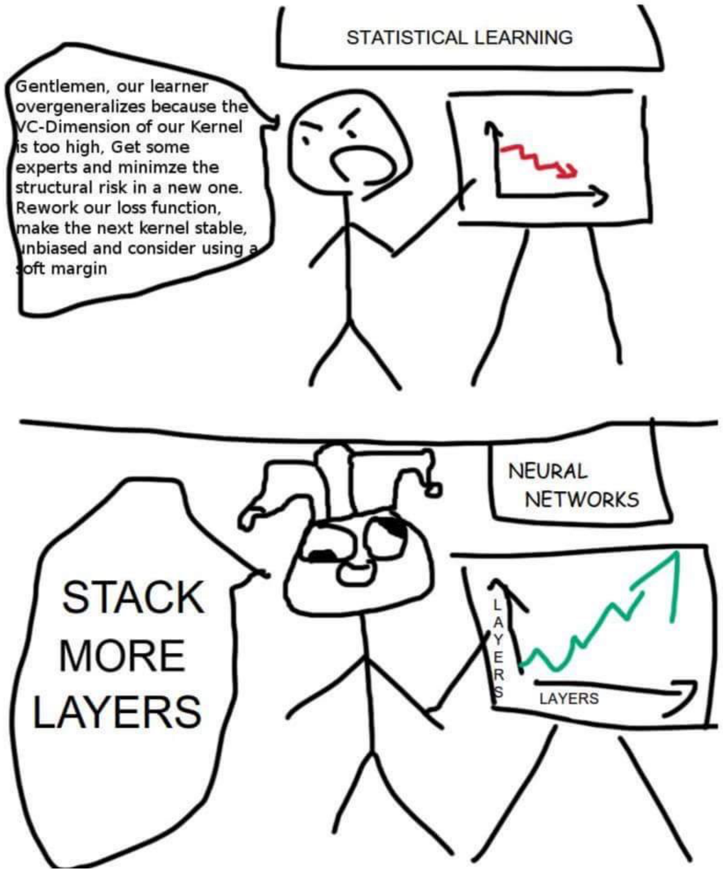
- Restricted Boltzmann machines
- Probabilistic Matrix Factorization

- From their blog:

To put these algorithms to use, we had to work to overcome some limitations, for instance that they were built to handle 100 million ratings, instead of the more than 5 billion that we have, and that they were not built to adapt as members added more ratings. But once we overcame those challenges, we put the two algorithms into production, where they are still used as part of our recommendation engine.

Course Topics

- Probabilistic Models
- Bayesian Methods
- Optimization
- Decision trees
- Unsupervised learning
- Latent variable models
- Neural networks
- Sampling methods
- Reinforcement learning
- Algorithmic fairness


STATISTICAL LEARNING
Gentlemen, our learner overgeneralizes because the VC-Dimension of our Kernel is too high, Get some experts and minimze the structural risk in a new one. Rework our loss function, make the next kernel stable, unbiased and consider using a soft margin
NEURAL NETWORKS
STACK MORE LAYERS
LAYERS
LAYERS

Course Information

- **Lectures:** This course has two identical sections each week:
  - Monday 2pm-5pm online
    - 2:10pm-3pm + 3:10pm-4pm + 4:10pm-5pm
  - Tuesday 7pm-10pm online
    - 7:10pm-8pm + 8:10pm-9pm + 9:10pm-10pm
- Announcements and zoom links will be sent through **quercus** on Sunday.
- **Course website:** erdogdu.github.io/sta414/
  - Contains all course information, slides, additional reading, assignments, announcements, OHs etc. Need to check regularly!

- **Piazza**: should be available through quercus.
- Your grade does not depend on your participation on Piazza. It's just a good way for asking questions, discussing with your instructor, TAs and your peers. We will only allow questions that are related to the course materials/assignments/exams.

- **Email policy:** Please use the Piazza site for most questions. For administrative issues that only concern you, email the course staff.

  Instructor email: sta414-2021-prof@cs.toronto.edu

  TA email: sta414-2021-tas@cs.toronto.edu

Course Evaluation

- 4 assignments: 50% (will not be equally weighted)
  - No collaboration.

- 2-hour midterm: 20% (date TBD, most likely on the week of Mar 1)

- 3-hour final: 30% (date determined by FAS)

- Information about these will be posted on the website.

Assignments and Computation

- For the assignments:
  - -Each student is responsible for his/her own work
  - -You must write your own code, and your own solutions
  - -You can discuss with instructor and TAs
- Computation :
  - -Assignments will involve programming
  - -You should use Python
  - -It is freely available online

Pre-requisites

- Make sure that you have the necessary pre-requisites.
- Linear algebra, probability, calculus
- STA314 is a pre-requisite!
  - But there will be some overlap.
  - Check the list of topics on course webpage!

Course Textbook

- No required textbooks.
- Christopher M. Bishop, Pattern Recognition and Machine Learning (2006)
- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, *Introduction to Statistical Learning (2017)*
- Trevor Hastie, Robert Tibshirani, Jerome Friedman,

  The Elements of Statistical Learning (2009)
- Kevin P. Murphy,

  Machine Learning: A Probabilistic Perspective
- Ian Goodfellow, Yoshua Bengio, Aaron Courville,

  Deep Learning (2016)

What is Machine Learning?

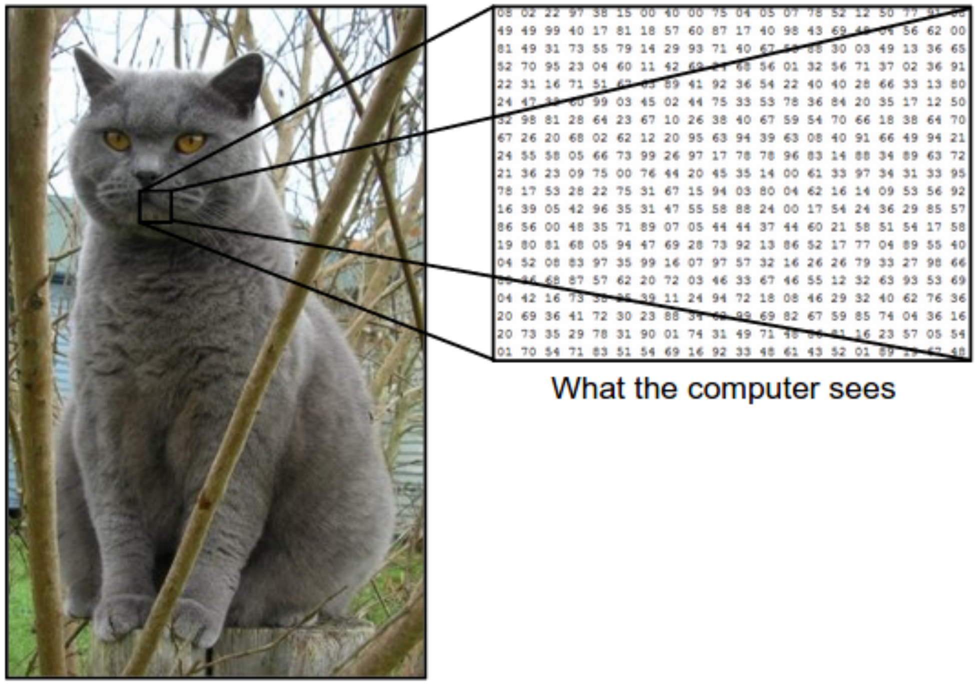
- It’s similar to statistics...
  - Both fields try to uncover patterns in data
  - Both fields draw heavily on calculus, probability, and linear algebra, and share many of the same core algorithms

- But it’s not statistics!
  - Stats is more concerned with helping scientists and policymakers draw good conclusions; ML is more concerned with building autonomous agents
  - Stats puts more emphasis on interpretability and mathematical rigor; ML puts more emphasis on predictive performance, scalability, and autonomy

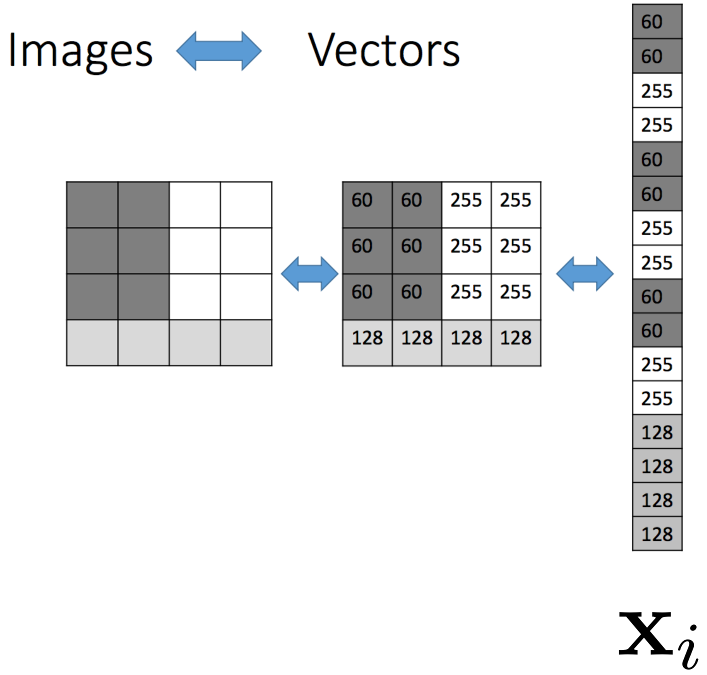
Input Vectors

Consider observing a series of input vectors (feature, covariate):

$$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \ldots.$$



What the computer sees



60	60	255	255
60	60	255	255
60	60	255	255
128	128	128	128

$$\mathbf{x}_i = \begin{bmatrix} 60 \\ 60 \\ 255 \\ 255 \\ 60 \\ 60 \\ 255 \\ 255 \\ 60 \\ 60 \\ 255 \\ 255 \\ 128 \\ 128 \\ 128 \\ 128 \end{bmatrix}$$

Types of Learning

Consider observing a series of input vectors (feature, covariate):

$$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \ldots.$$

- **Supervised Learning:** We are also given target outputs (labels, responses, output, classes): t_1, t_2,…, and the goal is to predict correct output given a new input.

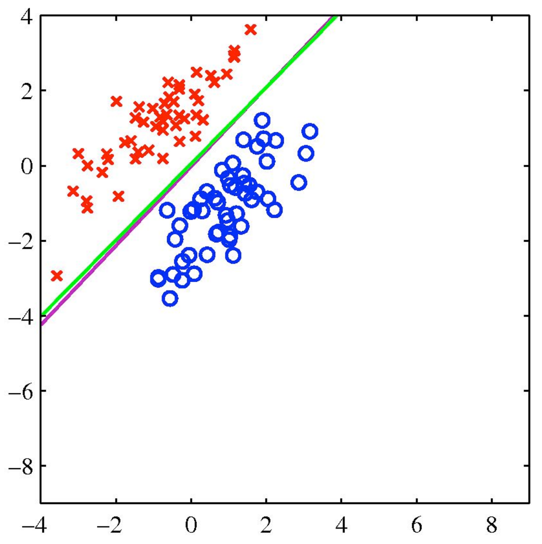
- **Unsupervised Learning:** The goal is to find relations in x’s, which can be used for making predictions, decisions.
  -There is no outcome variable, just a set of measurements.
  - Objective can vary: 1- find relationships between data points 2- find low dimensional representation of your data, etc

- **Semi-supervised Learning:** We are given only a limited amount of labels, but lots of unlabeled data.

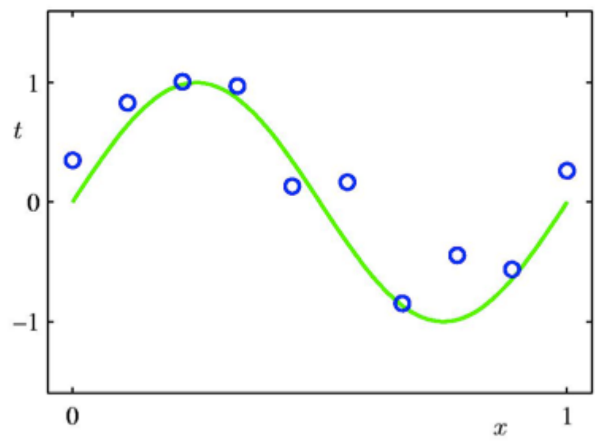
- **Reinforcement Learning:** Learning system receives a reward signal, tries to learn to maximize the reward.

Supervised Learning

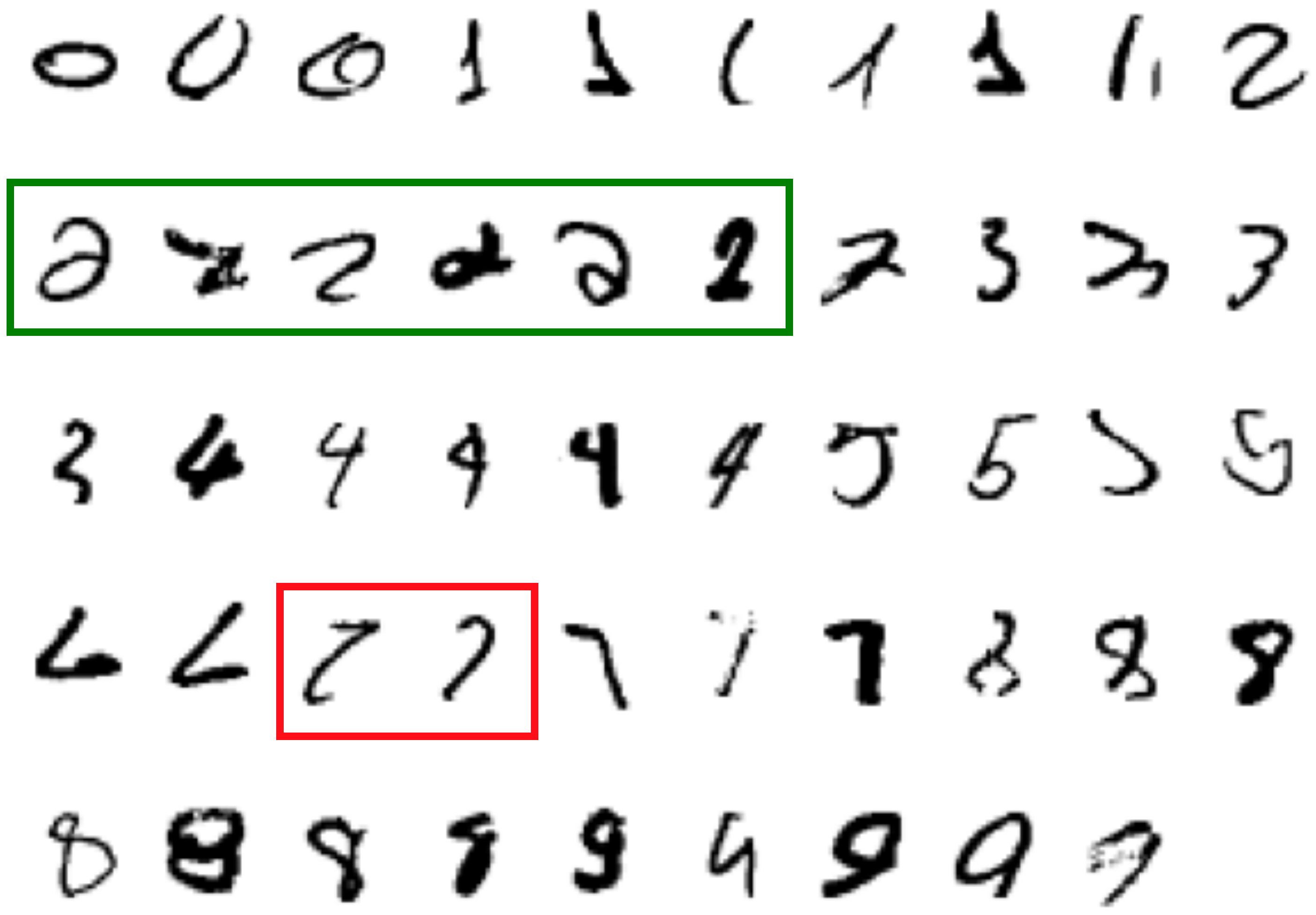
Classification: target outputs t_i are discrete class labels. The goal is to correctly classify new inputs.



Regression: target outputs t_i are continuous. The goal is to predict the output given new inputs.



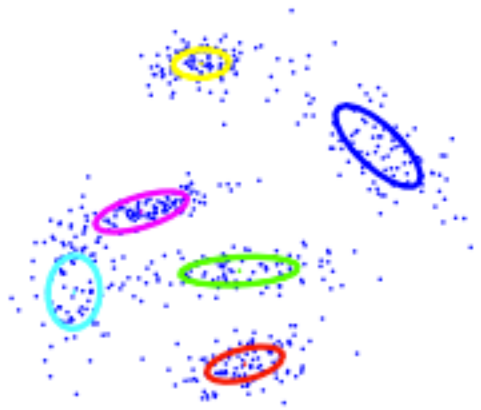
Handwritten Digit Classification



Unsupervised Learning

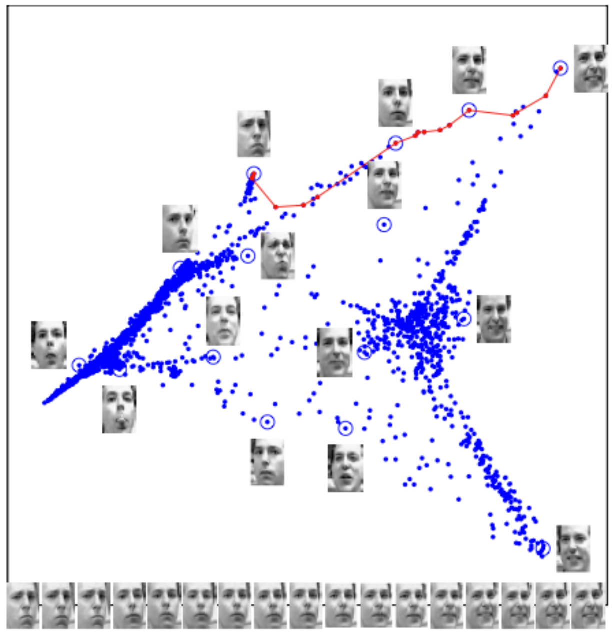
The goal is to construct statistical model that finds useful representation of data:

- Clustering
- Dimensionality reduction
- Modeling the data density
- Finding hidden causes (useful explanation) of the data

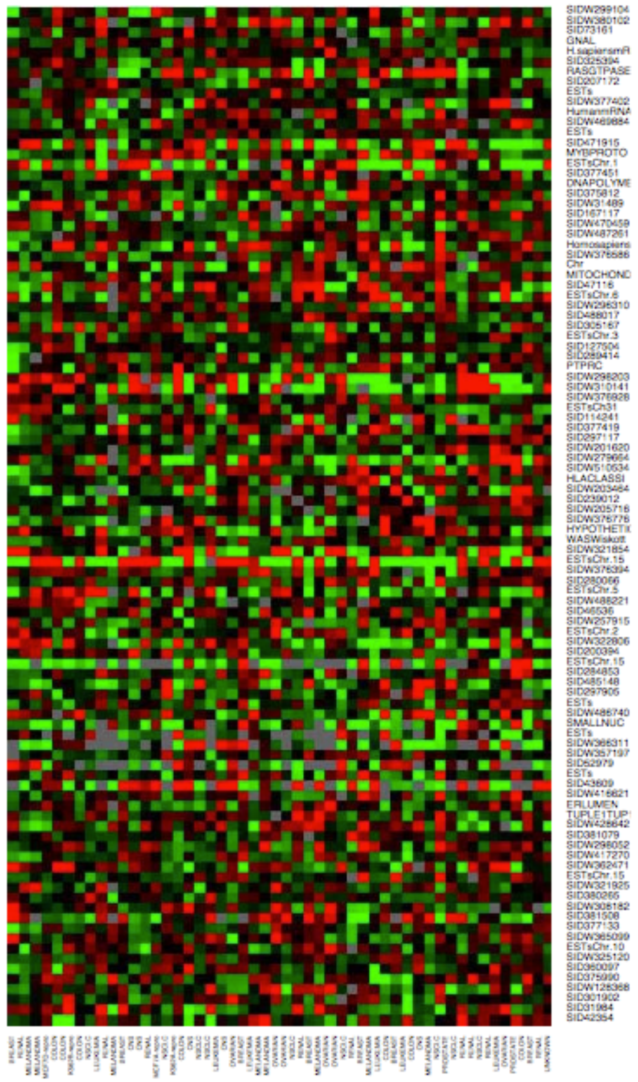


Unsupervised Learning can be used for:

- Structure discovery
- Anomaly detection / Outlier detection
- Data compression, Data visualization
- Used to aid classification/regression tasks (i.e. pre-processing)



DNA Microarray Data



Expression matrix of 6830 genes (rows) and 64 samples (columns) for the human tumor data.

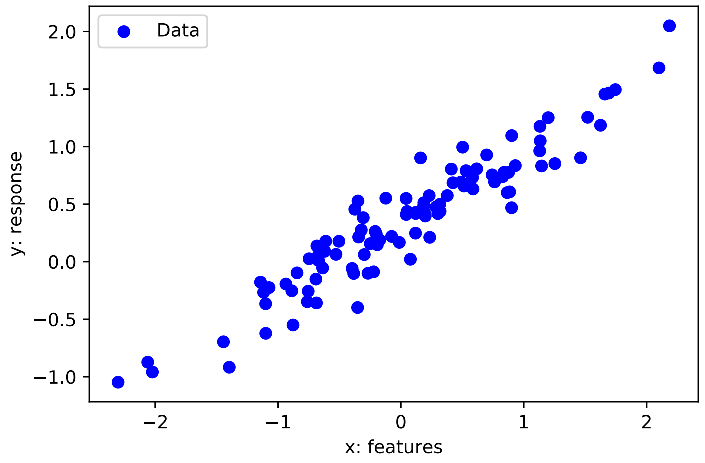
The display is a heat map ranging from bright green (under expressed) to bright red (over expressed).

Questions we may ask:

- Which samples are similar to other samples in terms of their expression levels across genes.
- Which genes are similar to each other in terms of their expression levels across samples.

Linear Least Squares

- Assume that you have a dataset $\mathcal{D} = \{(t_i, \mathbf{x}_i) \text{ for } i = 1, 2, \ldots, N\}$
- The pair $(t_i, \mathbf{x}_i)$ is called a training sample.
- t_i is target (response), $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T$ are features



- Looks linear! (d=1 for this Example)

- Find the "best" line that explains the relationship between the target and features.

Linear Least Squares

- Given a vector of d-dimensional inputs $\mathbf{x} = (x_1, x_2, ..., x_d)^T$, we want to predict the target t_i (response) using the linear model:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_d x_d = w_0 + \sum_{j=1}^{d} w_j x_j.$$

- The term w_0 is the intercept, or often called bias term. It will be convenient to include the constant variable 1 in x and write:

$$\mathbf{x} = (1, x_1, \ldots, x_d) \qquad y(\mathbf{x}, \mathbf{w}) = \mathbf{x}^T \mathbf{w}.$$

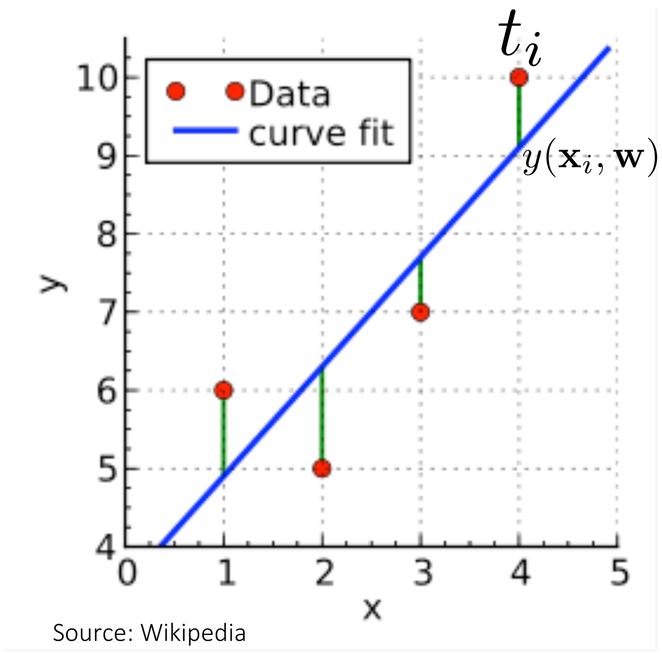
This subscript denotes coordinate index!

- We want to make $y(\mathbf{x}_i, \mathbf{w})$ close to t_i by minimizing a loss.

This subscript denotes sample index!

Linear Least Squares

One option is to minimize the sum of the squares of the errors between the predictions $y(\mathbf{x}_n, \mathbf{w})$ for each data point x_n and the corresponding real-valued targets t_n.



Source: Wikipedia

Loss (error) function: sum-of-squared error function:

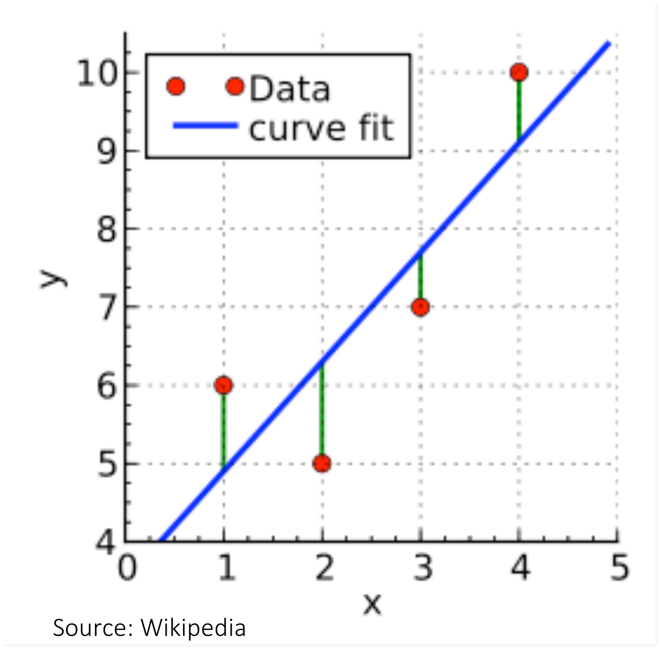
$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n^T\mathbf{w} - t_n)^2$$

Solve: $\min_w E(\mathbf{w})$

Linear Least Squares

In matrix notation, we can write $\mathbf{t} = (t_1, t_2, ..., t_N)^T$.

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1d} \\ 1 & x_{21} & x_{22} & \dots & x_{2d} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{N1} & x_{N2} & \dots & x_{Nd} \end{bmatrix}$$



Source: Wikipedia

Intercept is included in design matrix to write things in compact form.

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n^T\mathbf{w} - t_n)^2 \\ &= \frac{1}{2}(\mathbf{X}\mathbf{w} - \mathbf{t})^{\mathbf{T}}(\mathbf{X}\mathbf{w} - \mathbf{t}). \end{aligned}$$

Solve: $\min_w E(\mathbf{w})$

Linear Least Squares

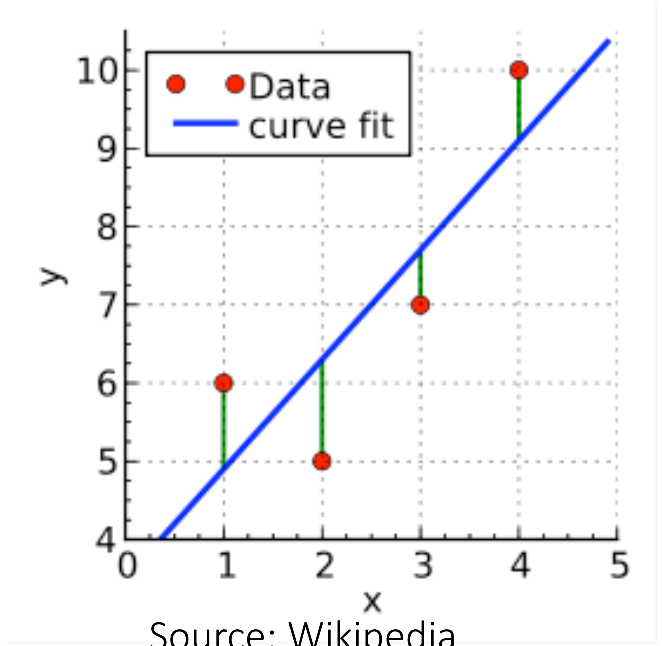
If $\mathbf{X^T X}$ is nonsingular, then the unique solution is given by:

optimal weights

vector of target values

$$\mathbf{w}^* = (\mathbf{X^T X})^{-1}\mathbf{X^T t}$$

the design matrix has one input vector per row

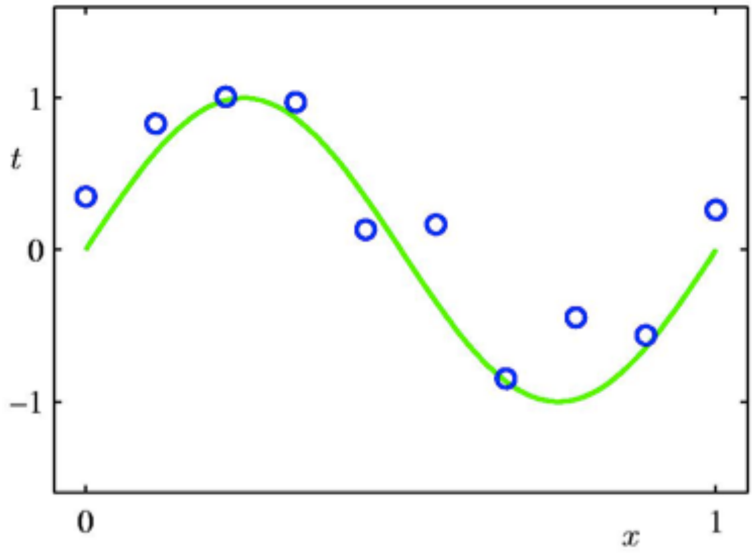


Source: Wikipedia

- At an arbitrary input $\mathbf{x}_0$, the prediction is $y(\mathbf{x}_0, \mathbf{w}) = \mathbf{x}_0^T \mathbf{w}^*$.
- The entire model is characterized by d+1 parameters w*.

Example: Polynomial Curve Fitting

Consider observing a training set consisting of N 1-dimensional observations: $\mathbf{x} = (x_1, x_2, ..., x_N)^T$, together with corresponding real-valued targets: $\mathbf{t} = (t_1, t_2, ..., t_N)^T$.



- The green plot is the true function $\sin(2\pi x)$.
- The training data was generated by taking x_n spaced uniformly between [0 1].
- The target set (blue circles) was obtained by first computing the corresponding values of the sin function, and then adding a small Gaussian noise.

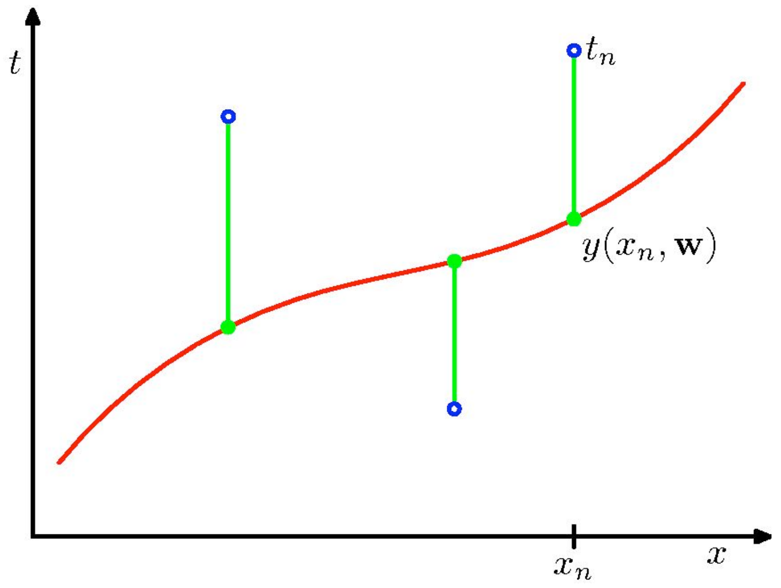
Goal: Fit the data using a polynomial function of the form:

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + ... + w_M x^M = \sum_{j=0}^{M} w_j x^j.$$

Note: the polynomial function is a nonlinear function of x, but it is a linear function of the coefficients w ! Still a linear model!

Example: Polynomial Curve Fitting

• As for the least squares example: we can minimize the sum of the squares of the errors between the predictions $y(x_n, \mathbf{w})$ for each data point x_n and the corresponding target values t_n.



Loss function: sum-of-squared error function:

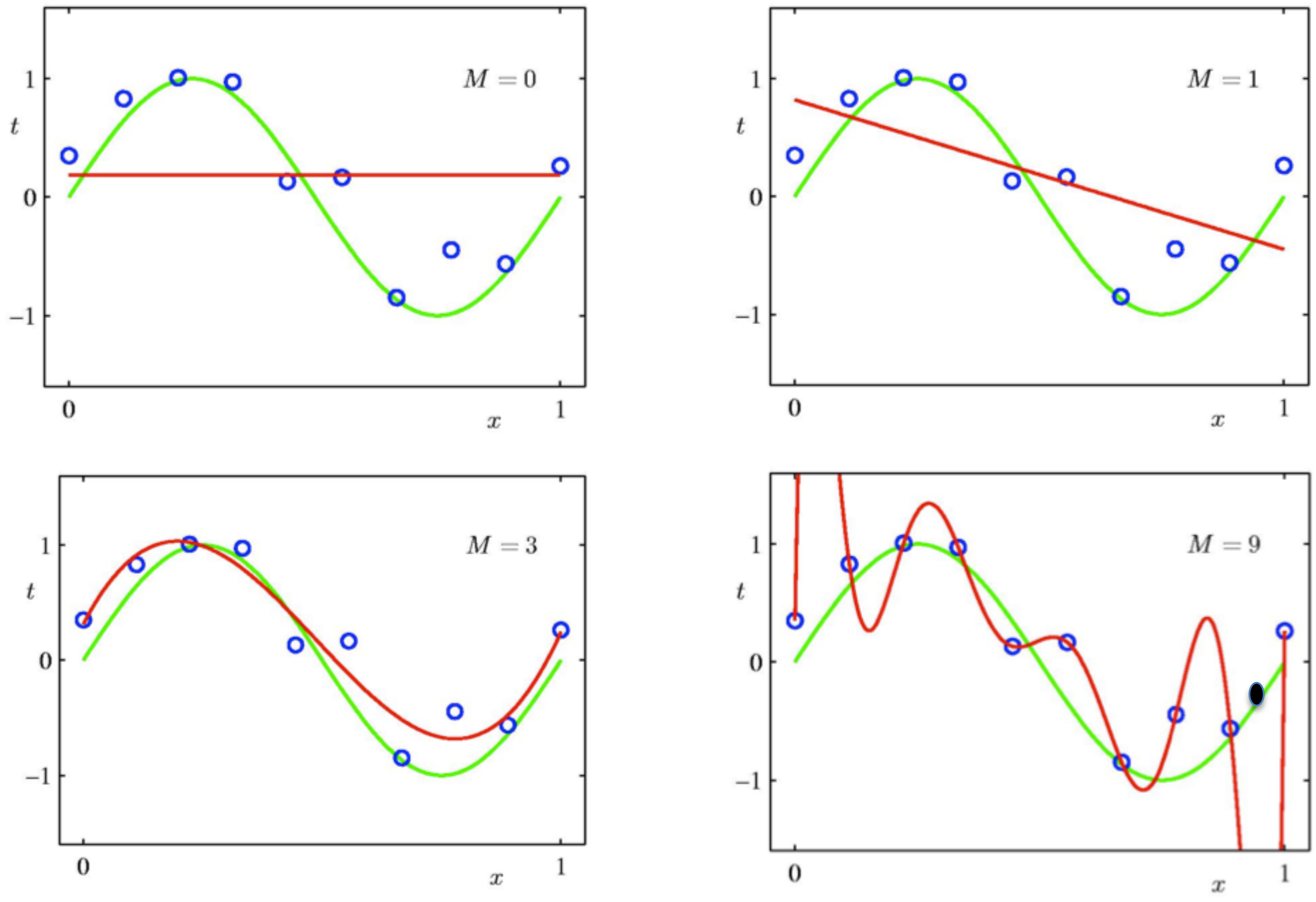
$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(y(x_n, \mathbf{w}) - t_n)^2.$$

• Similar to the linear least squares: Minimizing sum-of-squared error function has a unique solution w^*.

• The model is characterized by M+1 parameters w^*.

• How do we choose M? ! Model Selection.

Some Fits to the Data

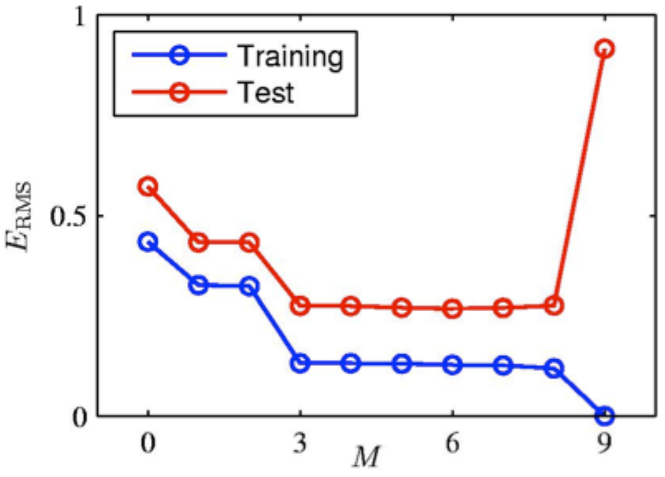


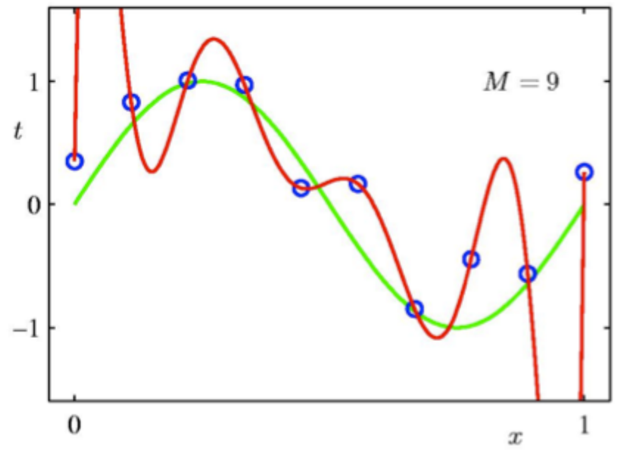
For M=9, we have fitted the training data perfectly!

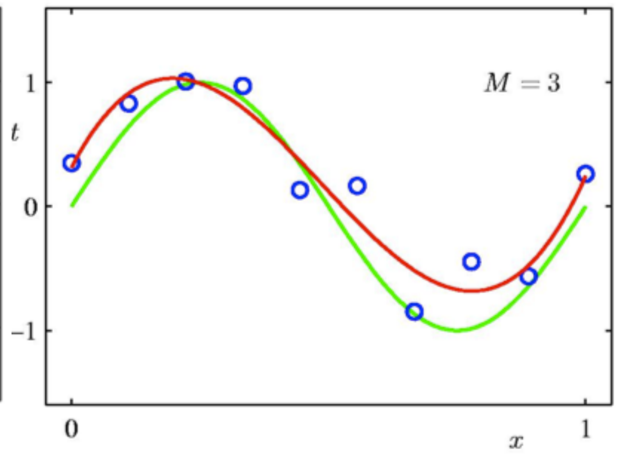
For the black point (which wasn't in the training data) your prediction is here!

Overfitting

- Consider a separate test set containing 100 new data points generated using the same procedure that was used to generate the training data.





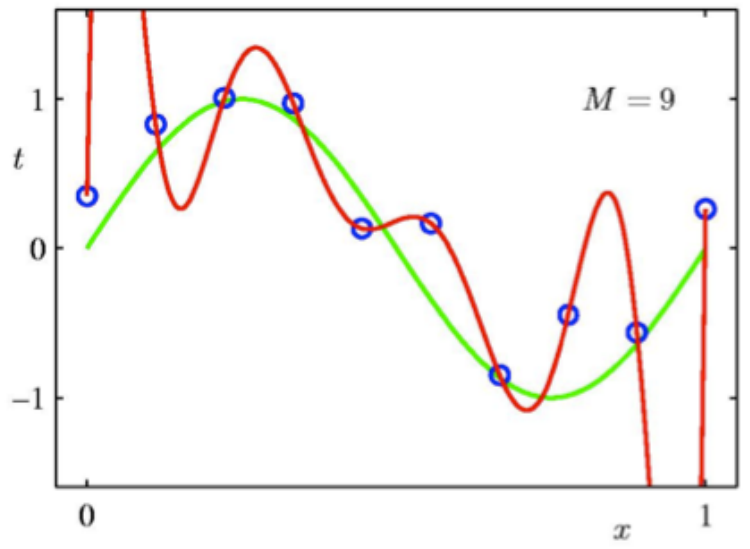


- For M=9, the training error is zero ! The polynomial contains 10 parameters w, and so can be fitted exactly to the 10 data points.

- However, the test error has become very large. Why?

Overfitting

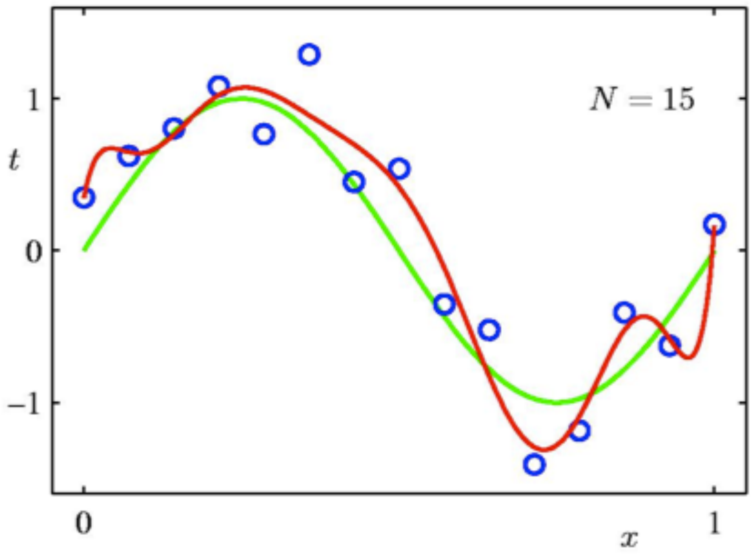
	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

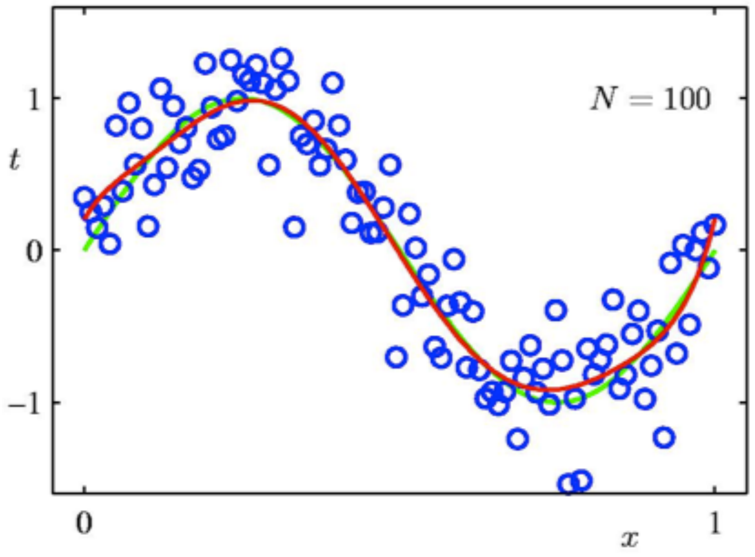


- As M increases, the magnitude of coefficients gets larger.
- For M=9, the coefficients have become finely tuned to the data.
- Between data points, the function exhibits large oscillations.

Varying the Size of the Data

9th order polynomial (M=9)





• For a given model complexity, the overfitting problem becomes less severe as the size of the dataset increases.

• **However, the number of parameters is not necessarily the most appropriate measure of the model complexity!**

Generalization

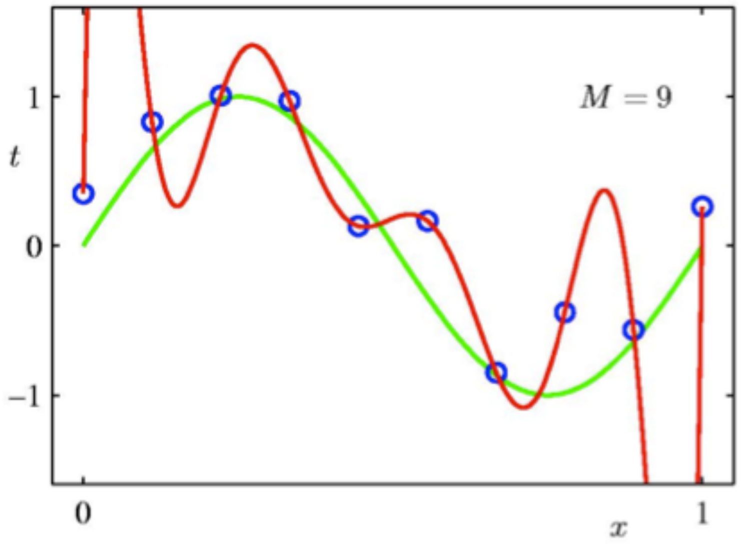
- The goal is to achieve good generalization by making accurate predictions for new test data that is not known during learning.

Generalization = do well on test data

- Choosing the values of parameters that minimize the loss function on the training data may not be the best option.

Generalization

- The goal is to achieve good generalization by making accurate predictions for new test data that is not known during learning.
- Choosing the values of parameters that minimize the loss function on the training data may not be the best option.
- We would like to model the true regularities in the data and ignore the noise in the data:
  - It is hard to know which regularities are real and which are accidental due to the particular training examples we happen to pick.



- Intuition: We expect the model to generalize if it explains the data well given the complexity of the model is low.

- A model can fit the data perfectly. But this is not very informative.

A Simple Way to Penalize Complexity

One technique for controlling over-fitting phenomenon is regularization, which amounts to adding a penalty term to the error function.

penalized error function

target value

regularization parameter

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

where $||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w} = w_1^2 + w_2^2 + \ldots + w_M^2$ called the regularization term. Note that we do not penalize the bias term w_0.

A Simple Way to Penalize Complexity

$$\widetilde{E}(\mathbf{w}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2}_{f(\mathbf{w})} + \underbrace{\frac{\lambda}{2}\|\mathbf{w}\|^2}_{\frac{\lambda}{2}\cdot r(\mathbf{w})}$$

Simple game: $\text{minimize}_w \; f(\mathbf{w}) + \frac{\lambda}{2}\cdot r(\mathbf{w})$

$\mathbf{w}$ is 1 dimensional and takes on values from the set: $\mathbf{w} \in \{0,1,2,3\}$

$$\left\{\begin{array}{c} f(0) = 4 \\ f(1) = 3 \\ f(2) = 2 \\ f(3) = 1 \end{array}\right\} + \frac{\lambda}{2}\cdot \left\{\begin{array}{c} r(0) = 0 \\ r(1) = 1 \\ r(2) = 2 \\ r(3) = 3 \end{array}\right\}$$

	minimizer
$\lambda = 0$	$\mathbf{w} = 3$
$\lambda = 1$	$\mathbf{w} = 3$
$\lambda = 2$	$\mathbf{w} \in \{0,1,2,3\}$
$\lambda = 4$	$\mathbf{w} = 0$

A Simple Way to Penalize Complexity

One technique for controlling over-fitting phenomenon is regularization, which amounts to adding a penalty term to the error function.

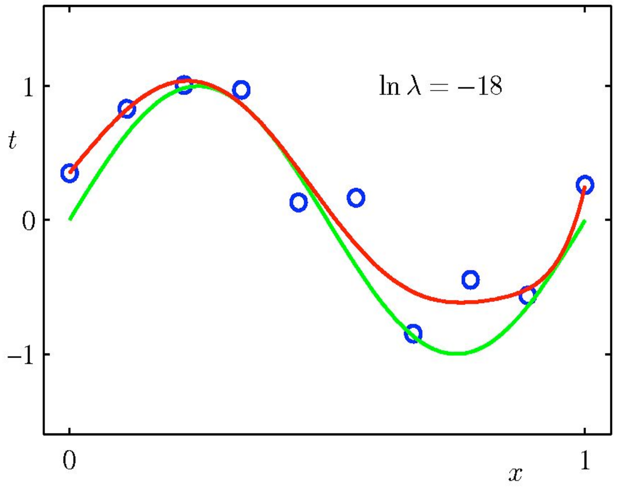
penalized error function

target value

regularization parameter

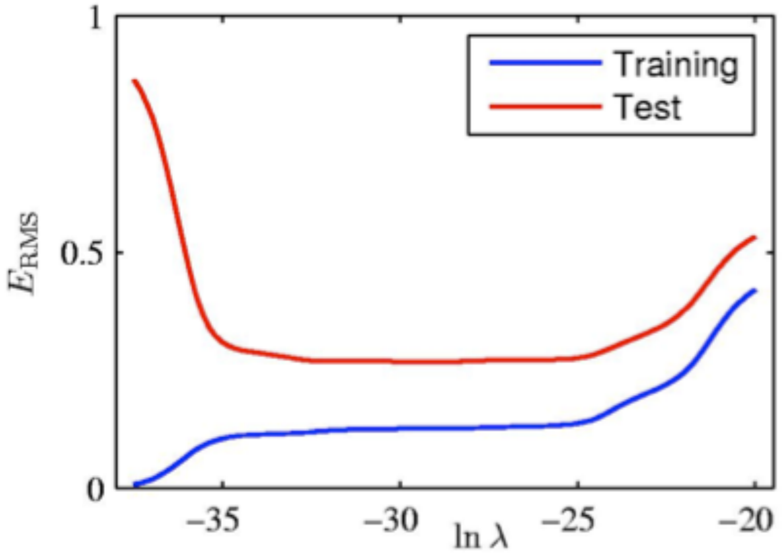
$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

where $||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w} = w_1^2 + w_2^2 + \ldots + w_M^2$ called the regularization term. Note that we do not penalize the bias term w_0.



- The idea is to “shrink” estimated parameters towards zero (or towards the mean of some other weights).
- Shrinking to zero: penalize coefficients based on their size. But doesn’t provide sparsity!
- For a penalty function which is the sum of the squares of the parameters, this is known as a “weight decay”, or “ridge regression”.

Regularization



	$\lambda = 0$	$\lambda = 1.523e-8$	$\lambda = 1$
	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Graph of the root-mean-squared training and test errors vs. lnλ for the M=9 polynomial.

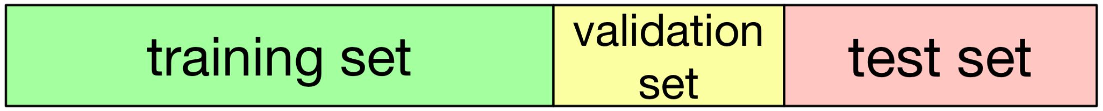
How to choose the regularization parameter λ?

Validation

If the data is plentiful, we can divide the dataset into three subsets:

- Training Data: used to fitting/learning the parameters of the model.
- Validation Data: not used for learning but for selecting the model, choosing the amount of regularization that works best (i.e. M or any other hyperparameter tuning).
- Test Data: used to get performance of the final model.

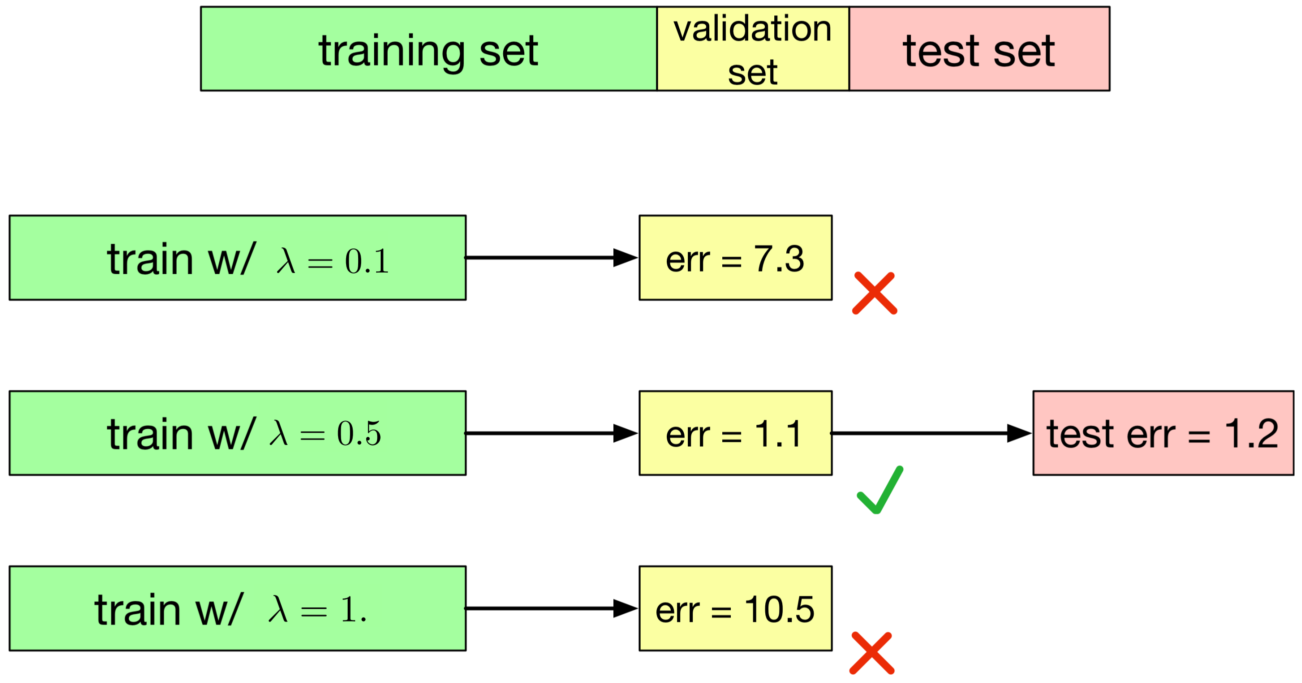
Rule of thumb: split 50% training, 25% validation, 25% test



- For a range of λ (say $\lambda = \{0.1, 0.5, 1, 1.5, 2\}$)
  - For each λ, train model on training data and compute its error on validation data set
- Choose λ that has the smallest error.
- Test the final performance of your model on test data.

Your model should never see test data!!

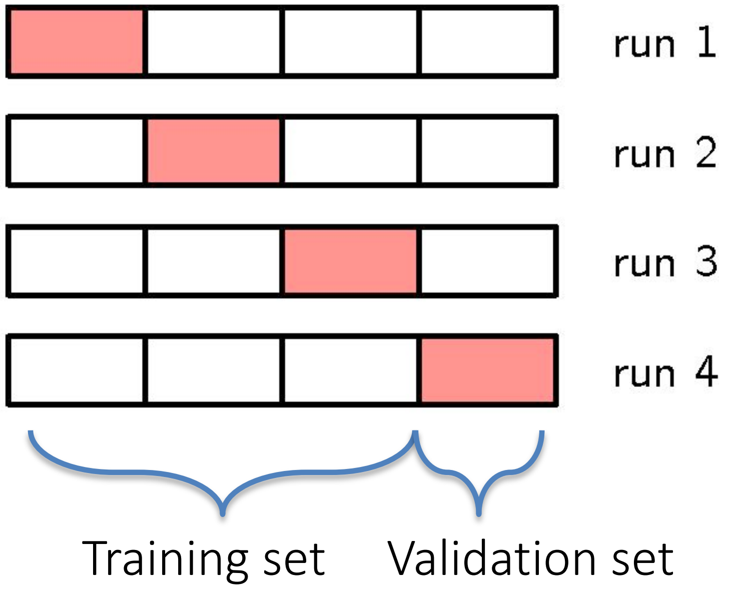
Validation


training set
validation set
test set
train w/ $\lambda = 0.1$
err = 7.3
train w/ $\lambda = 0.5$
err = 1.1
test err = 1.2
train w/ $\lambda = 1.$
err = 10.5

Cross Validation

For many applications, the supply of data for training and testing is limited.
To build good models, we may want to use as much training data as possible.
If the validation set is small, we get noisy estimate of the predictive performance.

S=4 fold cross-validation



- For a range of λ (say $\lambda = \{0.1, 0.5, 1, 1.5, 2\}$)
  - For each λ,
    - The data is partitioned into S groups.
    - Then S-1 of the groups are used for training the model, which is evaluated on the remaining group that serves as validation.
    - Repeat procedure for all S possible choices of the held-out group.
    - Errors from the S runs are averaged, which is the CV error corresponding to that λ.
    - Pick λ with smallest CV error.
    - Test the final performance of your model on test data.

The Rules of Probability

For discrete random variables X and Y

Joint distribution

Sum rule $$p(X) = \sum_{Y} p(X, Y)$$

Product Rule $$p(X, Y) = p(Y|X)p(X)$$

Conditional distribution

Bayes' Rule

- From the product rule,

$$p(Y|X)P(X) = p(X|Y)P(Y)$$

Bayes' rule: $$\boxed{p(Y|X) = \frac{p(X|Y)P(Y)}{P(X)}}$$

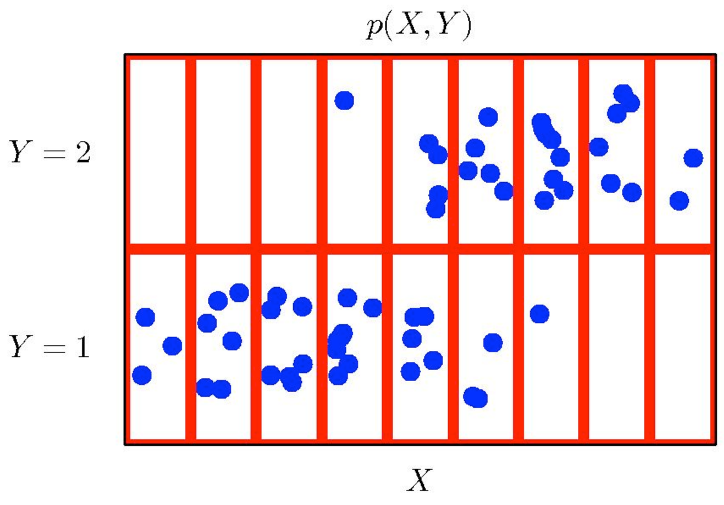
- Remember the sum rule:

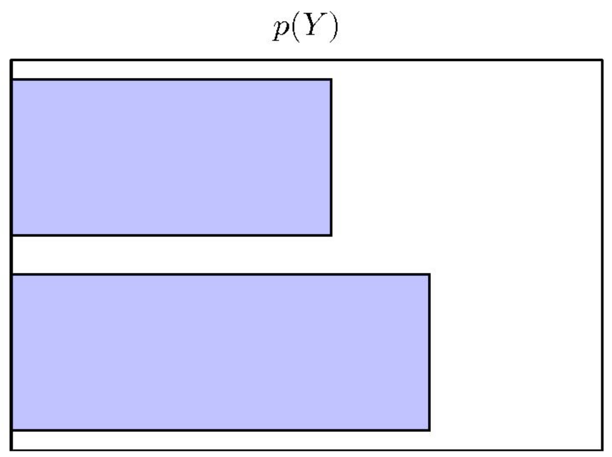
$$p(X) = \sum_Y p(X,Y)$$

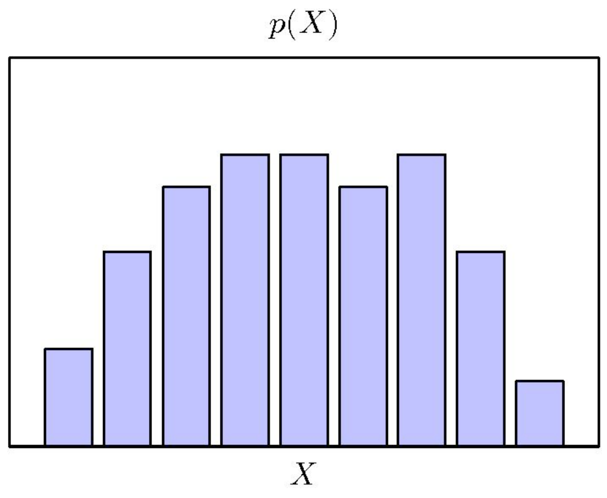
- We will revisit Bayes' Rule later in class.

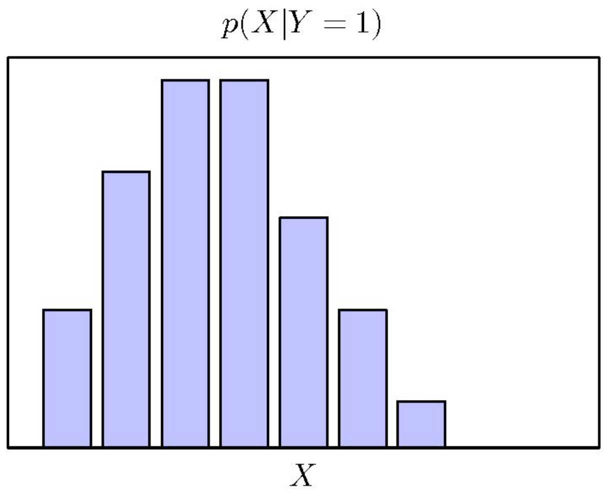
Illustrative Example

- Distribution over two variables: X takes on 9 possible values, and Y takes on 2 possible values.





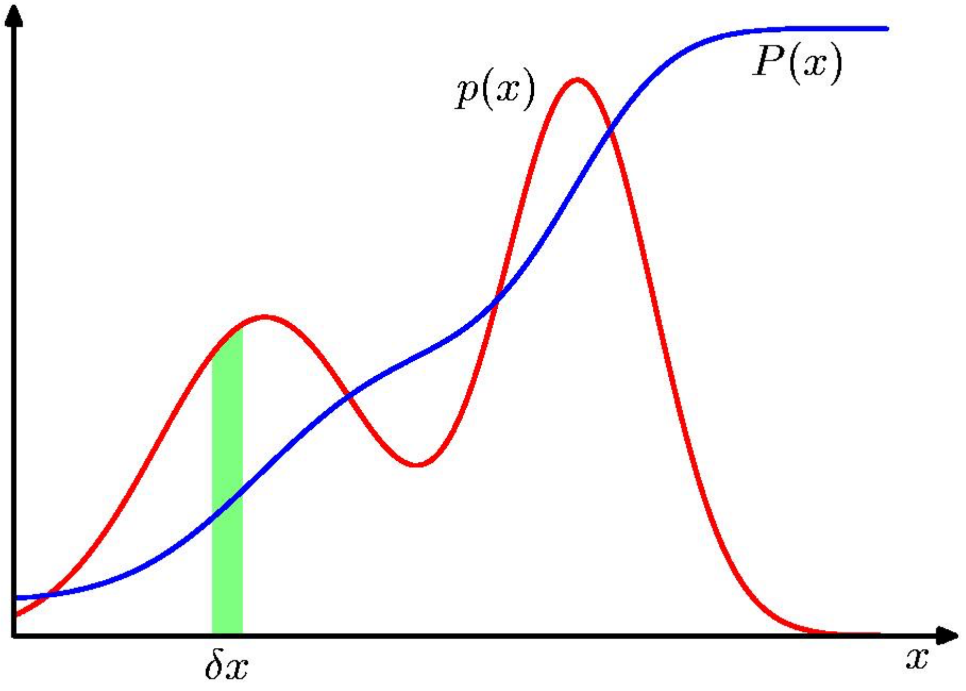




Probability Density

- Cumulative distribution function is defined as:

$$P(z) = \mathbb{P}(X \leq z)$$



The probability density is:

$$P'(z) = p(z)$$

$$P(z) = \int_{-\infty}^{z} p(x)\,\mathrm{d}x$$

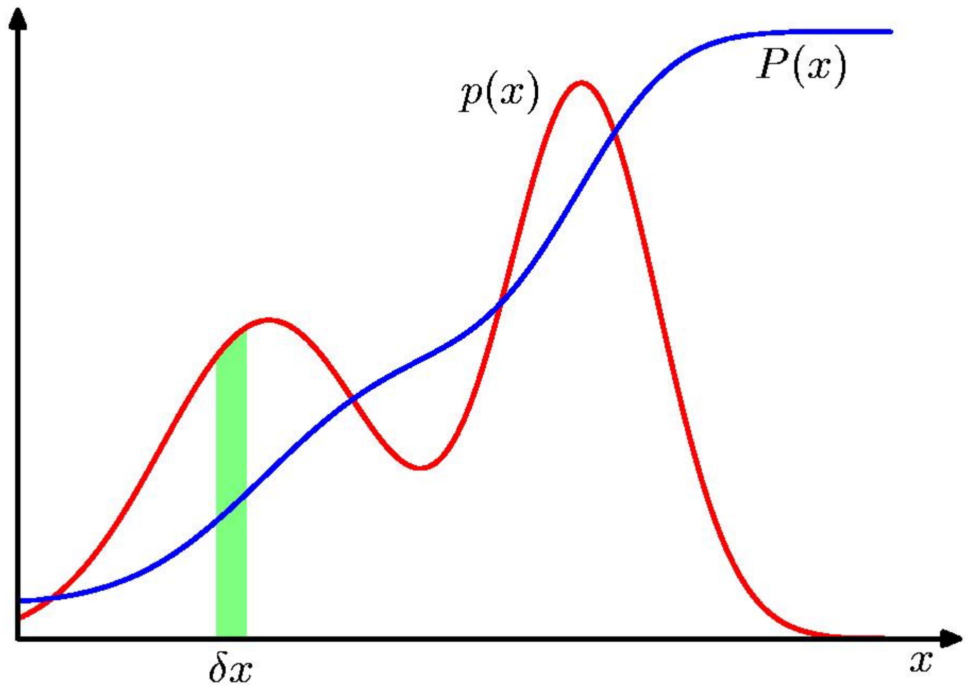
- The sum and product rules take similar forms:

$$\begin{aligned} p(x) &= \int p(x,y)dy \\ p(x,y) &= p(y|x)p(x) \end{aligned}$$

Probability Density

$$\mathbb{P}(x \in (a,b)) = \int_a^b p(x)\,\mathrm{d}x$$

- The probability density must satisfy the following two conditions



$$p(x) \geqslant 0$$

$$\int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1$$

Expectations

- The average value of some function f(x) under a probability distribution (density) p(x) is called the expectation of f(x):

$$\mathbb{E}[f] = \sum_x p(x) f(x) \qquad \mathbb{E}[f] = \int p(x) f(x) \, \mathrm{d}x$$

- If we are given a finite number N of points drawn from the probability distribution (density), then the expectation can be approximated as:

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^{N} f(x_n)$$

- Conditional Expectation with respect to the conditional distribution:

$$\mathbb{E}_x[f|y] = \sum_x p(x|y) f(x)$$

Variances and Covariances

- The variance of f(x) is defined as:

$$\mathrm{var}[f] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

which measures how much variability there is in f(x) around its mean value E[f(x)].

- Note that if f(x) = x, then

$$\mathrm{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2$$

Variances and Covariances

- For two random variables x and y, the covariance is defined as:

$$\begin{aligned}\mathrm{cov}[x,y] &= \mathbb{E}_{x,y}\left[\{x-\mathbb{E}[x]\}\{y-\mathbb{E}[y]\}\right] \\ &= \mathbb{E}_{x,y}[xy]-\mathbb{E}[x]\mathbb{E}[y]\end{aligned}$$

which measures the extent to which x and y vary together. If x and y are independent, then their covariance vanishes.

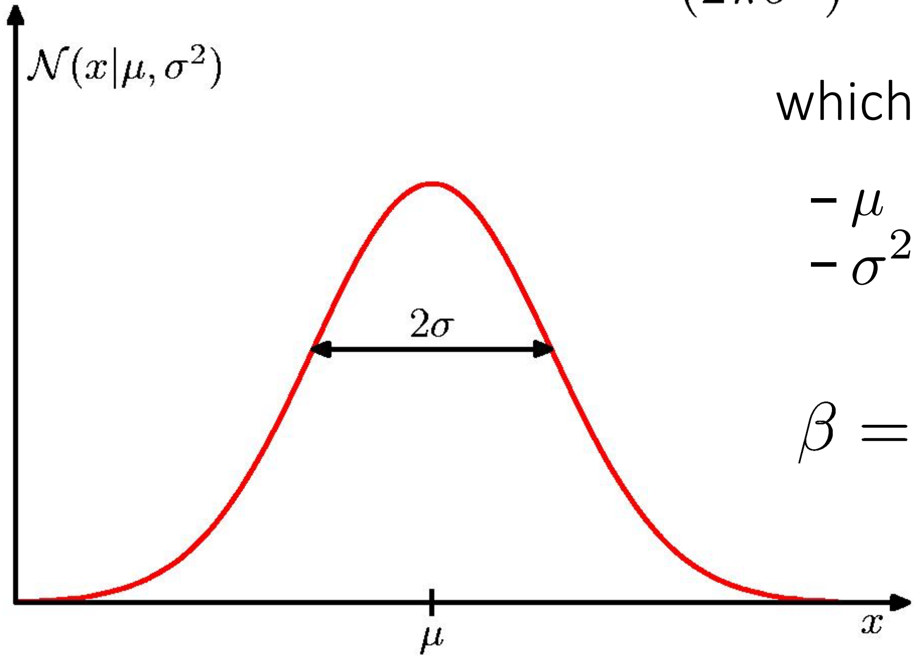
- For two vectors of random variables x and y, the covariance is a matrix:

$$\begin{aligned}\mathrm{cov}[\mathbf{x},\mathbf{y}] &= \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\{\mathbf{x}-\mathbb{E}[\mathbf{x}]\}\{\mathbf{y}^{\mathrm{T}}-\mathbb{E}[\mathbf{y}^{\mathrm{T}}]\}\right] \\ &= \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbf{x}\mathbf{y}^{\mathrm{T}}]-\mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^{\mathrm{T}}]\end{aligned}$$

The Gaussian Distribution

- For the case of single real-valued variable x, the Gaussian distribution is defined as:

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



which is governed by two parameters:

- μ (mean)
- σ^2 (variance)

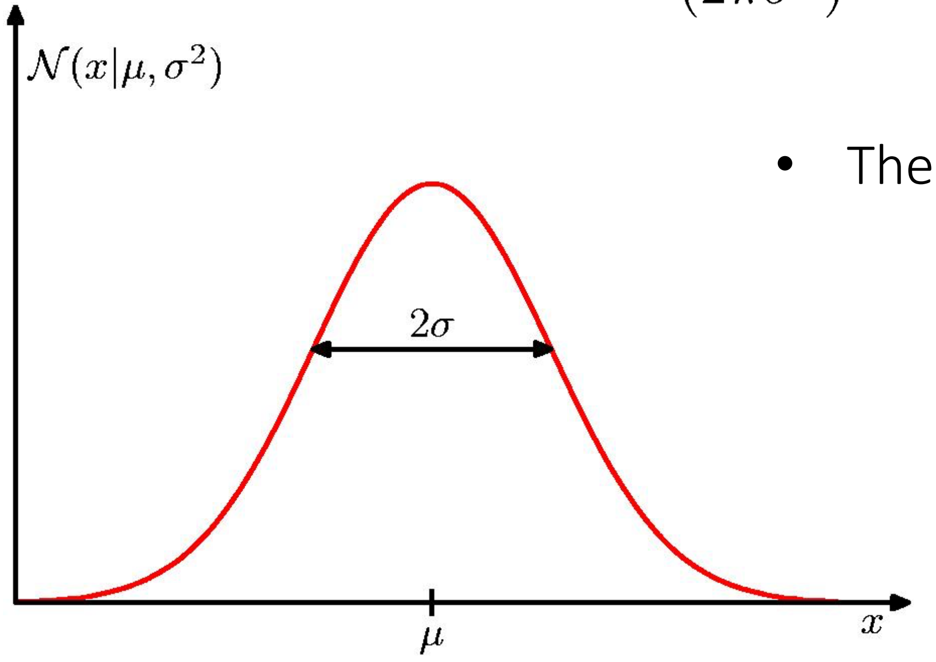
$\beta = 1/\sigma^2$ is called the precision.

- Next class, we will look at various distributions as well as at multivariate extension of the Gaussian distribution.

The Gaussian Distribution

- For the case of single real-valued variable x, the Gaussian distribution is defined as:

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



- The Gaussian distribution satisfies:

$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right)\,\mathrm{d}x = 1$$

which satisfies the two requirements for a valid probability density

Mean and Variance

- Expected value of x takes the following form:

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu, \sigma^2\right) x \, \mathrm{d}x = \mu$$

Because the parameter μ represents the average value of x under the distribution, it is referred to as the mean.

- Similarly, the second order moment takes form:

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu, \sigma^2\right) x^2 \, \mathrm{d}x = \mu^2 + \sigma^2$$

- It then follows that the variance of x is given by:

$$\mathrm{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2$$

Questions?