STA 414/2104:
Machine Learning

Russ Salakhutdinov

Department of Computer Science
Department of Statistics
rsalakhu@cs.toronto.edu
http://www.cs.toronto.edu/~rsalakhu/

Lecture 1

Evaluation

- 3 Assignments worth 40%.
- Midterm worth 20%.
- Undergrads: Final worth 40%
- Graduate: 10% oral presentation, 30% final

Text Books

- **Christopher M. Bishop (2006)**
Pattern Recognition and Machine Learning, **Springer.**

Additional Books
- Kevin Murphy, Machine Learning: A Probabilistic Perspective.

- Trevor Hastie, Robert Tibshirani, Jerome Friedman (2009)
The Elements of Statistical Learning

- David MacKay (2003)
Information Theory, Inference, and Learning Algorithms

- Most of the figures and material will come from these books.

Statistical Machine Learning

Statistical machine learning is a very dynamic field that lies at the intersection of Statistics and computational sciences.

The goal of statistical machine learning is to develop algorithms that can learn from data by constructing stochastic models that can be used for making predictions and decisions.

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.

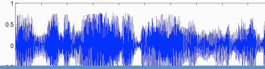Images & Video   Text & Language   Speech & Audio

Gene Expression

Develop statistical models that can discover underlying structure, cause, or statistical correlations from data.
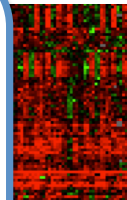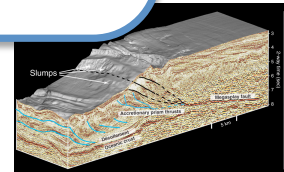
Example: Boltzmann Machine
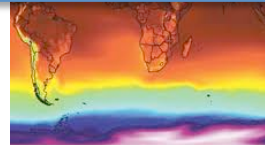
Model parameters

Latent (hidden) variables

$$P(\mathbf{x}, \mathbf{y}) = \frac{1}{\mathcal{Z}} \sum_{\mathbf{h}} \exp \left[\mathbf{x}^\top \mathbf{W}^{(1)} \mathbf{h} + \mathbf{y}^\top \mathbf{W}^{(2)} \mathbf{h} \right]$$

Input data (e.g. pixel intensities of an image, words from webpages, speech signal).

Target variables (response) (e.g. class labels, categories, phonemes).
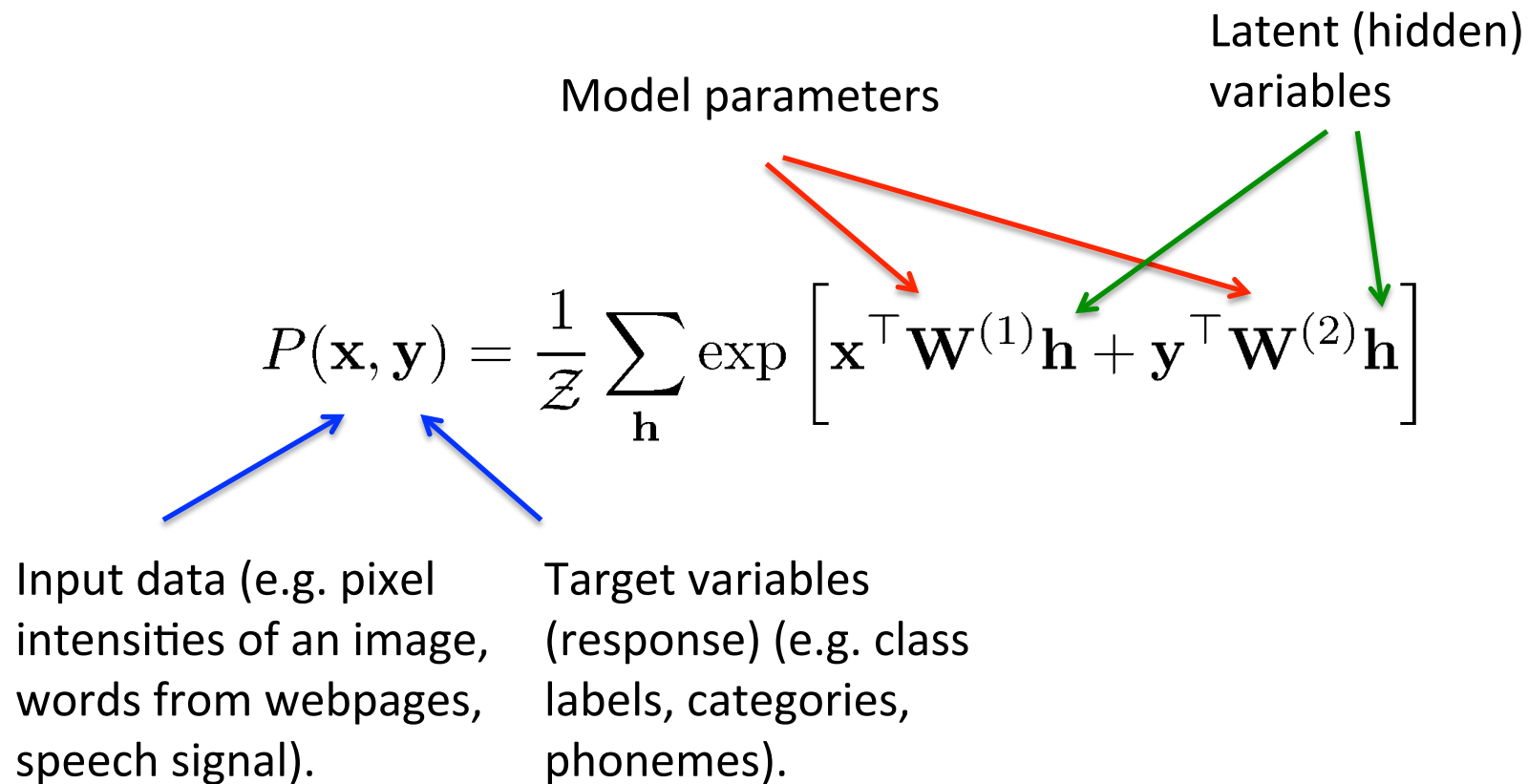
Markov Random Fields, Undirected Graphical Models.

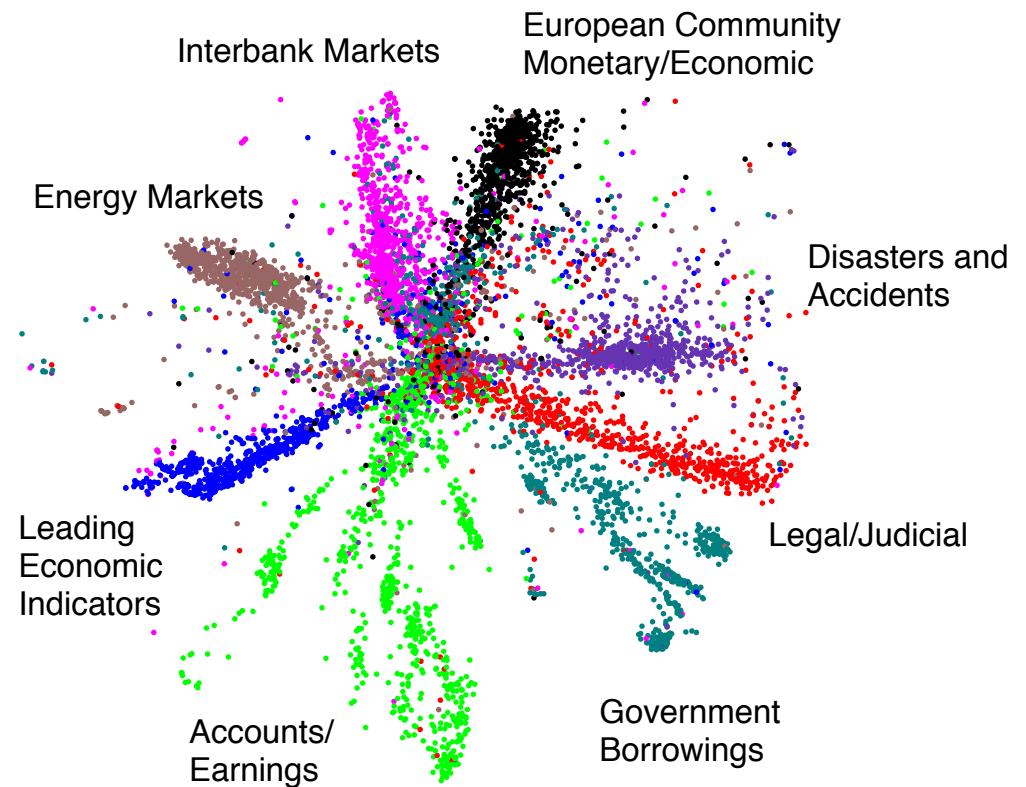# Finding Structure in Data

$$P(\mathbf{x}) = \frac{1}{\mathcal{Z}} \sum_{\mathbf{h}} \exp \left[\mathbf{x}^{\top} \mathbf{W} \mathbf{h} \right]$$

Vector of word counts
on a webpage

Latent variables:
hidden topics

804,414 newswire stories

Interbank Markets

European Community
Monetary/Economic

Energy Markets

Disasters and
Accidents

Leading
Economic
Indicators

Legal/Judicial

Accounts/
Earnings

Government
Borrowings

Matrix Factorization

Collaborative Filtering/
Matrix Factorization/

Hierarchical Bayesian Model

Rating value of
user i for item j

Latent user feature
(preference) vector

Latent item
feature vector

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

$$\mathbf{u}_i|\sigma_u \quad \sim \quad \mathcal{N}(\mathbf{0}, \sigma_u^2\mathbf{I}), \quad i = 1, ..., N.$$

$$\mathbf{v}_j|\sigma_v \quad \sim \quad \mathcal{N}(\mathbf{0}, \sigma_v^2\mathbf{I}), \quad j = 1, ..., M.$$

Latent variables that
we infer from
observed ratings.

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

$$P(r_{ij}^*|\mathbf{R}) = \iint P(r_{ij}^*|\mathbf{u}_i, \mathbf{v}_j)P(\mathbf{u}_i, \mathbf{v}_j|\mathbf{R})d\mathbf{u}_i d\mathbf{v}_j$$

Posterior over Latent Variables

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

Finding Structure in Data

Collaborative Filtering/
Matrix Factorization/
Product Recommendation

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).

Tentative List of Topics

- Linear methods for regression, Bayesian linear regression
- Linear models for classification
- Probabilistic Generative and Discriminative models
- Regularization methods
- Model Comparison and BIC
- Neural Networks
- Radial basis function networks
- Kernel Methods, Gaussian processes, Support Vector Machines
- Mixture models and EM algorithm
- Graphical Models and Bayesian Networks

Types of Learning

Consider observing a series of input vectors:

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

- **Supervised Learning**: We are also given target outputs (labels, responses): $y_1, y_2, \ldots$, and the goal is to predict correct output given a new input.

- **Unsupervised Learning**: The goal is to build a statistical model of x, which can be used for making predictions, decisions.

- **Reinforcement Learning**: the model (agent) produces a set of actions: $a_1, a_2, \ldots$ that affect the state of the world, and received rewards $r_1, r_2 \ldots$ The goal is to learn actions that maximize the reward (we will not cover this topic in this course).

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

Supervised Learning

Classification: target outputs y_i are discrete class labels. The goal is to correctly classify new inputs.
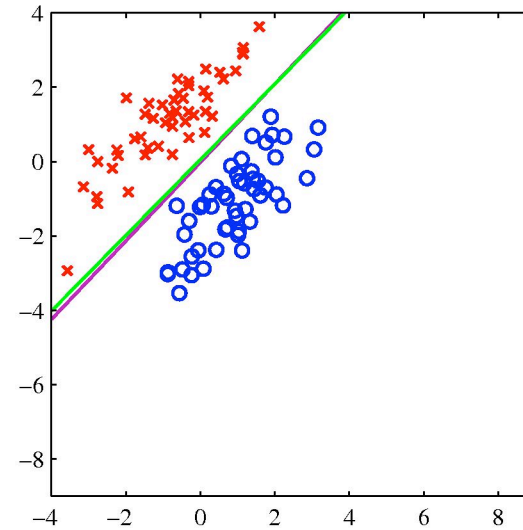
Regression: target outputs y_i are continuous. The goal is to predict the output given new inputs.
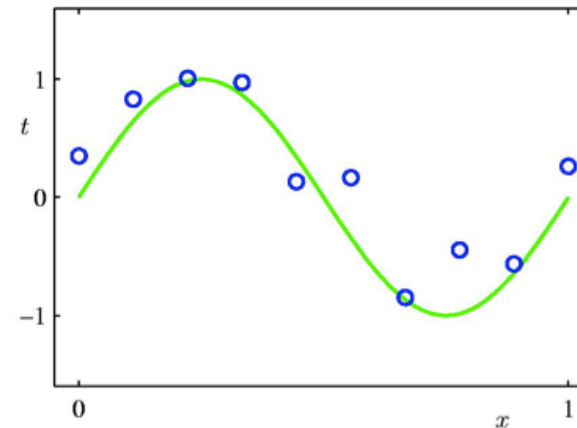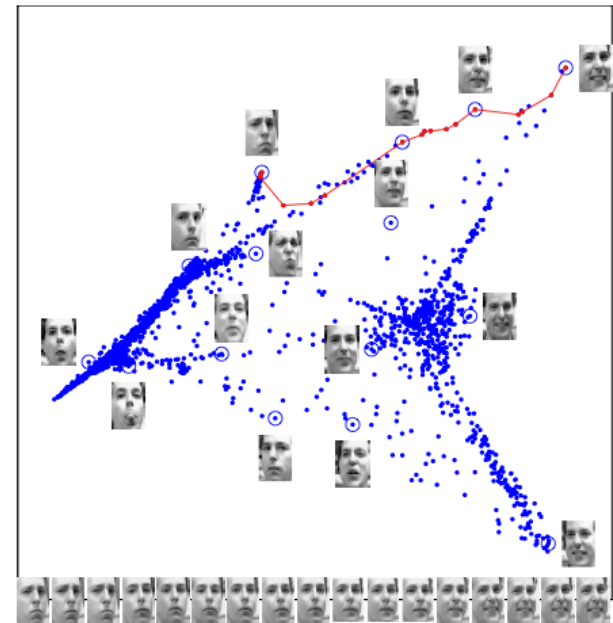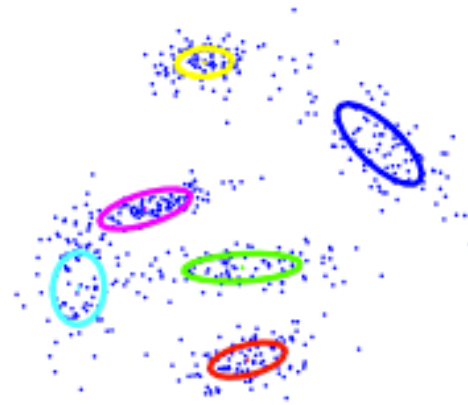
Handwritten Digit Classification

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

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

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

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

- 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:

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{x}^T \mathbf{w}.$$

- Observe a training set consisting of N observations $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)^T$, together with corresponding target values $\mathbf{t} = (t_1, t_2, ..., t_N)^T$.

- Note that **X** is an $N \times (d+1)$ matrix.

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.
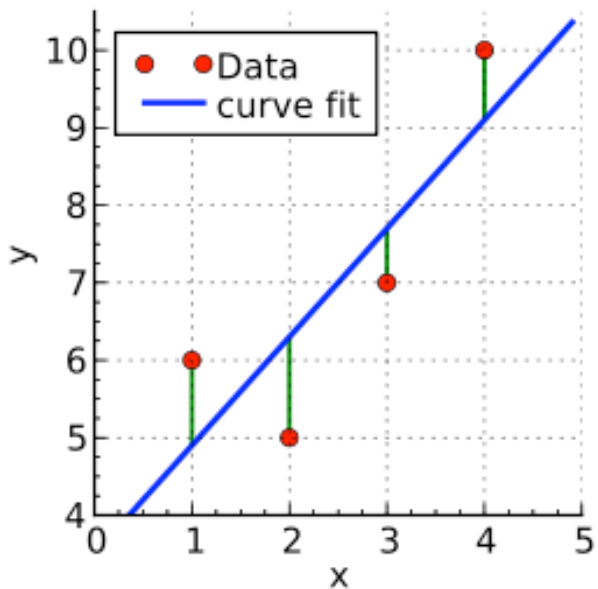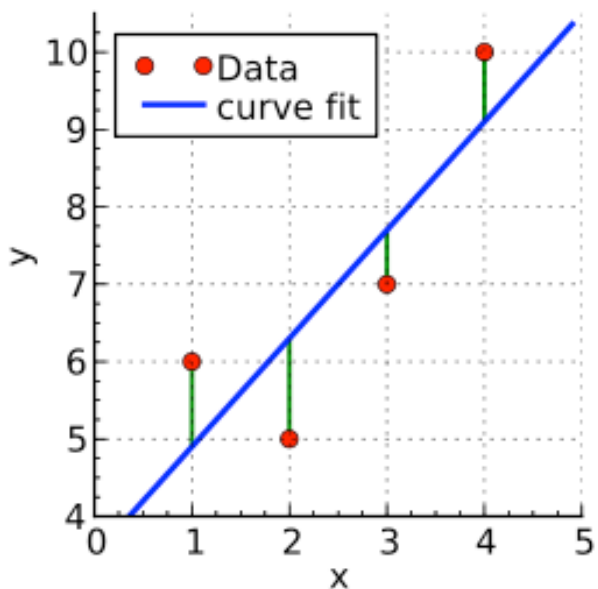

Source: Wikipedia

Loss function: sum-of-squared error function:

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

Linear Least Squares

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

Source: Wikipedia

optimal
weights

vector of
target values

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

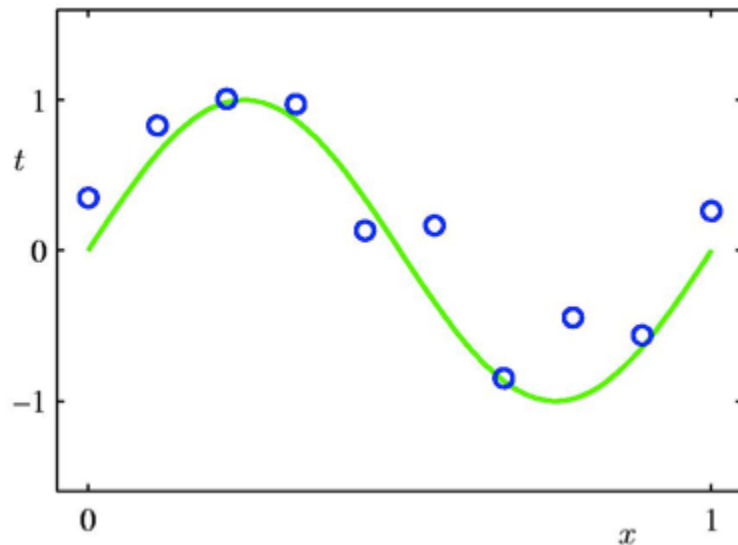the design matrix has one
input vector per row

- 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 $\mathbf{w}^*$.

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 $\mathbf{w} \rightarrow$ **Linear Models**.

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.
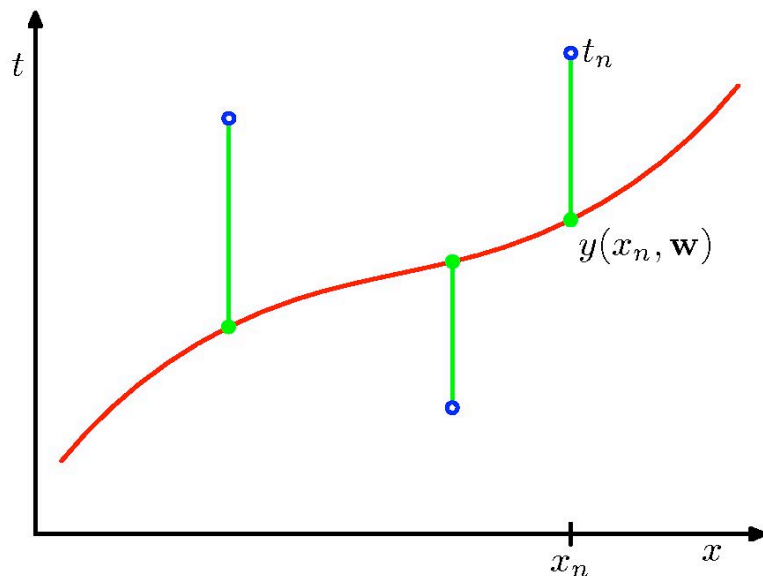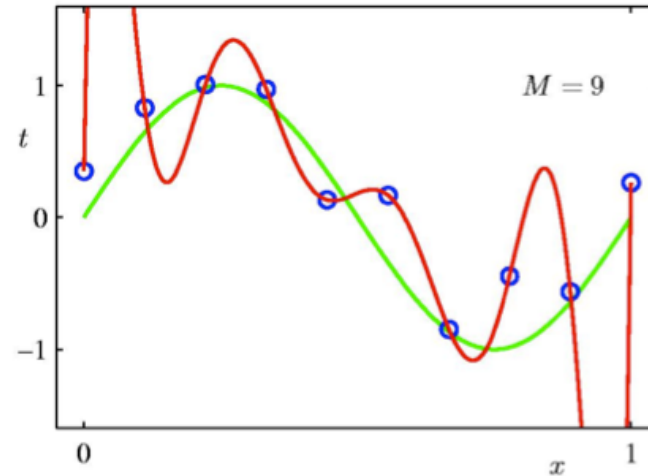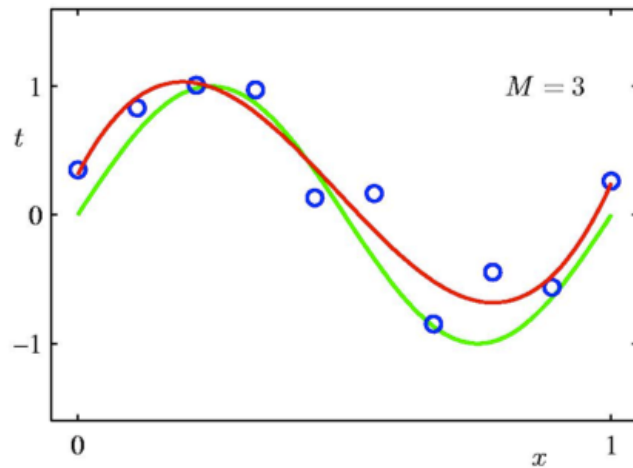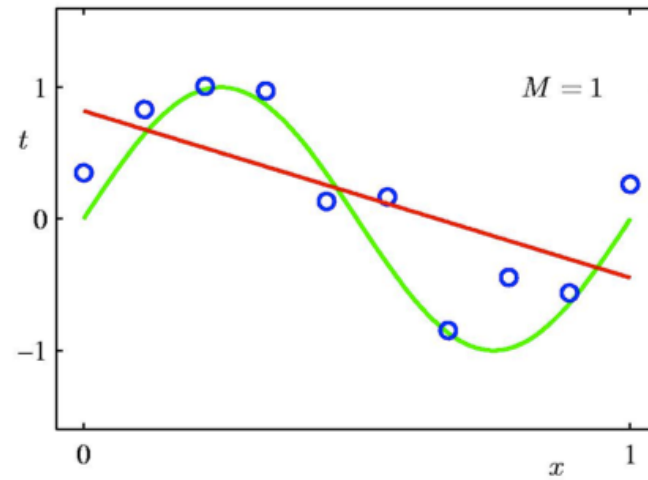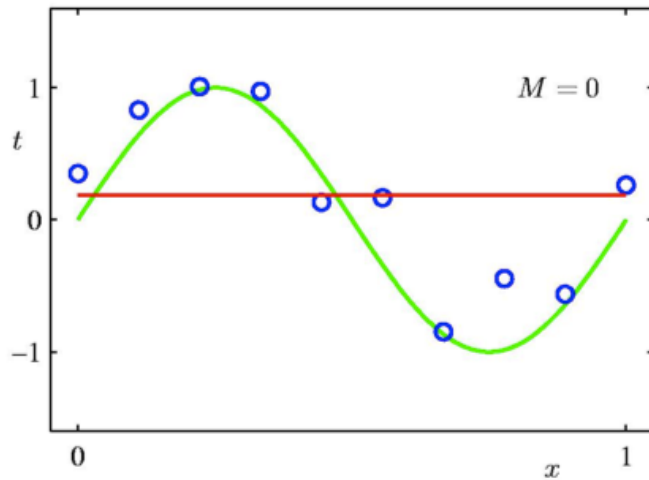

Loss function: sum-of-squared error function:

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

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

• The model is characterized by M+1 parameters $\mathbf{w}^*$.

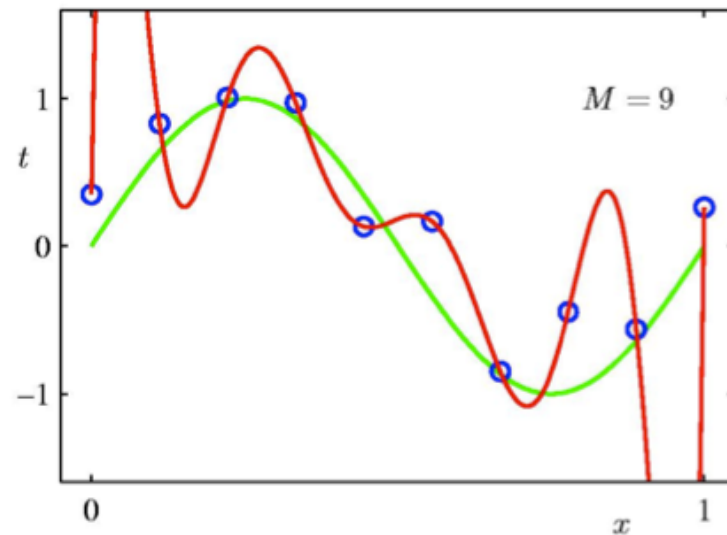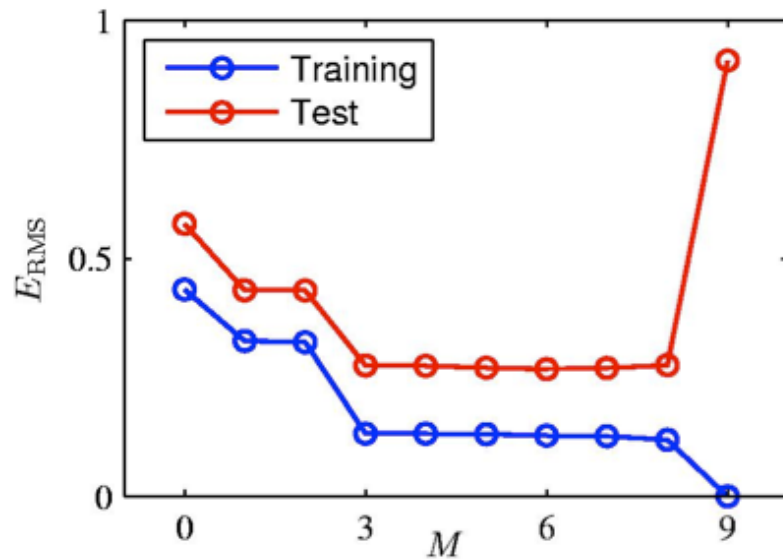• How do we choose M? $\rightarrow$ **Model Selection**.

Some Fits to the Data

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

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 $\rightarrow$ The polynomial contains 10 degrees of freedom corresponding to 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
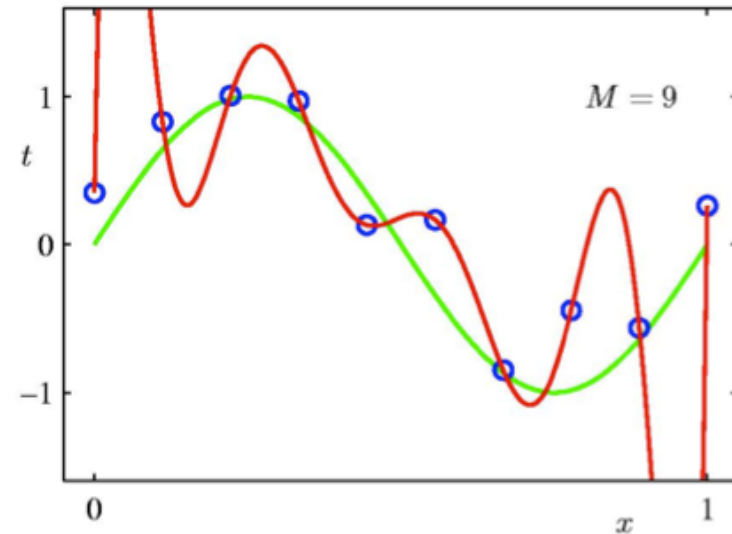
- As M increases, the magnitude of coefficients gets larger.

- For M=9, the coefficients have become finely tuned to the data.

- Between data points, the function exhibits large oscillations.

More flexible polynomials with larger M tune to the random noise on the target values.

# Varying the Size of the Data

9th order polynomial
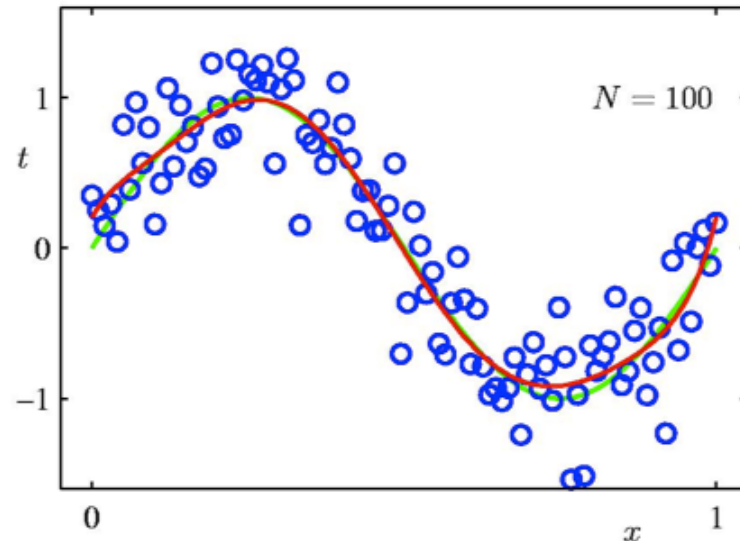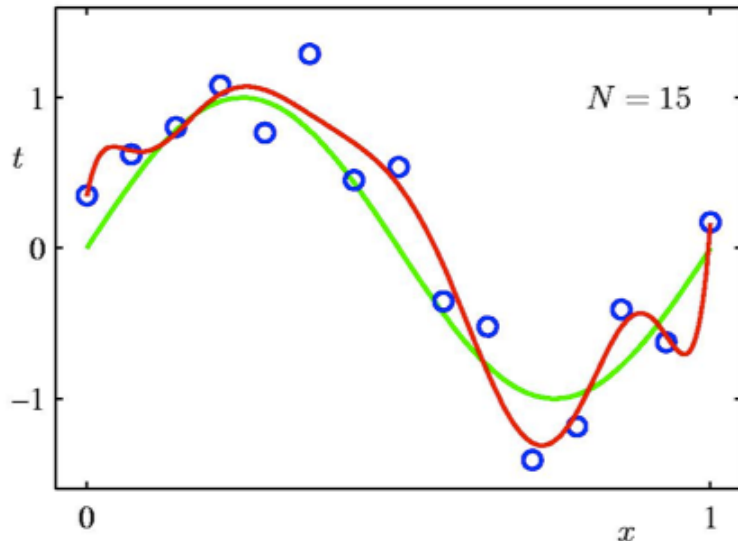
- For a given model complexity, the overfitting problem becomes less severe as the size of the dataset increases.

- However, the number of parameters is not necessarily the most appropriate measure of the model complexity.

Generalization

• The goal is 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.
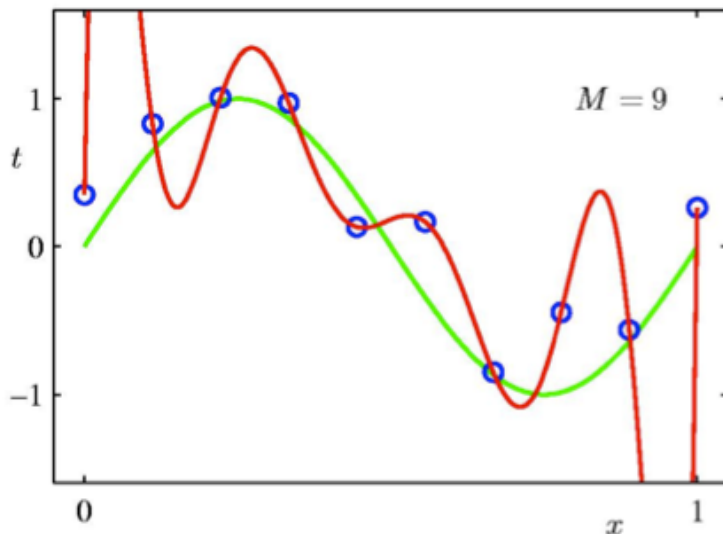
• Intuition: We expect the model to generalize if it explains the data well given the complexity of the model.
• If the model has as many degrees of freedom as the data, it can fit the data perfectly. But this is not very informative.
• Some theory on how to control model complexity to optimize generalization.

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.
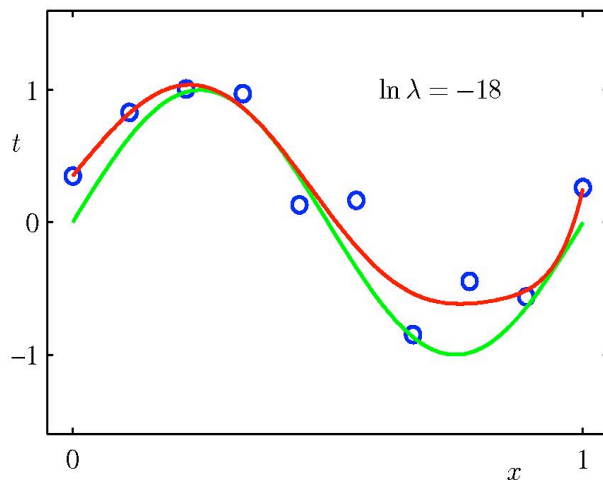
penalized error function

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

$\ln \lambda = -18$

- 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.
- 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
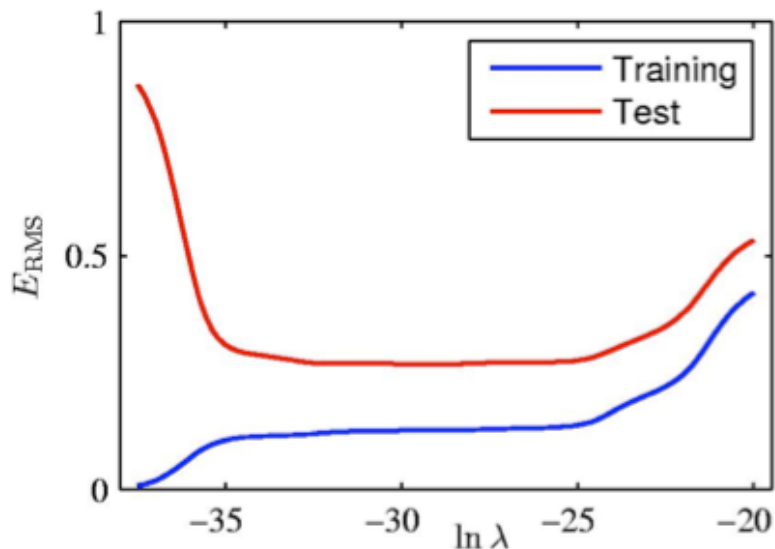
	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	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. $\ln \lambda$ for the M=9 polynomial.

How to choose λ?

Cross 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, or choosing the amount of regularization that works best.
- Test Data: used to get performance of the final model.

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 fold cross-validation

run 1
run 2
run 3
run 4
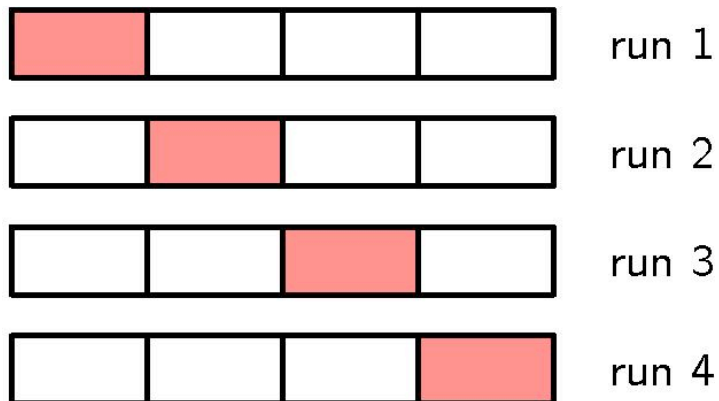
- The data is partitioned into S groups.
- Then S-1 of the groups are used for training the model, which is evaluated on the remaining group.
- Repeat procedure for all S possible choices of the held-out group.
- Performance from the S runs are averaged.

Basics of Probability Theory

- Consider two random variables X and Y:

    - X takes any values x_i, where i=1,..,M.
    - Y takes any values y_j, j=1,...,L.

- Consider a total of N trials and let the number of trials in which X = x_i and Y = y_j is n_{ij}.
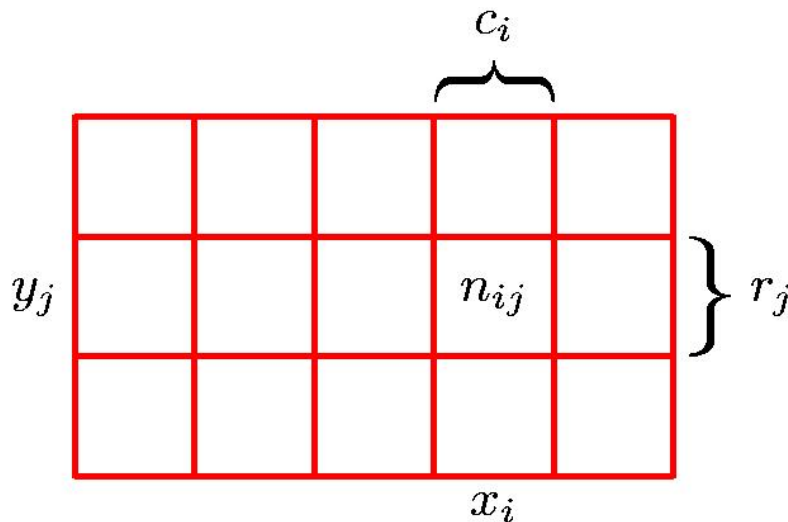
- Joint Probability:

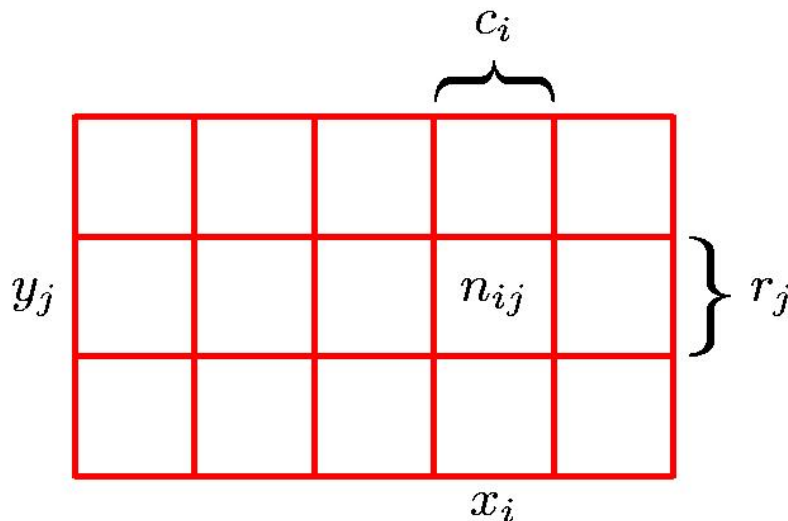$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

- Marginal Probability:

$$p(X = x_i) = \frac{c_i}{N}.$$

where $c_i = \sum_j n_{ij}.$

Basics of Probability Theory

- Consider two random variables X and Y:

    – X takes any values x_i, where i=1,..,M.
    – Y takes any values y_j, j=1,...,L.

- Consider a total of N trials and let the number of trials in which X = x_i and Y = y_j is n_{ij}.
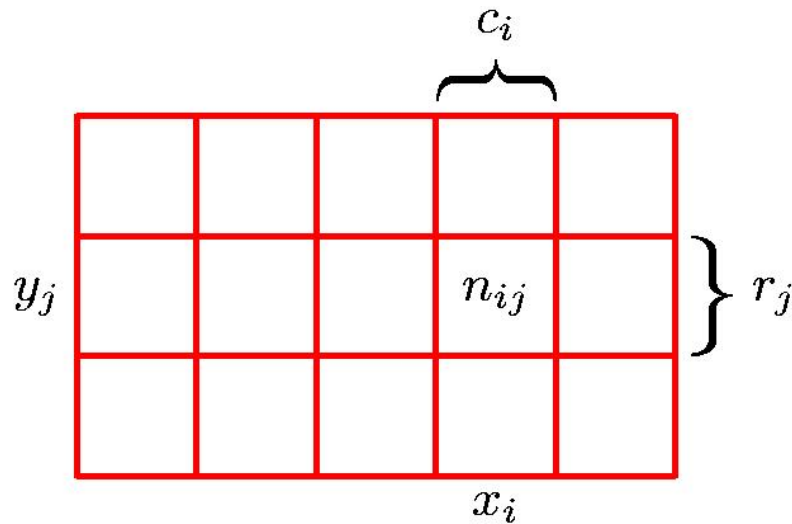

- Marginal probability can be written as:

$$p(X = x_i) = \sum_{j=1}^{L} p(X = x_i, Y = y_j)$$

- Called marginal probability because it is obtained by marginalizing, or summing out, the other variables.

Basics of Probability Theory

- Conditional Probability:

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

- We can derive the following relationship:

$$
\begin{aligned}
p(X = x_i, Y = y_j) &= \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \frac{c_i}{N} \\
&= p(Y = y_j | X = x_i) p(X = x_i)
\end{aligned}
$$

which is the product rule of probability.

The Rules of Probability

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

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

Bayes' Rule

• From the product rule, together with symmetric property:

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

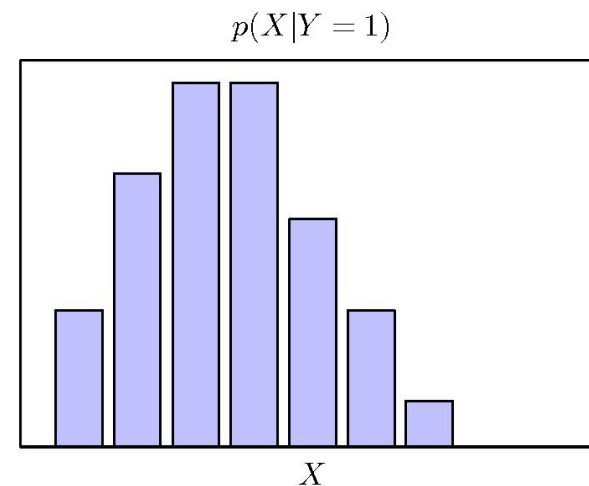$$p(Y|X)P(X) = p(X|Y)P(Y)$$

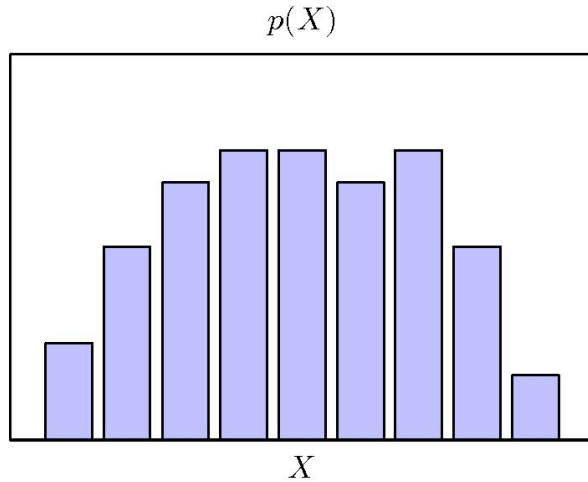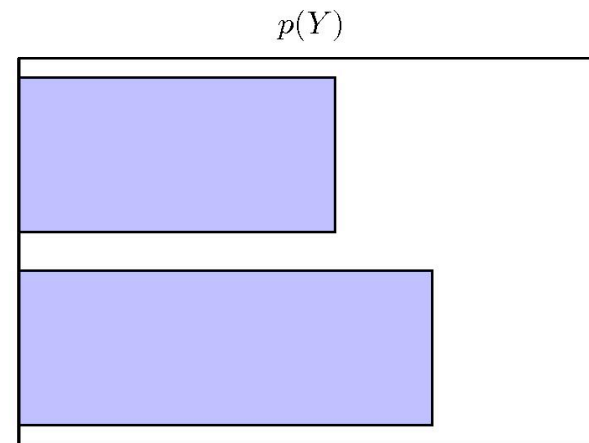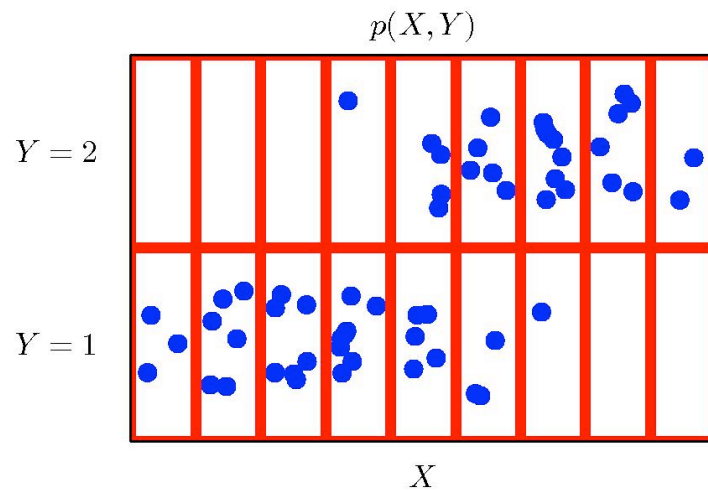$$p(Y|X) = \frac{p(X|Y)P(Y)}{P(X)}$$

• 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

• If the probability of a real-valued variable x falling in the interval

$(x, x + \delta x)$ is given by $p(x)\delta x$ for $\delta x \to 0,$ then p(x) is called the
probability density over x.
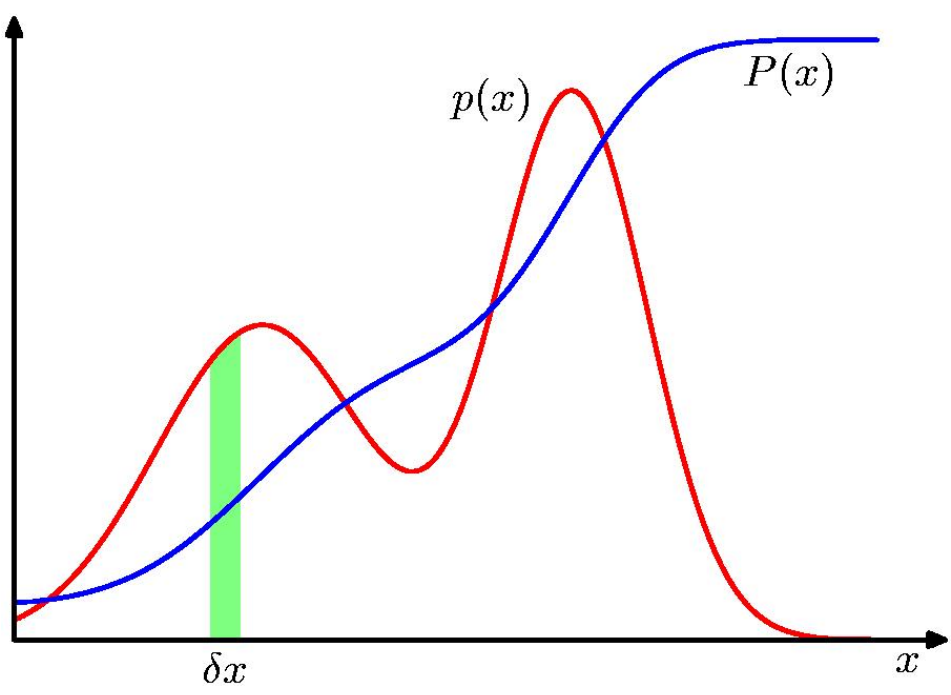
$$p(x \in (a, b)) = \int_a^b p(x)\,\mathrm{d}x$$

• The probability density must
satisfy the following two
conditions

$$p(x) \geqslant 0$$

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

Probability Density

• Cumulative distribution function is defined as:

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

which also satisfies:

$$P'(x) = p(x)$$

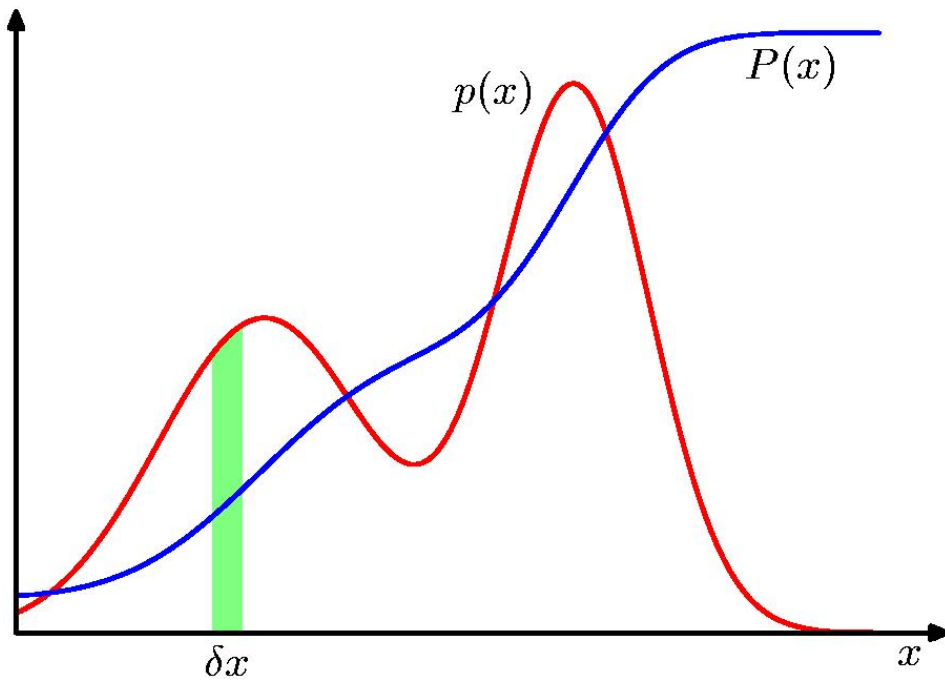• The sum and product rules take similar forms:

$$p(x) = \int p(x,y)\,dy$$

$$p(x,y) = p(y|x)p(x)$$

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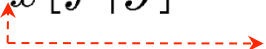
• The average value of some function f(x) under a probability distribution (density) p(x) is called the expectation of f(x):

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

• 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:

$$\text{cov}[x, y] = \mathbb{E}_{x,y}\left[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}\right]$$
$$= \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y]$$

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:

$$\text{cov}[\mathbf{x}, \mathbf{y}] = \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{y}^{\text{T}} - \mathbb{E}[\mathbf{y}^{\text{T}}]\}\right]$$
$$= \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbf{x}\mathbf{y}^{\text{T}}] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^{\text{T}}]$$

The Gaussian Distribution

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

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$
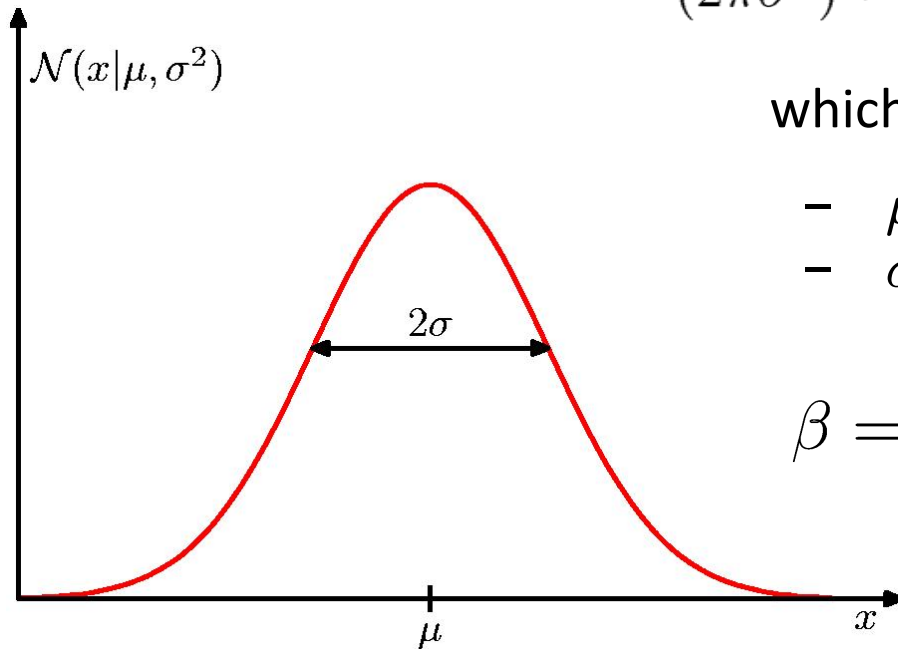
which is governed by two parameters:

- 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:

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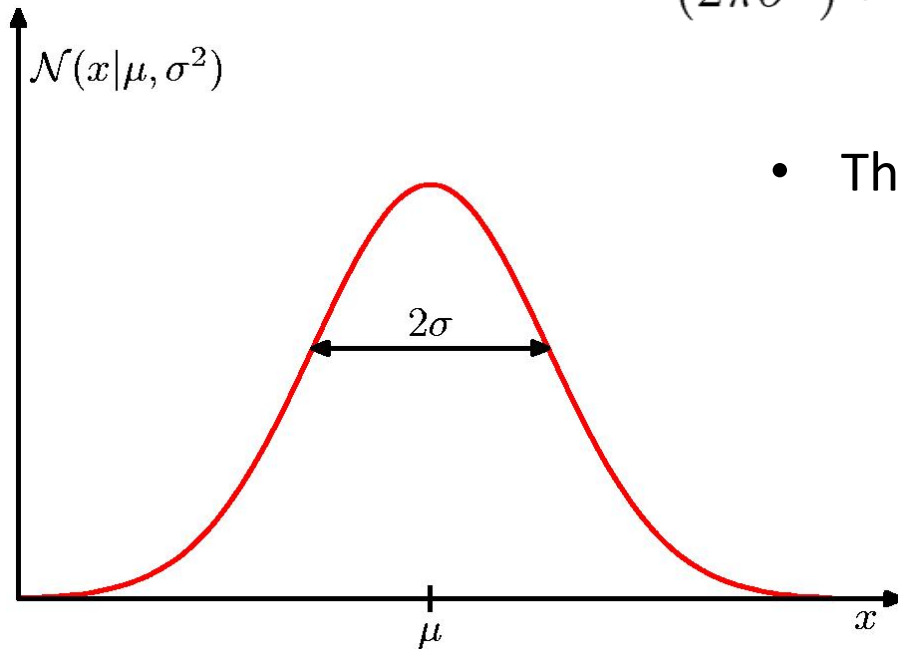
- The Gaussian distribution satisfies:

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

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

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:

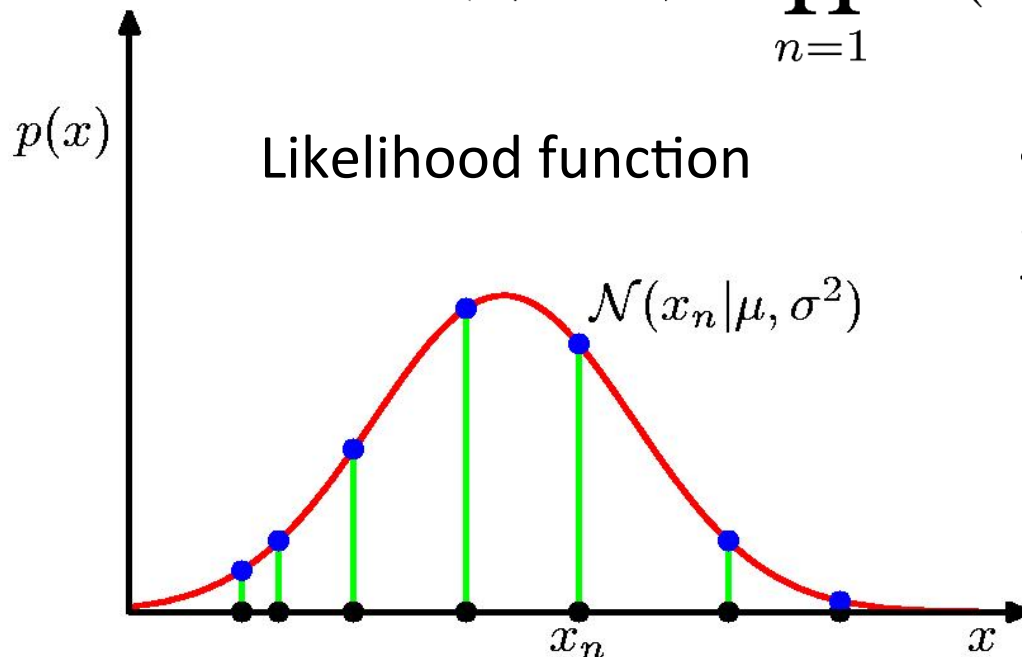$$\mathrm{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2$$

Sampling Assumptions

• Suppose we have a dataset of observations $\mathbf{x} = (x_1,...,x_N)^T$, representing N 1-dimensional observations.

• Assume that the training examples are drawn independently from the set of all possible examples, or from the same underlying distribution

• We also assume that the training examples are identically distributed $\rightarrow$ i.i.d assumption.

• Assume that the test samples are drawn in exactly the same way -- i.i.d from the same distribution as the training data.

• These assumptions make it unlikely that some strong regularity in the training data will be absent in the test data.

Gaussian Parameter Estimation

• Suppose we have a dataset of i.i.d. observations $\mathbf{x} = (x_1,...,x_N)^\mathsf{T}$, representing N 1-dimensional observations.

• Because out dataset $\mathbf{x}$ is i.i.d., we can write down the joint probability of all the data points as given μ and σ^2:

$$p(\mathbf{x}|\mu, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(x_n|\mu, \sigma^2\right)$$

$p(x)$

Likelihood function

$\mathcal{N}(x_n|\mu, \sigma^2)$

x_n

x

• When viewed as a function of μ and σ^2, this is called the likelihood function for the Gaussian.
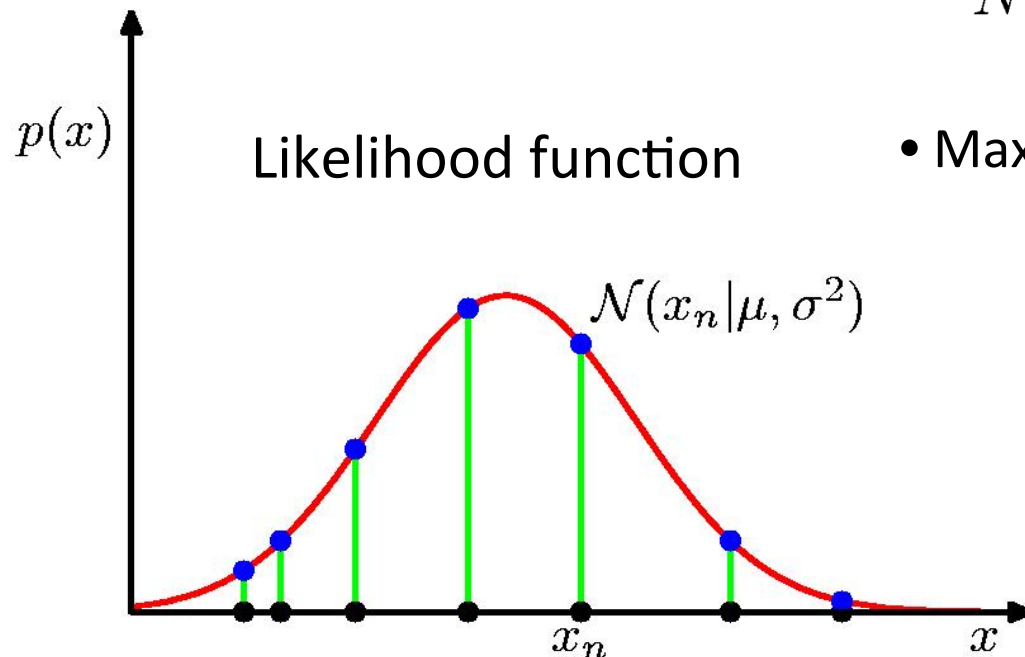
Maximum (log) likelihood

- The log-likelihood can be written as:

$$\ln p\left(\mathbf{x}|\mu, \sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

- Maximizing w.r.t μ gives:

Sample mean

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}x_n$$

$p(x)$

Likelihood function

$\mathcal{N}(x_n|\mu, \sigma^2)$

x_n

x

- Maximizing w.r.t σ^2 gives:

Sample variance

$$\sigma_{\mathrm{ML}}^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{\mathrm{ML}})^2$$

Properties of the ML estimation

- ML approach systematically underestimates the variance of the distribution.

- This is an example of a phenomenon called bias.

- It is straightforward to show that:

$$\mathbb{E}[\mu_{\mathrm{ML}}] = \mu \qquad \mathbb{E}[\sigma^2_{\mathrm{ML}}] = \left(\frac{N-1}{N}\right)\sigma^2$$

- It follows that the following estimate is unbiased:

$$\widetilde{\sigma}^2 = \frac{N}{N-1}\sigma^2_{\mathrm{ML}}$$

$$= \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu_{\mathrm{ML}})^2$$

Properties of the ML estimation

• Example of how bias arises in using ML to determine the variance of a Gaussian distribution:

• The green curve shows the true Gaussian distribution.

• Fit three datasets, each consisting of two blue points.

• Averaged across 3 datasets, the mean is correct.

• But the variance is under-estimated because it is measured relative to the sample (and not the true) mean.
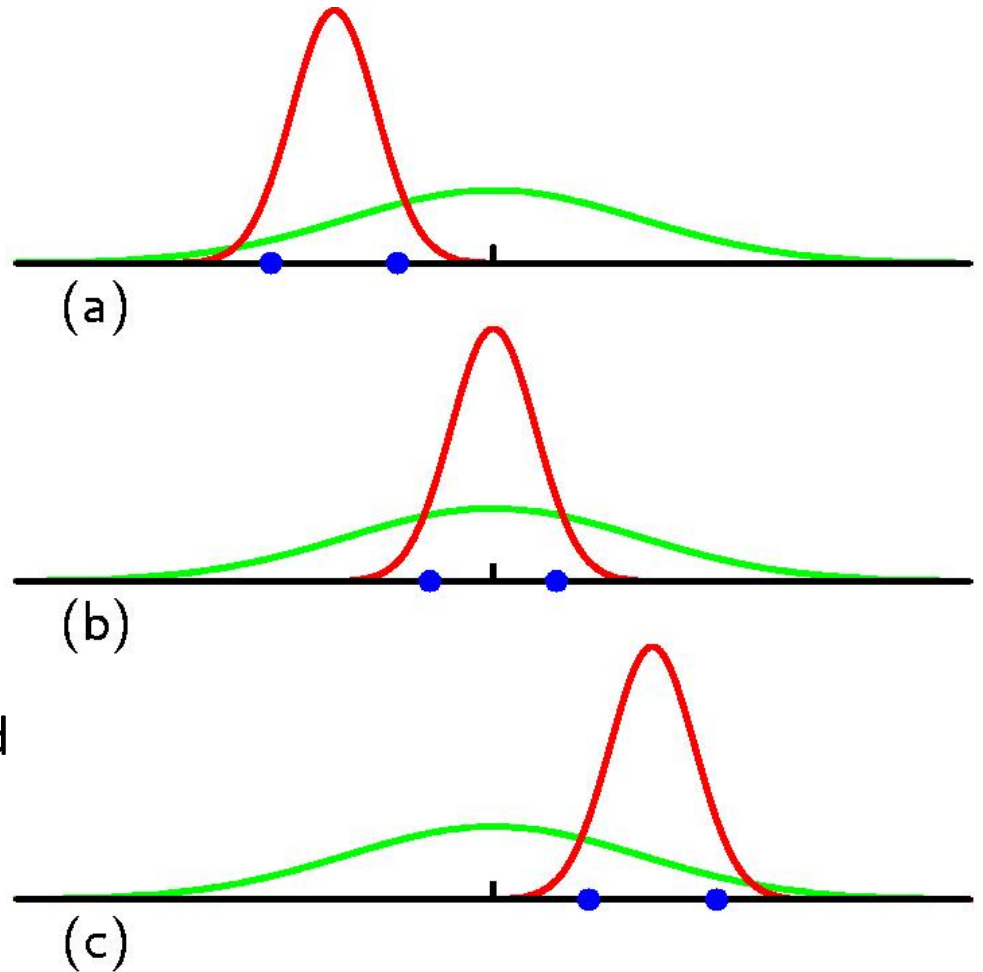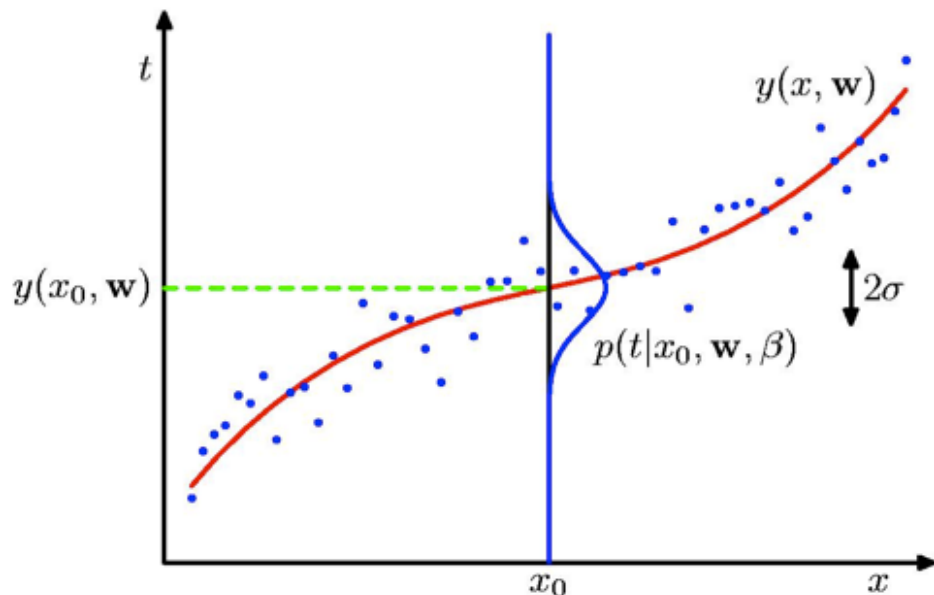
Probabilistic Perspective

• So far we saw that polynomial curve fitting can be expressed in terms of error minimization. We now view it from probabilistic perspective.

• Suppose that our model arose from a statistical model:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon,$$

where ϵ is a random error having Gaussian distribution with zero mean, and is independent of **x**.

Thus we have:

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}),$$

where β is a precision parameter, corresponding to the inverse variance.

I will use probability distribution and probability density interchangeably. It should be obvious from the context.

Maximum Likelihood

If the data are assumed to be independently and identically distributed (*i.i.d assumption*), the likelihood function takes form:

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{i=1}^{N} \mathcal{N}(t_n|y(\mathbf{x}_n, \mathbf{w}), \beta^{-1}).$$

It is often convenient to maximize the log of the likelihood function:

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2} \underbrace{\sum_{n=1}^{N} (y(\mathbf{x}_n, \mathbf{w}) - t_n)^2}_{\beta E(\mathbf{w})} + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi).$$

• Maximizing log-likelihood with respect to **w** (under the assumption of a Gaussian noise) is equivalent to minimizing the *sum-of-squared error* function.

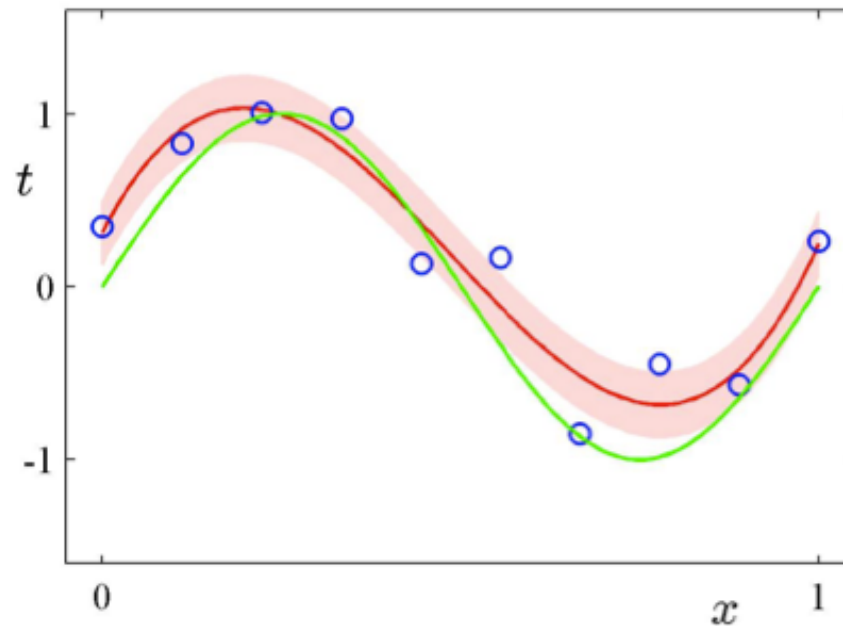• Determine $\mathbf{w}_{ML}$ by maximizing log-likelihood. Then maximizing w.r.t. β:

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n} (y(\mathbf{x}_n, \mathbf{w}_{ML}) - t_n)^2.$$

Predictive Distribution

Once we determined the parameters **w** and β, we can make prediction for new values of **x**:

$$p(t|\mathbf{x}, \mathbf{w}_{ML}, \beta_{ML}) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}_{ML}), \beta_{ML}^{-1}).$$

Later we will consider Bayesian linear regression.

Statistical Decision Theory

• We now develop a small amount of theory that provides a framework for developing many of the models we consider.

• Suppose we have a real-valued input vector **x** and a corresponding target (output) value t with joint probability distribution: $p(\mathbf{x}, t)$.

• Our goal is predict target t given a new value for **x**:
  - for regression: t is a real-valued continuous target.
  - for classification: t a categorical variable representing class labels.

The joint probability distribution $p(\mathbf{x}, t)$ provides a complete summary of uncertainties associated with these random variables.

Determining $p(\mathbf{x}, t)$ from training data is known as the inference problem.