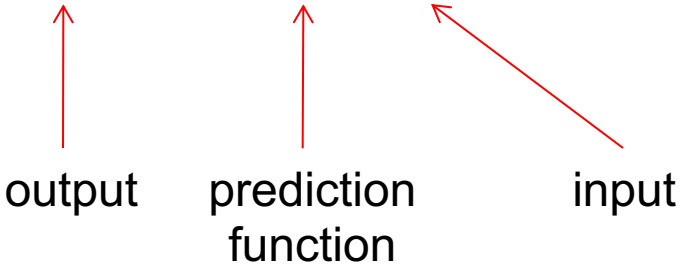
Intro to ML/DL with a Brief LinAlg/Calc Review

ML/DL in a Nutshell

$$y' = f(\mathbf{x})$$



- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y' = f(\mathbf{x})$

Slide credit: L. Lazebnik

ML/DL in a Nutshell

- Example:
  - Predict whether an email is spam or not:

Sebring, Tracy
To: Batra, Dhruv
ECE 4424 proposal

CUSP has approved ECE 4424 with the following changes: Can you
copy of the proposal with these items addressed? (see below)
Thanks!!!
Tracy

vs

nadia bamba
To: undisclosed recipients: ;
Reply-To: nadia bamba
From Miss Nadia BamBa,

January 19, 2015 5:57 AM
Hide Details

From Miss Nadia BamBa,

Greeting, Permit me to inform you of my desire of going into business relationship with you. I am Nadia BamBa the only Daughter of late Mr and Mrs James BamBa, My father was a director of cocoa merchant in Abidjan, the economic capital of Ivory Coast before he was poisoned to death by his business associates on one of their outing to discus on a business deal. When my mother died on the 21st October 2002, my father took me very special because i am motherless.

Before the death of my father in a private hospital here in Abidjan, He secretly called me on his bedside and told me that he had a sum of $6, 8000.000(SIX Million EIGHT HUNDRED THOUSAND), Dollars) left in a suspense account in a Bank here in Abidjan, that he used my name as his first Daughter for the next of kin in deposit of the fund.

He also explained to me that it was because of this wealth and some huge amount of money That his business associates supposed to balance him from the deal they had that he was poisoned by his business associates, that I should seek for a God fearing foreign partner in a country of my choice where I will transfer this money and use it for investment purposes, (such as real estate Or Hotel management).please i am honourably seeking your assistance in the following ways.

1) To provide a Bank account where this money would be transferred to.
2) To serve as the guardian of this Money since I am a girl of 19 years old.
3)Your private phone number's and your family background' s that we can know each order more.

Figures from Dhruv Batra

ML/DL in a Nutshell

- Example:
  - Predict whether an email is spam or not.
  - **x** = words in the email, one-hot representation of size |V|x1, where V is the full vocabulary and x(j) = 1 iff word *j* is mentioned
  - y = 1 (if spam) or 0 (if not spam)
  - y' = f(**x**) = $\mathbf{w}^T$ **x**
    - **w** is a vector of the same size as **x**
    - One weight per dimension of **x** (i.e. one weight per word)
    - Weight can be positive, zero, negative…
    - What might these weights look like?

Simple strategy: Let's count!

This is X

This is Y

nadia bamba
To: undisclosed recipients: ;
Reply-To: nadia bamba
From Miss Nadia BamBa,

From Miss Nadia BamBa,

Greeting, Permit me to inform you of my desire of going i
Nadia BamBa the only Daughter of late Mr and Mrs Jame
cocoa merchant in Abidjan, the economic capital of Ivory
his business associates on one of their outing to discus c
on the 21st October 2002, my father took me very specia

Before the death of my father in a private hospital here i
bedside and told me that he had a sum of $6, 8000.000(S
Dollars) left in a suspense account in a Bank here in Abic
Daughter for the next of kin in deposit of the fund.

$$\begin{pmatrix} \text{free} & 100 \\ \text{money} & 2 \\ \vdots & \vdots \\ \text{account} & 2 \\ \vdots & \vdots \end{pmatrix}$$

= 1 or 0?

Sebring, Tracy
To: Batra, Dhruv
ECE 4424 proposal

CUSP has approved ECE 4424 with the following changes: Can
copy of the proposal with these items addressed? (see below)
Thanks!!!
Tracy

$$\begin{pmatrix} \text{free} & 1 \\ \text{money} & 1 \\ \vdots & \vdots \\ \text{account} & 2 \\ \vdots & \vdots \end{pmatrix}$$

Adapted from Dhruv Batra, Fei Sha

Weigh counts and sum to get prediction

nadia bamba
To: undisclosed recipients: ;
Reply-To: nadia bamba
From Miss Nadia BamBa,

From Miss Nadia BamBa,

Greeting, Permit me to inform you of
Nadia BamBa the only Daughter of la
cocoa merchant in Abidjan, the econ
his business associates on one of th
on the 21st October 2002, my father

Before the death of my father in a pr
bedside and told me that he had a su
Dollars) left in a suspense account in
Daughter for the next of kin in depos

$$\begin{pmatrix} 100 \times 0.2 \\ 2 \times 0.3 \\ \vdots \\ 2 \times 0.3 \\ \vdots \end{pmatrix}$$

$$\begin{pmatrix} \text{free} & 100 \\ \text{money} & 2 \\ \vdots & \vdots \\ \text{account} & 2 \\ \vdots & \vdots \end{pmatrix}$$

Adapted from Dhruv Batra, Fei Sha

ML/DL in a Nutshell

- Example:
  - Apply a prediction function to an image to get the desired label output:

f() = “apple”

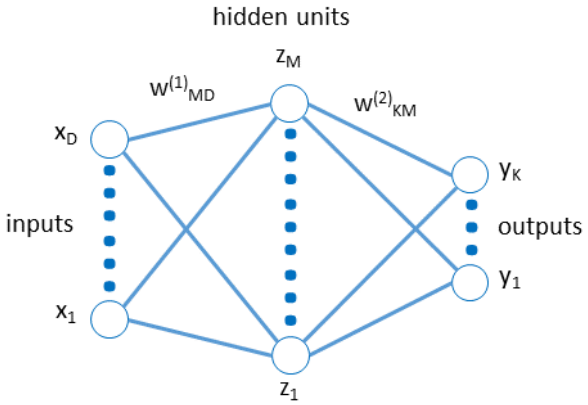
f() = “tomato”

f() = “cow”

ML/DL in a Nutshell

- Example:
  - **x** = pixels of the image (concatenated to form a vector)
  - y = integer (1 = apple, 2 = tomato, etc.)
  - $y' = f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$
    - **w** is a vector of the same size as **x**
    - One weight per each dimension of **x** (i.e. one weight per pixel)

DL in a Nutshell



- Input → network → outputs
- Input X is raw (e.g. raw image, one-hot representation of text)
- Network extracts features: abstraction of input
- Output is the labels Y
- All parameters of the network trained by checking how well predicted/true Y agree, using labels in the training set

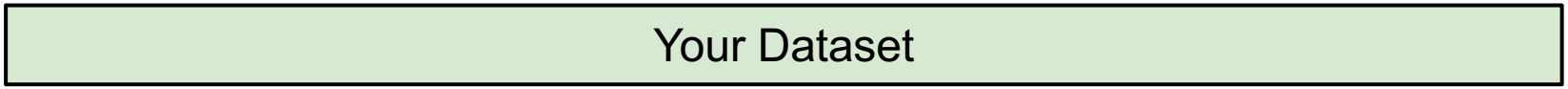
Validation strategies

- Ultimately, for our application, what do we want?
  - High accuracy on training data?
  - No, high accuracy on *unseen/new/test data!*
  - Why is this tricky?
- Training data
  - Features (x) and labels (y) used to learn mapping f
- Test data
  - Features used to make a prediction
  - Labels only used to see how well we've learned f!!!
- Validation data
  - Held-out set of the *training data*
  - Can use both features and labels to tune model *hyperparameters*
  - *Hyperparameters* are "knobs" of the algorithm tuned by the designer: number of iterations for learning, learning rate, etc.
  - We train multiple model (one per hyperparameter setting) and choose the best one, on the validation set

Validation strategies

Idea #1: Choose hyperparameters that work best on the data

BAD: Overfitting; e.g. in K-nearest neighbors, K = 1 always works perfectly on training data



Idea #2: Split data into **train** and **test**, choose hyperparameters that work best on test data

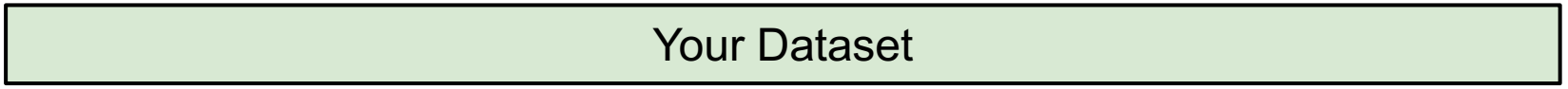
BAD: No idea how algorithm will perform on new data; cheating

Idea #3: Split data into **train**, **val**, and **test**; choose hyperparameters on val and evaluate on test

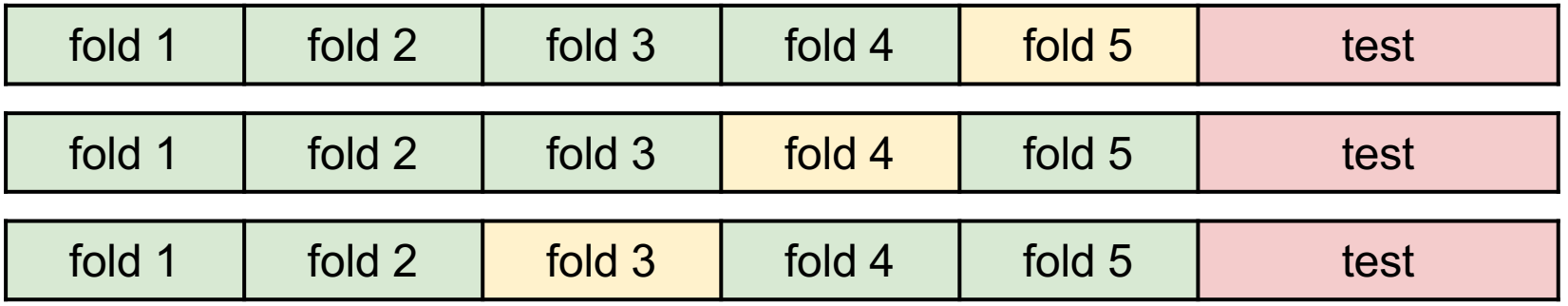
Better!

Adapted from Fei-Fei, Johnson, Yeung

Validation strategies



Idea #4: **Cross-Validation**: Split data into **folds**,
try each fold as validation and average the results



Useful for small datasets, but not used too frequently in deep learning

Adapted from Fei-Fei, Johnson, Yeung

Why do we hope this would work?

- Statistical estimation view:
  - x and y are *random variables*
  - D = (x_1,y_1), (x_2,y_2), …, (x_N,y_N) ~ P(X,Y)
  - Both training & testing data sampled IID from P(X,Y)
    - IID: Independent and Identically Distributed
  - Learn on training set, have some hope of *generalizing* to test set

Adapted from Dhruv Batra

Elements of Machine *Learning*

- Every machine learning algorithm has:
  - Data representation (x, y)
  - Problem representation (network)
  - Evaluation / objective function
  - Optimization (solve for parameters of network)

Adapted from Pedro Domingos

Data representation

- Let's brainstorm what our "X" should be for various "Y" prediction tasks…

Problem representation

- Instances
- Decision trees
- Sets of rules / Logic programs
- Support vector machines
- Graphical models (Bayes/Markov nets)
- **Neural networks**
- Model ensembles
- Etc.

Slide credit: Pedro Domingos

Evaluation / objective function

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.

Slide credit: Pedro Domingos

Loss functions

- Measure error
- Can be defined for discrete or continuous outputs
- E.g. if task is classification – could use cross-entropy loss
- If task is regression – use L2 loss i.e. $||y-y'||$

Optimization

- Optimization means we need to solve for the parameters **w** of the model
- For a (non-linear) neural network, there is no closed-form solution to solve for **w**; cannot set up linear system with **w** as the unknowns
- Thus, all optimization solutions look like this:
  1. Initialize **w** (e.g. randomly)
  2. Check error (ground-truth vs predicted labels on training set) under current model
  3. Use gradient (derivative) of error wrt **w** to update **w**
  4. Repeat from 2 until convergence

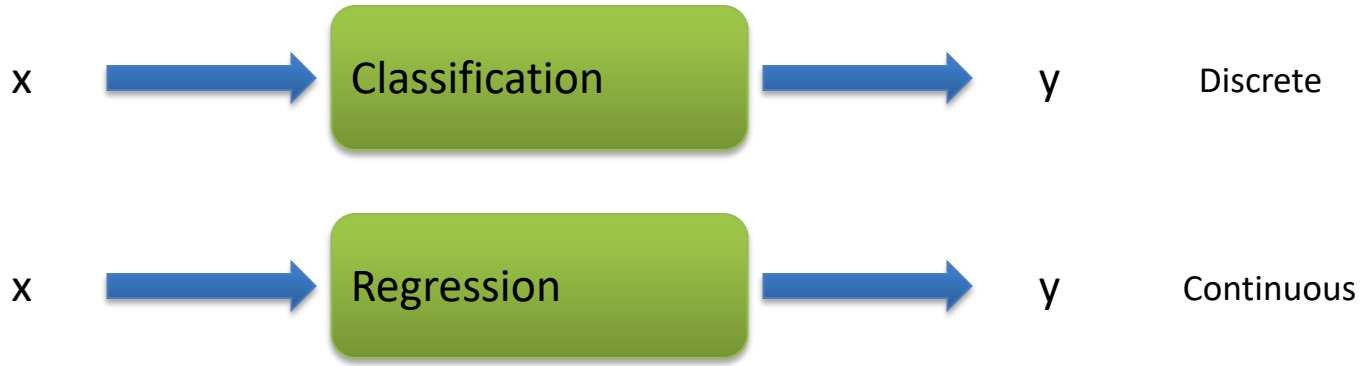
Types of Learning

- Supervised learning
  - Training data includes desired outputs
- Unsupervised learning
  - Training data does not include desired outputs
- Weakly or Semi-supervised learning
  - Training data includes a few desired outputs, or contains labels that only approximate the labels desired at test time
- Reinforcement learning
  - Rewards from sequence of actions

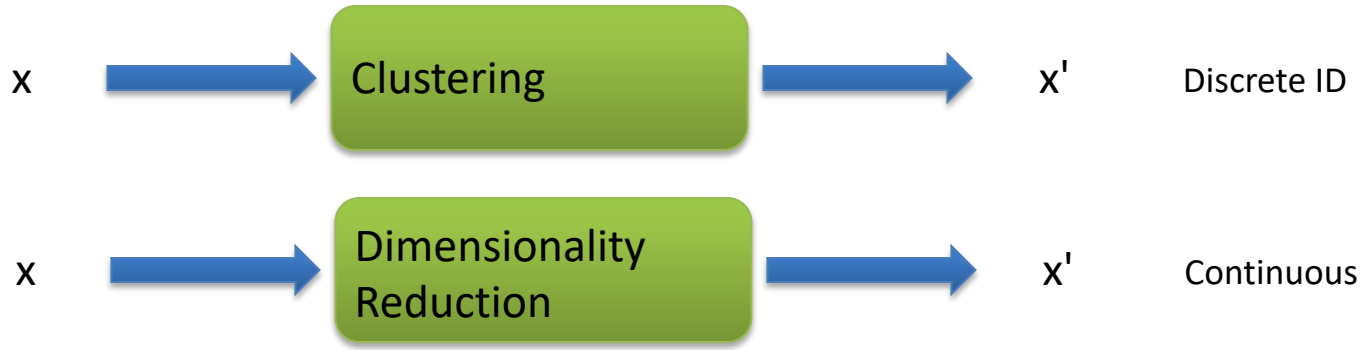
Adapted from: Dhruv Batra

Types of Prediction Tasks

Supervised Learning



Unsupervised Learning



Adapted from Dhruv Batra

Recall:
Example of Solving a ML Problem

- Spam or not?

Sebring, Tracy
To: Batra, Dhruv
ECE 4424 proposal

CUSP has approved ECE 4424 with the following cha
copy of the proposal with these items addressed? (se
Thanks!!!
Tracy

vs

nadia bamba
To: undisclosed recipients: ;
Reply-To: nadia bamba
From Miss Nadia BamBa,

January 19, 2015 5:57 AM
Hide Details

From Miss Nadia BamBa,

Greeting, Permit me to inform you of my desire of going into business relationship with you. I am Nadia BamBa the only Daughter of late Mr and Mrs James BamBa, My father was a director of cocoa merchant in Abidjan, the economic capital of Ivory Coast before he was poisoned to death by his business associates on one of their outing to discus on a business deal. When my mother died on the 21st October 2002, my father took me very special because i am motherless.

Before the death of my father in a private hospital here in Abidjan, He secretly called me on his bedside and told me that he had a sum of $6, 8000.000(SIX Million EIGHT HUNDRED THOUSAND), Dollars) left in a suspense account in a Bank here in Abidjan, that he used my name as his first Daughter for the next of kin in deposit of the fund.

He also explained to me that it was because of this wealth and some huge amount of money That his business associates supposed to balance him from the deal they had that he was poisoned by his business associates, that I should seek for a God fearing foreign partner in a country of my choice where I will transfer this money and use it for investment purposes, (such as real estate Or Hotel management).please i am honourably seeking your assistance in the following ways.

1) To provide a Bank account where this money would be transferred to.
2) To serve as the guardian of this Money since I am a girl of 19 years old.
3)Your private phone number's and your family background' s that we can know each order more.

Moreover i am willing to offer you 15% of the total sum as compensation for effort input after the successful transfer of this fund to your designated account overseas,

Anticipating to hear from you soon.
Thanks and God Bless.
Best regards.

Slide credit: Dhruv Batra

Simple strategy: Let's count!

nadia bamba
To: undisclosed recipients: ;
Reply-To: nadia bamba
From Miss Nadia BamBa,

From Miss Nadia BamBa,

Greeting, Permit me to inform you of my desire of going i
Nadia BamBa the only Daughter of late Mr and Mrs Jame
cocoa merchant in Abidjan, the economic capital of Ivory
his business associates on one of their outing to discus c
on the 21st October 2002, my father took me very specia

Before the death of my father in a private hospital here i
bedside and told me that he had a sum of $6, 8000.000(S
Dollars) left in a suspense account in a Bank here in Abic
Daughter for the next of kin in deposit of the fund.

Sebring, Tracy
To: Batra, Dhruv
ECE 4424 proposal

CUSP has approved ECE 4424 with the following changes: Can
copy of the proposal with these items addressed? (see below)
Thanks!!!
Tracy

This is X

$$\begin{pmatrix} \text{free} & 100 \\ \text{money} & 2 \\ \vdots & \vdots \\ \text{account} & 2 \\ \vdots & \vdots \end{pmatrix}$$

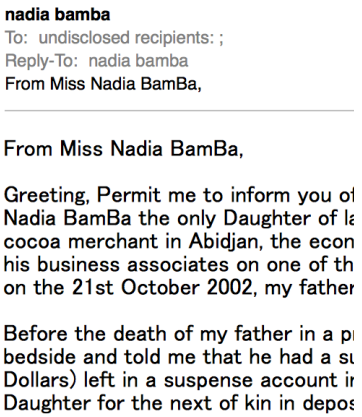
$$\begin{pmatrix} \text{free} & 1 \\ \text{money} & 1 \\ \vdots & \vdots \\ \text{account} & 2 \\ \vdots & \vdots \end{pmatrix}$$

This is Y

= 1 or 0?

Adapted from Dhruv Batra, Fei Sha

Weigh counts and sum to get prediction



$$\begin{pmatrix} \text{free} & 100 \\ \text{money} & 2 \\ \vdots & \vdots \\ \text{account} & 2 \\ \vdots & \vdots \end{pmatrix}$$

$$\begin{pmatrix} 100 \times 0.2 \\ 2 \times 0.3 \\ \vdots \\ 2 \times 0.3 \\ \vdots \end{pmatrix}$$

$$\begin{pmatrix} 100 \times 0.01 \\ 2 \times 0.02 \\ \vdots \\ 2 \times 0.01 \\ \vdots \end{pmatrix}$$

Where do the weights come from?

Adapted from Dhruv Batra, Fei Sha

Why not just hand-code these weights?

- We're letting the data do the work rather than develop hand-code classification rules
  - The *machine* is *learning* to program itself

- But there are challenges…

Challenges

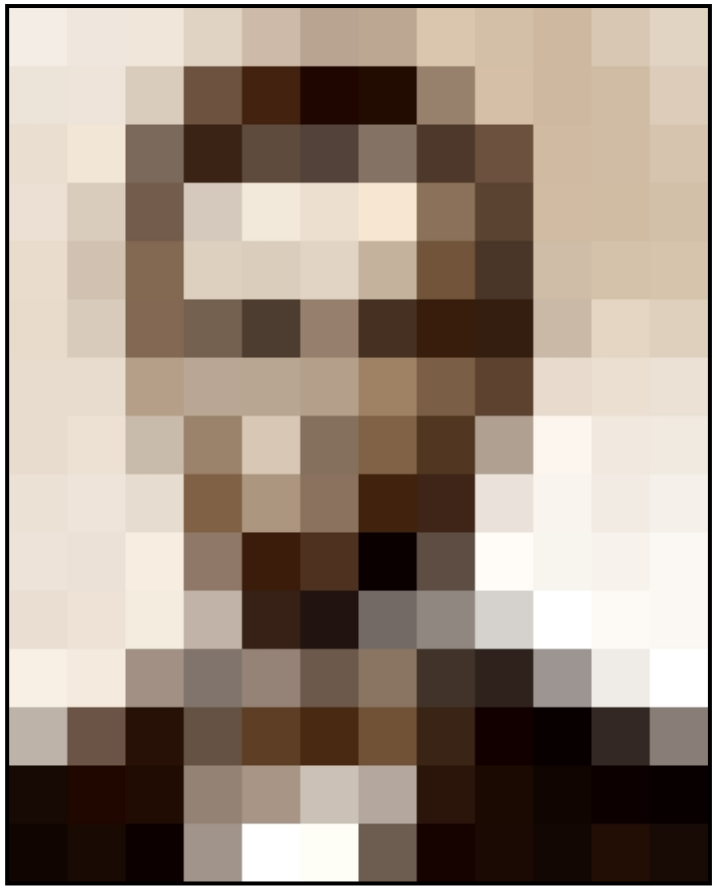
- Some challenges: ambiguity and context
- Machines take data representations too literally
- Humans are much better than machines at generalization, which is needed since test data will rarely look exactly like the training data

Klingon vs Mlingon Classification

- Imagine we have two different languages: Klingon and Mlingon
- Training Data
  - Klingon: klix, kour, koop
  - Mlingon: moo, maa, mou
- Testing Data: kap
- Which language? Why?

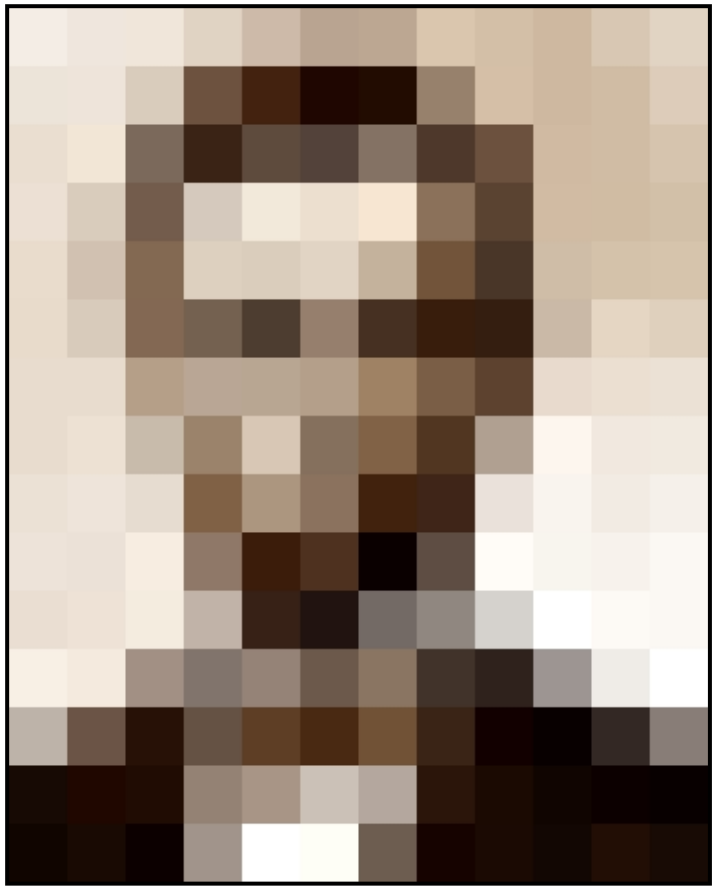
Slide credit: Dhruv Batra

What humans see



Slide credit: Larry Zitnick

What computers see



Slide credit: Larry Zitnick

Generalization

Training set (labels known)

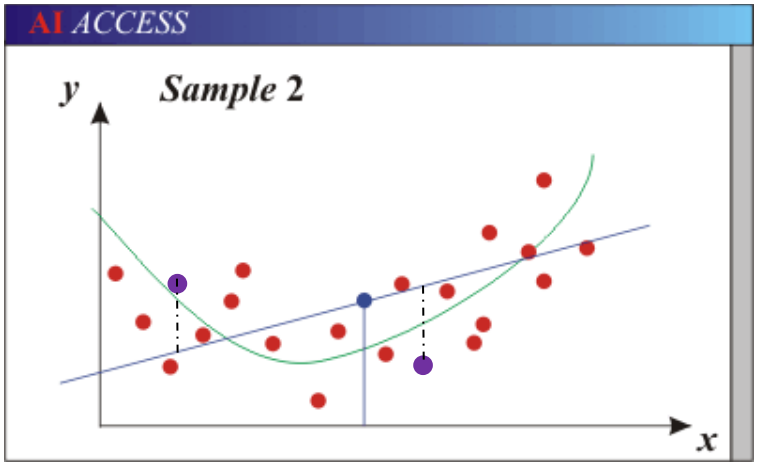
Test set (labels unknown)

- How well does a learned model generalize from the data it was trained on to a new test set?

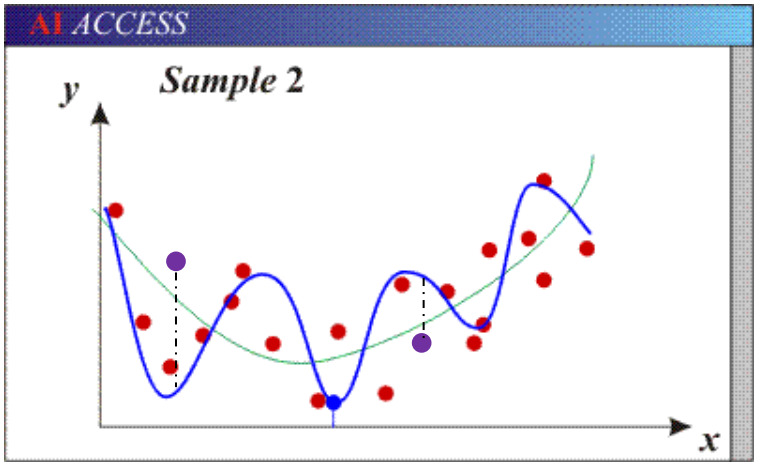
Well! It depends on the model we build!

Slide credit: L. Lazebnik

Generalization



- Underfitting: Models with too few parameters are inaccurate because of a large bias (not enough flexibility).



- Overfitting: Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).

Purple dots = possible test points

Red dots = training data (all that we see before we ship off our model!)

Green curve = true underlying model

Blue curve = our predicted model/fit

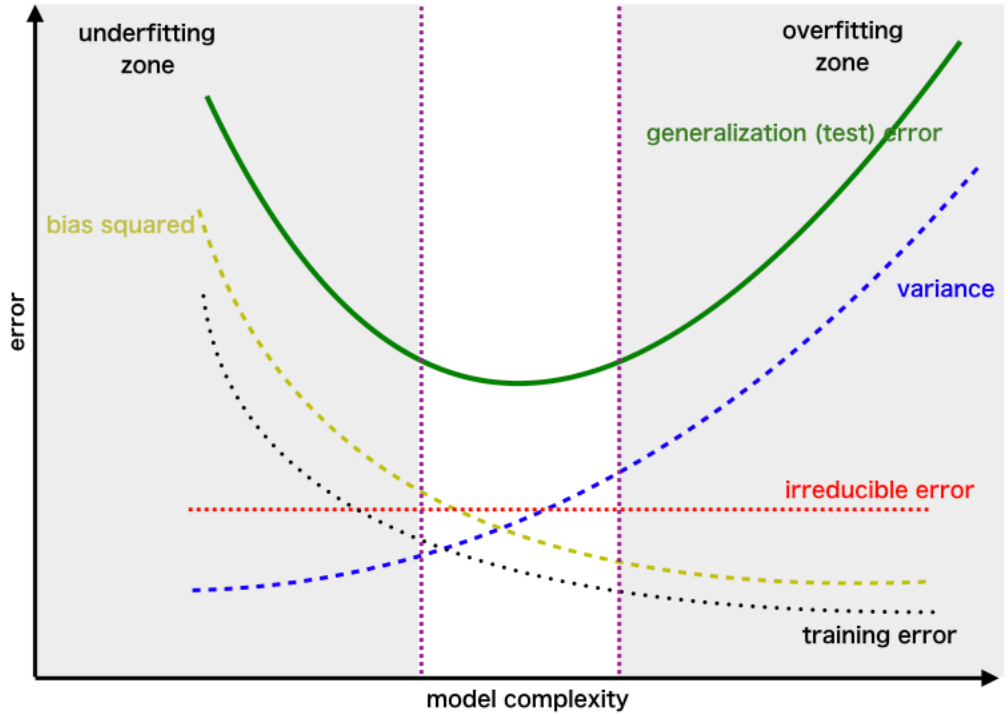
Adapted from D. Hoiem

Generalization

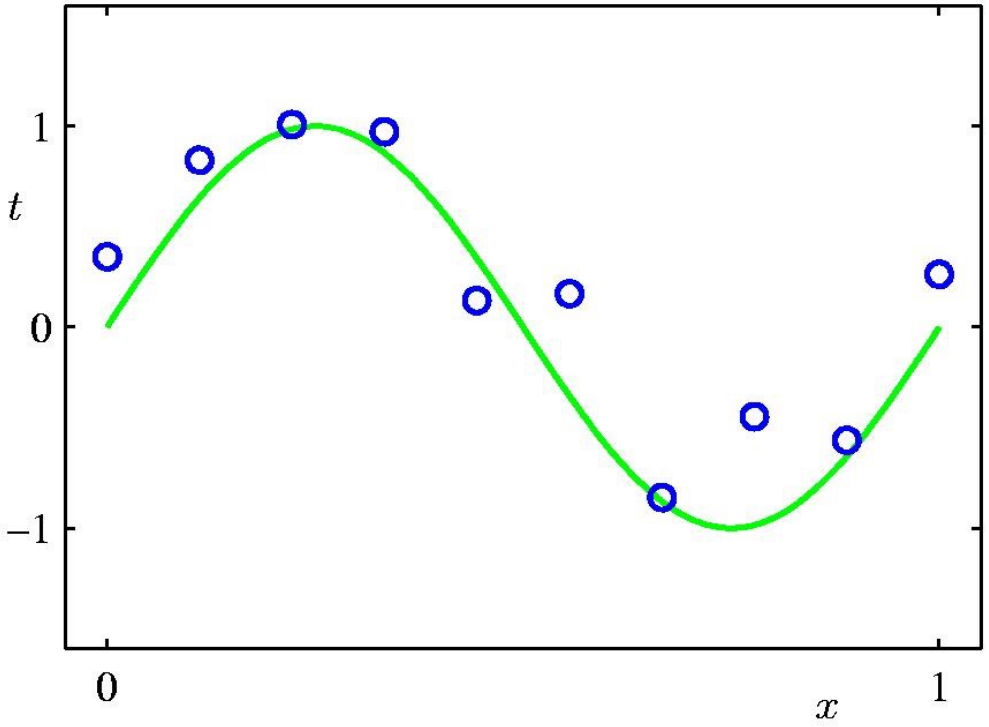
- Components of generalization error
  - **Noise** in our observations: unavoidable
  - **Bias:** how much the average model over all training sets differs from the true model
    - Inaccurate assumptions/simplifications made by the model
  - **Variance:** how much models estimated from different training sets differ from each other
- **Underfitting:** model is too "simple" to represent all the relevant class characteristics
  - High bias and low variance
  - High training error and high test error
- **Overfitting:** model is too "complex" and fits irrelevant characteristics (noise) in the data
  - Low bias and high variance
  - Low training error and high test error

Slide credit: L. Lazebnik

Bias-Variance Tradeoff



Polynomial Curve Fitting

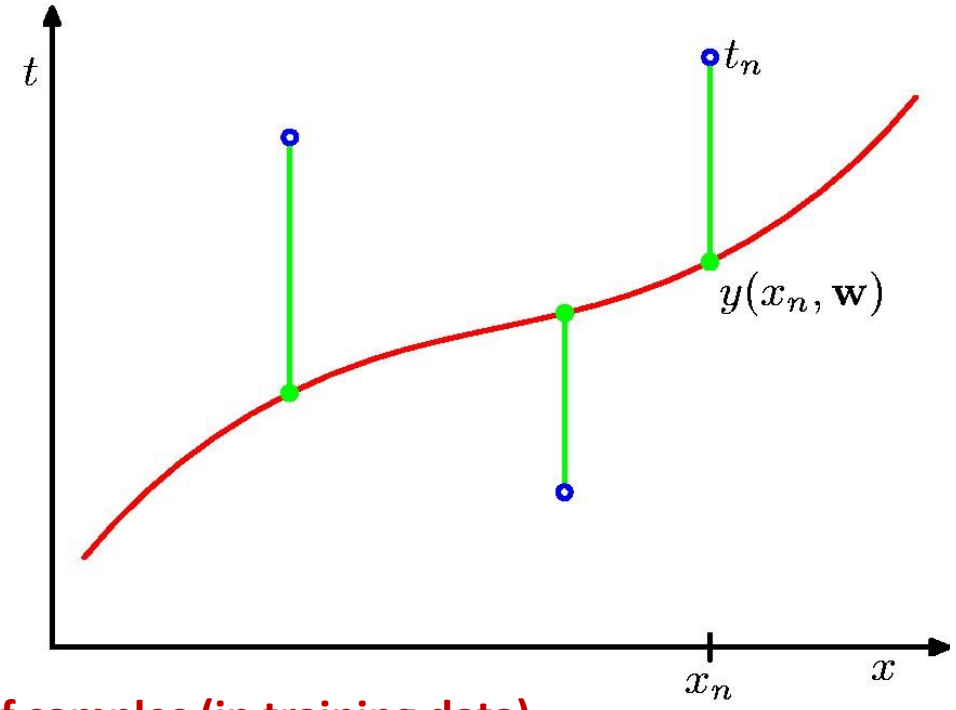


M: Polynomial Order

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

Slide credit: Chris Bishop

Sum-of-Squares Error Function

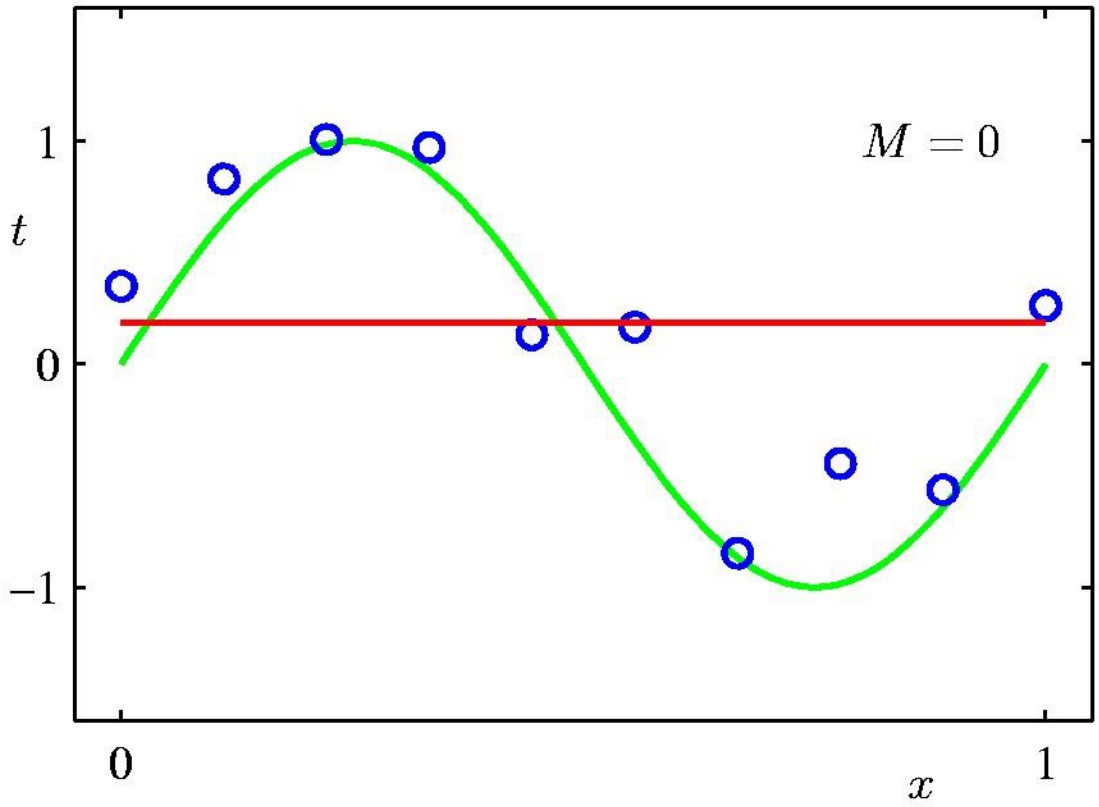


N: Number of samples (in training data)

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

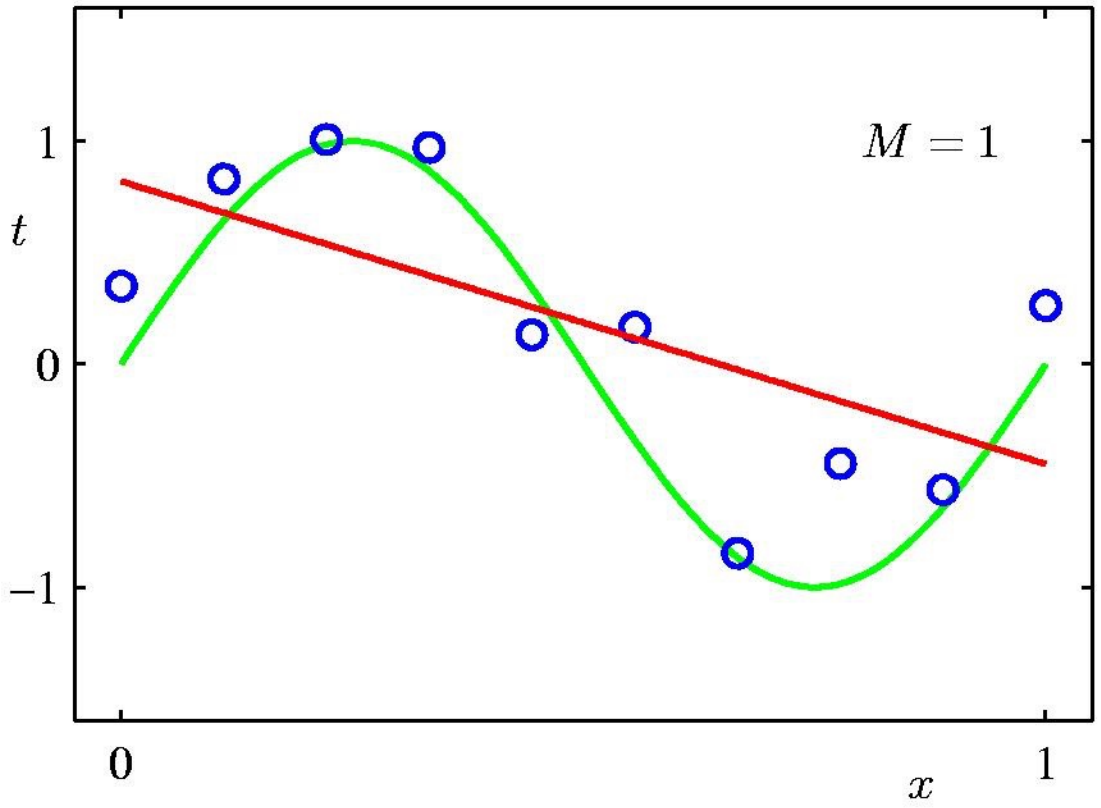
Slide credit: Chris Bishop

0^{th} Order Polynomial



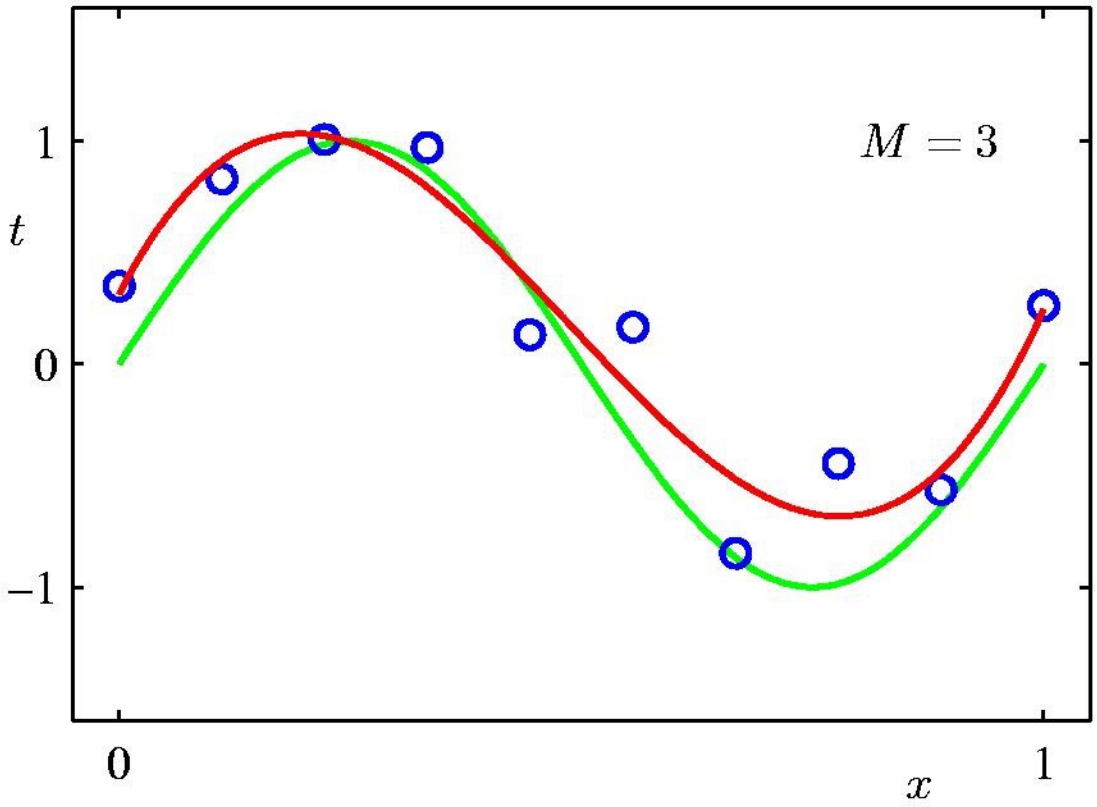
Slide credit: Chris Bishop

1st Order Polynomial



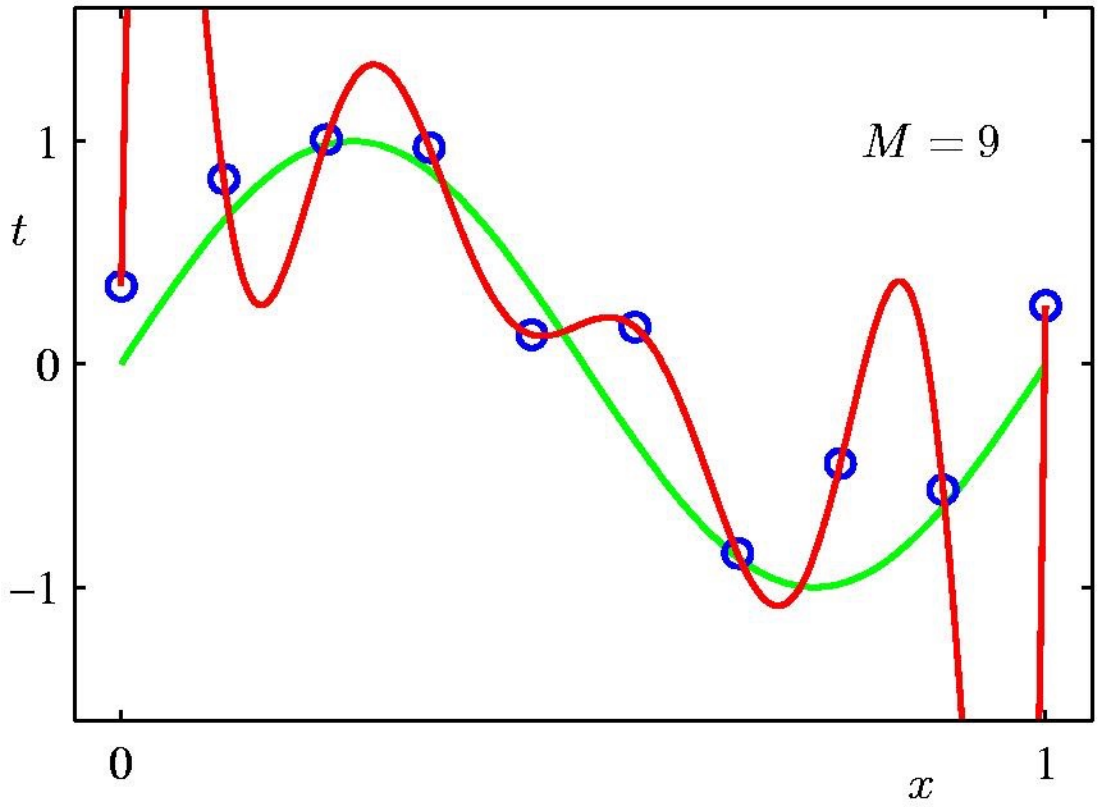
Slide credit: Chris Bishop

3rd Order Polynomial



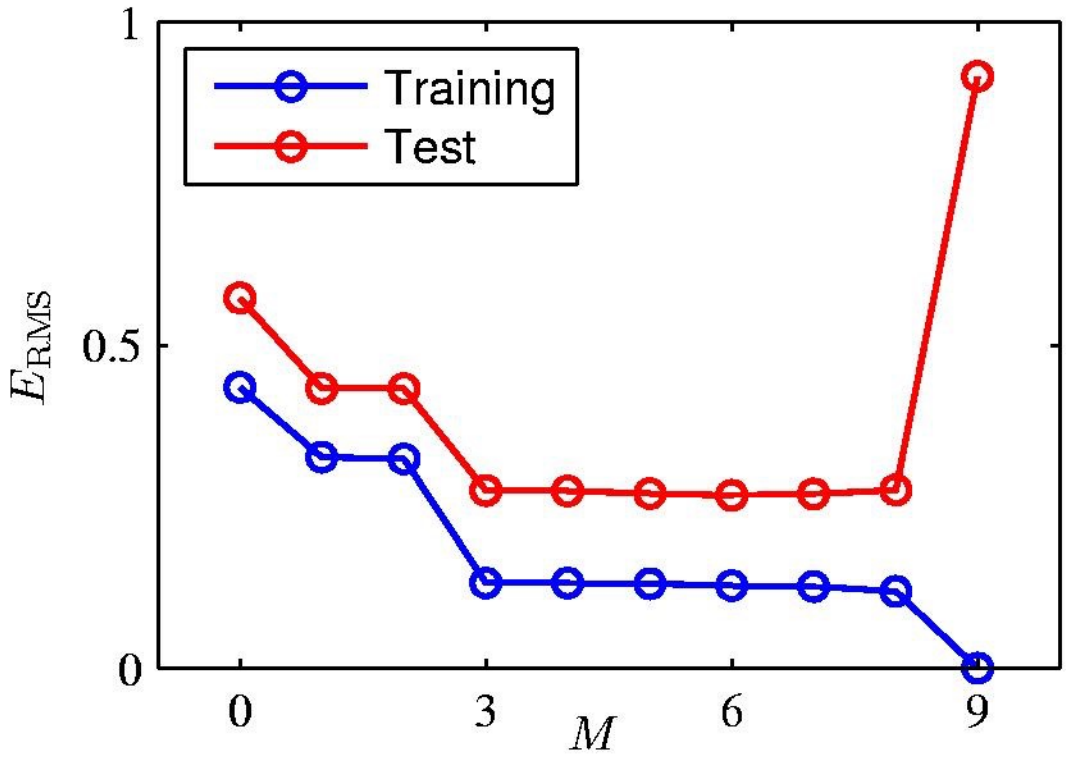
Slide credit: Chris Bishop

9th Order Polynomial



Slide credit: Chris Bishop

Over-fitting

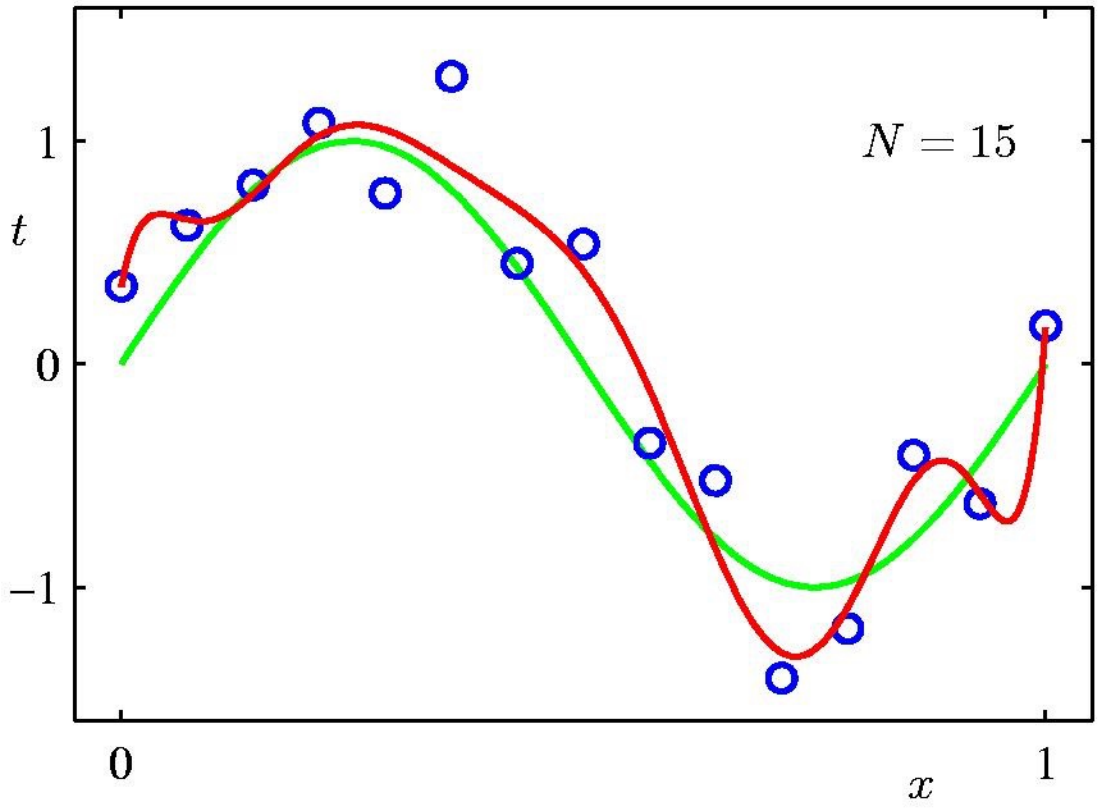


Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^{\star})/N}$

Slide credit: Chris Bishop

Data Set Size: $N = 15$

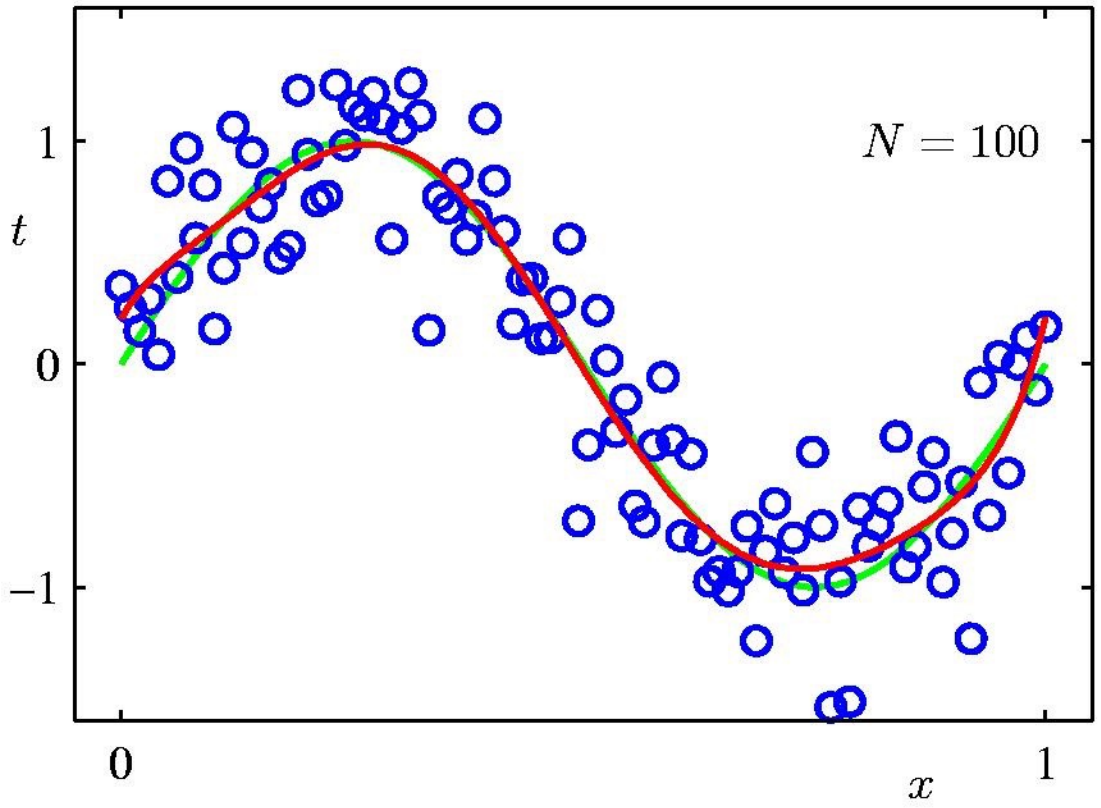
9th Order Polynomial



Slide credit: Chris Bishop

Data Set Size: $N = 100$

9th Order Polynomial



Slide credit: Chris Bishop

Regularization

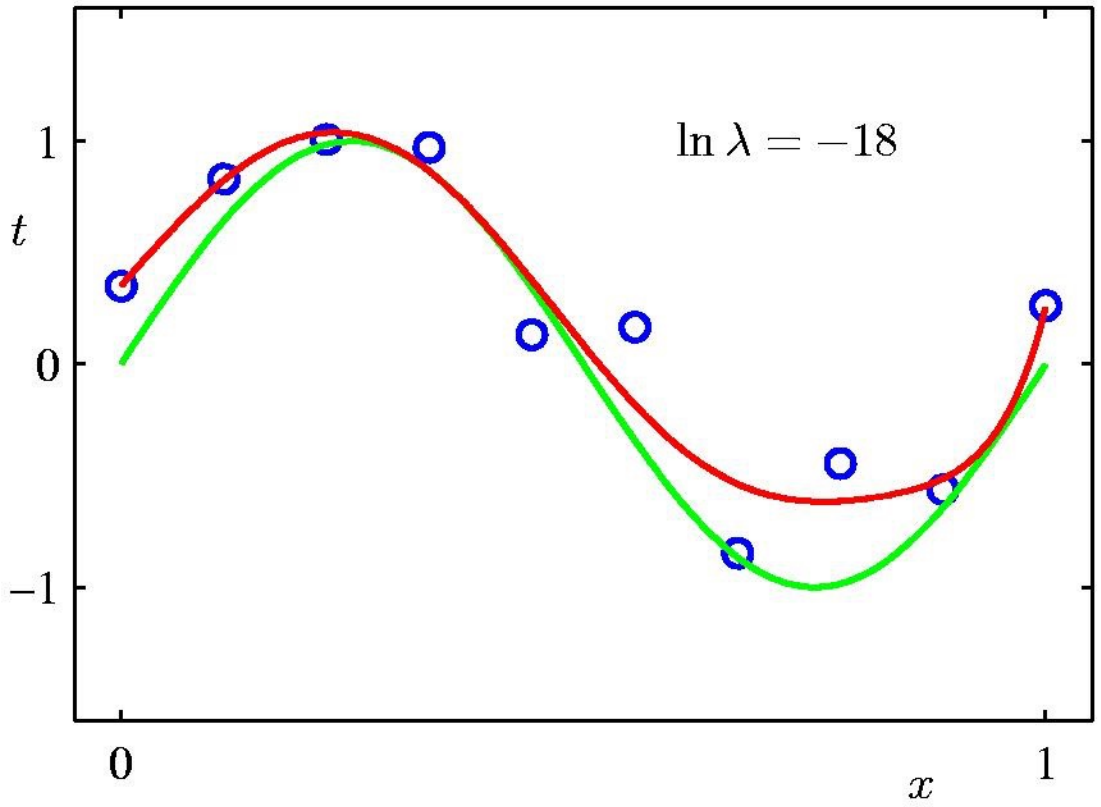
Penalize large coefficient values

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

(Remember: We want to minimize this expression.)

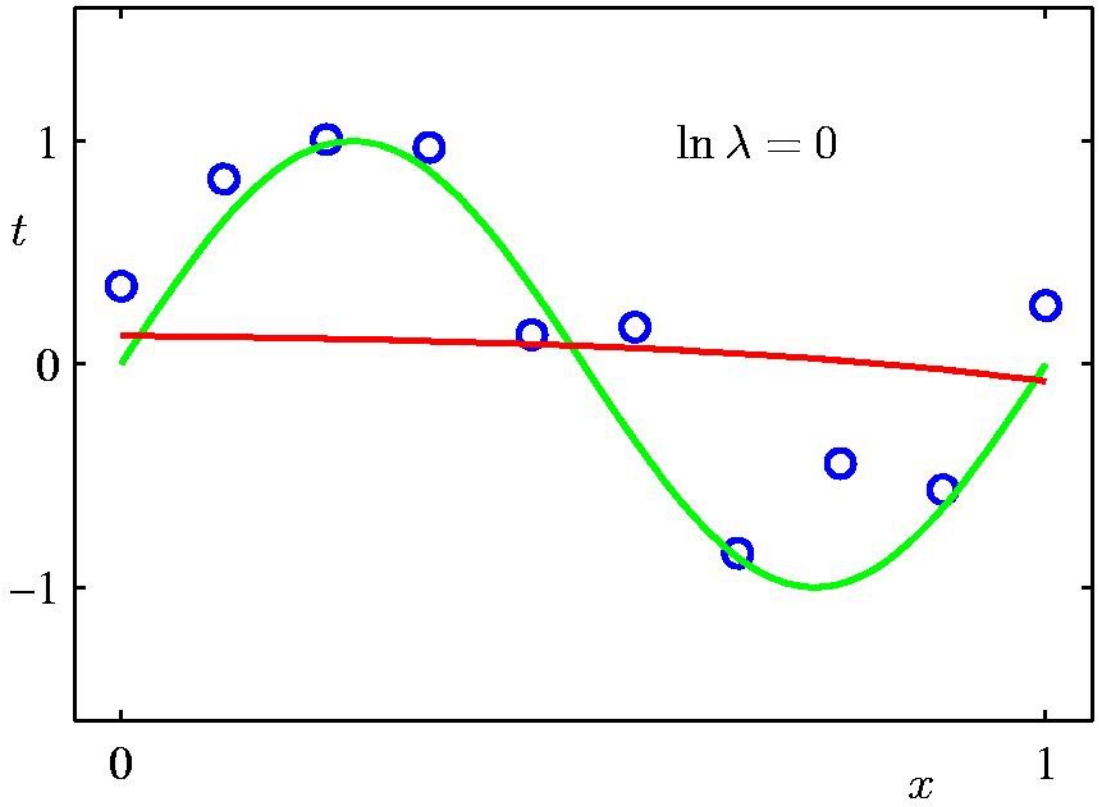
Lambda: Regularization Coefficient (Typically a small number, depends on M)

Regularization: $\ln \lambda = -18$



Slide credit: Chris Bishop

Regularization: $\ln \lambda = 0$



Slide credit: Chris Bishop

Polynomial Coefficients

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	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

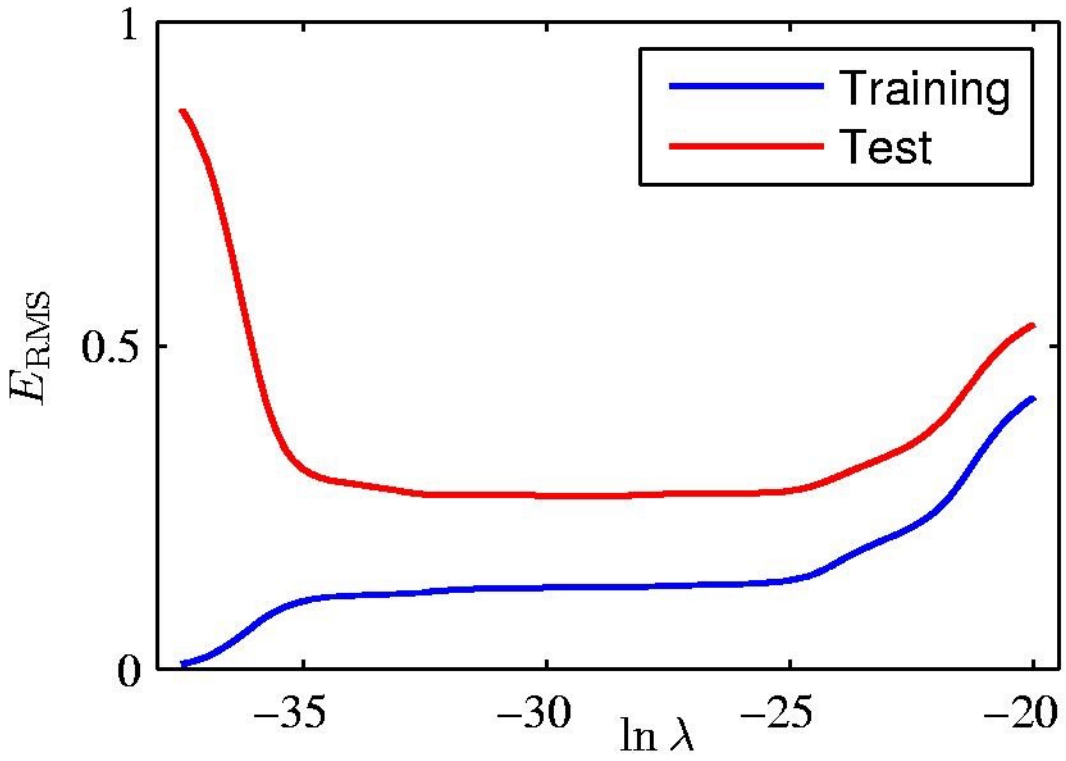
Slide credit: Chris Bishop

Polynomial Coefficients

	No regularization		Huge 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

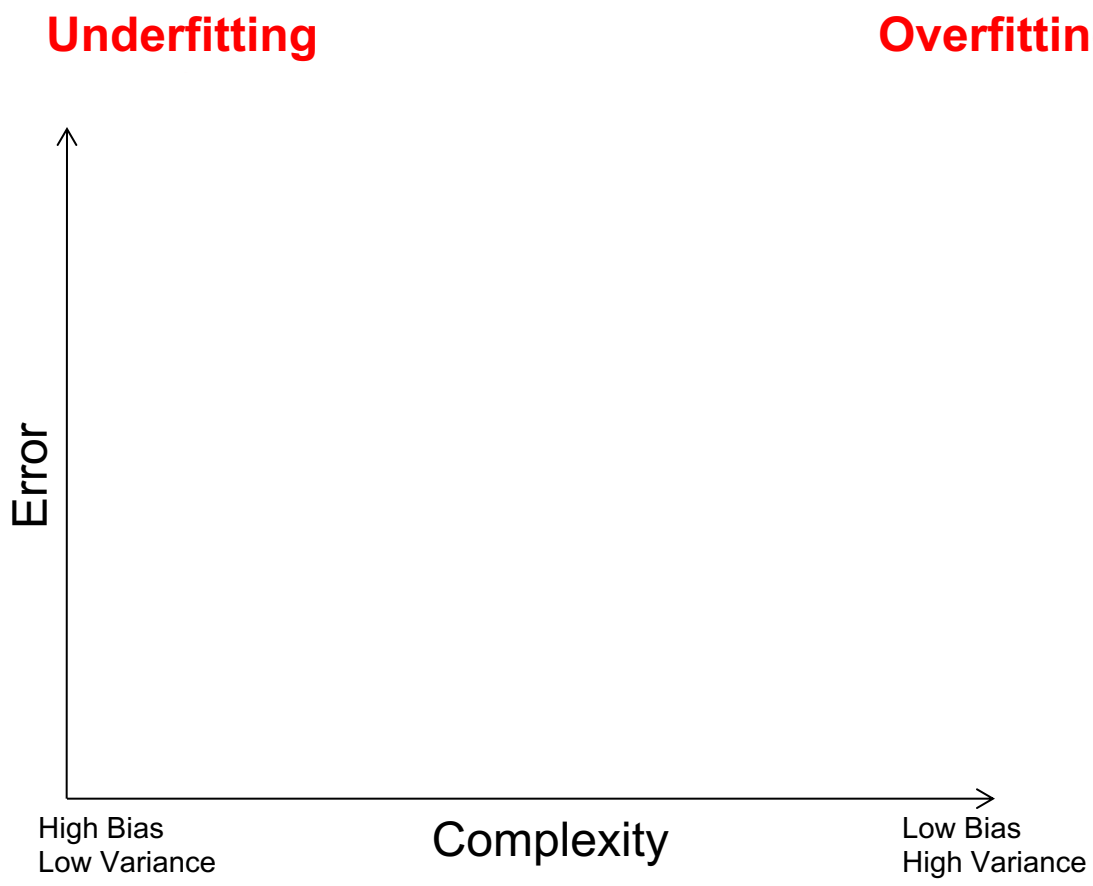
Adapted from Chris Bishop

Regularization: E_{RMS} vs. $\ln \lambda$

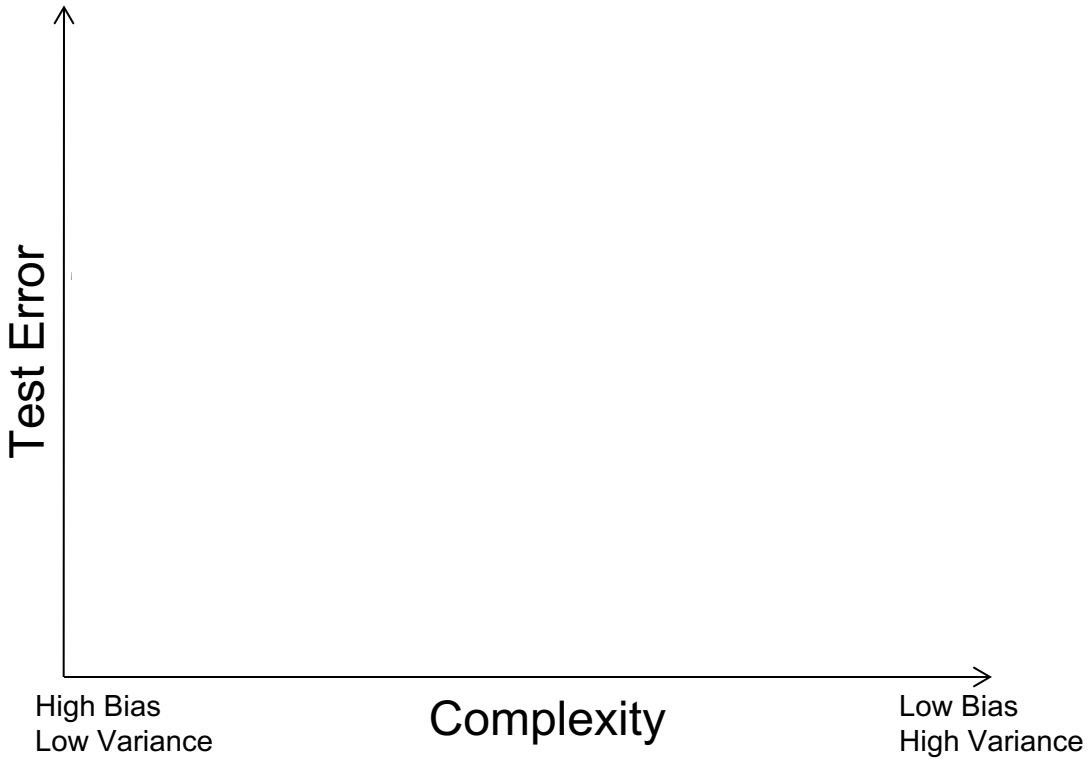


Slide credit: Chris Bishop

Training vs test error



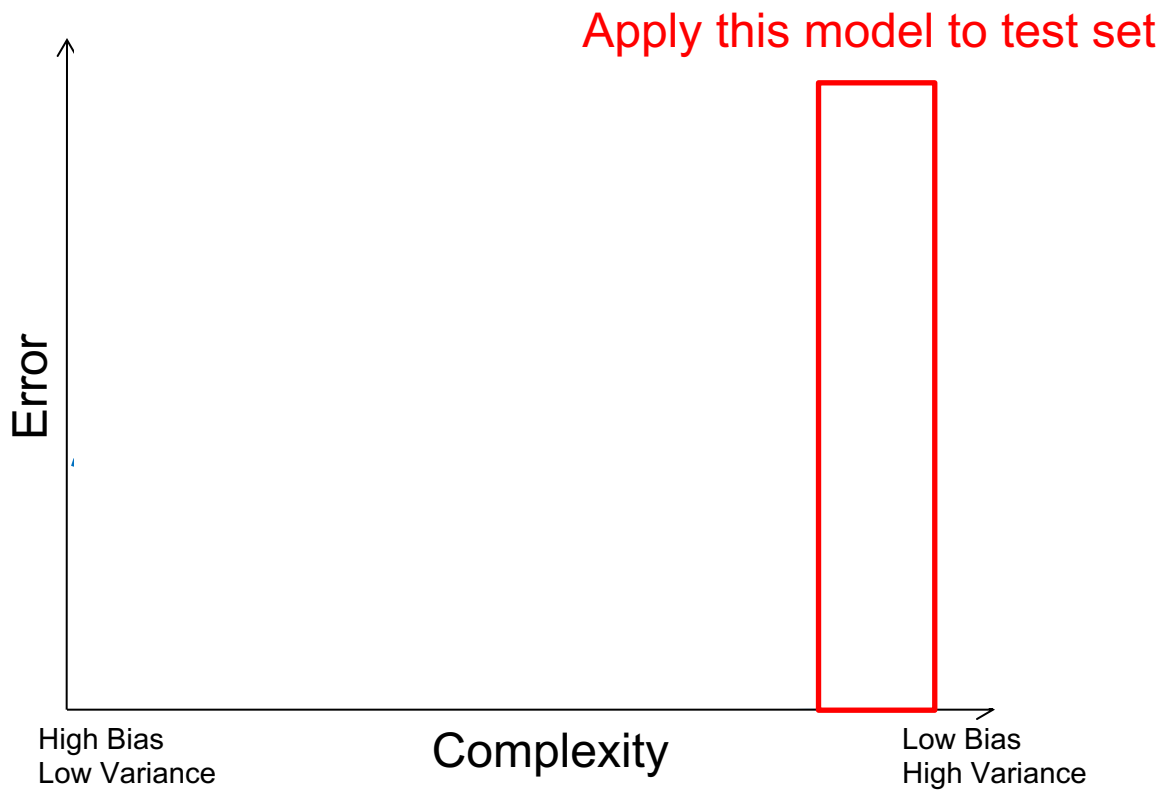
The effect of training set size



Slide credit: D. Hoiem

Choosing the trade-off between bias and variance

- Need validation set (separate from the test set)



Summary of generalization

- Try simple classifiers (models) first
- Better to have smart features and simple classifiers than simple features and smart classifiers
- Use increasingly powerful classifiers with more training data
- As an additional technique for reducing variance, try regularizing the parameters

Slide credit: D. Hoiem

Linear algebra review

Vectors and Matrices

- Vectors and matrices are just collections of ordered numbers that represent something: movements in space, scaling factors, word counts, movie ratings, pixel brightnesses, etc.
- We'll define some common uses and standard operations on them.

Vector

- A column vector $\mathbf{v} \in \mathbb{R}^{n \times 1}$ where

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

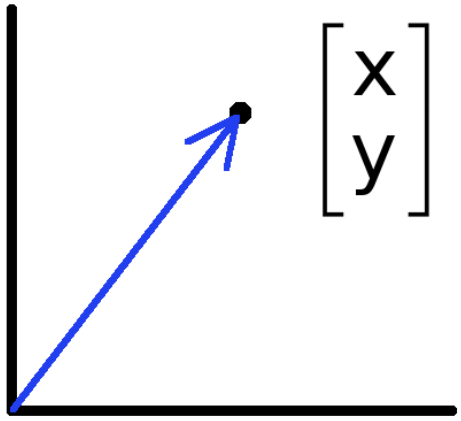
- A row vector $\mathbf{v}^T \in \mathbb{R}^{1 \times n}$ where

$$\mathbf{v}^T = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$

T denotes the transpose operation

- You need to keep track of orientation

Vectors have two main uses



- Vectors can represent an offset in 2D or 3D space
- Points are just vectors from the origin

- Data can also be treated as a vector
- Such vectors don't have a geometric interpretation, but calculations like "distance" still have value

Matrix

- A matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is an array of numbers with size $m \downarrow$ by $n \rightarrow$, i.e. m rows and n columns.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & & & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

- If $m = n$, we say that $\mathbf{A}$ is square.

Matrix Operations

- Addition

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} a+1 & b+2 \\ c+3 & d+4 \end{bmatrix}$$

  - Can only add a matrix with matching dimensions, or a scalar.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + 7 = \begin{bmatrix} a+7 & b+7 \\ c+7 & d+7 \end{bmatrix}$$

- Scaling

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times 3 = \begin{bmatrix} 3a & 3b \\ 3c & 3d \end{bmatrix}$$

Different types of product

- $\boldsymbol{x}$, $\boldsymbol{y}$ = column vectors (nx1)
- $\boldsymbol{X}$, $\boldsymbol{Y}$ = matrices (mxn)
- x, y = scalars (1x1)

- $\boldsymbol{x}^T \boldsymbol{y} = \boldsymbol{x} \cdot \boldsymbol{y}$ = inner product (1xn x nx1 = scalar)
- $\boldsymbol{x} \otimes \boldsymbol{y} = \boldsymbol{x}\, \boldsymbol{y}^T$ = outer product (nx1 x 1xn = matrix)

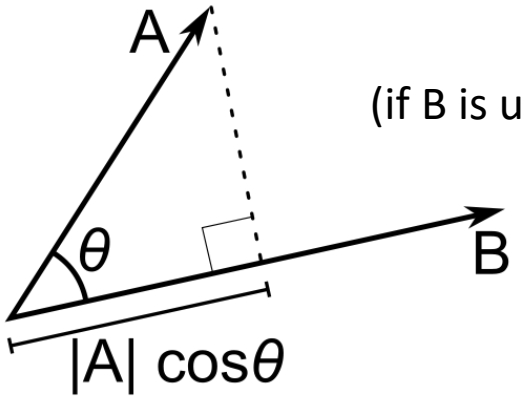
- $\boldsymbol{X} * \boldsymbol{Y}$ = matrix product
- $\boldsymbol{X} .* \boldsymbol{Y}$ = element-wise product

Inner Product

- Multiply corresponding entries of two vectors and add up the result

$$\mathbf{x}^T\mathbf{y} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i \qquad (\text{scalar})$$

- x·y is also |x||y|Cos(*angle between x and y*)
- If B is a unit vector, then A·B gives the length of A which lies in the direction of B (projection)



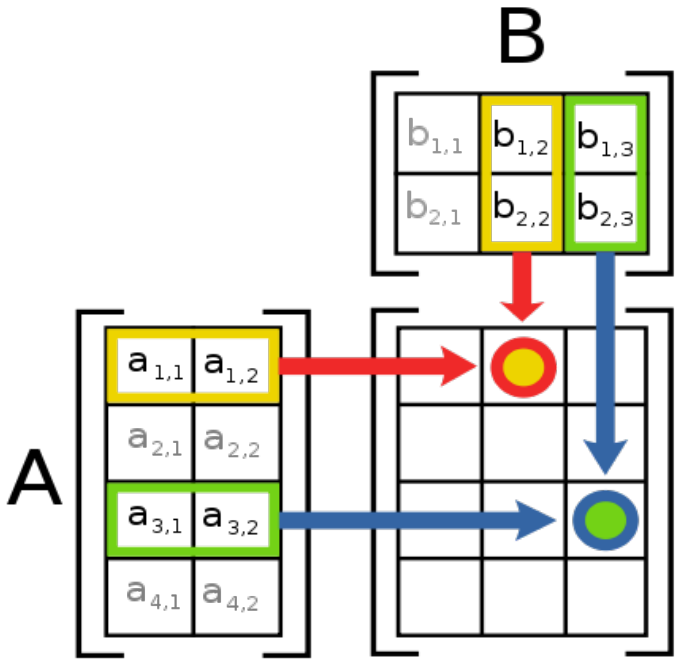
(if B is unit-length hence norm is 1)

Matrix Multiplication

- Let X be an *a*x*b* matrix, Y be an *b*x*c* matrix
- Then Z = X*Y is an *a*x*c* matrix
- Second dimension of first matrix, and first dimension of first matrix have to be the same, for matrix multiplication to be possible

Matrix Multiplication

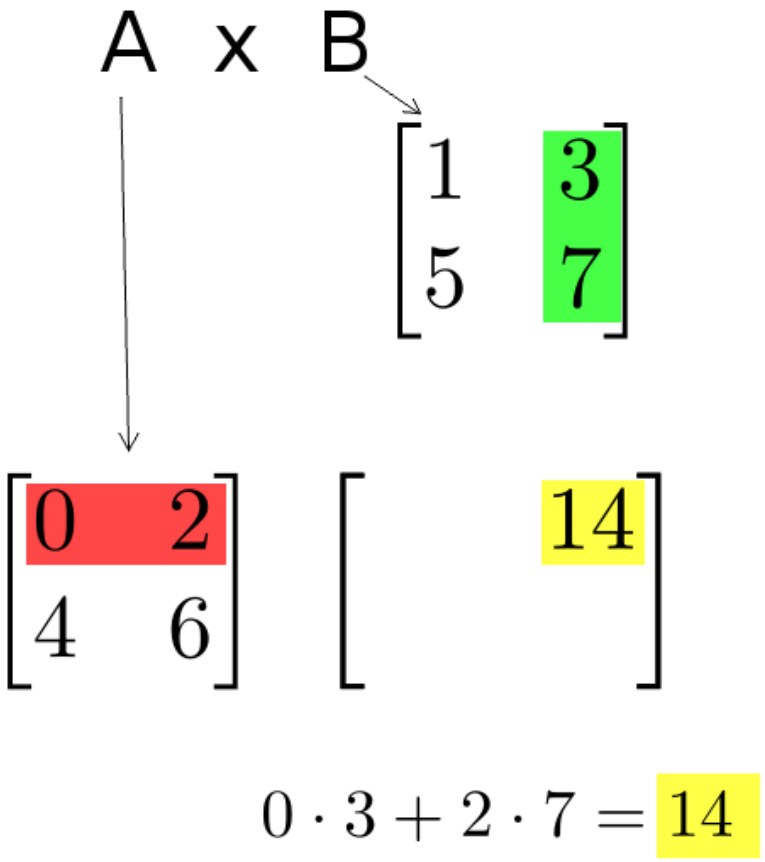
- The product AB is:



- Each entry in the result is (that row of A) dot product with (that column of B)

Matrix Multiplication

- Example:



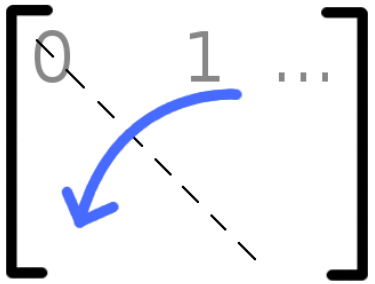
– Each entry of the matrix product is made by taking the dot product of the corresponding row in the left matrix, with the corresponding column in the right one.

Matrix Operation Properties

- Matrix addition is commutative and associative
  - A + B = B + A
  - A + (B + C) = (A + B) + C
- Matrix multiplication is associative and distributive but *not* commutative
  - A(B*C) = (A*B)C
  - A(B + C) = A*B + A*C
  - A*B != B*A

Matrix Operations

- Transpose – flip matrix, so row 1 becomes column 1



$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}^T = \begin{bmatrix} 0 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix}$$

- A useful identity:

$$(ABC)^T = C^T B^T A^T$$

Inverse

- Given a matrix **A**, its inverse $\mathbf{A^{-1}}$ is a matrix such that $\mathbf{AA^{-1} = A^{-1}A = I}$

- E.g. $\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$

- Inverse does not always exist. If $\mathbf{A^{-1}}$ exists, **A** is *invertible* or *non-singular*. Otherwise, it's *singular*.

The following are properties of the inverse; all assume that $A, B \in \mathbb{R}^{n \times n}$ are non-singular:

- $(A^{-1})^{-1} = A$
- $(AB)^{-1} = B^{-1}A^{-1}$
- $(A^{-1})^T = (A^T)^{-1}$. For this reason this matrix is often denoted A^{-T}.

Special Matrices

- Identity matrix **`I`**
  - Square matrix, 1's along diagonal, 0's elsewhere
  - **`I`** · [another matrix] = [that matrix]

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Diagonal matrix
  - Square matrix with numbers along diagonal, 0's elsewhere
  - A diagonal · [another matrix] scales the rows of that matrix

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}$$

Special Matrices

- Symmetric matrix $\mathbf{A}^T = \mathbf{A}$

$$\begin{bmatrix} 1 & 2 & 5 \\ 2 & 1 & 7 \\ 5 & 7 & 1 \end{bmatrix}$$

- Orthogonal matrix (the inverse of an orthogonal matrix is its transpose)

$$U^T U = I = U U^T$$

Another nice property of orthogonal matrices is that operating on a vector with an orthogonal matrix will not change its Euclidean norm, i.e.,

$$\|Ux\|_2 = \|x\|_2$$

for any $x \in \mathbb{R}^n$, $U \in \mathbb{R}^{n \times n}$ orthogonal.

Norms (A measure of the "length" of the vector)

- L1 norm: $\|x\|_1 = \sum_{i=1}^{n} |x_i|$
- L2 norm: $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}.$
- L^p norm

(for real numbers $p \geq 1$) $\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$

- L_∞ $\|x\|_\infty = \max_i |x_i|.$

System of Linear Equations

$$AX = B$$

$$A = \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

```
>> X = np.linalg.solve(A, B)
>> X
[1.0000,-0.5000]
```

Matrix Rank

- Column/row rank

$$\text{col-rank}(\mathbf{A}) = \text{ the maximum number of linearly independent column vectors of } \mathbf{A}$$
$$\text{row-rank}(\mathbf{A}) = \text{ the maximum number of linearly independent row vectors of } \mathbf{A}$$

- Column rank always equals row rank
- Matrix rank $\text{rank}(\mathbf{A}) \triangleq \text{col-rank}(\mathbf{A}) = \text{row-rank}(\mathbf{A})$
- If a matrix is not full rank, inverse doesn't exist
  - Inverse also doesn't exist for non-square matrices

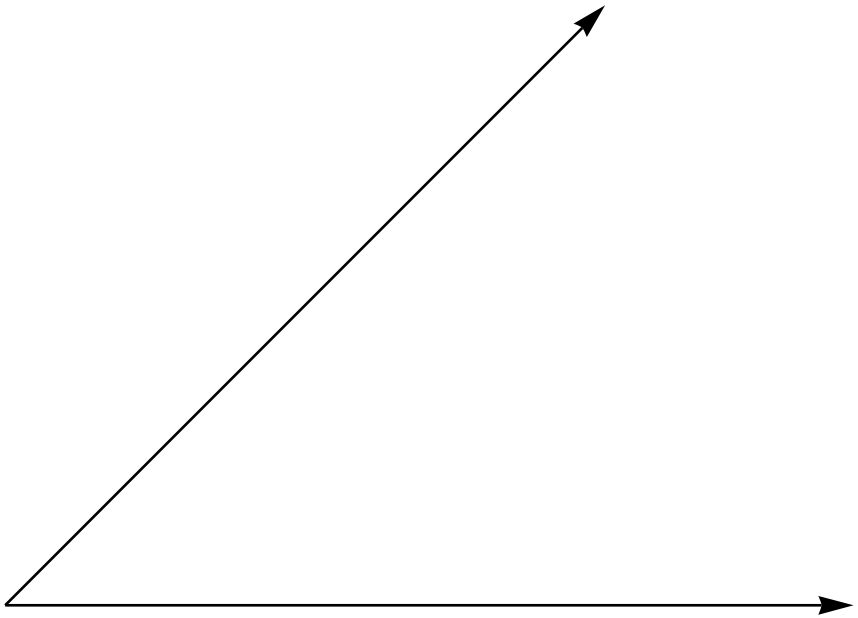
- For $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) \leq \min(m, n)$. If $\text{rank}(A) = \min(m, n)$, then A is said to be ***full rank***.
- For $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = \text{rank}(A^T)$.
- For $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))$.
- For $A, B \in \mathbb{R}^{m \times n}$, $\text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B)$.

Linear independence

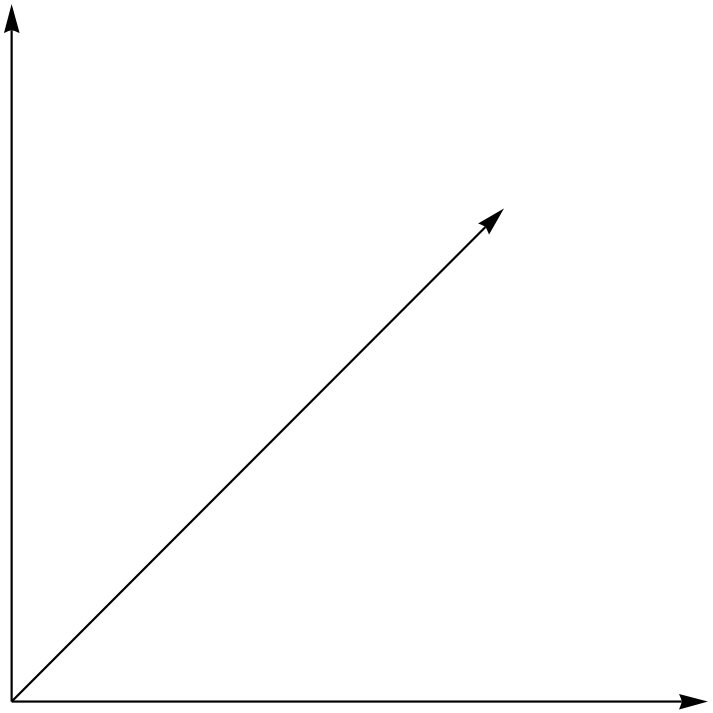
- Suppose we have a set of vectors $\mathbf{v_1}, \ldots, \mathbf{v_n}$
- If we can express $\mathbf{v}_1$ as a linear combination of the other vectors $\mathbf{v}_2 \ldots \mathbf{v}_n$, then $\mathbf{v}_1$ is linearly *dependent* on the other vectors.
  - The direction $\mathbf{v}_1$ can be expressed as a combination of the directions $\mathbf{v}_2 \ldots \mathbf{v}_n$. (E.g. $\mathbf{v}_1 = .7\ \mathbf{v}_2 - .5\ \mathbf{v}_4$)
- If no vector is linearly dependent on the rest of the set, the set is linearly *independent*.
  - Common case: a set of vectors $\mathbf{v_1}, \ldots, \mathbf{v_n}$ is always linearly independent if each vector is perpendicular to every other vector (and non-zero)

Linear independence

Linearly independent set



Not linearly independent



Span, Range, and Nullspace

The ***span*** of a set of vectors $\{x_1, x_2, \ldots x_n\}$ is the set of all vectors that can be expressed as a linear combination of $\{x_1, \ldots, x_n\}$. That is,

$$\text{span}(\{x_1, \ldots x_n\}) = \left\{ v : v = \sum_{i=1}^{n} \alpha_i x_i, \quad \alpha_i \in \mathbb{R} \right\}.$$

The ***range*** (sometimes also called the columnspace) of a matrix $A \in \mathbb{R}^{m \times n}$, denoted $\mathcal{R}(A)$, is the the span of the columns of A. In other words,

$$\mathcal{R}(A) = \{v \in \mathbb{R}^m : v = Ax, x \in \mathbb{R}^n\}.$$

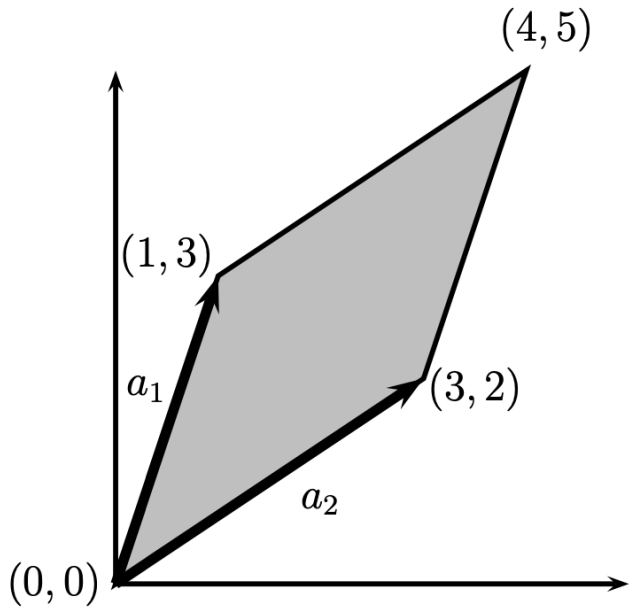
The ***nullspace*** of a matrix $A \in \mathbb{R}^{m \times n}$, denoted $\mathcal{N}(A)$ is the set of all vectors that equal 0 when multiplied by A, i.e.,

$$\mathcal{N}(A) = \{x \in \mathbb{R}^n : Ax = 0\}.$$

Matrix Determinant

$$A = \begin{bmatrix} 1 & 3 \\ 3 & 2 \end{bmatrix}$$

$$a_1 = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad a_2 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}.$$



The absolute value of the determinant, |det *A*| = 7, is the area of the parallelogram.

1D, determinant = length
2D, determinant = surface
3D, determinant = volume!

Matrix Determinant Properties

1. The determinant of the identity is 1, $|I| = 1$. (Geometrically, the volume of a unit hypercube is 1).

2. Given a matrix $A \in \mathbb{R}^{n \times n}$, if we multiply a single row in A by a scalar $t \in \mathbb{R}$, then the determinant of the new matrix is $t|A|$,

$$\left| \begin{bmatrix} — & t\, a_1^T & — \\ — & a_2^T & — \\ & \vdots & \\ — & a_m^T & — \end{bmatrix} \right| = t|A|.$$

   (Geometrically, multiplying one of the sides of the set S by a factor t causes the volume to increase by a factor t.)

3. If we exchange any two rows a_i^T and a_j^T of A, then the determinant of the new matrix is $-|A|$, for example

$$\left| \begin{bmatrix} — & a_2^T & — \\ — & a_1^T & — \\ & \vdots & \\ — & a_m^T & — \end{bmatrix} \right| = -|A|.$$

Matrix Determinant Properties

- For $A \in \mathbb{R}^{n \times n}$, $|A| = |A^T|$.
- For $A, B \in \mathbb{R}^{n \times n}$, $|AB| = |A||B|$.
- For $A \in \mathbb{R}^{n \times n}$, $|A| = 0$ if and only if A is singular (i.e., non-invertible). (If A is singular then it does not have full rank, and hence its columns are linearly dependent. In this case, the set S corresponds to a "flat sheet" within the n-dimensional space and hence has zero volume.)
- For $A \in \mathbb{R}^{n \times n}$ and A non-singular, $|A^{-1}| = 1/|A|$.

Eigenvalues and Eigenvectors

Given a square matrix $A \in \mathbb{R}^{n \times n}$, we say that $\lambda \in \mathbb{C}$ is an ***eigenvalue*** of A and $x \in \mathbb{C}^n$ is the corresponding ***eigenvector***[3] if

$$Ax = \lambda x, \quad x \neq 0.$$

Intuitively, this definition means that multiplying A by the vector x results in a new vector that points in the same direction as x, but scaled by a factor λ.

We can rewrite the equation above to state that (λ, x) is an eigenvalue-eigenvector pair of A if,

$$(\lambda I - A)x = 0, \quad x \neq 0.$$

But $(\lambda I - A)x = 0$ has a non-zero solution to x if and only if $(\lambda I - A)$ has a non-empty nullspace, which is only the case if $(\lambda I - A)$ is singular, i.e.,

$$|(\lambda I - A)| = 0.$$

Eigen Properties

- The trace of a A is equal to the sum of its eigenvalues,

$$\mathrm{tr}A = \sum_{i=1}^{n} \lambda_i.$$

- The determinant of A is equal to the product of its eigenvalues,

$$|A| = \prod_{i=1}^{n} \lambda_i.$$

- The rank of A is equal to the number of non-zero eigenvalues of A.
- If A is non-singular then $1/\lambda_i$ is an eigenvalue of A^{-1} with associated eigenvector x_i, i.e., $A^{-1}x_i = (1/\lambda_i)x_i$. (To prove this, take the eigenvector equation, $Ax_i = \lambda_i x_i$ and left-multiply each side by A^{-1}.)
- The eigenvalues of a diagonal matrix $D = \mathrm{diag}(d_1, \ldots d_n)$ are just the diagonal entries $d_1, \ldots d_n$.

Singular Value Decomposition (SVD)

- There are several computer algorithms that can "factor" a matrix, representing it as the product of some other matrices
- The most useful of these is the Singular Value Decomposition
- Represents any matrix **A** as a product of three matrices: $\mathbf{U\Sigma V^T}$

Singular Value Decomposition (SVD)

$$\mathbf{U\Sigma V^T = A}$$

- Where **U** and **V** are rotation matrices, and **Σ** is a scaling matrix. For example:

$$\underset{U}{\begin{bmatrix} -.40 & .916 \\ .916 & .40 \end{bmatrix}} \times \underset{\Sigma}{\begin{bmatrix} 5.39 & 0 \\ 0 & 3.154 \end{bmatrix}} \times \underset{V^T}{\begin{bmatrix} -.05 & .999 \\ .999 & .05 \end{bmatrix}} = \underset{A}{\begin{bmatrix} 3 & -2 \\ 1 & 5 \end{bmatrix}}$$

Singular Value Decomposition (SVD)

- In general, if **A** is *m* x *n*, then **U** will be *m* x *m*, **Σ** will be *m* x *n*, and $\mathbf{V}^{\mathbf{T}}$ will be *n* x *n*.

$$\underset{U}{\begin{bmatrix} -.39 & -.92 \\ -.92 & .39 \end{bmatrix}} \times \underset{\Sigma}{\begin{bmatrix} 9.51 & 0 & 0 \\ 0 & .77 & 0 \end{bmatrix}} \times \underset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} = \underset{A}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}}$$

Singular Value Decomposition (SVD)

- **U** and **V** are always rotation matrices.
  - Geometric rotation may not be an applicable concept, depending on the matrix. So we call them “unitary” matrices – each column is a unit vector.
- **Σ** is a diagonal matrix
  - The number of nonzero entries = rank of **A**
  - The algorithm always sorts the entries high to low

$$\underset{U}{\begin{bmatrix} -.39 & -.92 \\ -.92 & .39 \end{bmatrix}} \times \underset{\Sigma}{\begin{bmatrix} 9.51 & 0 & 0 \\ 0 & .77 & 0 \end{bmatrix}} \times \underset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} = \underset{A}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}}$$

Singular Value Decomposition (SVD)

$$\mathbf{M = U\Sigma V^T}$$

Illustration from Wikipedia

SVD is crucial

- PCA
- Dimension reduction
- Computer vision
- Image processing

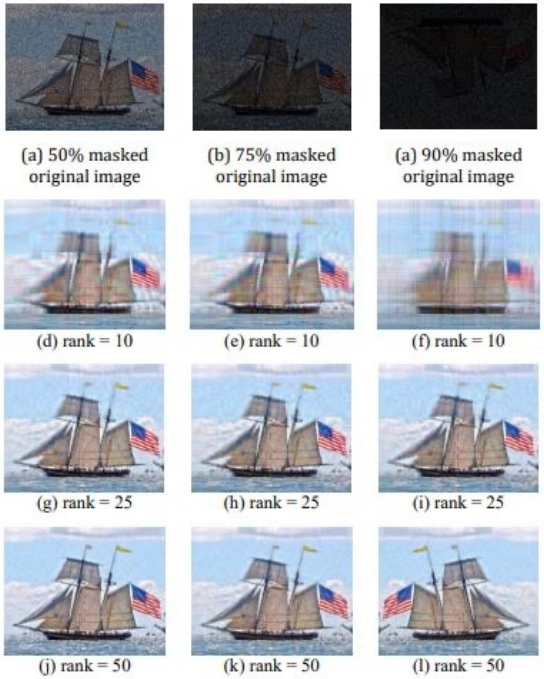
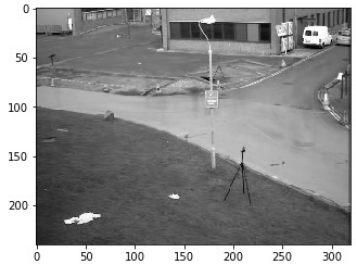
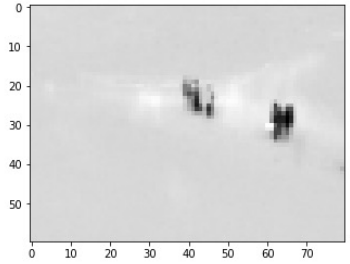


Figure 4. Recovery for a 960 × 1200 RGB image in easy, medium and hard mode

BACKGROUND REMOVAL - RANK 2 APPOXIMATION



LOW RANK BACKGROUND



FOREGROUND

Calculus review

Differentiation

The derivative provides us information about the rate of change of a function.

The derivative of a function is also a function.

Example:

The derivative of the rate function is the acceleration function.

Derivative = rate of change

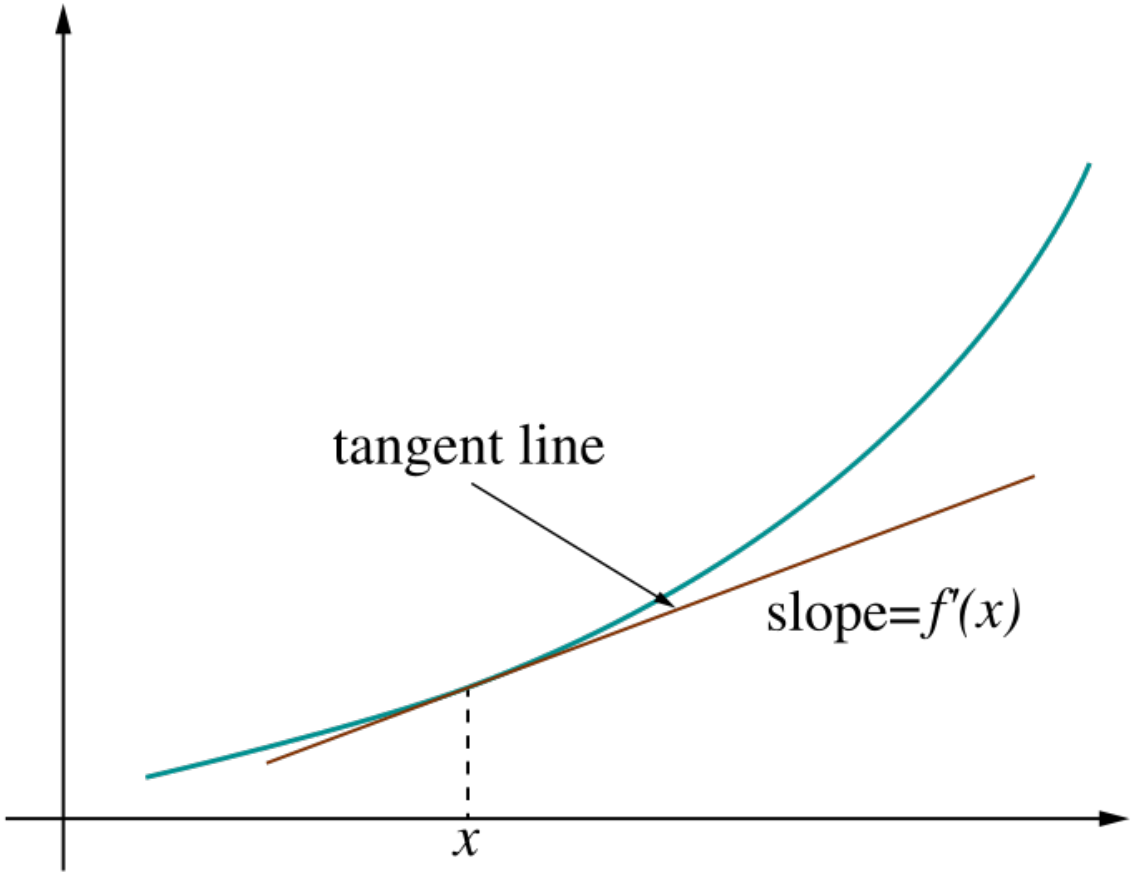


Image: Wikipedia

Derivative = rate of change

- Linear function y = mx + b
- Slope $m = \dfrac{\text{change in } y}{\text{change in } x} = \dfrac{\Delta y}{\Delta x},$

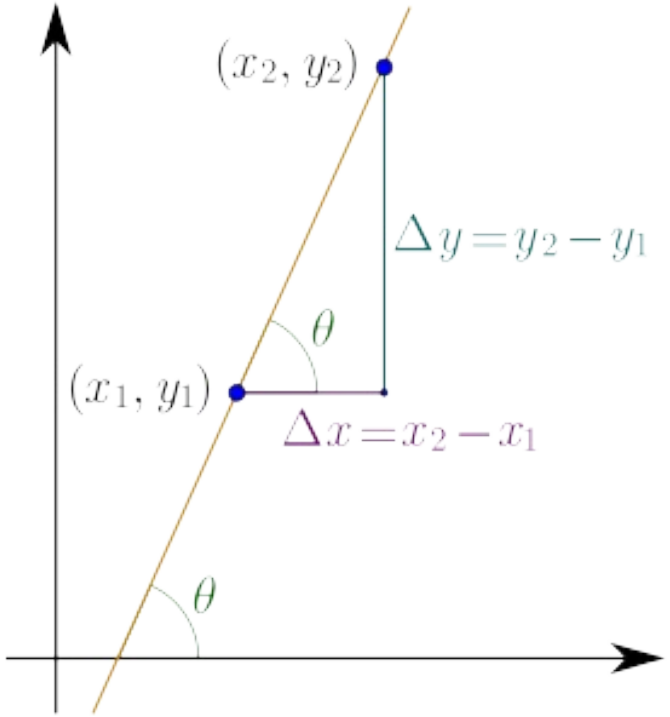


Image: Wikipedia

Ways to Write the Derivative

Given the function f(x), we can write its derivative in the following ways:

- f'(x)
- $\frac{d}{dx}$f(x)

The derivative of x is commonly written dx.

Differentiation Formulas

The following are common differentiation formulas:

- The derivative of a constant is 0.

$$\frac{d}{du}c = 0$$

- The derivative of a sum is the sum of the derivatives.

$$\frac{d}{du}(f(u)+g(u)) = f'(u)+g'(u)$$

Examples

- The derivative of a constant is 0.

$$\frac{d}{du}7 =$$

- The derivative of a sum is the sum of the derivatives.

$$\frac{d}{dt}(t+4) =$$

More Formulas

- The derivative of *u* to a constant power:

$$\frac{d}{du}u^n = n * u^{n-1}du$$

- The derivative of *e*:

$$\frac{d}{du}e^u = e^u du$$

- The derivative of *log*:

$$\frac{d}{du}\log(u) = \frac{1}{u}du$$

More Examples

- The derivative of *u* to a constant power:

$$\frac{d}{dx} 3x^3 =$$

- The derivative of *e*:

$$\frac{d}{dy} e^{4y} =$$

- The derivative of *log*:

$$\frac{d}{dx} 3\log(x) =$$

Product and Quotient

The product rule and quotient rules are commonly used in differentiation.

- Product rule:

$$\frac{d}{du}(f(u)*g(u)) = f(u)g'(u) + g(u)f'(u)$$

- Quotient rule:

$$\frac{d}{du}\left(\frac{f(u)}{g(u)}\right) = \frac{g(u)f'(u) - f(u)g'(u)}{(g(u))^2}$$

Chain Rule

The chain rule allows you to combine any of the differentiation rules we have already covered.

- First, do the derivative of the outside and then do the derivative of the inside.

$$\frac{d}{du} f(g(u)) = f'(g(u)) \times g'(u) \times du$$

Matrix Calculus: The Gradient

$$\nabla_A f(A) \in \mathbb{R}^{m \times n} = \begin{bmatrix} \frac{\partial f(A)}{\partial A_{11}} & \frac{\partial f(A)}{\partial A_{12}} & \cdots & \frac{\partial f(A)}{\partial A_{1n}} \\ \frac{\partial f(A)}{\partial A_{21}} & \frac{\partial f(A)}{\partial A_{22}} & \cdots & \frac{\partial f(A)}{\partial A_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(A)}{\partial A_{m1}} & \frac{\partial f(A)}{\partial A_{m2}} & \cdots & \frac{\partial f(A)}{\partial A_{mn}} \end{bmatrix}$$

Matrix Calculus: The Hessian

$$\nabla_x^2 f(x) \in \mathbb{R}^{n \times n} = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

Hessian is always symmetric!

Matrix Calculus: Least Squares

Let's apply the equations we obtained in the last section to derive the least squares equations. Suppose we are given matrices $A \in \mathbb{R}^{m \times n}$ (for simplicity we assume A is full rank) and a vector $b \in \mathbb{R}^m$ such that $b \notin \mathcal{R}(A)$. In this situation we will not be able to find a vector $x \in \mathbb{R}^n$, such that $Ax = b$, so instead we want to find a vector x such that Ax is as close as possible to b, as measured by the square of the Euclidean norm $\|Ax - b\|_2^2$.

Using the fact that $\|x\|_2^2 = x^T x$, we have

$$\begin{aligned} \|Ax - b\|_2^2 &= (Ax - b)^T (Ax - b) \\ &= x^T A^T A x - 2b^T A x + b^T b \end{aligned}$$

Taking the gradient with respect to x we have, and using the properties we derived in the previous section

$$\begin{aligned} \nabla_x (x^T A^T A x - 2b^T A x + b^T b) &= \nabla_x x^T A^T A x - \nabla_x 2b^T A x + \nabla_x b^T b \\ &= 2A^T A x - 2A^T b \end{aligned}$$

Setting this last expression equal to zero and solving for x gives the normal equations

$$x = (A^T A)^{-1} A^T b$$

which is the same as what we derived in class.

Matrix Calculus: Eigenvalues as Optimization

Finally, we use matrix calculus to solve an optimization problem in a way that leads directly to eigenvalue/eigenvector analysis. Consider the following, equality constrained optimization problem:

$$\max_{x \in \mathbb{R}^n} \; x^T A x \quad \text{subject to } \|x\|_2^2 = 1$$

for a symmetric matrix $A \in \mathbb{S}^n$. A standard way of solving optimization problems with equality constraints is by forming the ***Lagrangian***, an objective function that includes the equality constraints.[5] The Lagrangian in this case can be given by

$$\mathcal{L}(x, \lambda) = x^T A x - \lambda x^T x$$

where λ is called the Lagrange multiplier associated with the equality constraint. It can be established that for x^* to be a optimal point to the problem, the gradient of the Lagrangian has to be zero at x^* (this is not the only condition, but it is required). That is,

$$\nabla_x \mathcal{L}(x, \lambda) = \nabla_x (x^T A x - \lambda x^T x) = 2A^T x - 2\lambda x = 0.$$

Notice that this is just the linear equation $Ax = \lambda x$. This shows that the only points which can possibly maximize (or minimize) $x^T A x$ assuming $x^T x = 1$ are the eigenvectors of A.