Deep Learning for Computer Vision

Spring 2019

http://vllab.ee.ntu.edu.tw/dlcv.html (primary)

https://ceiba.ntu.edu.tw/1072CommE5052_ (grade, etc.)

FB: DLCV Spring 2019

Yu-Chiang Frank Wang 王鈺強, Associate Professor

Dept. Electrical Engineering, National Taiwan University

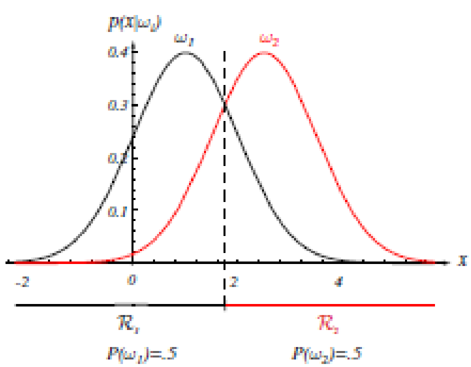
2019/02/27

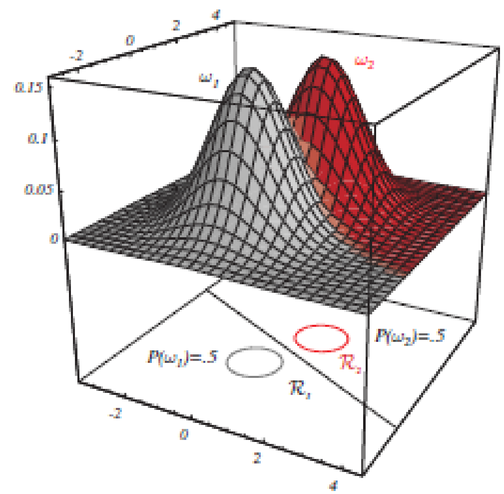
Tight (yet tentative) Schedule

Week	Date	Topic	Remarks
0	2/20	Course Logistics + Intro to Computer Vision	
1	2/27	**Machine Learning 101**	
2	3/06	**Image Representation: From Recognition to Tracking**	**HW #1 out**
3	3/13	Intro to Neural Networks + CNN (I)	
4	3/20	Intro to Neural Networks + CNN (II) Tutorial on Python, GitHub, etc.	**HW #1 due**
5	3/27	Detection & Segmentation	HW #2 out
6	4/03	Spring Break!	
7	4/10	Generative Models	
8	4/17	Visualization and Understanding NNs	HW #3 out, HW #2 due
9	4/24	Transfer Learning for Visual Analysis	
10	5/01	Recurrent NNs and Seq-to-Seq Models (I)	Team Up for Final Projects
11	5/08	Guest Lecture	HW #4 out, HW #3 due
12	5/15	Recurrent NNs and Seq-to-Seq Models (I)	
13	5/22	Learning Beyond Images (2D/3D, depth, etc.)	
14	5/29	Deep Reinforcement Learning for Visual Apps	HW #4 due
15	6/05	Final Project Checkpoint	
16	6/12 or 6/19	Final Presentation	

What’s to Be Covered Today…

- From Probability to Bayes Decision Rule
- Brief Review of Linear Algebra & Linear System
- Unsupervised vs. Supervised Learning
    - Clustering & Dimension Reduction
    - Training, testing, & validation
    - Linear Classification





Bayesian Decision Theory

- Fundamental statistical approach to classification/detection tasks
- Example (for a 2-class scenario):
  - Let's see if a student would pass or fail the course of DLCV.
  - Define a probabilistic variable ω describe the case of pass or fail.
  - That is, $\omega = \omega_1$ for pass, and $\omega = \omega_2$ for fail.
- Prior Probability
  - The **a priori** or **prior** probability reflects the knowledge of how likely we expect a certain state of nature before observation.
  - $P(\omega = \omega_1)$ or simply $P(\omega_1)$ as the prior that the next student would pass DLCV.
  - The priors must exhibit *exclusivity* and *exhaustivity*, i.e.,

Prior Probability (cont'd)

- Equal priors
  - If we have *equal* numbers of students pass/fail DLCV, then the priors are equal; in other words, the priors are uniform.
- Decision rule based on priors only
  - If the only available info is the prior and the cost of any incorrect classification is equal, what is a reasonable decision rule?
  - Decide ω_1 if ;

    otherwise decide ω_2 .
  - What's the incorrect classification rate (or error rate) P_e?

Class-Conditional Probability Density (or Likelihood)

- The probability density function (PDF) for input/observation ***x*** given a state of nature ω is written as:

- Here's (hopefully) the hypothetical class-conditional densities reflecting the studying time of students who pass/fail DLCV.

Posterior Probability & Bayes Formula

- If we know the prior distribution and the class-conditional density, can we come up with a better decision rule?
- Posterior probability:
  - The probability of a certain state of nature ω given an observable $\boldsymbol{x}$.
- Bayes formula:

$$P\left(\omega_j, \boldsymbol{x}\right)$$

$$P(\omega_j|\boldsymbol{x})$$

And, we have $\sum_{j=1}^{C} P(\omega_j|\boldsymbol{x}) = 1$.

Decision Rule & Probability of Error

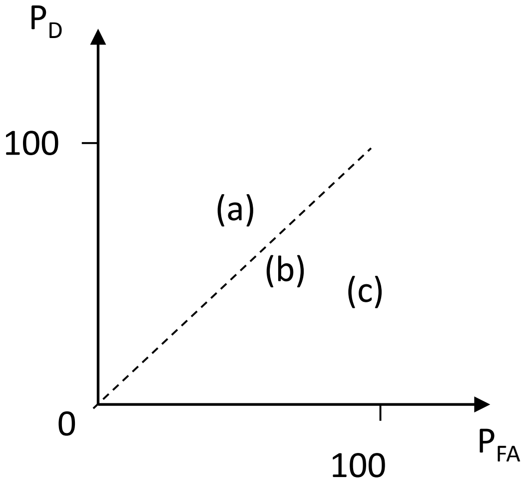
- For a given observable ***x***, the decision rule will be now based on:

- What's the probability of error P(error) (or P_e)?

From Bayes Decision Rule to Detection Theory

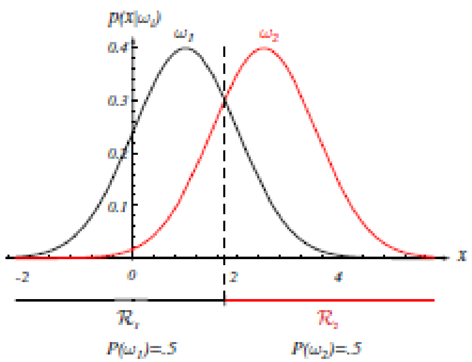
- Hit (or detection), false alarm, miss (or false reject), & correct rejection

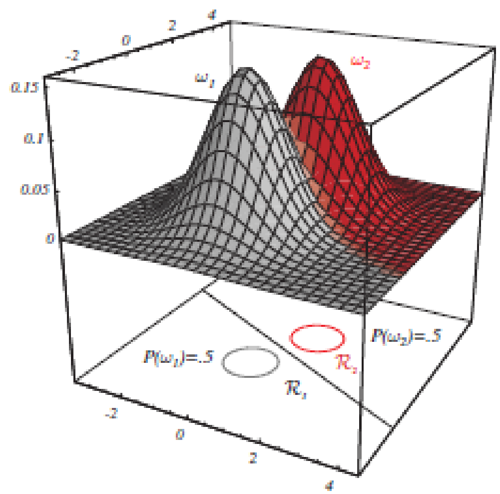
- Receiver Operating Characteristics (ROC)
  - To assess the effectiveness of the designed features/classifiers/detectors
  - False alarm (P_{FA}) vs. detection (P_D) rates
  - Which curve/line makes sense? (a), (b), or (c)?



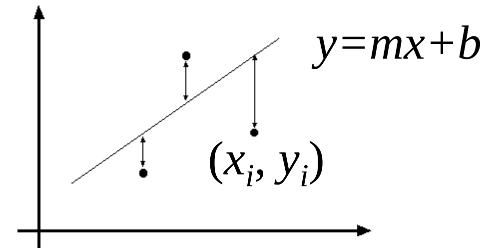
What's to Be Covered Today…

- From Probability to Bayes Decision Rule
- Brief Review of Linear Algebra & Linear System
- Unsupervised vs. Supervised Learning
    - Dimension Reduction & Clustering
    - Linear Classification
    - Training, testing, & validation

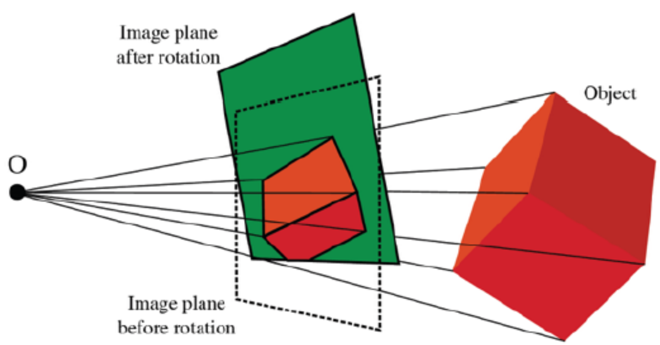


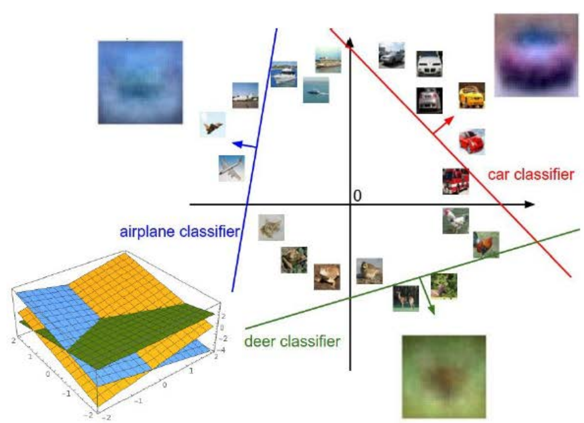


Why Review Linear Systems?



- Aren't DL models considered to be non-linear?
- Yes, but there are lots of things (e.g., problem setting, feature representation, regularization, etc.) in the fields of learning and vision starting from linear formulations.





$f(x,W) = Wx + b$

Array of **32x32x3** numbers (3072 numbers total)

Image credit: CMU 16720 / Stanford CS231n

Rank of a Matrix

- Consider **A** as a *m* x *n* matrix, the rank of matrix **A**, or rank(**A**), is determined as the maximum number of linearly independent row/column vectors.

  - Thus, we have rank(**A**) = rank($\mathbf{A}^T$) and rank(**A**) ≤ or ≥ (?) min(*m*, *n*).
- If rank(**A**) = min(*m*, *n*), **A** is a full-rank matrix.
- If *m* = *n* = rank(**A**) (i.e., **A** is a squared matrix and full-rank), what can we say about $\mathbf{A}^{-1}$?
- If *m* = *n* but rank(**A**) < m, what can we do to compute $\mathbf{A}^{-1}$ in practice?

Solutions to Linear Systems

- Let **A x** = **b**, where **A** is a *m* x *n* matrix, **x** is *n* x 1 vector (to be determined), and **b** is a *m* x 1 vector (observation).
- We consider **A x** = **b** as a linear system with *m* equations & *n* unknowns.

- If m = n & rank(**A**) = m, we can solve **x** by Gauss Elimination, or simply **x** = $\mathbf{A}^{-1}\mathbf{b}$. ($\mathbf{A}^{-1}$ is the inverse matrix of **A**.)

Pseudoinverse of a Matrix (I)

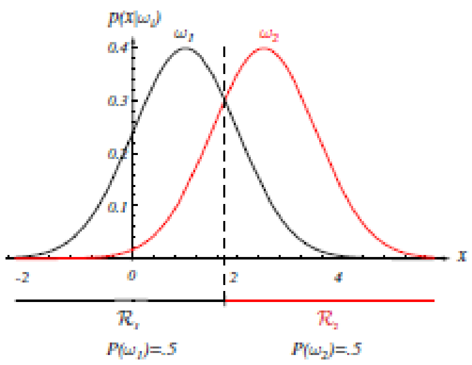
- Given **A x** = **b**, if m < n & rank(**A**) = m…
  - We have more # of unknowns than # of equations, i.e., underdetermined.
  - Which ***x*** is typically desirable?
    - Ever heard of / used the optimization technique of Lagrange multipliers?

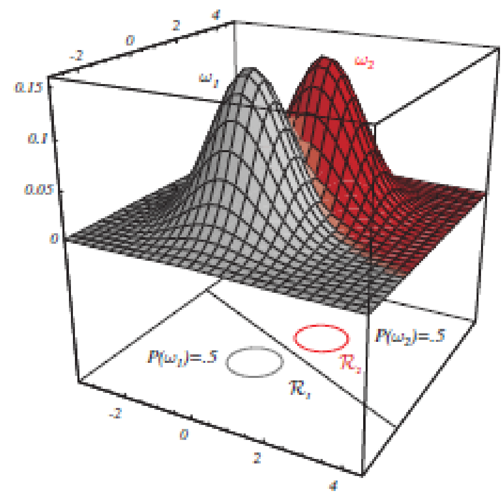
Pseudoinverse of a Matrix (II)

- Given **A x** = **b**, if m > n & rank(**A**) = n…
  - We have more # of equations than # of unknowns, i.e., overdetermined.
  - How to get a desirable **x**?

What's to Be Covered Today…

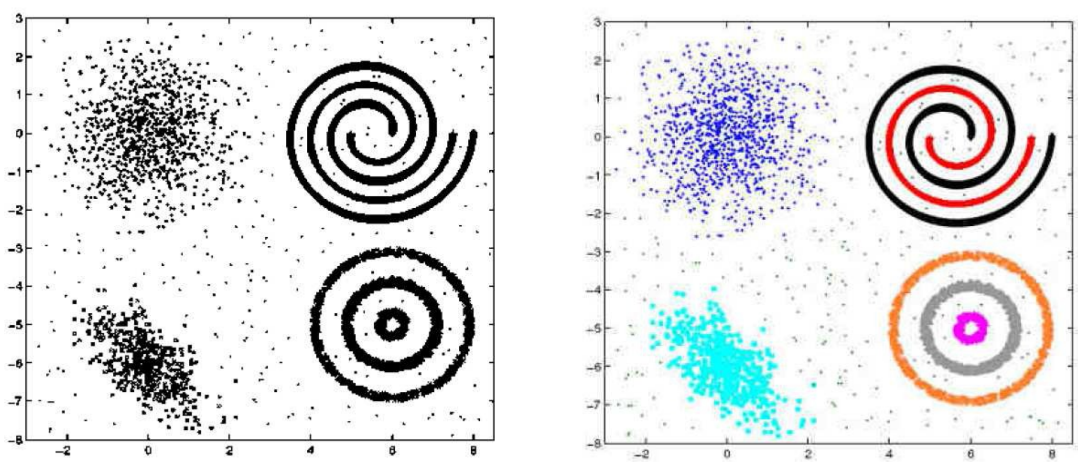
- From Probability to Bayes Decision Rule
- Brief Review of Linear Algebra & Linear System
- Unsupervised vs. Supervised Learning
  - Clustering & Dimension Reduction
  - Training, testing, & validation
  - Linear Classification





Clustering

- Clustering is an unsupervised algorithm.
  - Given:
    a set of N unlabeled instances $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; # of clusters K
  - Goal: group the samples into K partitions
- Remarks:
  - High within-cluster (intra-cluster) similarity
  - Low between-cluster (inter-cluster) similarity
  - But…how to determine a proper similarity measure?



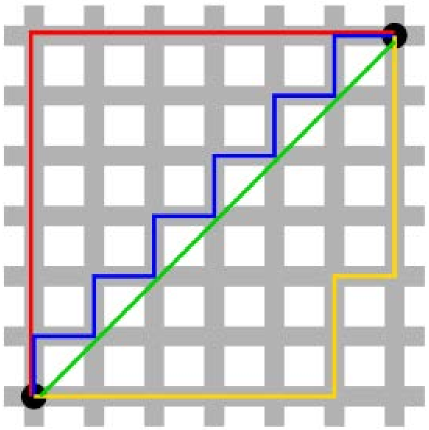
(a) Input data (b) Desired clustering

Similarity is NOT Always Objective…

Clustering (cont'd)

- Similarity:
  - A key component/measure to perform data clustering
  - Inversely proportional to distance
  - Example distance merrics:
    - Euclidean distance (L2 norm): $d(x, z) = \|x - z\|_2 = \sqrt{\sum_{i=1}^{D}(x_i - z_i)^2}$
    - Manhattan distance (L1 norm): $d(x, z) = \|x - z\|_1 = \sum_{i=1}^{D}|x_i - z_i|$

    - Note that *p*-norm of *x* is denoted as:



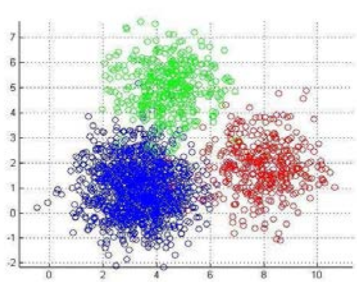
Clustering (cont'd)

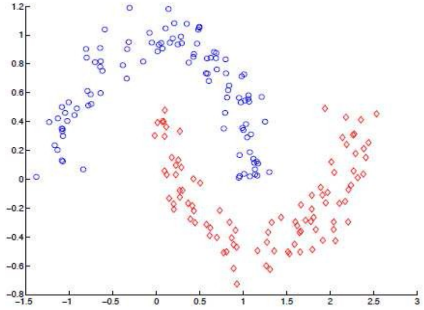
- Similarity:
  - A key component/measure to perform data clustering
  - Inversely proportional to distance
  - Example distance metrics:
    - Kernelized (non-linear) distance:

$$d(x,z) = \|\Phi(x) - \Phi(z)\|_2^2 = \|\Phi(x)\|_2^2 + \|\Phi(z)\|_2^2 - 2\Phi(x)^T\Phi(z)$$

    - Taking Gaussian kernel for example: $K(x,z) = \Phi(x)^T\Phi(z) = exp\left(-\frac{\|x-z\|_2^2}{2\sigma^2}\right)$, we have

      And, distance is more sensitive to larger/smaller σ. Why?
    - For example, L2 or kernelized distance metrics for the following two cases?



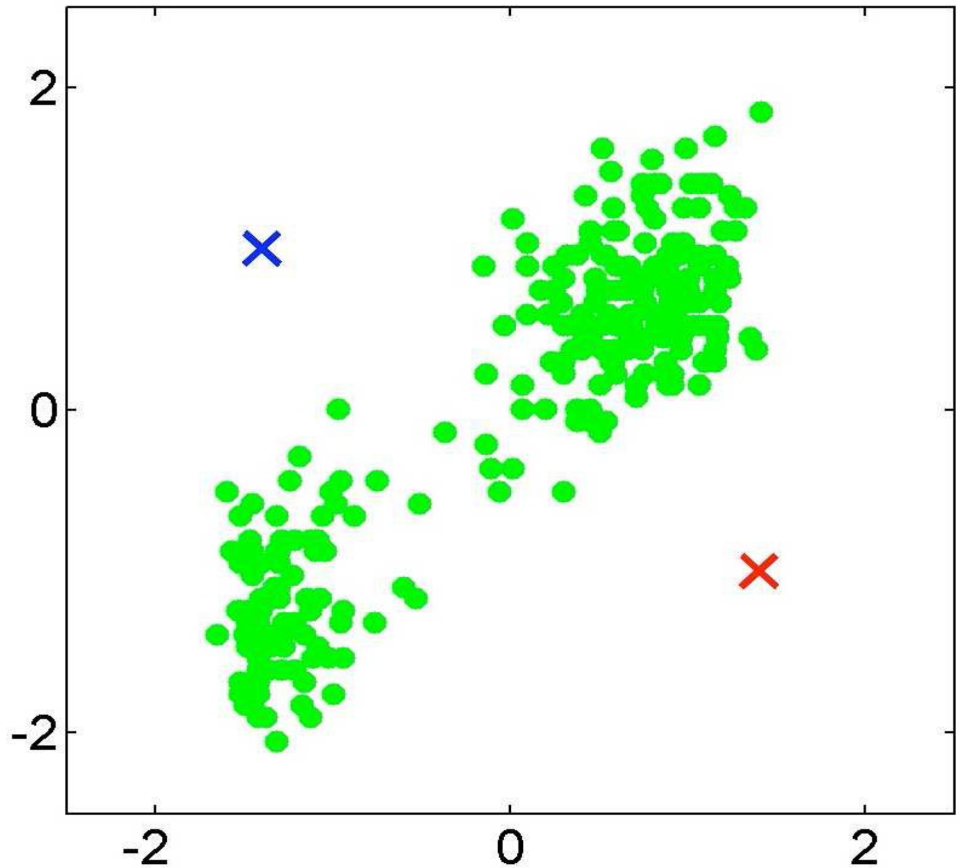


K-Means Clustering

- Input: N examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ($\boldsymbol{x}_n \in \mathrm{R}^D$); number of partitions K
- Initialize: K cluster centers $\mu_1, \ldots, \mu_K$. Several initialization options:
  - Randomly initialize $\mu_1, \ldots, \mu_K$ anywhere in R^D
  - Or, simply choose any K examples as the cluster centers
- Iterate:
  - Assign each of example $\boldsymbol{x}_\mathrm{n}$ to its closest cluster center
  - Recompute the new cluster centers μ_k (mean/centroid of the set C_k)
  - Repeat while not converge
- Possible convergence criteria:
  - Cluster centers do not change anymore
  - Max. number of iterations reached
- Output:
  - K clusters (with centers/means of each cluster)

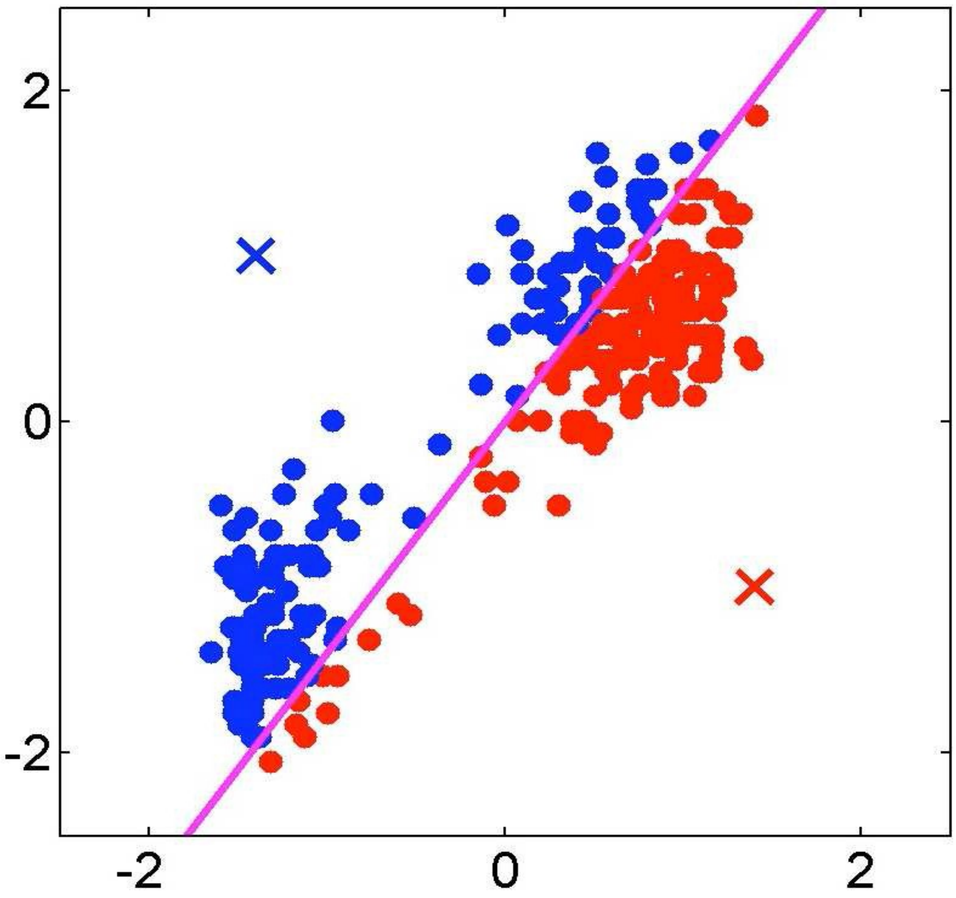
K-Means Clustering

- Example (K = 2): Initialization, iteration #1: pick cluster centers



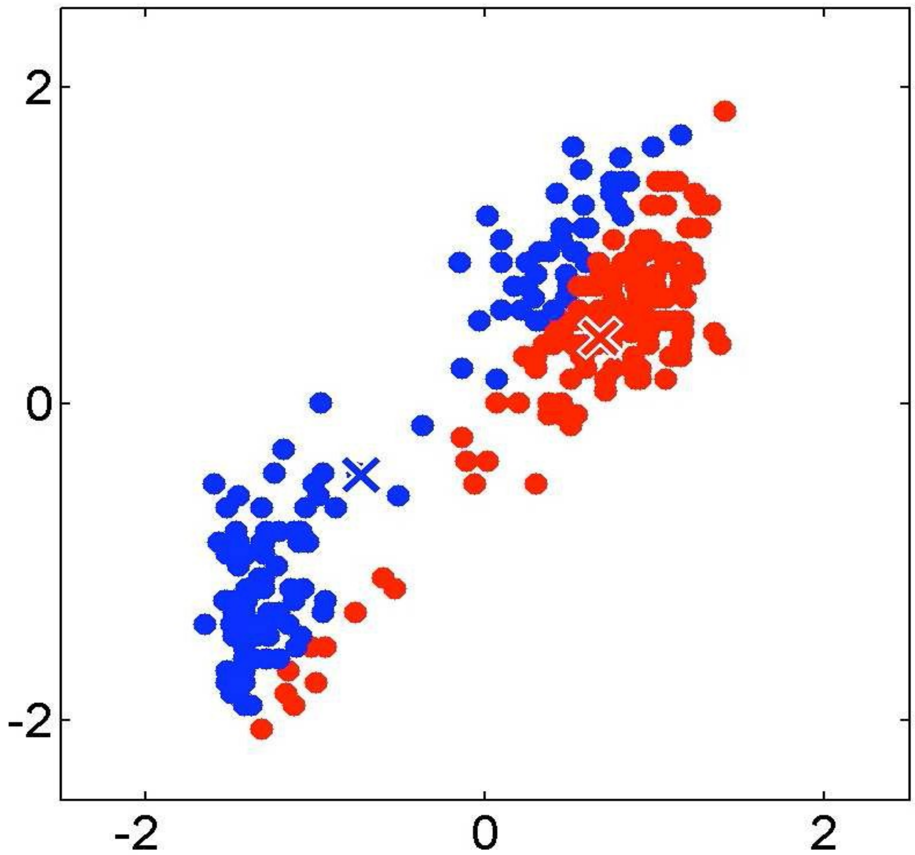
K-Means Clustering

- Example (K = 2): iteration #1-2, assign data to each cluster



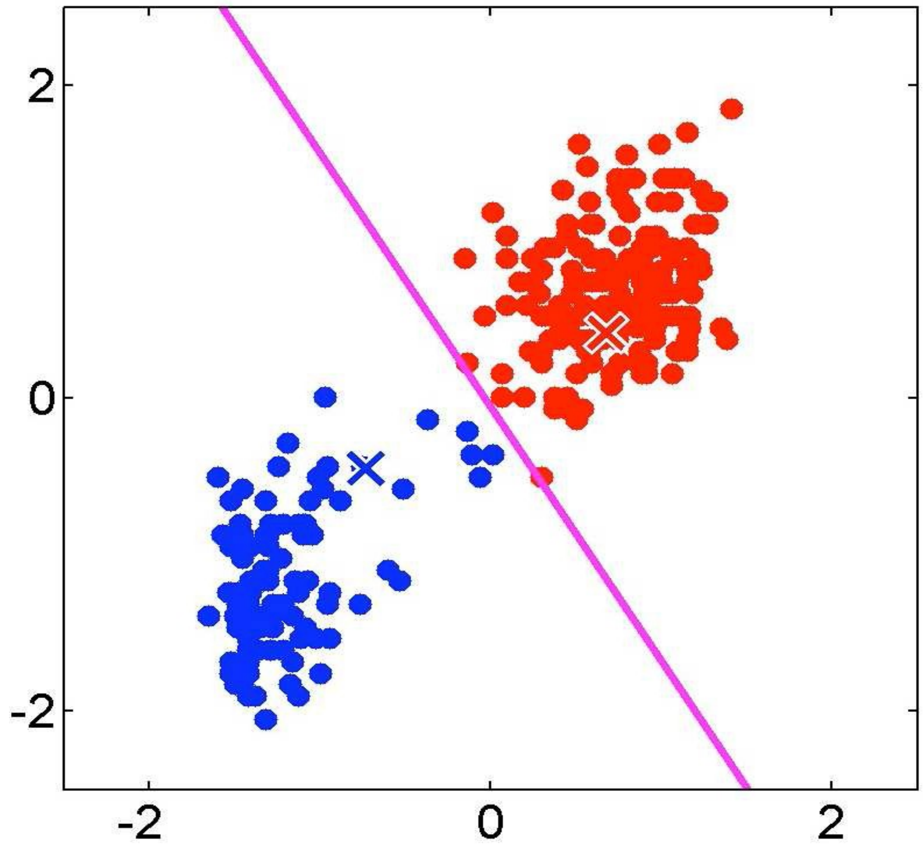
K-Means Clustering

- Example (K = 2): iteration #2-1, update cluster centers



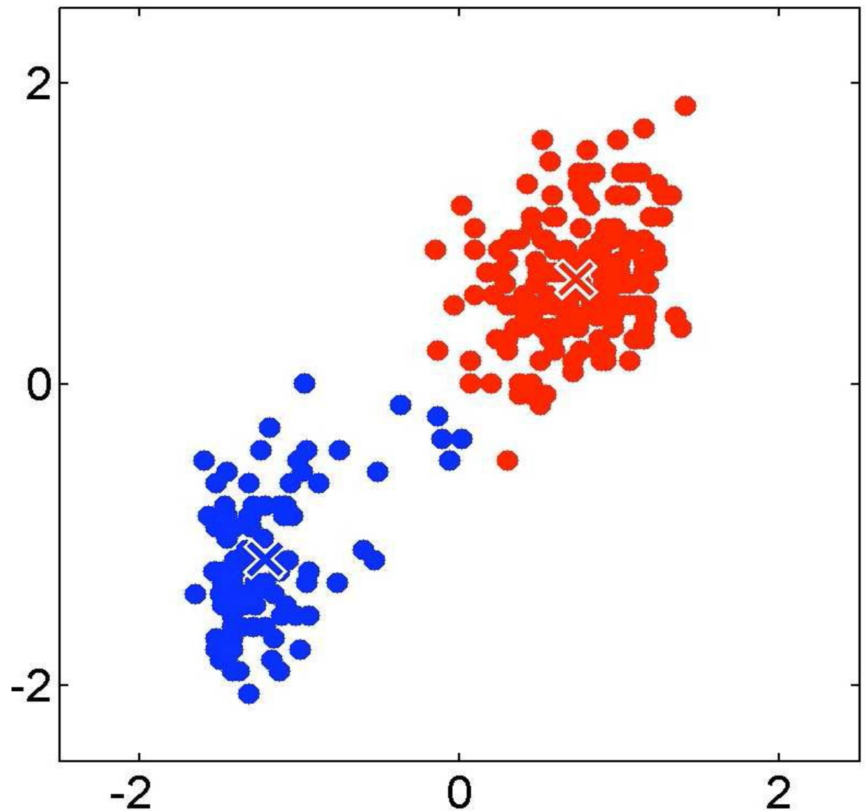
K-Means Clustering

- Example (K = 2): iteration #2, assign data to each cluster



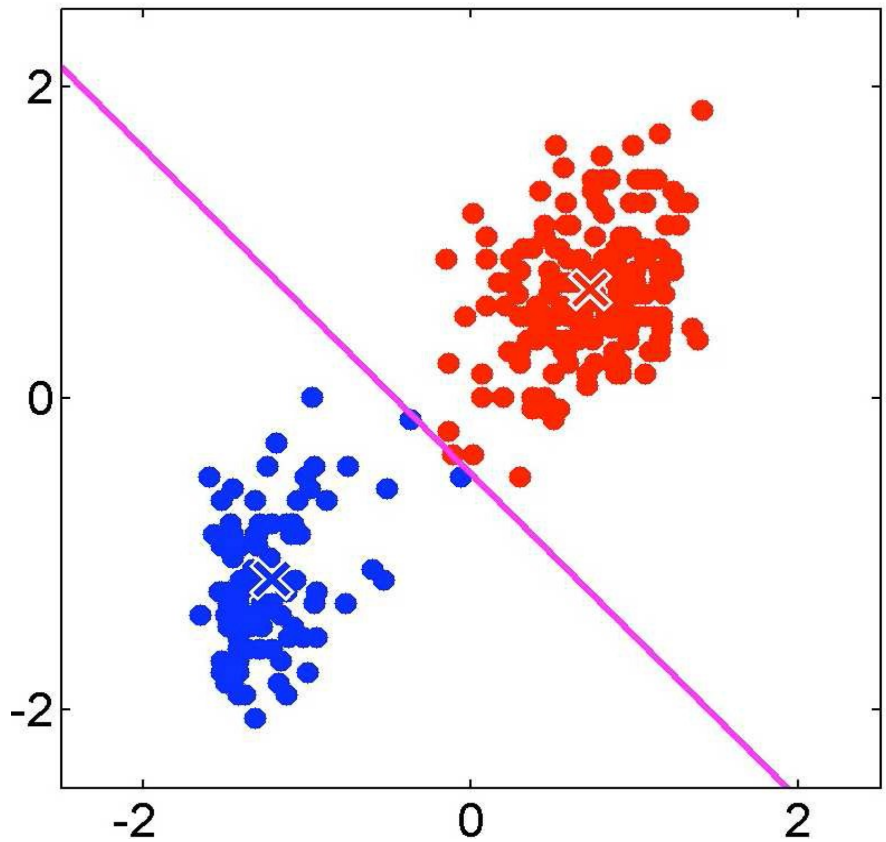
K-Means Clustering

- Example (K = 2): iteration #3-1



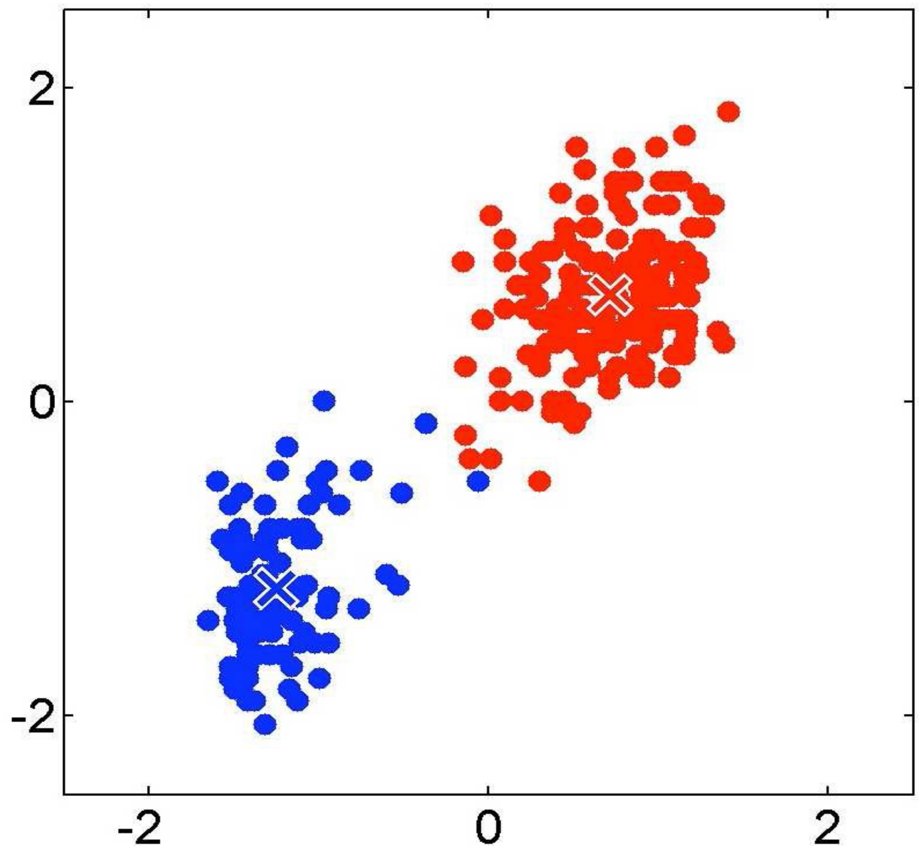
K-Means Clustering

- Example (K = 2): iteration #3-2



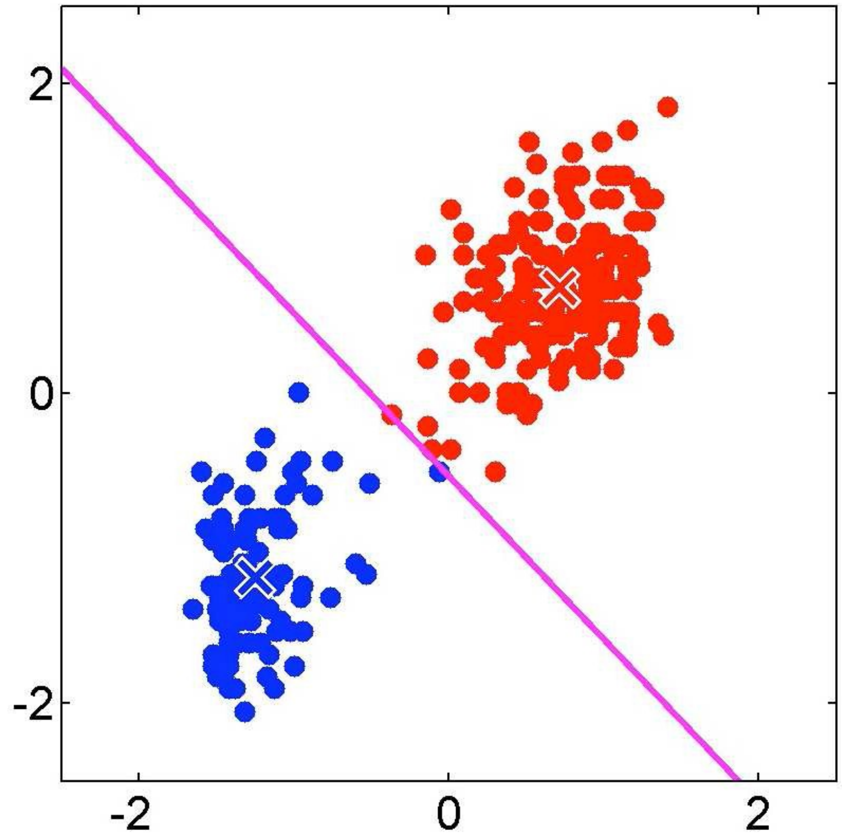
K-Means Clustering

- Example (K = 2): iteration #4-1



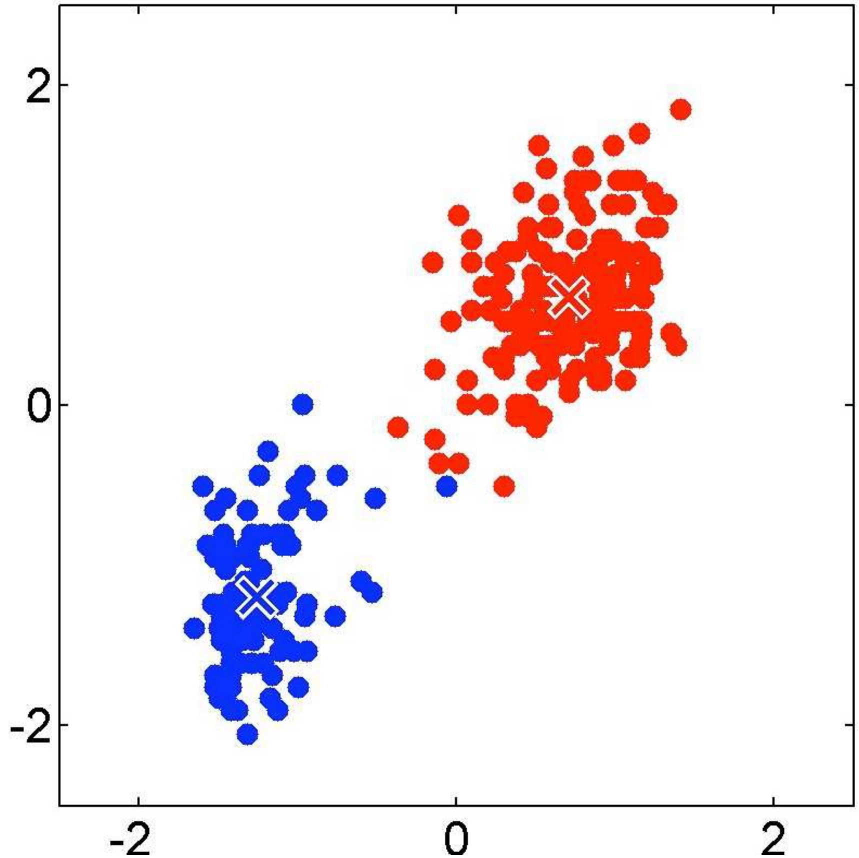
K-Means Clustering

- Example (K = 2): iteration #4-2



K-Means Clustering

- Example (K = 2): iteration #5, cluster means are not changed.

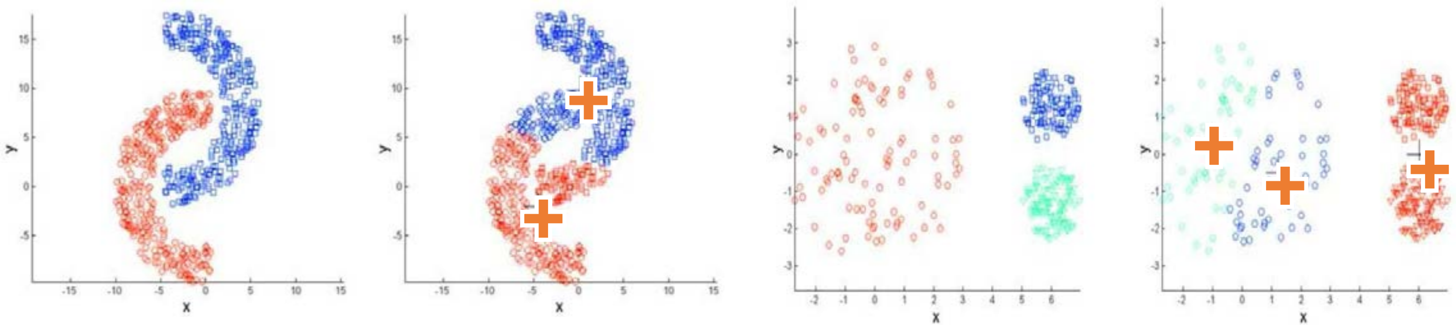


K-Means Clustering (cont'd)

- Proof in 1-D case if time permits...

K-Means Clustering (cont'd)

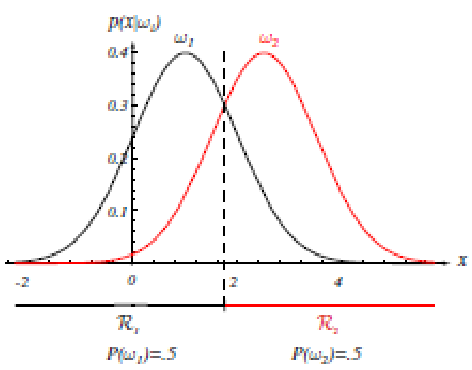
- Limitation
  - Sensitive to initialization; how to alleviate this problem?
  - Sensitive to outliers; possible change from K-means to…
  - Hard assignment only. Mathematically, we have…
  - Preferable for round shaped clusters with similar sizes

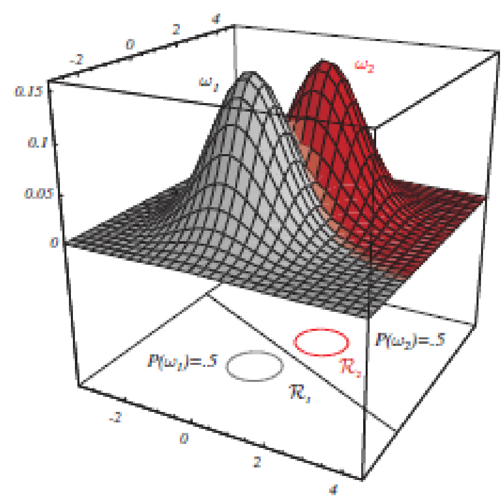


- Remarks
  - Speed-up possible by hierarchical clustering
  - Expectation–maximization (EM) algorithm

What's to Be Covered Today…

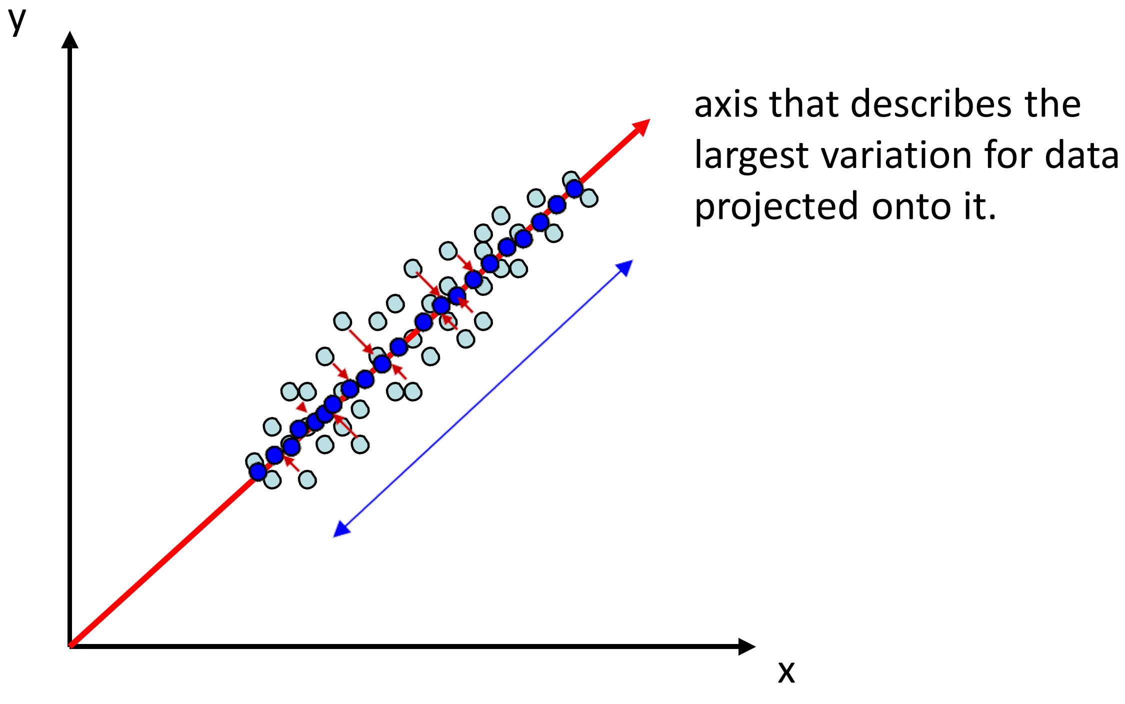
- From Probability to Bayes Decision Rule
- Brief Review of Linear Algebra & Linear System
- Unsupervised vs. Supervised Learning
    - Clustering & Dimension Reduction
    - Training, testing, & validation
    - Linear Classification





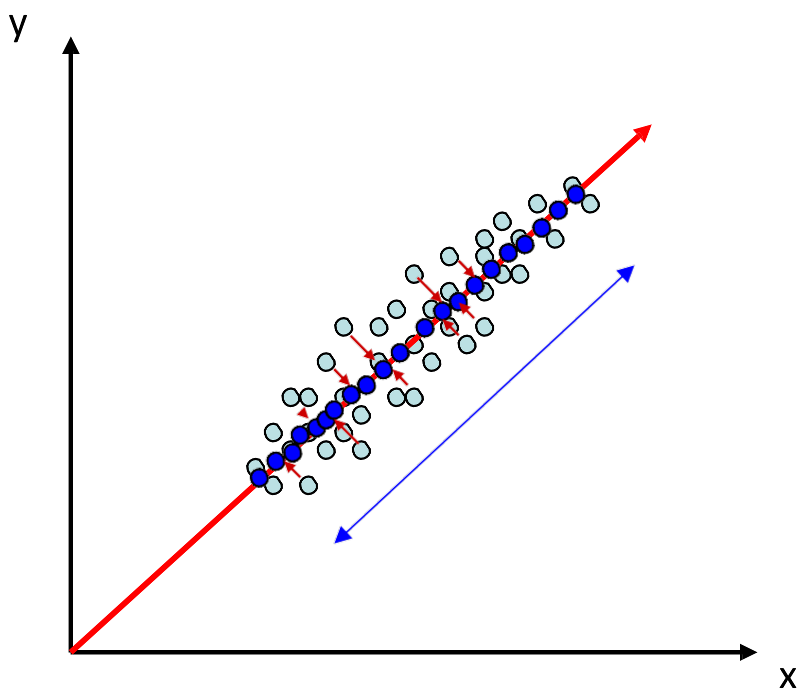
Dimension Reduction

- Principal Component Analysis (PCA)
  - Unsupervised & linear dimension reduction
  - Related to Eigenfaces, etc. feature extraction and classification techniques
  - Still very popular despite of its simplicity and effectiveness.
  - Goal:
    - Determine the projection, so that the variation of projected data is maximized.



Formulation & Derivation for PCA

- Input: a set of instances $\boldsymbol{x}$ without label info
- Output: a projection vector $\boldsymbol{\omega}$ maximizing the variance of the projected data
- In other words, we need to maximize $\mathrm{var}(\boldsymbol{\omega}^T\boldsymbol{x})$ with $\|\boldsymbol{\omega}\| = 1$.



Formulation & Derivation for PCA (cont'd)

- Lagrangian optimization for PCA

Eigenanalysis & PCA

- Eigenanalysis for PCA…find the eigenvectors $\boldsymbol{e}_i$ and the corresponding eigenvalues λ_i
  - In other words, the direction $\boldsymbol{e}_i$ captures the variance of λ_i.
  - But, which eigenvectors to use though? All of them?
- A d x d covariance matrix contains a maximum of d eigenvector/eigenvalue pairs.
  - Do we need to compute all of them? Which $\boldsymbol{e}_i$ and λ_i pairs to use?
  - Assuming you have N images of size M x M pixels, we have dimension $d = M^2$.
  - What is the rank of Σ?
  - Thus, at most non-zero eigenvalues can be obtained.
  - How dimension reduction is realized? how to reconstruct the input data?

Eigenanalysis & PCA (cont'd)

- A d x d covariance matrix contains a maximum of d eigenvector/eigenvalue pairs.
  - Assuming you have N images of size M x M pixels, we have dimension d = M^2.
  - With the rank of ∑ as , we have at most non-zero eigenvalues.
  - How dimension reduction is realized? how to reconstruct the input data?

- Expanding a signal via eigenvectors as bases
  - With symmetric matrices (e.g., covariance matrix), eigenvectors are orthogonal.
  - They can be regarded as unit basis vectors to span any instance in the d-dim space.

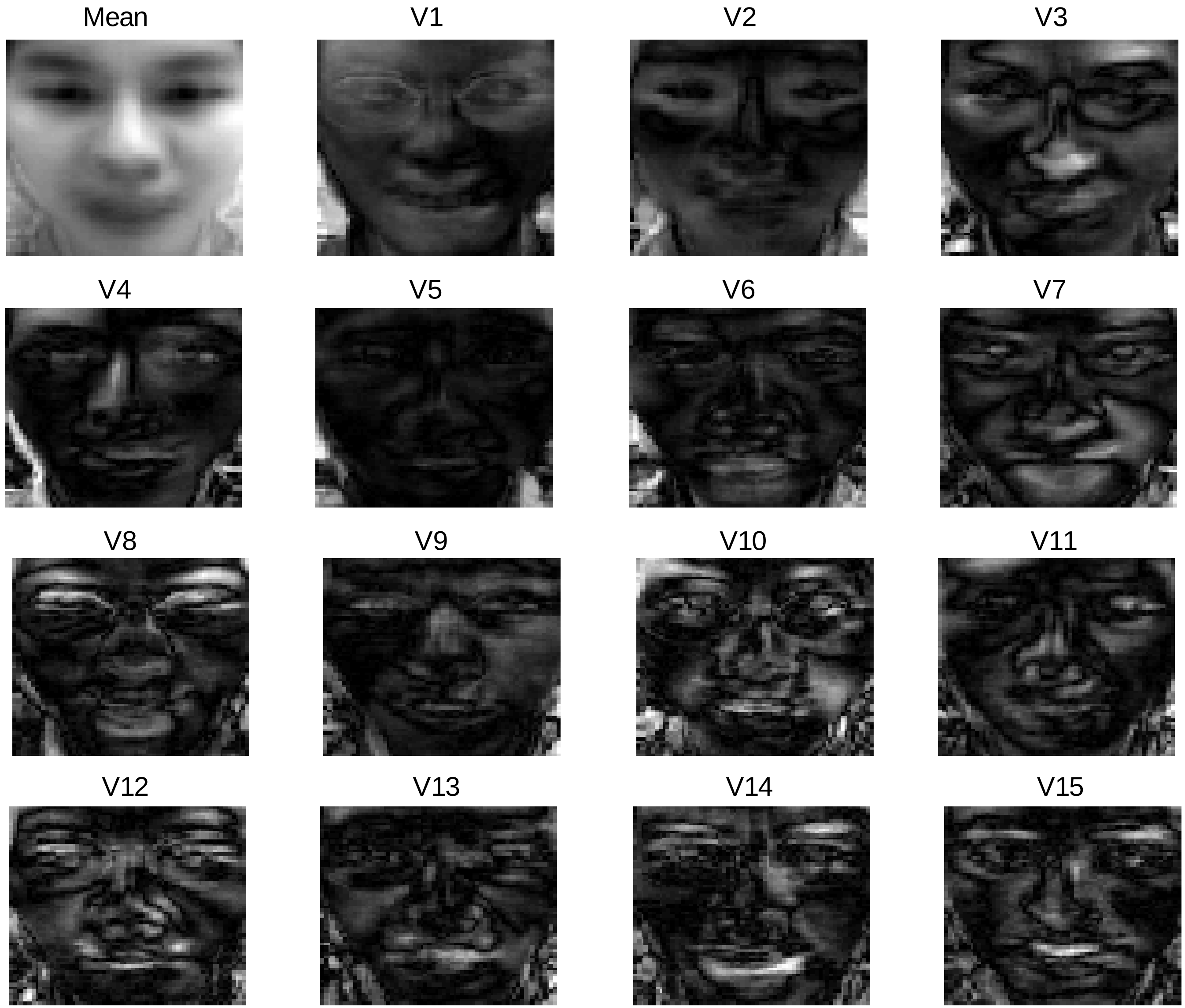
Practical Issues in PCA

- Assume we have N = 100 images of size 200 x 200 pixels (i.e., d = 40000).
- What is the size of the covariance matrix? What's its rank?
- What can we do? Gram Matrix Trick!

Let's See an Example (CMU AMP Face Database)

- Let's take 5 face images x 13 people = 65 images, each is of size 64 x 64 = 4096 pixels.
- # of eigenvectors are expected to use for perfectly reconstructing the input = 64.
- Let's check it out!

What Do the Eigenvectors/Eigenfaces Look Like?



All 64 Eigenvectors, do we need them all?

Use only 1 eigenvector, MSE = 1233

Use 2 eigenvectors, MSE = 1027

MSE=1027.63

Use 3 eigenvectors, MSE = 758

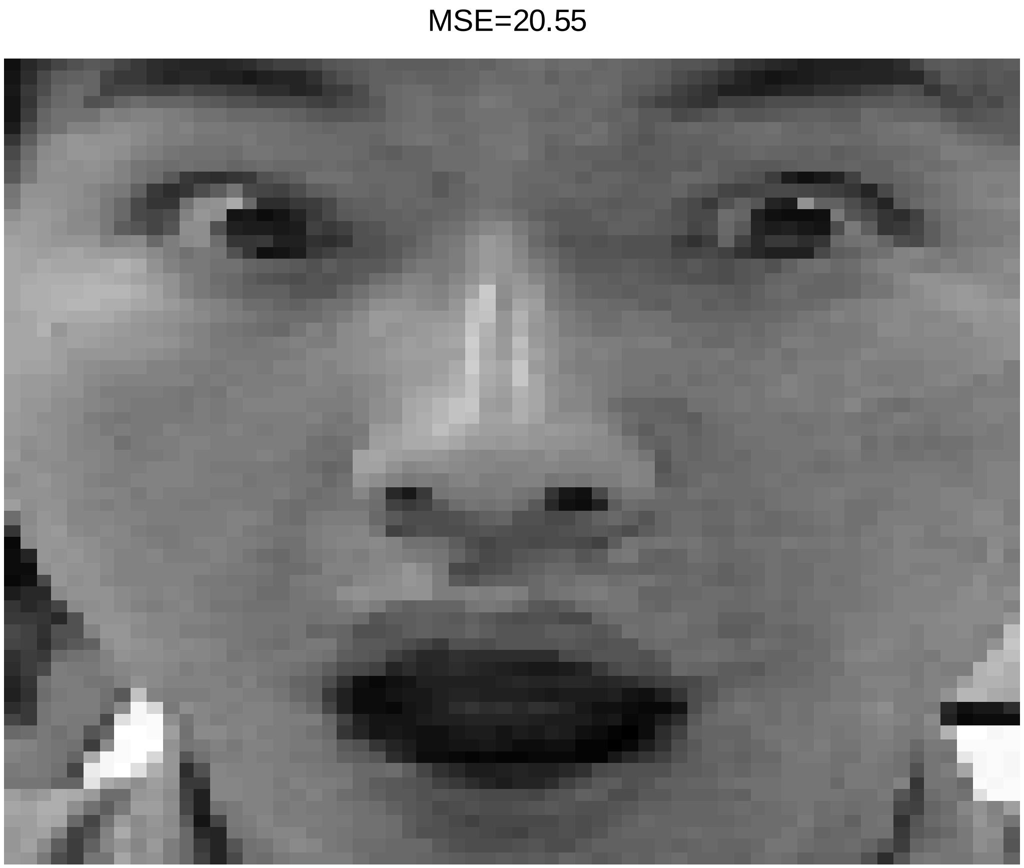
Use 4 eigenvectors, MSE = 634

MSE=634.54

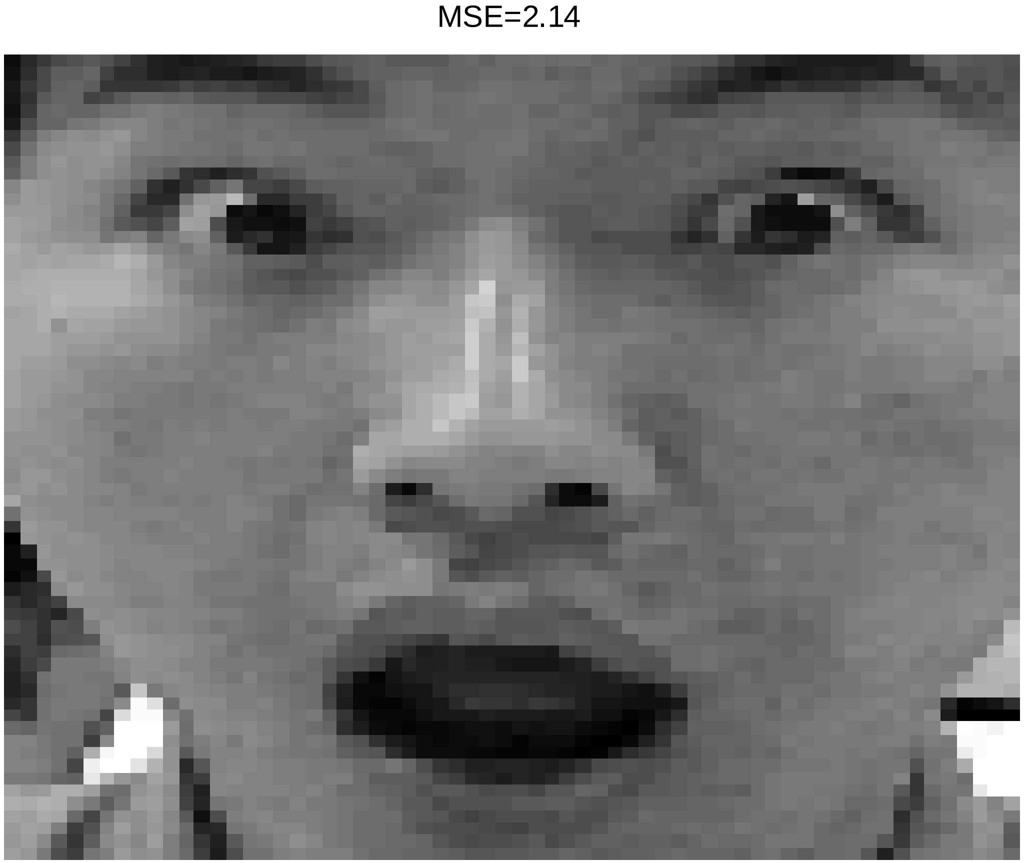
Use 8 eigenvectors, MSE = 285

With 20 eigenvectors, MSE = 87

With 30 eigenvectors, MSE = 20

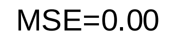


With 50 eigenvectors, MSE = 2.14



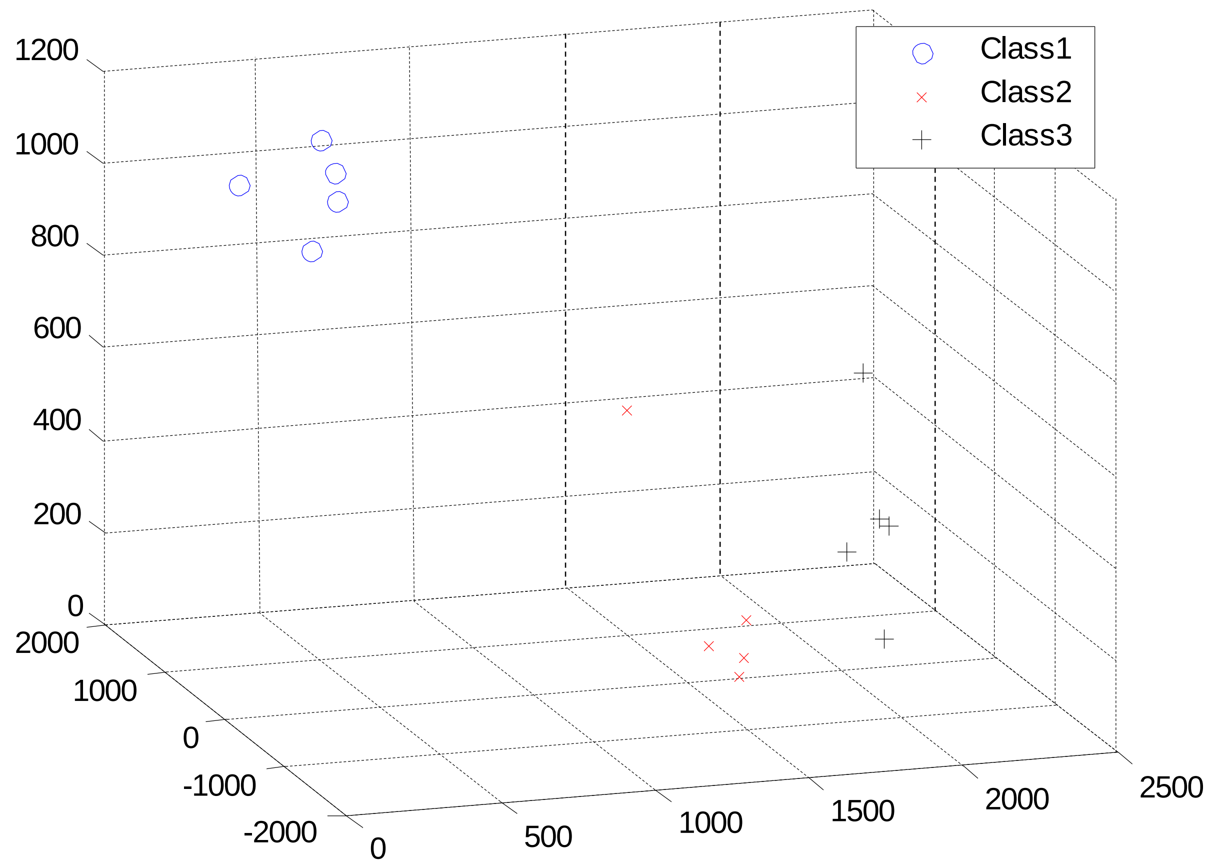
With 60 eigenvectors, MSE = 0.06

All 64 eigenvectors, MSE = 0



Final Remarks

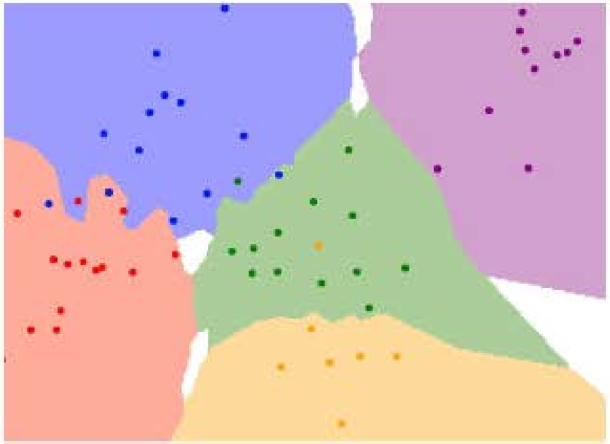
- Linear & unsupervised dimension reduction
- PCA can be applied as a feature extraction/preprocessing technique.
  - E.g,, Use the top 3 eigenvectors to project data into a 3D space for classification.



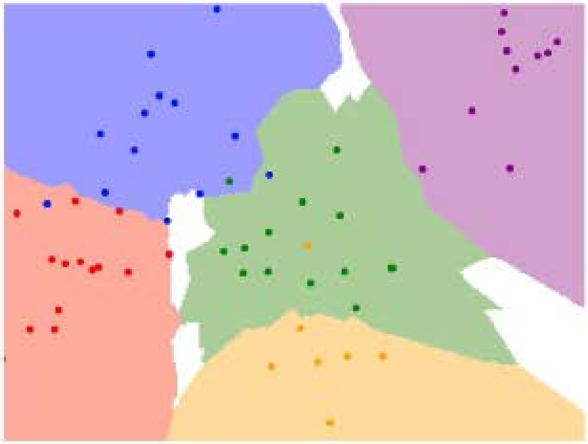
Final Remarks (cont'd)

- How do we classify? For example…
  - Given a test face input, project into the same 3D space (by the same 3 eigenvectors).
  - The resulting vector in the 3D space is the feature for this test input.
  - We can do a simple Nearest Neighbor (NN) classification with Euclidean distance, which calculates the distance to all the projected training data in this space.
  - If NN, then the label of the closest training instance determines the classification output.
  - If k-nearest neighbors (k-NN), then k-nearest neighbors need to vote for the decision.

k = 1



k = 3



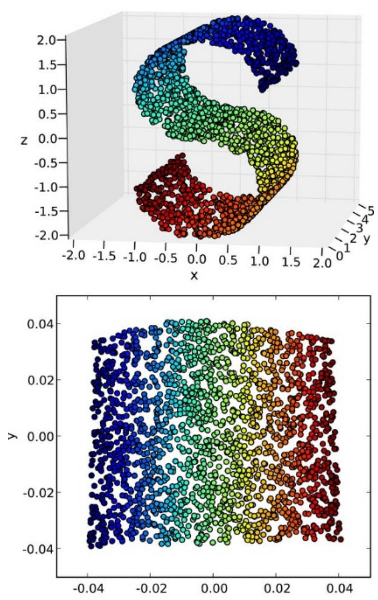
k = 5

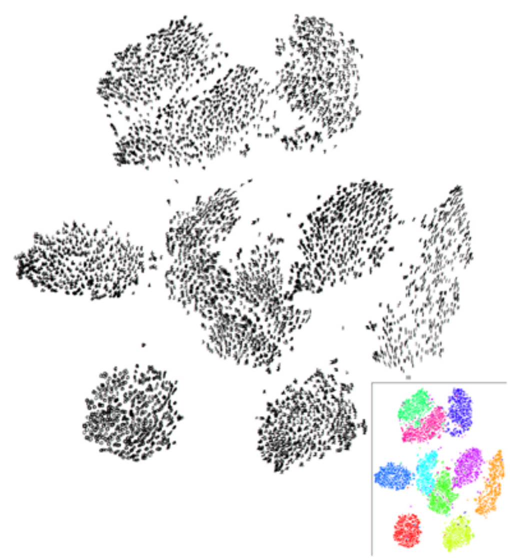
Demo available at http://vision.stanford.edu/teaching/cs231n-demos/knn/

Image credit: Stanford CS231n

Final Remarks (cont'd)

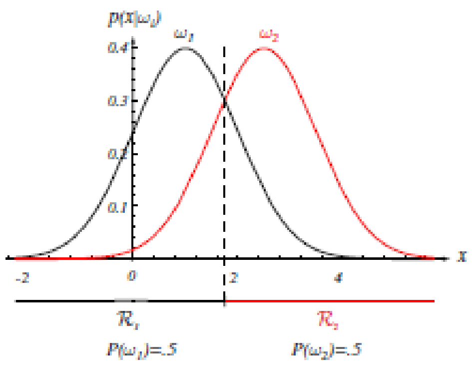
- If labels for each data is provided → Linear Discriminant Analysis (LDA)
  - LDA is also known as Fisher's discriminant analysis.
  - Eigenface vs. Fisherface (IEEE Trans. PAMI 1997)
- If linear DR is not sufficient, and non-linear DR is of interest…
  - Isomap, locally linear embedding (LLE), etc.
  - **t-distributed stochastic neighbor embedding (t-SNE)** (by G. Hinton & L. van der Maaten)

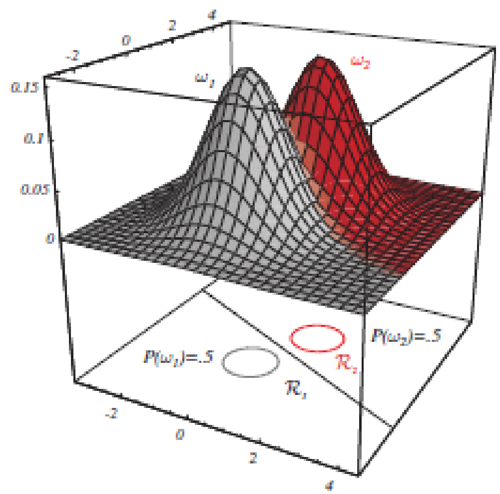




What's to Be Covered Today…

- From Probability to Bayes Decision Rule
- Brief Review of Linear Algebra & Linear System
- Unsupervised vs. Supervised Learning
  - Clustering & Dimension Reduction
  - Training, testing, & validation
  - Linear Classification

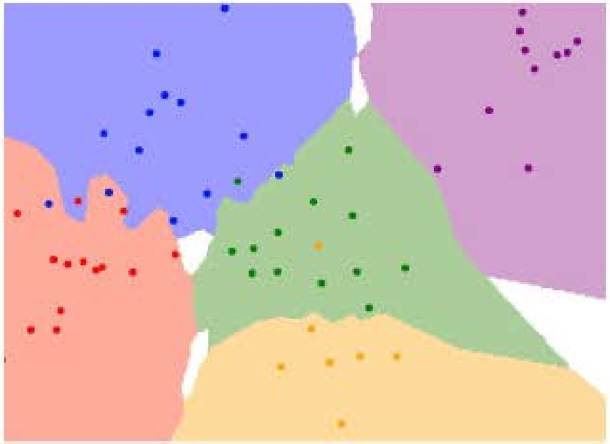




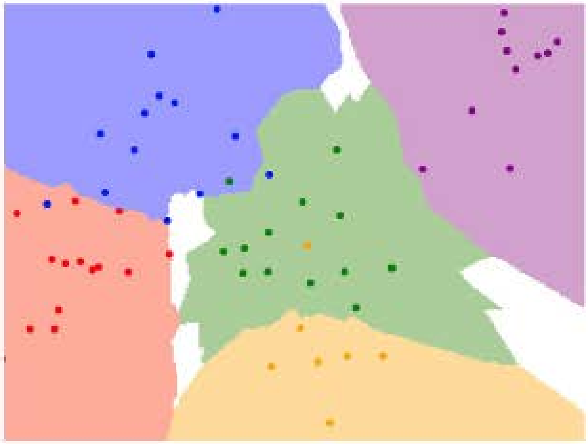
Hyperparameters in ML

- Recall that for k-NN, we need to determine the k value in advance.
  - What is the best k value?
  - And, what is the best distance/similarity metric?
  - Similarly, take PCA for example, what is the best reduced dimension number?
- **Hyperparameters**: choices about the learning model/algorithm of interest
  - We need to determine such hyperparameters instead of learn them.
  - Let’s see what we can do and cannot do…

k = 1



k = 3



k = 5

Image credit: Stanford CS231n

How to Determine Hyperparameters?

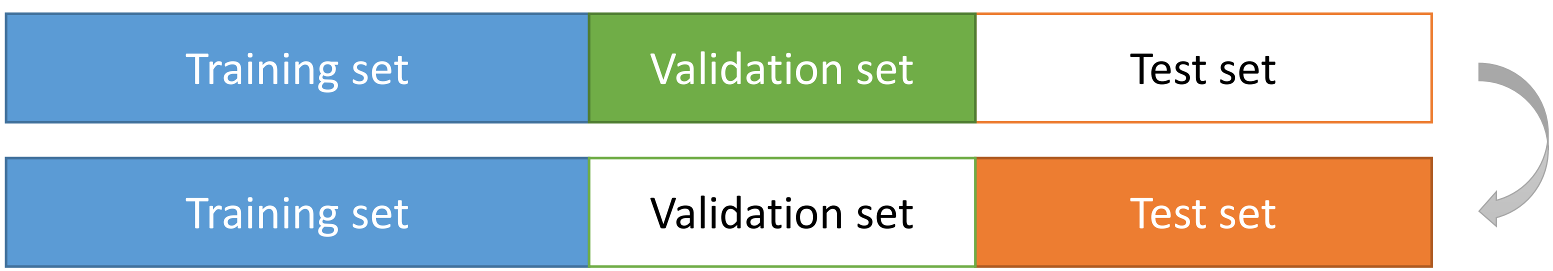
- Idea #1
  - Let’s say you are working on face recognition.
  - You come up with your very own feature extraction/learning algorithm.
  - You take a dataset to train your model, and select your hyperparameters based on the resulting performance.
  - 

How to Determine Hyperparameters? (cont'd)

- Idea #2
  - Let's say you are working on face recognition.
  - You come up with your very own feature extraction/learning algorithm.
  - For a dataset of interest, you split it into training and test sets.
  - You train your model with possible hyperparameter choices, and select those work best on test set data.
  - 

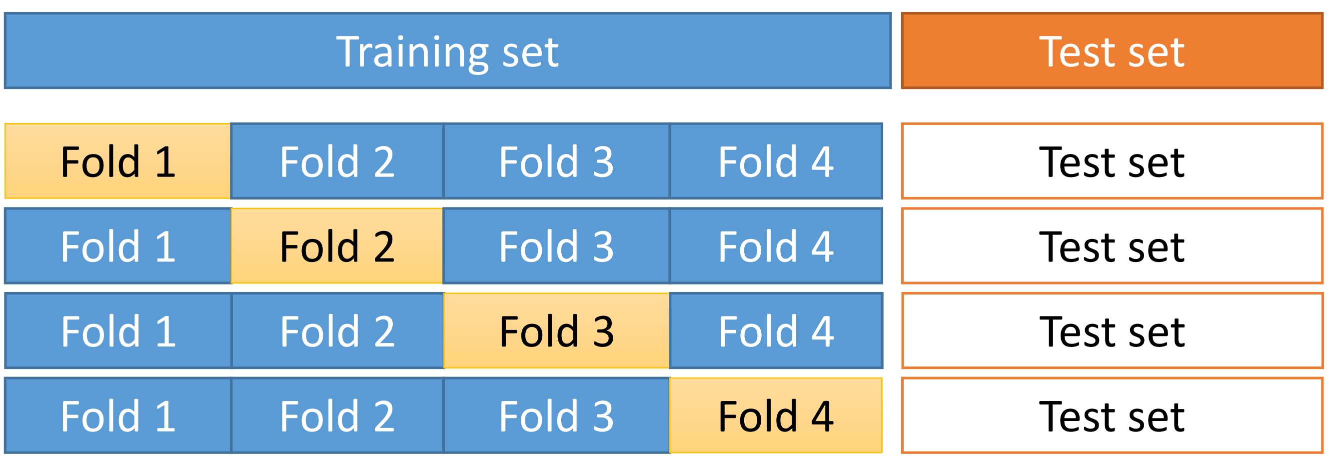
How to Determine Hyperparameters? (cont'd)

- Idea #3
  - Let's say you are working on face recognition.
  - You come up with your very own feature extraction/learning algorithm.
  - For the dataset of interest, it is split it into training, validation, and test sets.
  - You train your model with possible hyperparameter choices, and select those work best on the validation set.
  - 



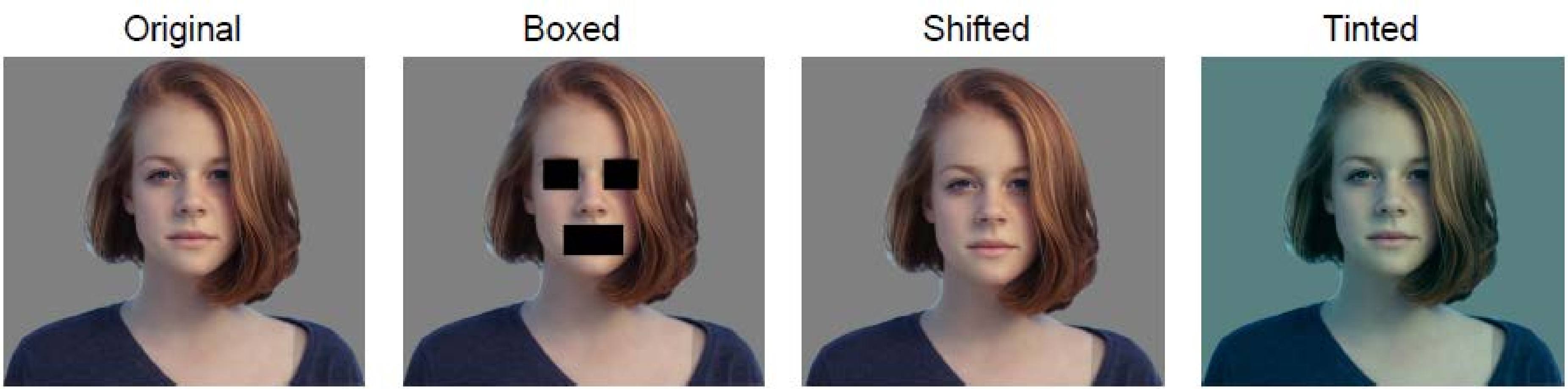
How to Determine Hyperparameters? (cont'd)

- Idea #3.5
  - What if only training and test sets are given, not the validation set?
  - Cross-validation (or *k-fold* cross validation)
    - Split the training set into k folds with a hyperparameter choice
    - Keep 1 fold as validation set and the remaining k-1 folds for training
    - After each of k folds is evaluated, report the average validation performance.
    - Choose the hyperparameter(s) which result in the highest average validation performance.
  - Take a 4-fold cross-validation as an example…



Minor Remarks on NN-based Methods

- In fact, k-NN (or even NN) is not of much interest in practice. Why?
  - Choice of distance metrics might be an issue. See example below.
  - Measuring distances in high-dimensional spaces might not be a good idea.
  - Moreover, NN-based methods require lots of and !
    (That is why NN-based methods are viewed as *data-driven* approaches.)

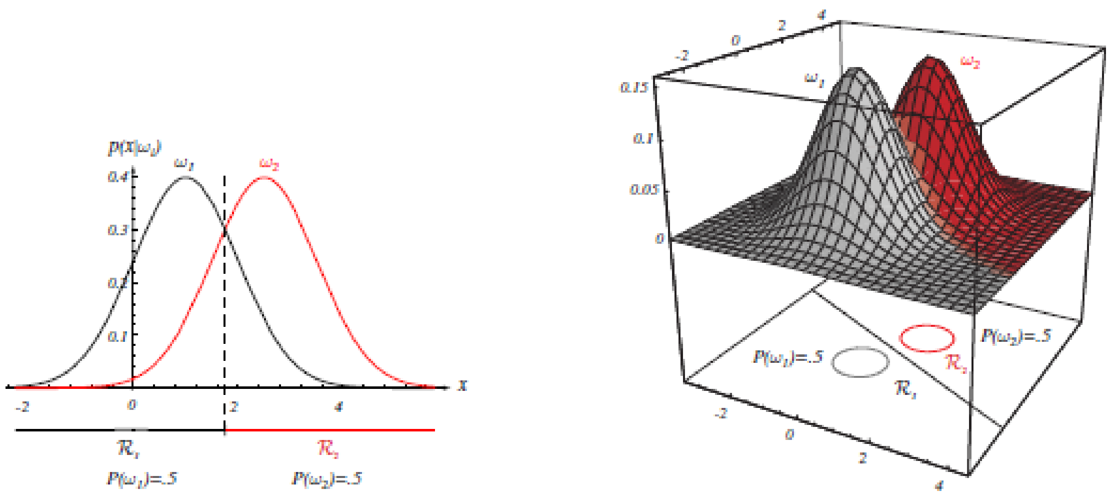


All three images have the same Euclidean distance to the original one.

Image credit: Stanford CS231n

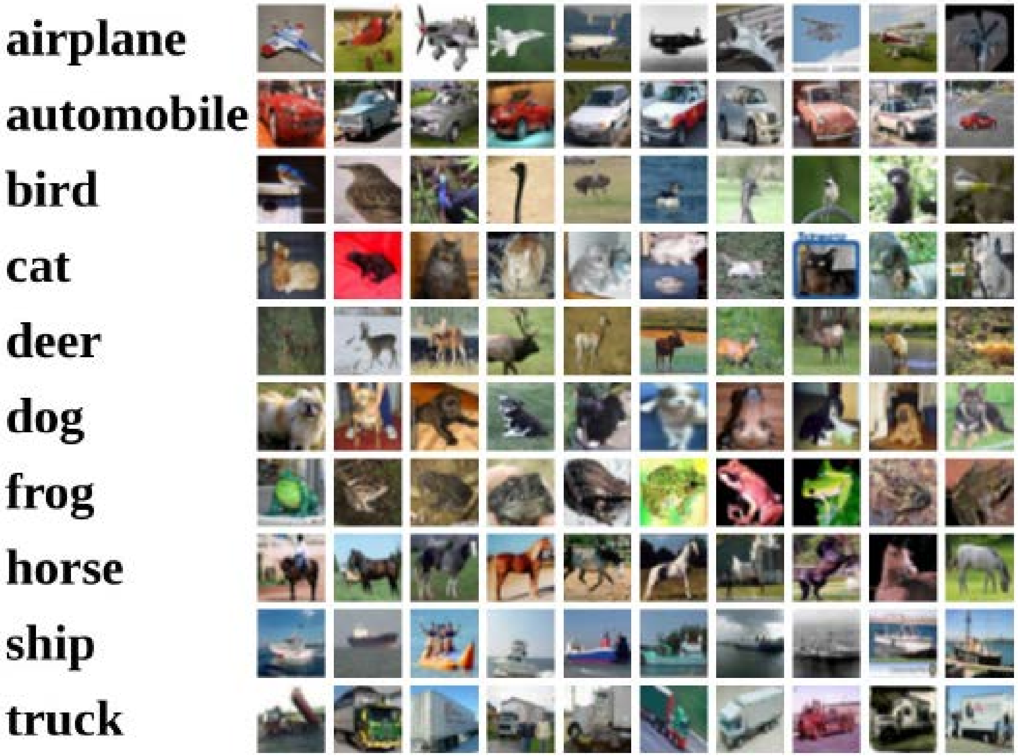
What's to Be Covered Today…

- From Probability to Bayes Decision Rule
- Brief Review of Linear Algebra & Linear System
- Unsupervised vs. Supervised Learning
  - Clustering & Dimension Reduction
  - Training, testing, & validation
  - Linear Classification



Linear Classification

- Linear Classifier
  - Can be viewed as a parametric approach. Why?
  - Assuming that we need to recognize 10 object categories of interest
  - E.g., CIFAR10 with 50K training & 10K test images of 10 categories. And, each image is of size 32 x 32 x 3 pixels.



Linear Classification (cont'd)

- Linear Classifier
  - Can be viewed as a parametric approach. Why?
  - Assuming that we need to recognize 10 object categories of interest (e.g., CIFAR10).
  - Let's take the input image as **x**, and the linear classifier as **W**. We hope to see that **y** = **Wx** + **b** as a 10-dimensional output indicating the score for each class.

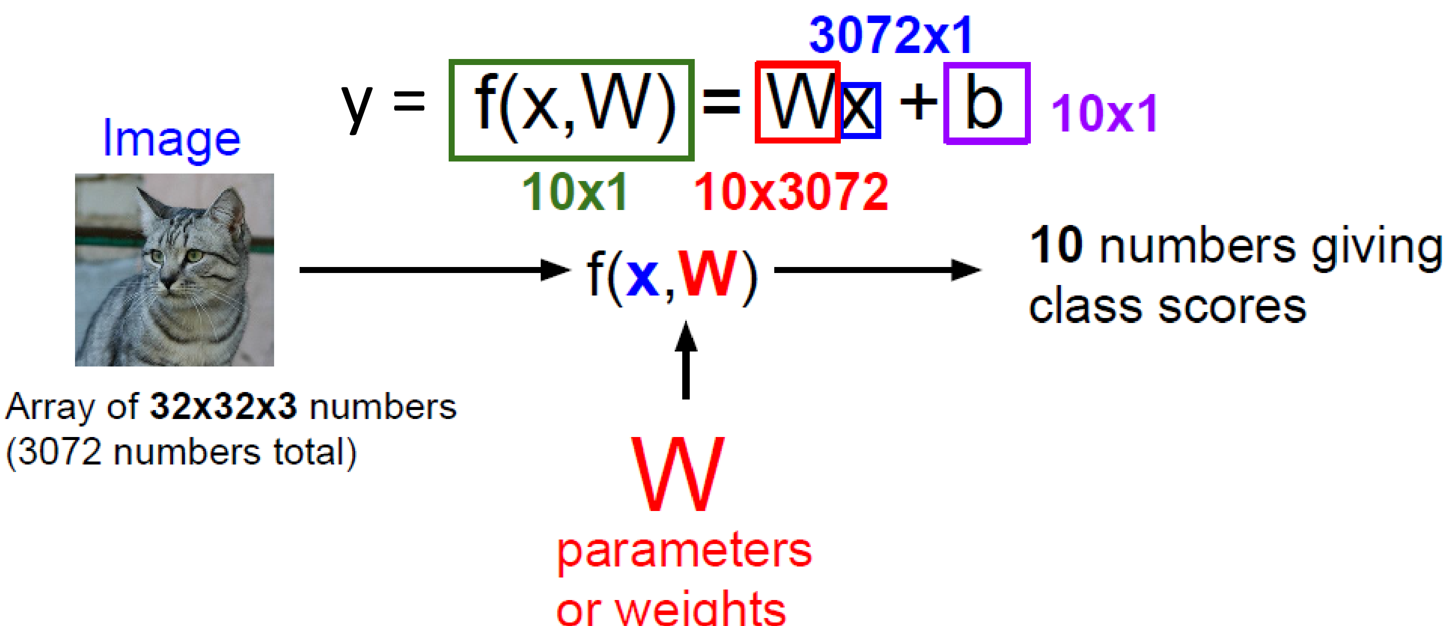


Image credit: Stanford CS231n

Linear Classification (cont'd)

- Linear Classifier
  - Can be viewed as a parametric approach. Why?
  - Assuming that we need to recognize 10 object categories of interest (e.g., CIFAR10).
  - Let's take the input image as **x**, and the linear classifier as **W**. We hope to see that **y** = **Wx** + **b** as a 10-dimensional output indicating the score for each class.
  - Take an image with 2 x 2 pixels & 3 classes of interest as example: we need to learn linear transformation/classifer **W** and bias **b**, so that desirable outputs **y** = **Wx** + **b** can be expected.

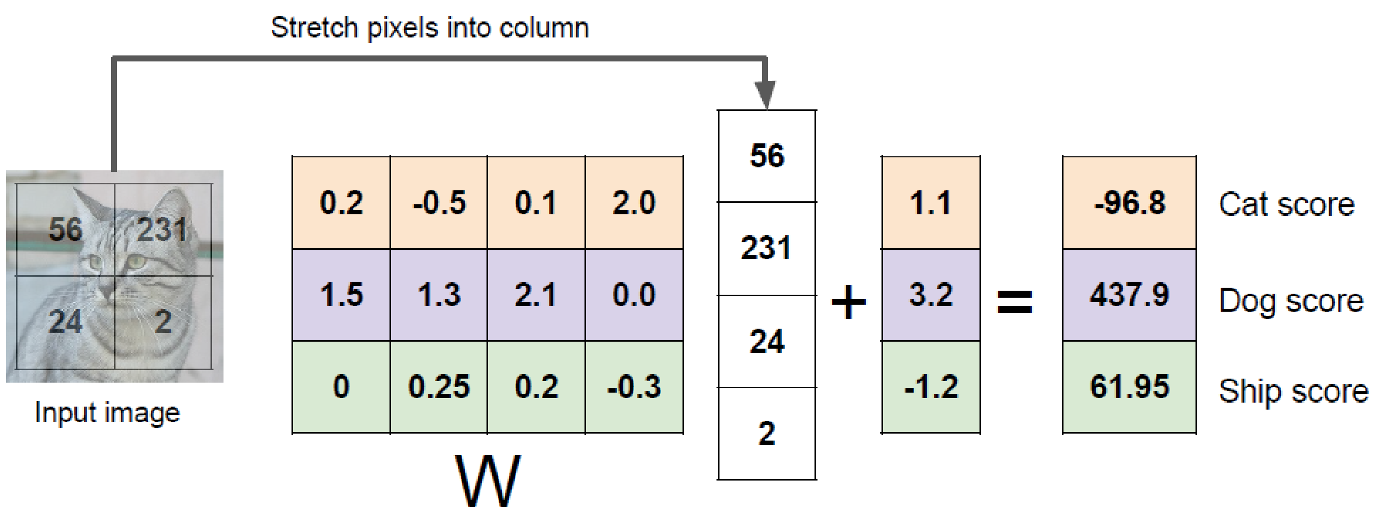
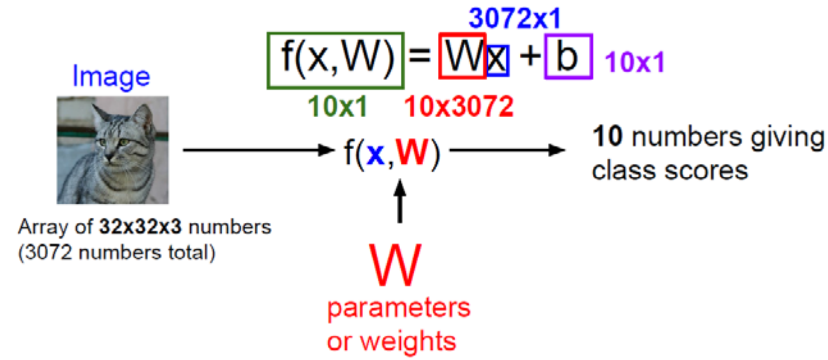
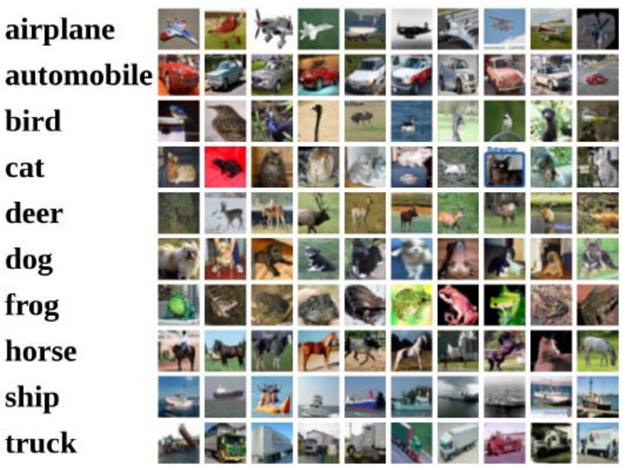


Image credit: Stanford CS231n

Some Remarks



- Interpreting **y** = **Wx** + **b**
  - What can we say about the learned **W**?
  - The weights in **W** are trained by observing training data **X** and their ground truth **Y**.
  - Each column in **W** can be viewed as an exemplar of the corresponding class.
  - Thus, **Wx** basically performs inner product (or correlation) between the input **x** and the exemplar of each class. (Signal & Systems!)



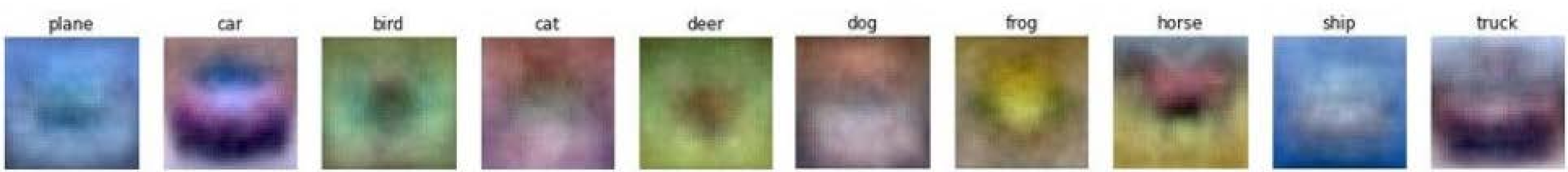
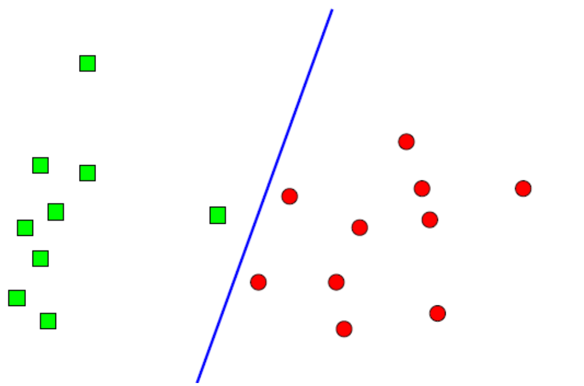


Image credit: Stanford CS231n

From Binary to Multi-Class Classification



e.g., Support Vector Machine (SVM)

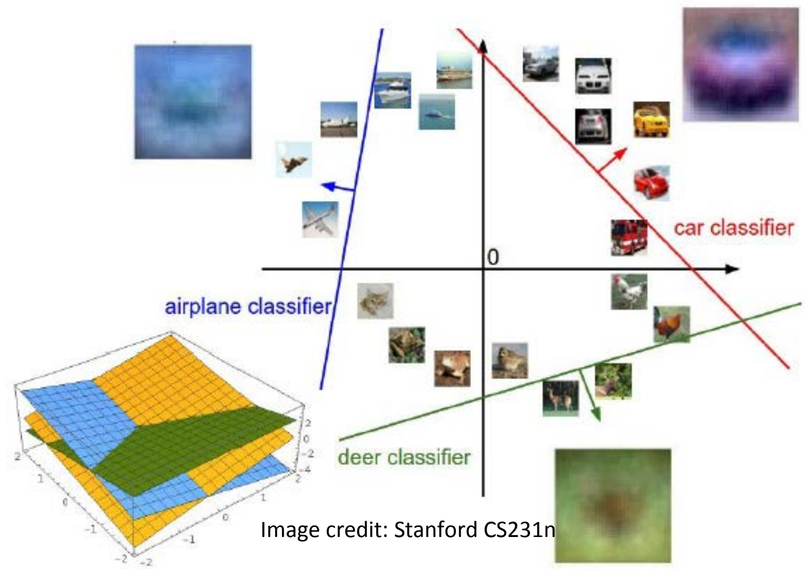
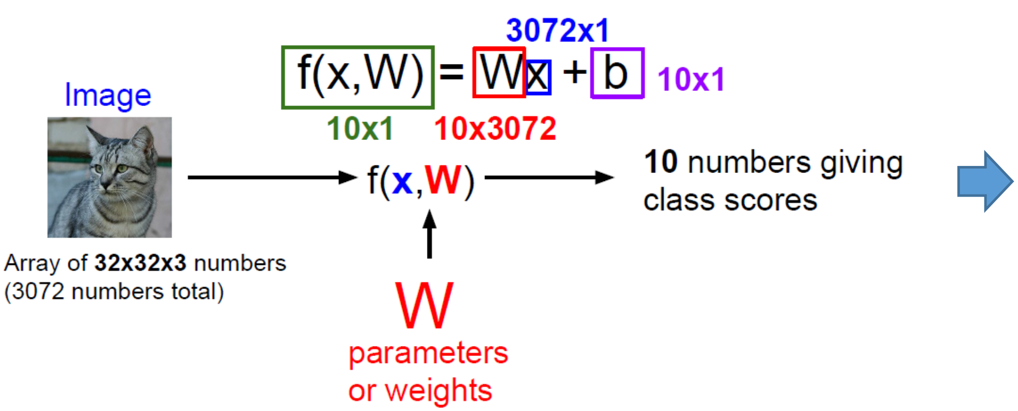


Image credit: Stanford CS231n

- How to extend binary classifiers for solving multi-class classification?
  - 1-vs.-1 (1-against-1) vs. 1-vs.-all (1-against-rest)

Linear Classification

- Remarks
  - Starting points for many multi-class or complex/nonlinear classifier
  - How to determine a proper loss function for matching **y** and **Wx+b**, and thus how to learn the model **W** (including the bias **b**), are the keys to the learning of an effective classification model.



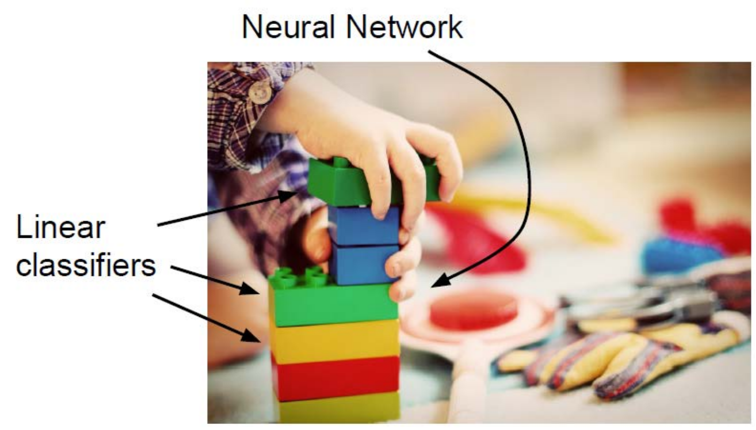


Image credit: Stanford CS231n

What We Learned Today…

- From Probability to Bayes Decision Rule
- Brief Review of Linear Algebra & Linear System
- Unsupervised vs. Supervised Learning
  - Dimension Reduction, Clustering
  - Training, testing, & validation
  - Linear Classification