Lecture 4

Math 178

Nonlinear Data Analytics

Prof. Weiqing Gu

Today's topics

- **Data Challenges**
- **How to analyze data on a (curved) manifold?**
- **The k-means clustering algorithm**
- **Mixtures of Gaussians**
- **The EM (Expectation-Maximization) Algorithm**
- **Support Vector Machine**
- **Probability Review**

1.1. The Challenge. The current 'data deluge' inundating science and technology is remarkable not merely for the often-mentioned *volumes* of data, but also for the rapid proliferation in new data *types*. In addition to the old standby of simple numerical arrays, we are starting to see arrays where the entries have highly structured values obeying nonlinear constraints.

Many such examples can be given. We have in mind data arrays of the form $p(t)$, $p(x, y)$, or $p(x, y, z)$ where t, x, y, z run through equispaced values in a cartesian grid, and p takes values in a manifold M. Consider these examples:

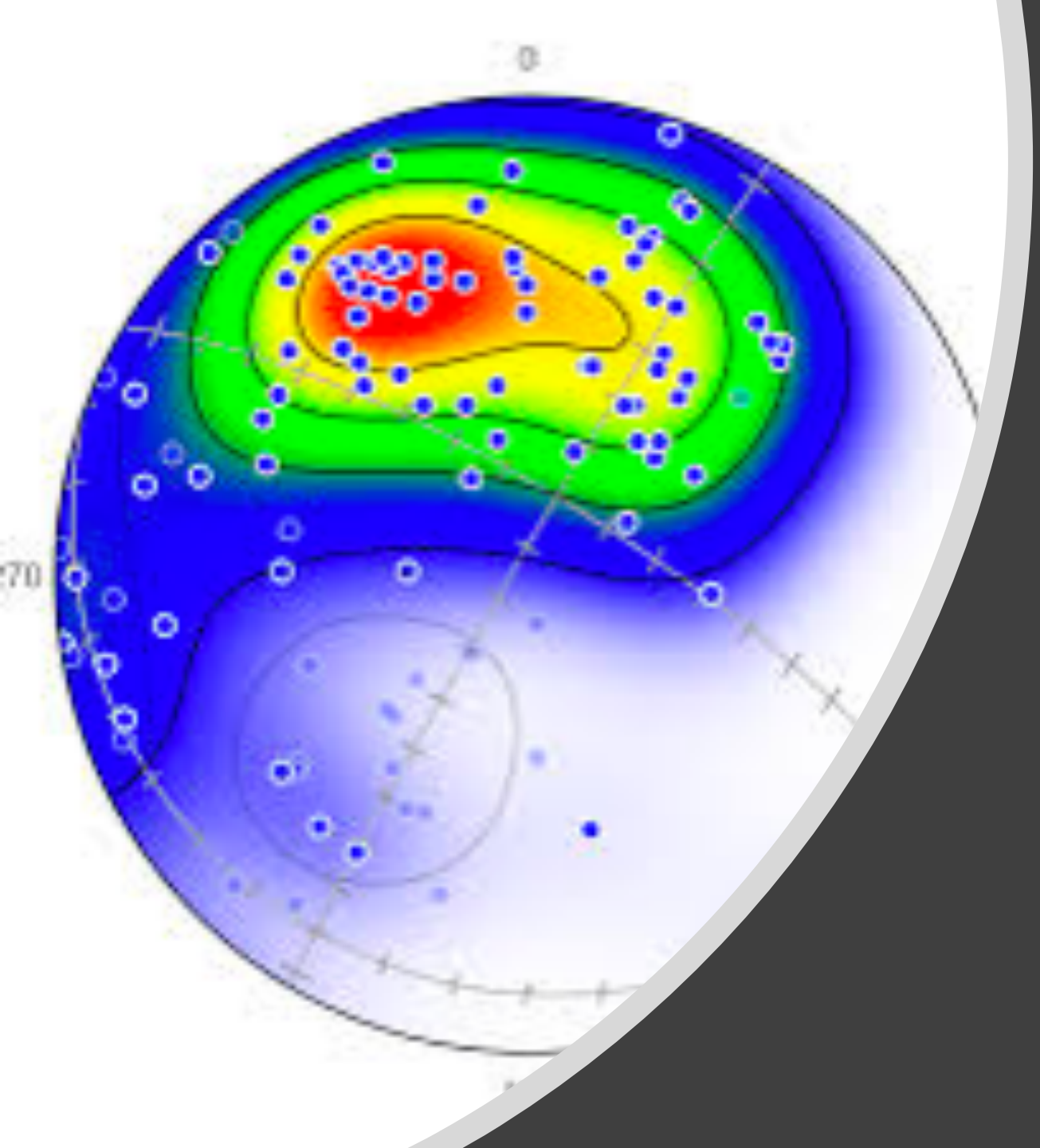
- *Headings.* Here p specifies directions in $\mathbf{R}^2$ or $\mathbf{R}^3$, and so M is either the unit circle $S^1 \subset \mathbf{R}^2$ or the unit sphere $S^2 \subset \mathbf{R}^3$. Such data can arise as a time series of observations of vehicle headings.
- *Orientations.* Here p gives 'tripods', i.e. orientations belonging to $M = SO(3)$. Such data can arise as a time series of aircraft orientations (pitch, roll, yaw).
- *Rigid Motions.* Here p specifies rigid motions in the special Euclidean group $M = SE(3)$. Such data can arise as a time series of placements of an ob-

ject in space (position, orientation), or as a spatially-organized array giving the displacements and orientations of marker particles having undergone a deformation.

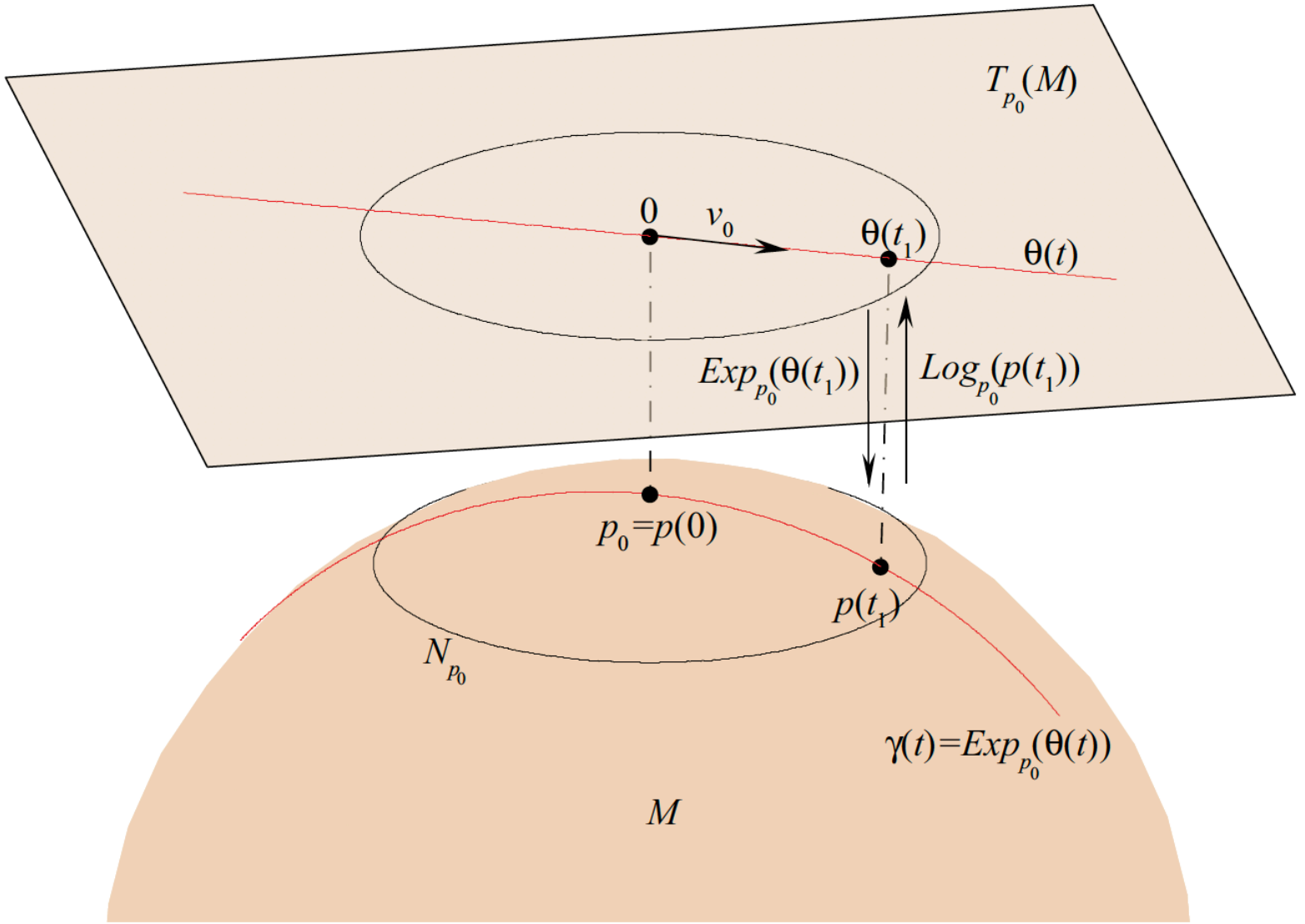
- *Deformation Tensors.* Here p is a symmetric positive definite matrix in $M = SPD(n)$. Spatially-organized data of this kind can arise from measurements of strain/stress and deformation in materials science and earth science. Arrays of this kind also arise in cosmological measurements of gravitational lensing.
- *Distance Matrices.* Here each p is an n by n matrix giving the pairwise distances between all pairs in a cloud of n points. Time series of this kind can arise as representing the state of a swarm of maneuvering vehicles, each of which can sense its distance to all other members of the swarm.
- *Projections, Subspaces.* Here p is a projector with k-dimensional range, or what is the same thing, a k-subspace of $\mathbf{R}^n$. Such values belong to the Grassmann manifold $G(k, n)$. Time series of this kind can arise in array signal processing, where the subspace is associated with the signal-generating sources.

MULTISCALE REPRESENTATIONS FOR MANIFOLD-VALUED DATA

INAM UR RAHMAN*, IDDO DRORI*, VICTORIA C. STODDEN*
DAVID L. DONOHO*, PETER SCHRÖDER†



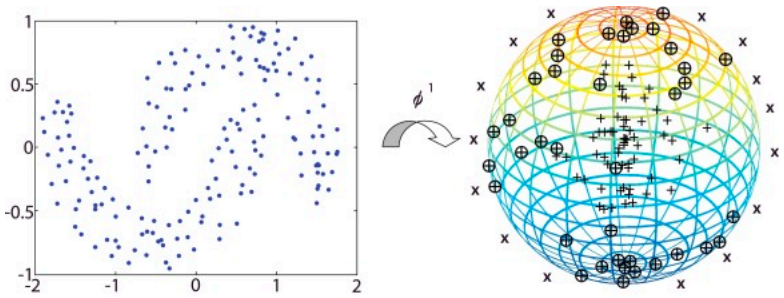
How to analyze data on a manifold?



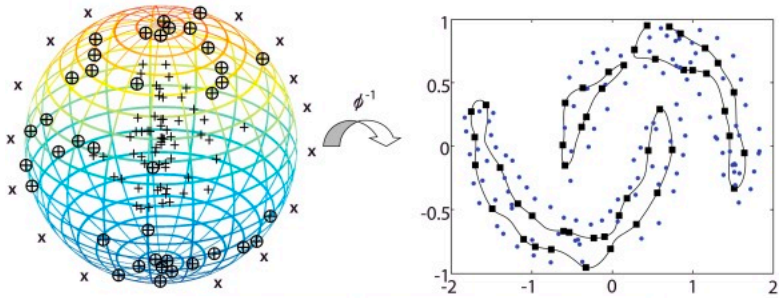
Details later

FIG. 3.1. *A manifold, its tangent plane, and the correspondence between a line in the tangent plane and a geodesic in the manifold.*

A Rough–Fuzzy approach for Support Vector Clustering



(a) Projection from Original Data Space to Higher-Dimensional Space

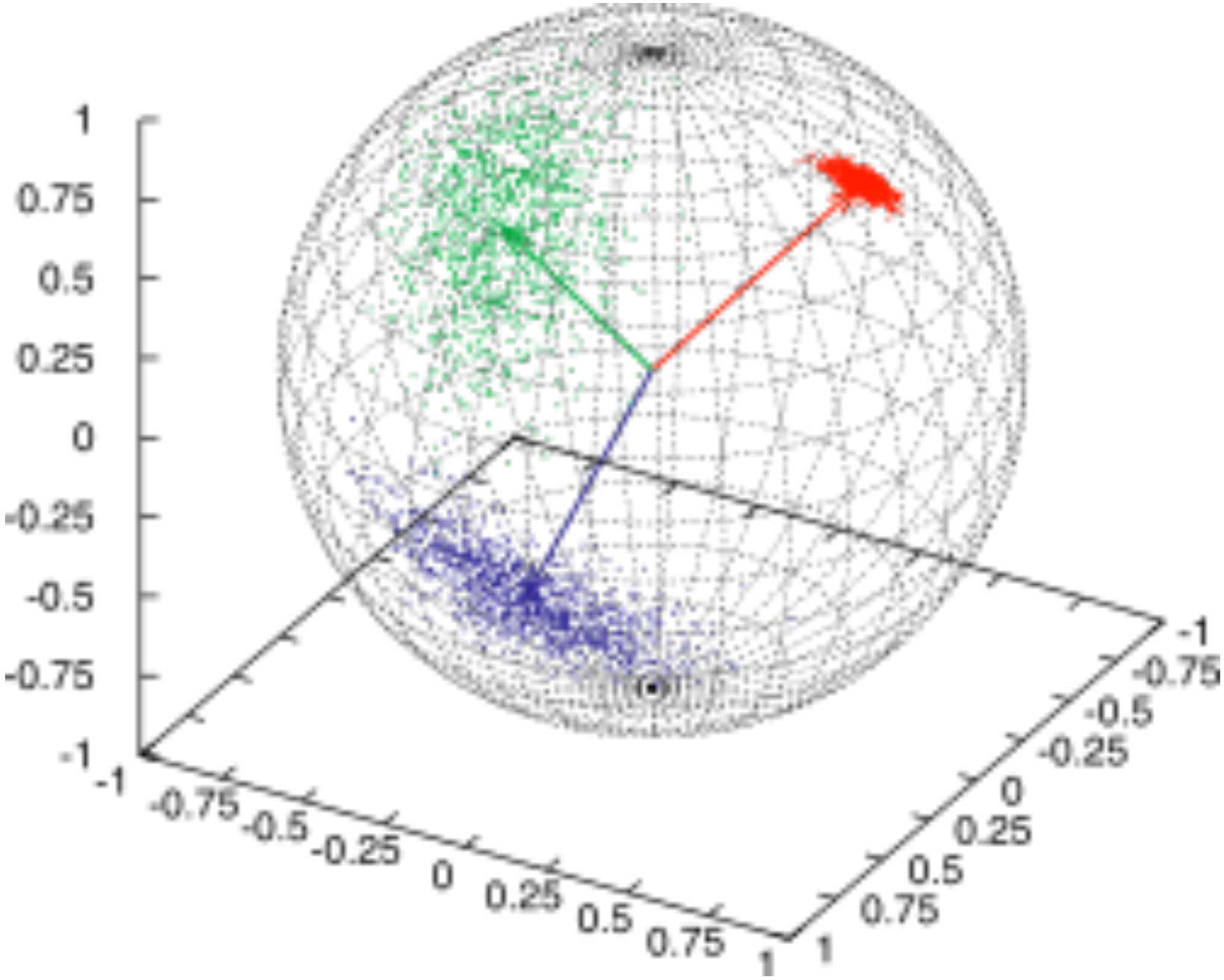


(b) Inverse Projection from Enclosing Sphere to Cluster Contours

Highlights

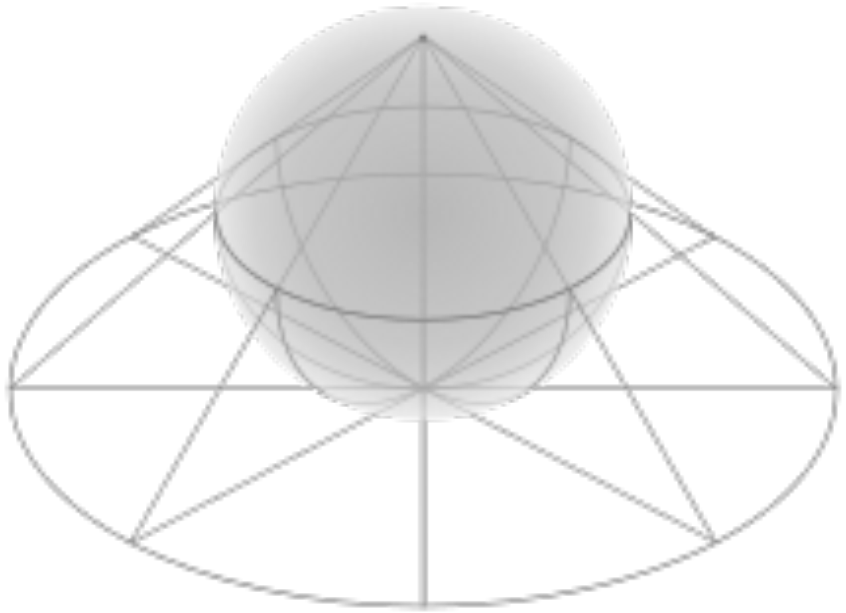
- We present a novel soft clustering approach based on Support Vector Clustering.
- Data points outside the clusters found form a fuzzy boundary region.
- Clusters with any shape as well as outliers can be identified.
- Membership degrees were calculated in a natural way.

https://www.sciencedirect.com/science/article/pii/S0020025516000189


1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
-1
-0.75
-0.5
-0.25
0
0.25
0.5
0.75
1
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1

Data lies or closes to a low dimensional manifold.

Stereographic Projection



3D illustration of a stereographic projection from the north pole onto a plane below the sphere

The unit sphere in three-dimensional space $\mathbf{R}^3$ is the set of points (x, y, z) such that $x^2 + y^2 + z^2 = 1$. Let $N = (0, 0, 1)$ be the "north pole", and let $\mathcal{M}$ be the rest of the sphere. The plane $z = 0$ runs through the center of the sphere; the "equator" is the intersection of the sphere with this plane.

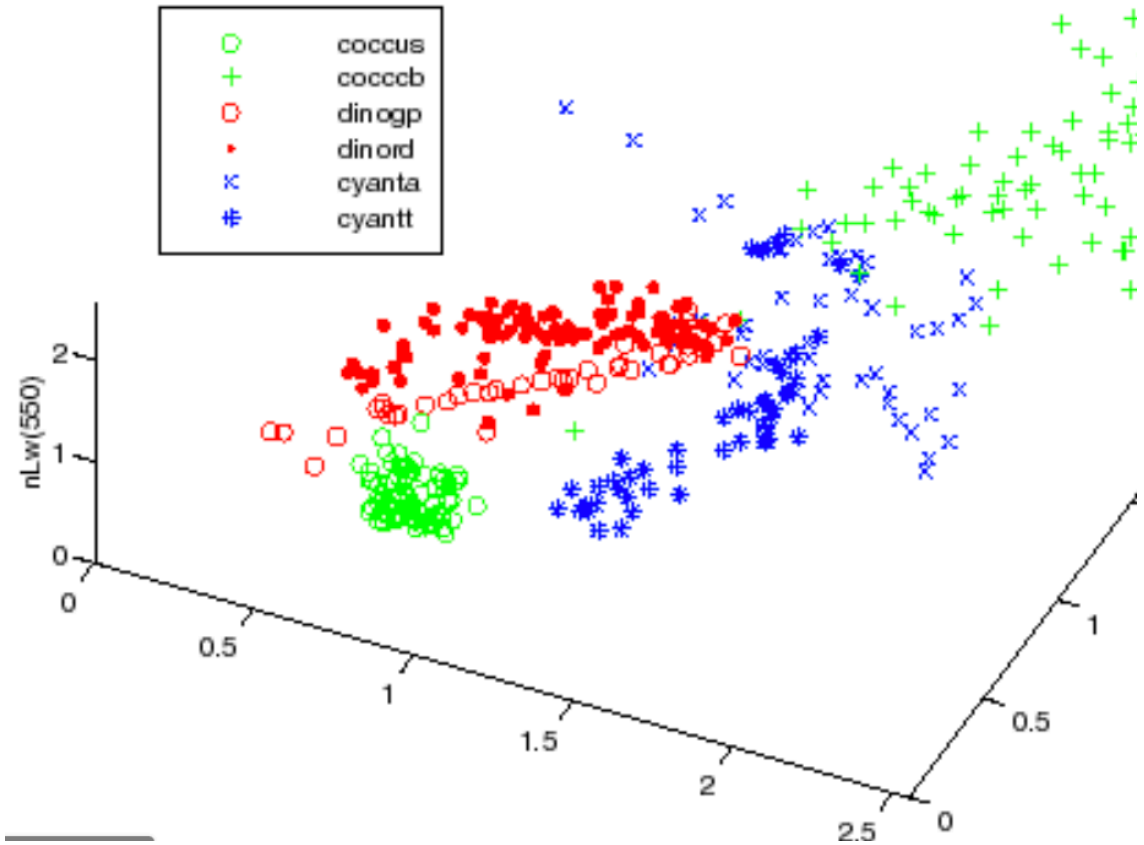
For any point P on $\mathcal{M}$, there is a unique line through N and P, and this line intersects the plane $z = 0$ in exactly one point P'. Define the **stereographic projection** of P to be this point P' in the plane.

In Cartesian coordinates (x, y, z) on the sphere and (X, Y) on the plane, the projection and its inverse are given by the formulas

$$(X, Y) = \left(\frac{x}{1-z}, \frac{y}{1-z}\right),$$

$$(x, y, z) = \left(\frac{2X}{1+X^2+Y^2}, \frac{2Y}{1+X^2+Y^2}, \frac{-1+X^2+Y^2}{1+X^2+Y^2}\right).$$

Step1: Project the data from sphere to a plane



Step 2: Apply K-mean cluster algorithm or Gaussian mixture algorithm on the plane to find the clusters and label them in different colors.

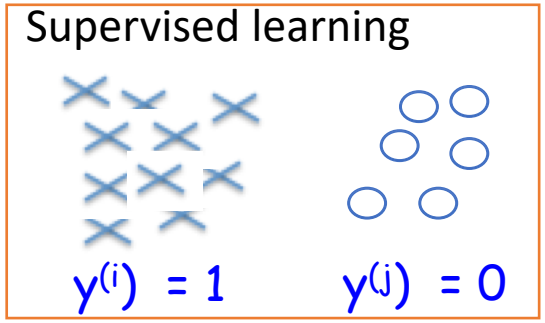
- Step 3: Use the inverse Stereographic Projection to project the data back to the sphere.

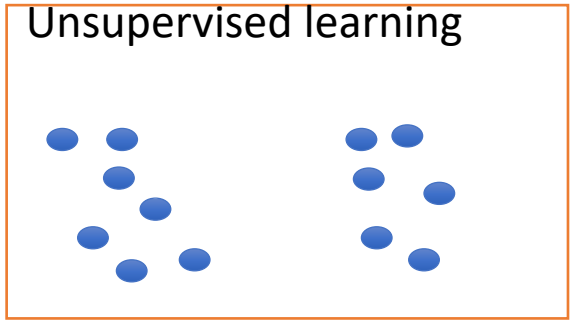
Review: K-mean cluster algorithm and Gaussian mixture algorithm in Euclidean space.

What is a clustering problem?

A clustering problem is an unsupervised learning problem

- Given a training set $\{x^{(1)}, \ldots, x^{(m)}\}$, here each $x^{(i)}$ is in $\mathbf{R}^n$.
- Goal: want to group the data into a few cohesive "clusters."
- Note: the difference between unsupervised learning and supervised learning is that no labels y(i) are given.





In general, if only $\{x^{(1)}, \ldots, x^{(m)}\}$ given for a problem, but no labels $y^{(i)}$ are given, the the problem is an unsupervised learning problem!

The k-means clustering algorithm

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.

2. Repeat until convergence: {

   For every i, set

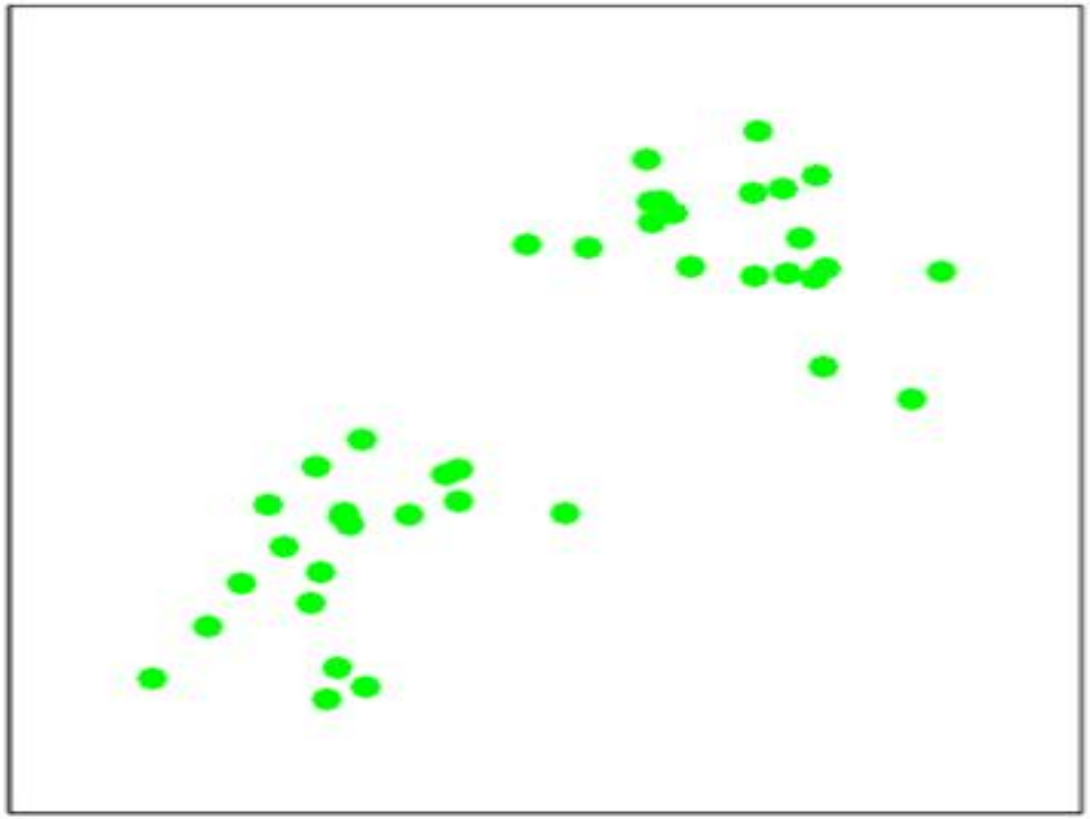
   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   For each j, set

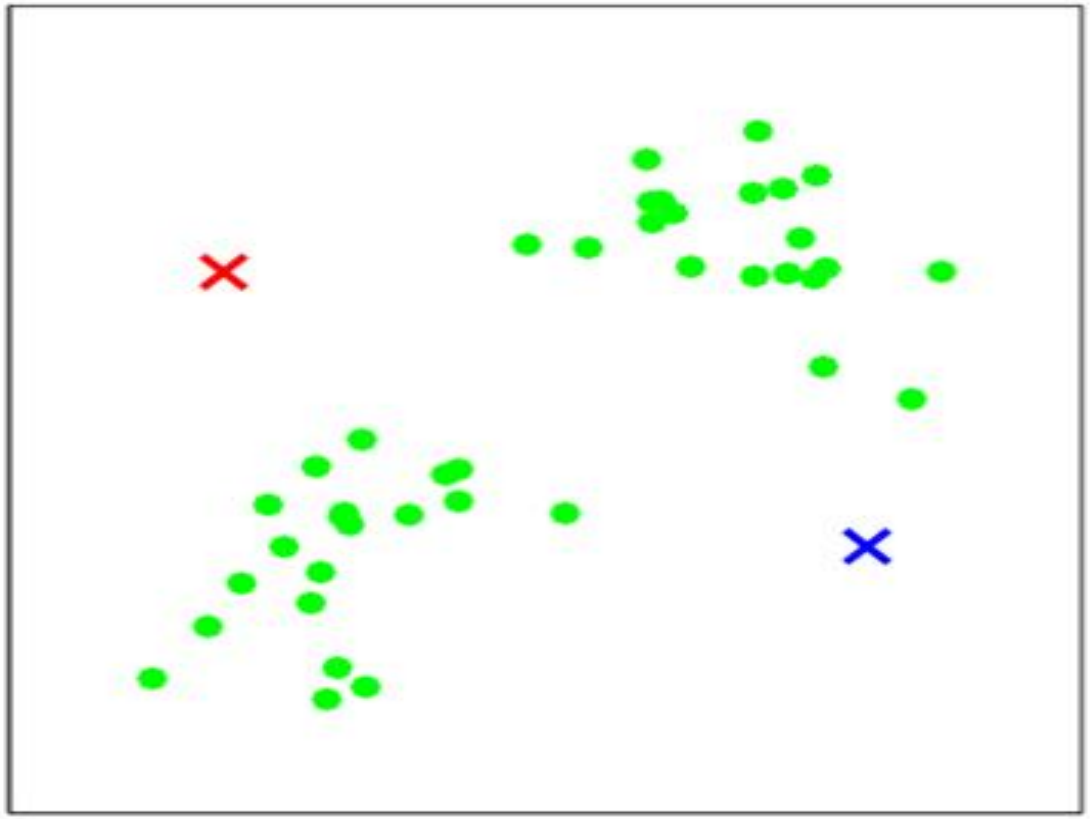
   $$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}.$$

   }

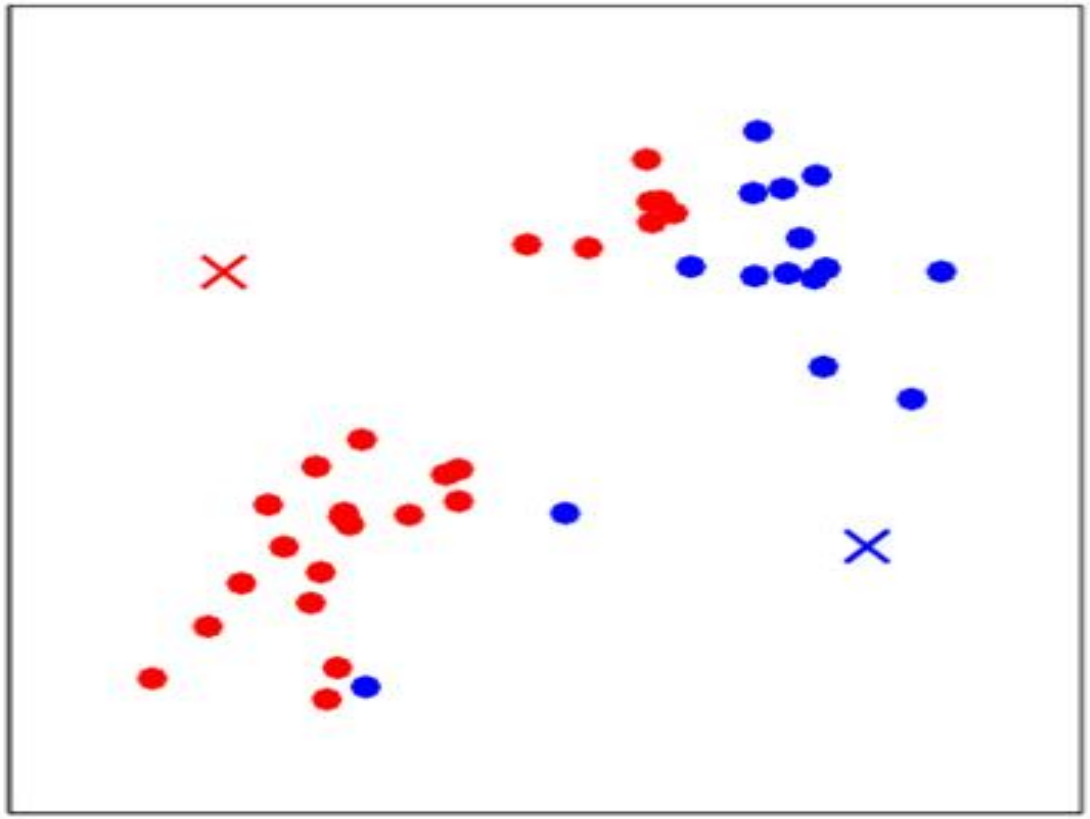
Example of K-mean clustering



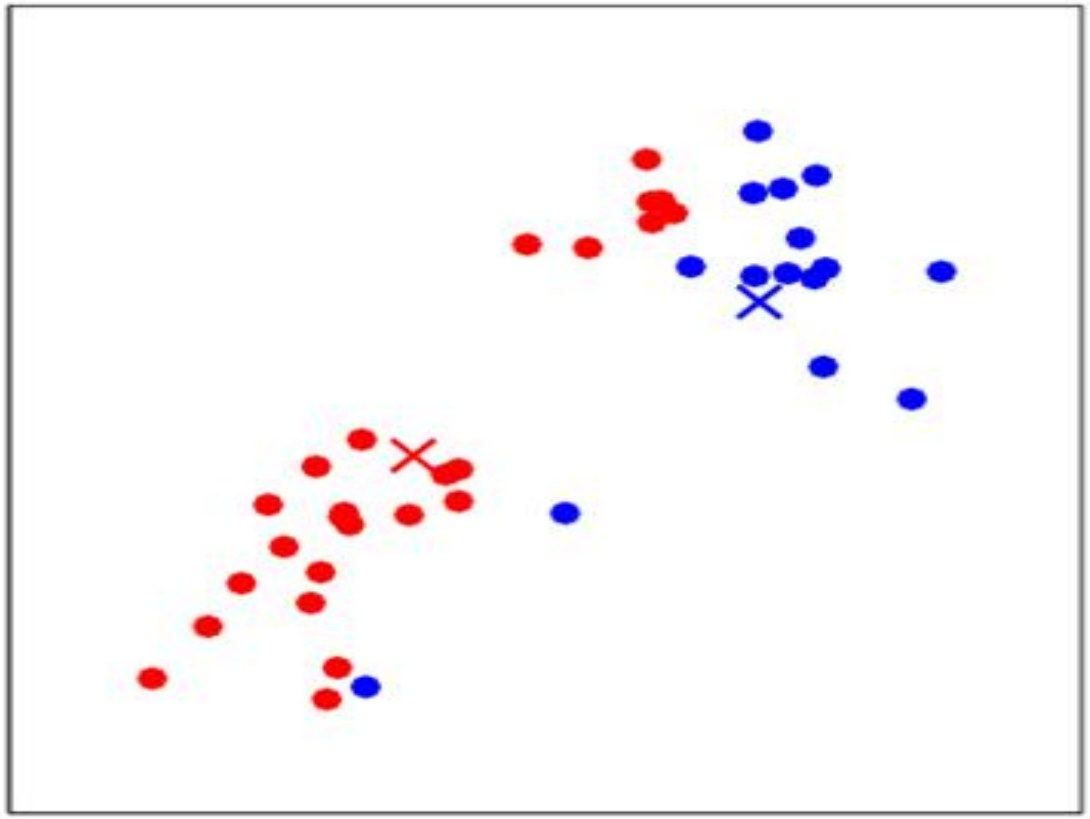
Example of K-mean clustering



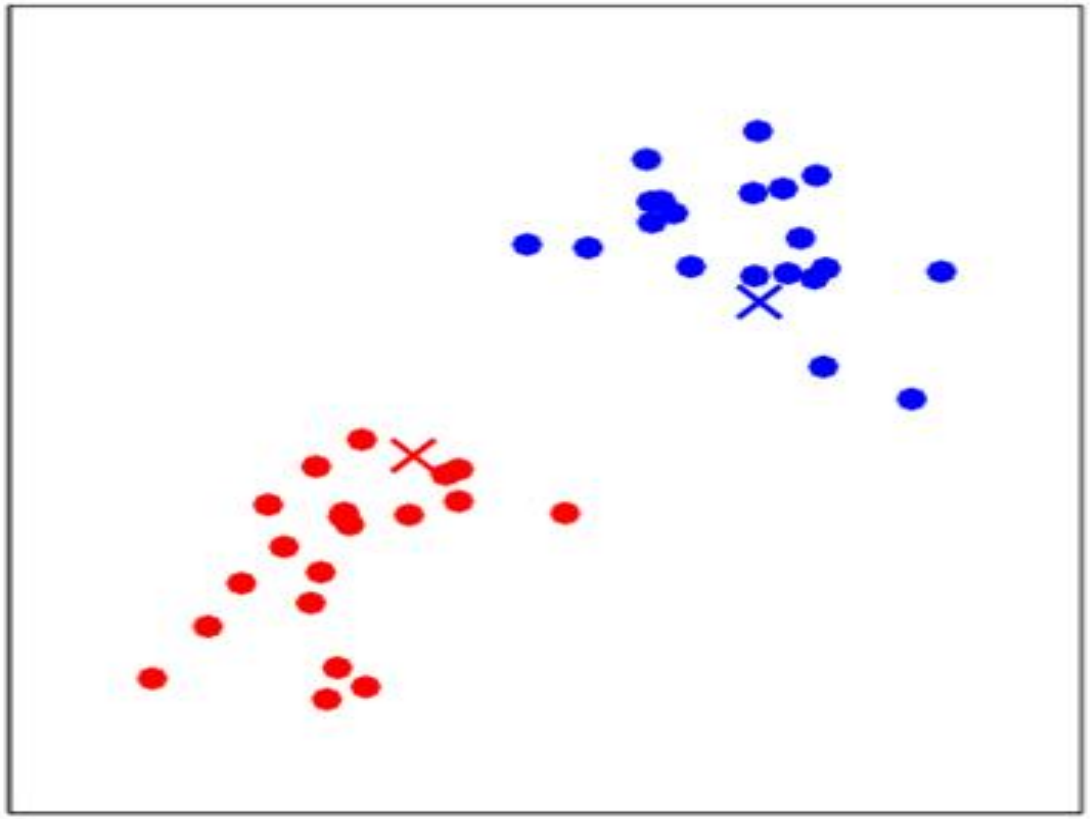
Example of K-mean clustering



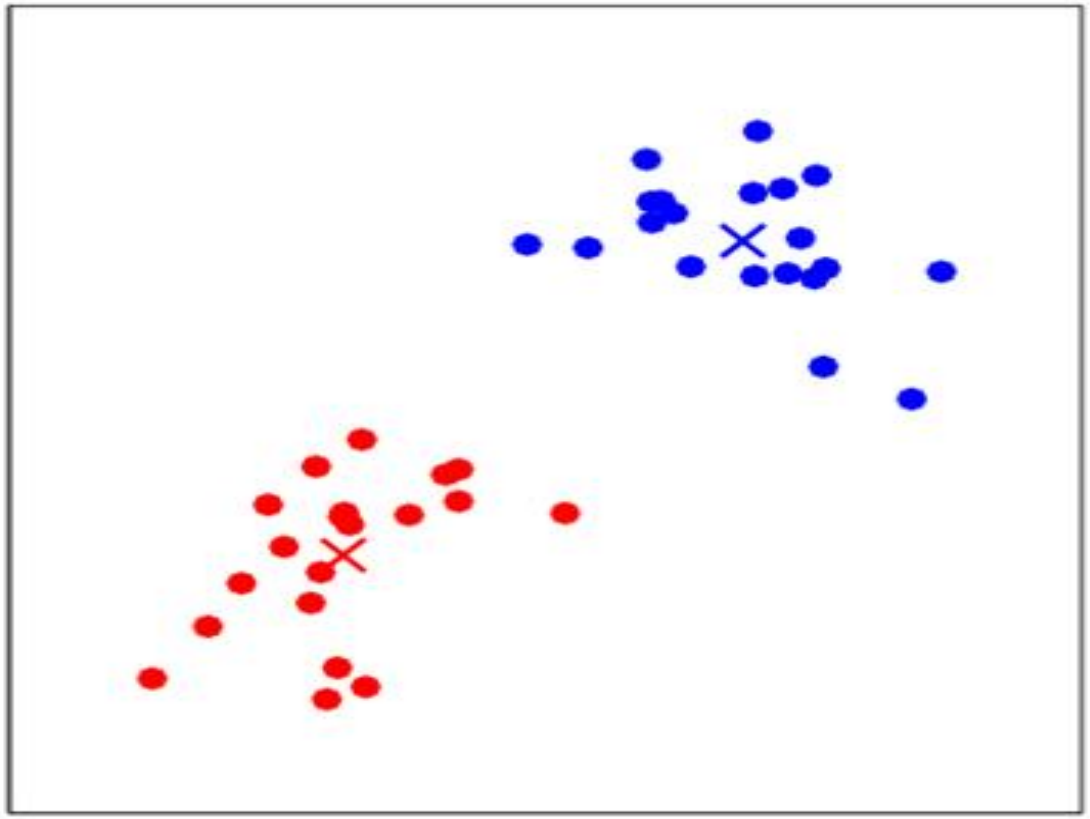
Example of K-mean clustering



Example of K-mean clustering



Example of K-mean clustering



Example of K-mean clustering

- In the Figure above for K-means algorithm: Training examples are shown as dots, and cluster centroids are shown as crosses.
- (a) Original dataset.
- (b) Random initial cluster centroids (in this instance, not chosen to be equal to two training examples).
- (c-f) Illustration of running two iterations of k-means.

- In each iteration, we assign each training example to the closest cluster centroid (shown by “painting” the training examples the same color as the cluster centroid to which is assigned); then we move each cluster centroid to the mean of the points assigned to it. (Best viewed in color.)
- Images courtesy Michael Jordan.

Q: Is the k-means algorithm guaranteed to converge?

- Yes it is, but it might convergent to a local optimization point instead a global one in following sense.

- Define the distortion function to be: $$J(c, \mu) = \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

- Here J measures the sum of squared distances between each training example $x^{(i)}$ and the cluster centroid $\mu_{c(i)}$ to which it has been assigned.
- It can be shown that k-means is exactly coordinate descent on J.
- This means that the inner-loop of k-means repeatedly minimizes J with respect to c while holding μ fixed, and then minimizes J with respect to μ while holding c fixed.
- Thus, J must monotonically decrease, and the value of J must converge. (Usually, this implies that c and μ will converge too.)
- In theory, it is possible k-means to oscillate between a few different clusterings—i.e., a few different values for c and/or μ—that have exactly the same value of J, but this almost never happens in practice.)

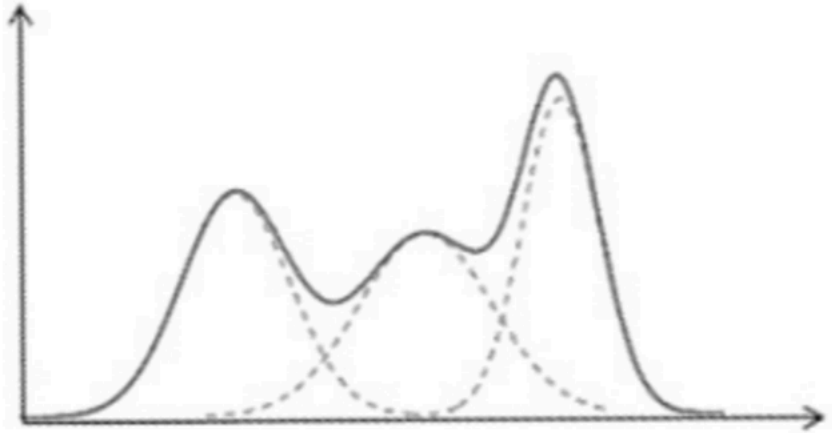
Note: The distortion function J is non-convex, so no global minimum is guaranteed.

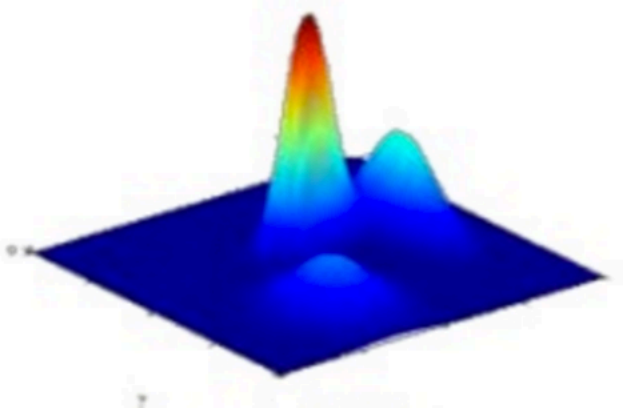
- That is to say the coordinate descent on J is not guaranteed to converge to the global minimum: k-means can be susceptible to local optima.
- But very often k-means will work fine and come up with very good clusterings despite this.
- Try heuristic method if you are worried about getting stuck in bad local minima:
- Run k-means many times (using different random initial values for the cluster centroids μ_j).
- Then, out of all the different clusterings found, select

  the one that gives the lowest distortion J(c, μ).

Change gears:

Multivariate Gaussian Mixture Model





Mimic linear combination of vectors, here is a convex combination.

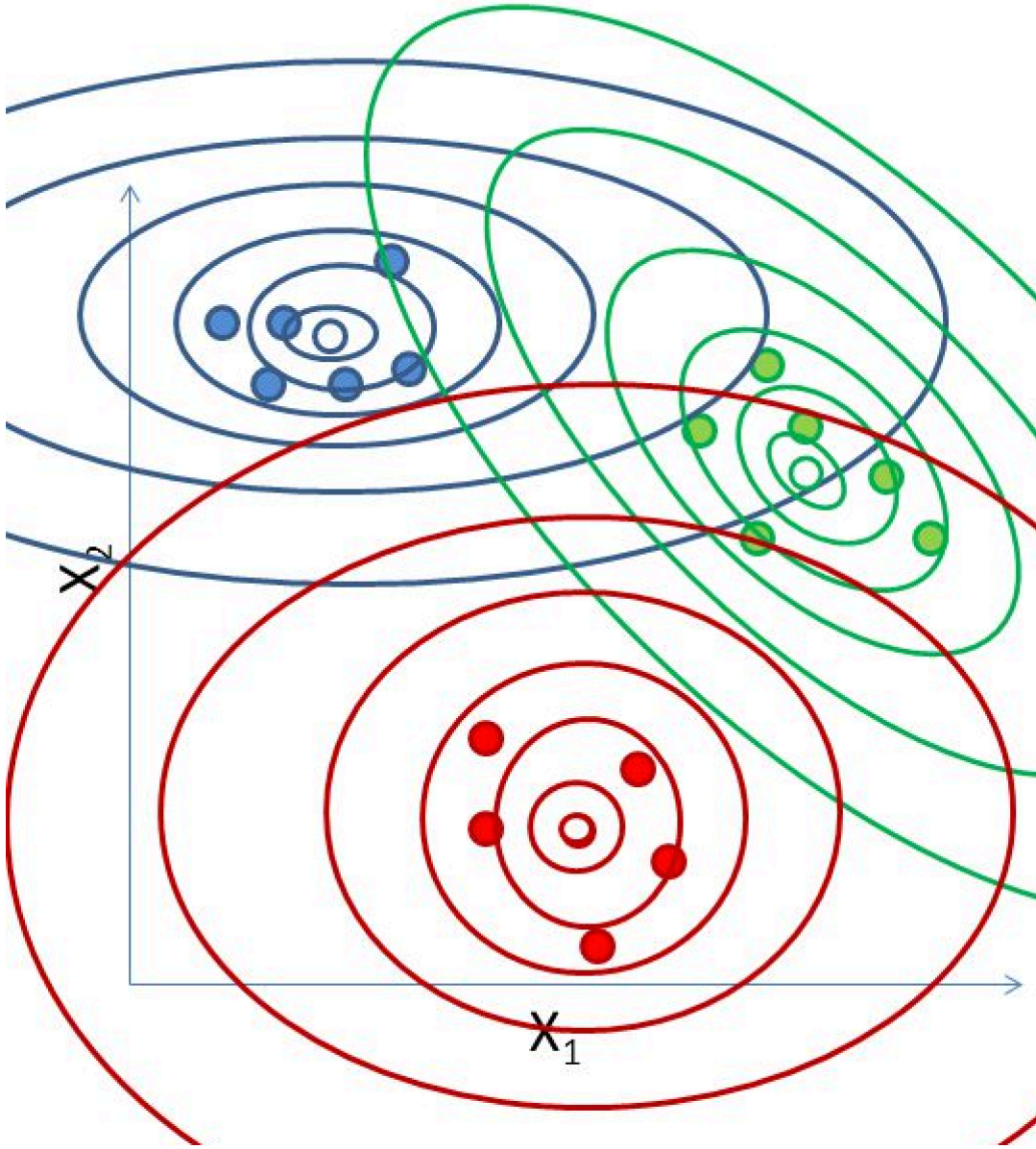
$$p(\boldsymbol{\theta}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$$

$$\phi_j \geq 0, \sum_{j=1}^{k} \phi_j = 1$$

where the i^{th} vector component is characterized by normal distributions

weights ϕ_i, means $\boldsymbol{\mu_i}$ and covariance matrices $\boldsymbol{\Sigma_i}$.

Gaussian Mixture Models



Like K-Means, GMM clusters have centers.

In addition, they have probability distributions that indicate the probability that a point belongs to the cluster.

These ellipses show "level sets": lines with equal probability of belonging to the cluster.

Notice that green points still have SOME probability of belonging to the blue cluster, but it's much lower than the blue points.

This is a more complex model than K-Means: distance from the center can matter more in one direction than another.

How Gaussian mixture model and Expectation-Maximization (EM) related?

Key: the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$ is *also* a Gaussian mixture mode **!**

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \sum_{i=1}^{K} \tilde{\phi}_i \mathcal{N}(\tilde{\boldsymbol{\mu}}_i, \tilde{\boldsymbol{\Sigma}}_i)$$

with new parameters $\tilde{\phi}_i$, $\tilde{\boldsymbol{\mu}}_i$ and $\tilde{\boldsymbol{\Sigma}}_i$ that are updated using the EM algorithm.

What is an EM algorithm?

EM = Expectation-Maximization

The EM (Expectation-Maximization) Algorithm

- Expectation of what?
- Maximization of what?

- *Work out details with students on the board.*

EM (Expectation-Maximization) Algorithm

- Given a training set $\{x^{(1)}, \ldots, x^{(m)}\}$
- Note: since we are in the unsupervised learning setting, so these points do not come with any labels.
- Goal: Model the data by specifying a joint distribution

$$p(x^{(i)}, z^{(i)}) = p(x^{(i)}|z^{(i)})p(z^{(i)}).$$

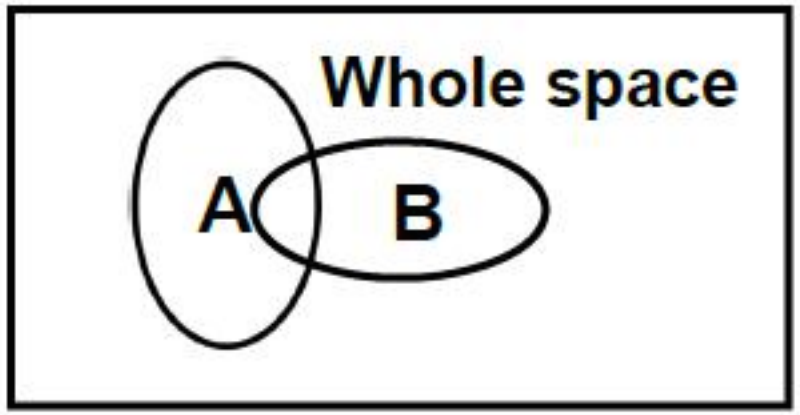
Here, $z^{(i)} \sim \text{Multinomial}(\phi)$ $\phi_j \geq 0, \sum_{j=1}^{k} \phi_j = 1$

and the parameter ϕ_j gives $p(z^{(i)} = j$

and $x^{(i)}|z^{(i)} = j \sim \mathcal{N}(\mu_j, \Sigma_j)$

k denote the number of values that the $z^{(i)}$'s can take on.

Visualize Bayes' Theorem



$$P(A) = \frac{\text{A}}{\text{Whole space}} \qquad P(B) = \frac{\text{B}}{\text{Whole space}}$$

$$P(A|B) = \frac{\text{A and B}}{\text{B}} \qquad P(B|A) = \frac{\text{A and B}}{\text{A}}$$

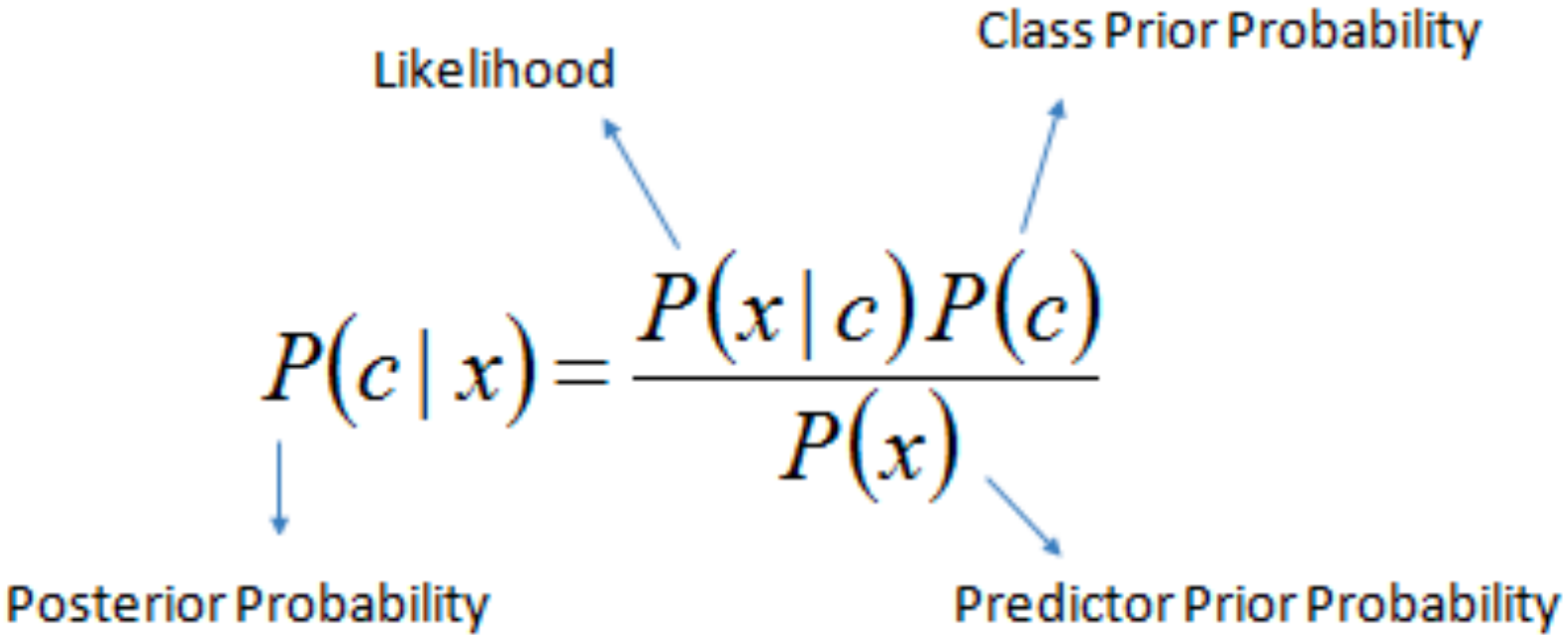
$$P(A \cap B) = \frac{\text{A and B}}{\text{Whole space}}$$

$$P(A) \times P(B|A) = \frac{\text{A}}{\text{Whole space}} \times \frac{\text{A and B}}{\text{A}} = \frac{\text{A and B}}{\text{Whole space}} = P(A \cap B)$$

$$P(B) \times P(A|B) = \frac{\text{B}}{\text{Whole space}} \times \frac{\text{A and B}}{\text{B}} = \frac{\text{A and B}}{\text{Whole space}} = P(A \cap B)$$

$$\Rightarrow P(B|A) = P(A|B) \times P(B) / P(A)$$

Recall:



$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Recall:
Conditional Normal/Gaussian Distribution

Conditional distributions [edit]

If *N*-dimensional **x** is partitioned as follows

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \text{ with sizes } \begin{bmatrix} q \times 1 \\ (N-q) \times 1 \end{bmatrix}$$

and accordingly **μ** and **Σ** are partitioned as follows

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} \text{ with sizes } \begin{bmatrix} q \times 1 \\ (N-q) \times 1 \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \text{ with sizes } \begin{bmatrix} q \times q & q \times (N-q) \\ (N-q) \times q & (N-q) \times (N-q) \end{bmatrix}$$

then the distribution of $\mathbf{x}_1$ conditional on $\mathbf{x}_2 = \mathbf{a}$ is multivariate normal ($\mathbf{x}_1 \mid \mathbf{x}_2 = \mathbf{a}$) ~ $N(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}})$ where

$$\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\left(\mathbf{a} - \boldsymbol{\mu}_2\right)$$

and covariance matrix

$$\overline{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}.$$ [18]

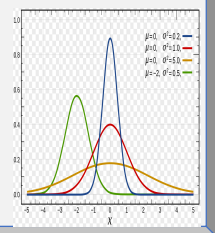
A Big Picture of Probability Theory

$$0 \le P(X) \le 1$$
$$\sum P(X) = 1$$

Key Characteristics:

Single rv	Muliti-rv
E(X) & Condit'l Expec'n	Cov (X, Y)
Variance/Stan. Devi.	Corrl(X, Y)
Moments	Cov. Matrix
Skewness etc.	Corrl Matrix

Probability Distributions (Discrete & Continuous) **and their Geometric Meanings**



Other known distrib'ns
Bernoulli
Beta $\theta \sim \text{Beta}(a, b)$
Chi-square
Poisson
Student's t
Uniform

Gaussian Distrib.

Discrete distrib'n → Taking limit → Conti. distrib'n

Probability Rules for Events:
Product rule/iid
Joint probability
Conditional Independence

	Discrete	Continuous
Single-rv	Binomial: $\binom{n}{k} p^k (1-p)^{n-k}$	Gaussian/Normal: $\frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
Multi-rv	Multinomial: $\frac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}$	Multivari-Gaussian: $(2\pi)^{-\frac{1}{2}k} \lvert\Sigma\rvert^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)}$

Condi. Prob & **Bayesian Rules**

$$p(A|B) = \frac{p(A,B)}{p(B)} \text{ if } p(B) > 0$$
$$p(X = x|Y = y) = \frac{p(X = x, Y = y)}{p(Y = y)}$$
$$= \frac{p(X = x)p(Y = y|X = x)}{\sum_{x'} p(X = x')p(Y = y|X = x')}$$

Besides **pmf/pdf,** + 3 key fcns:
- **cdf** (cumulative distri. fcn)
- **cf** (characteristic fcn $E(e^{itX})$)
- **mgf** (moment generating fcn) $m_X(t) = E(e^{tX})$

- **Central Limit Theorem**

Other Key Tech: Making connection to derivative/Jacobian/integrations.

Key: View everything as functions. P eats an observation x of a random variable X and spits out a value P(X=x) in [0,1], & the sum of all p(x) is 1.
- ***X is a random variable. P(X=x) = p(x).***

Like the variables in calculus, we can add, subtract, make linear combinations; or make new functions f(x), also can take derivatives/integrations.

X→ f(X). For e.g.s
$f(X) = \sum a_i X_i$
$f(\mathbf{X}) = A\mathbf{X} + b$
$f(X) = X^n$
f(X) = Taylor exp.

what is *E*(f(X))?

$$\mathbf{y} = f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$$
$$\mathbb{E}[\mathbf{y}] = \mathbb{E}[\mathbf{A}\mathbf{x} + \mathbf{b}] = \mathbf{A}\boldsymbol{\mu} + \mathbf{b}$$
$$\text{cov}[\mathbf{y}] = \text{cov}[\mathbf{A}\mathbf{x} + \mathbf{b}] = \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T$$
$$p_y(\mathbf{y}) = p_x(\mathbf{x}) \left| \det \left(\frac{\partial \mathbf{x}}{\partial \mathbf{y}} \right) \right| = p_x(\mathbf{x}) \left| \det \mathbf{J}_{\mathbf{y} \to \mathbf{x}} \right|$$
$$\therefore \mathbf{y} = f(\mathbf{x})$$

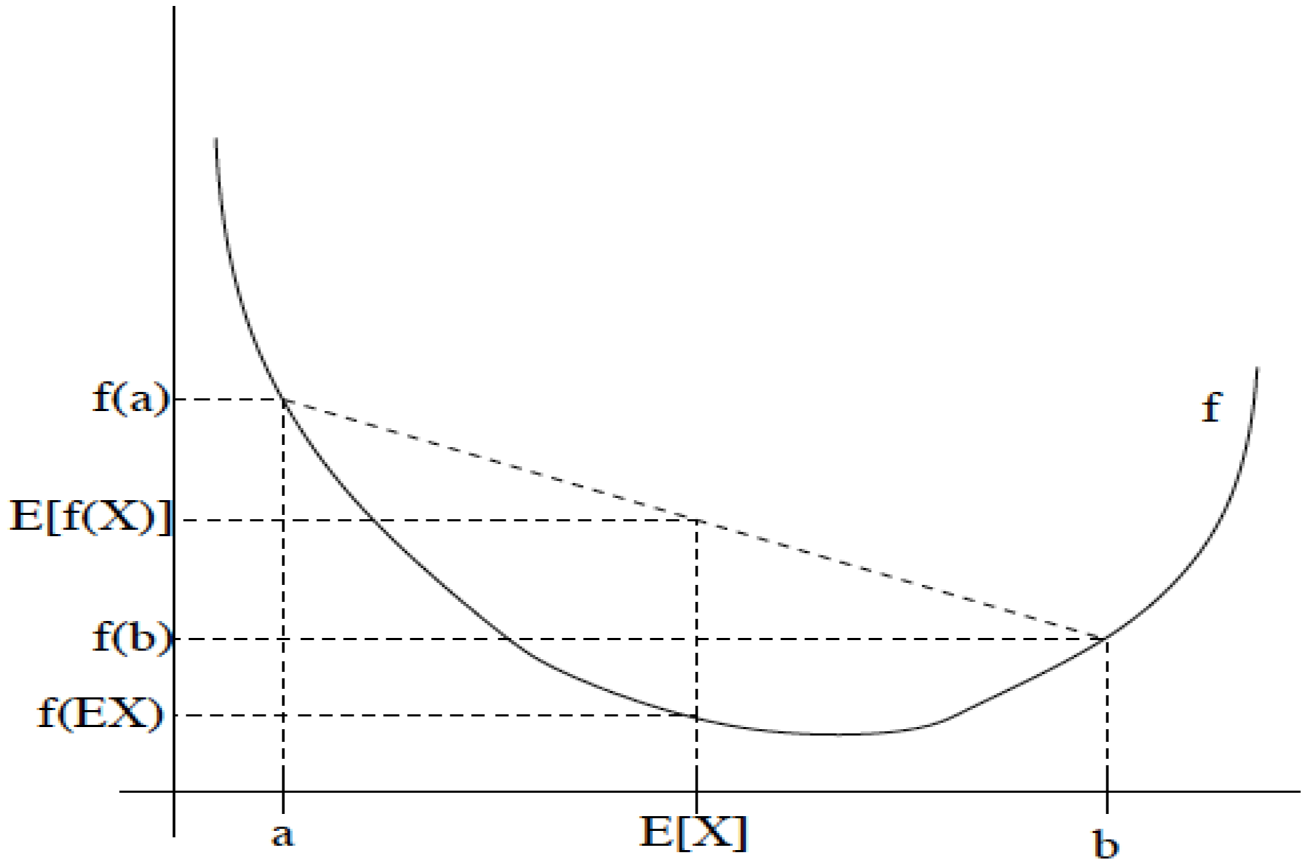
- The parameters of our model are thus ϕ, μ and $\sum$. To estimate them, write:

Jensen's inequality

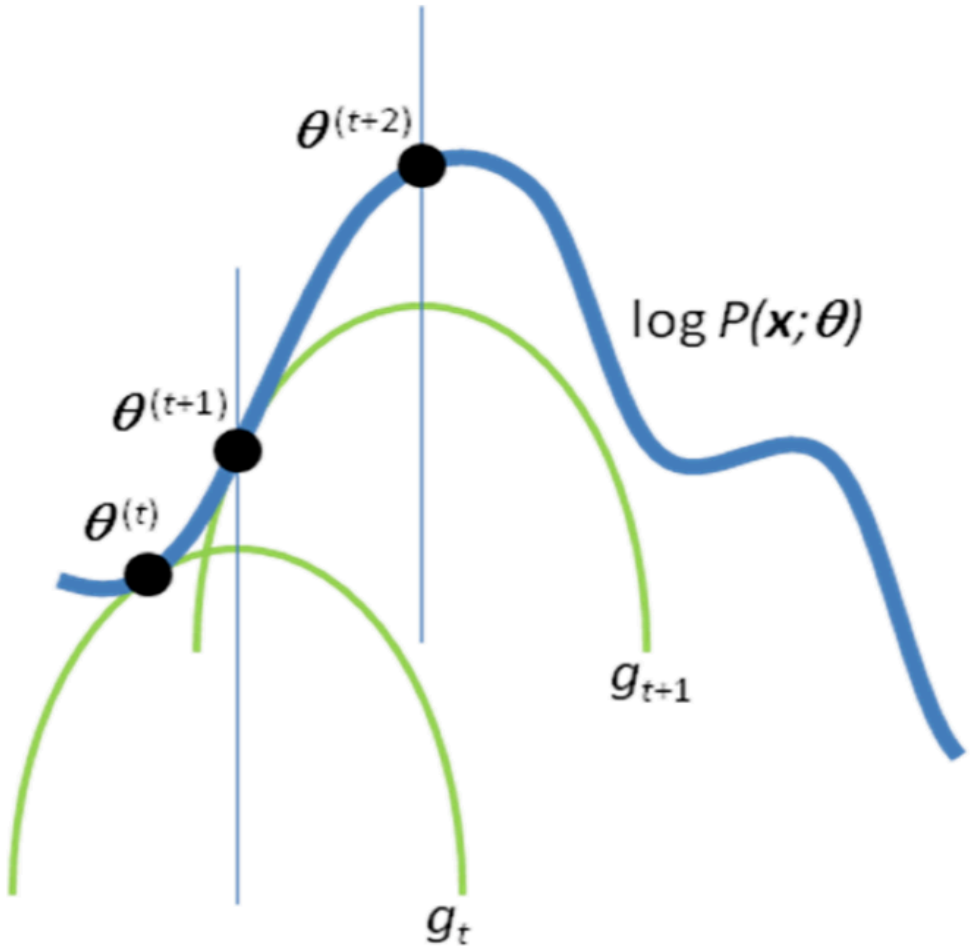
Theorem. Let f be a convex function, and let X be a random variable. Then:

$$\mathrm{E}[f(X)] \geq f(\mathrm{E}X).$$

Moreover, if f is strictly convex, then $\mathrm{E}[f(X)] = f(\mathrm{E}X)$ holds true if and only if $X = \mathrm{E}[X]$ with probability 1 (i.e., if X is a constant).

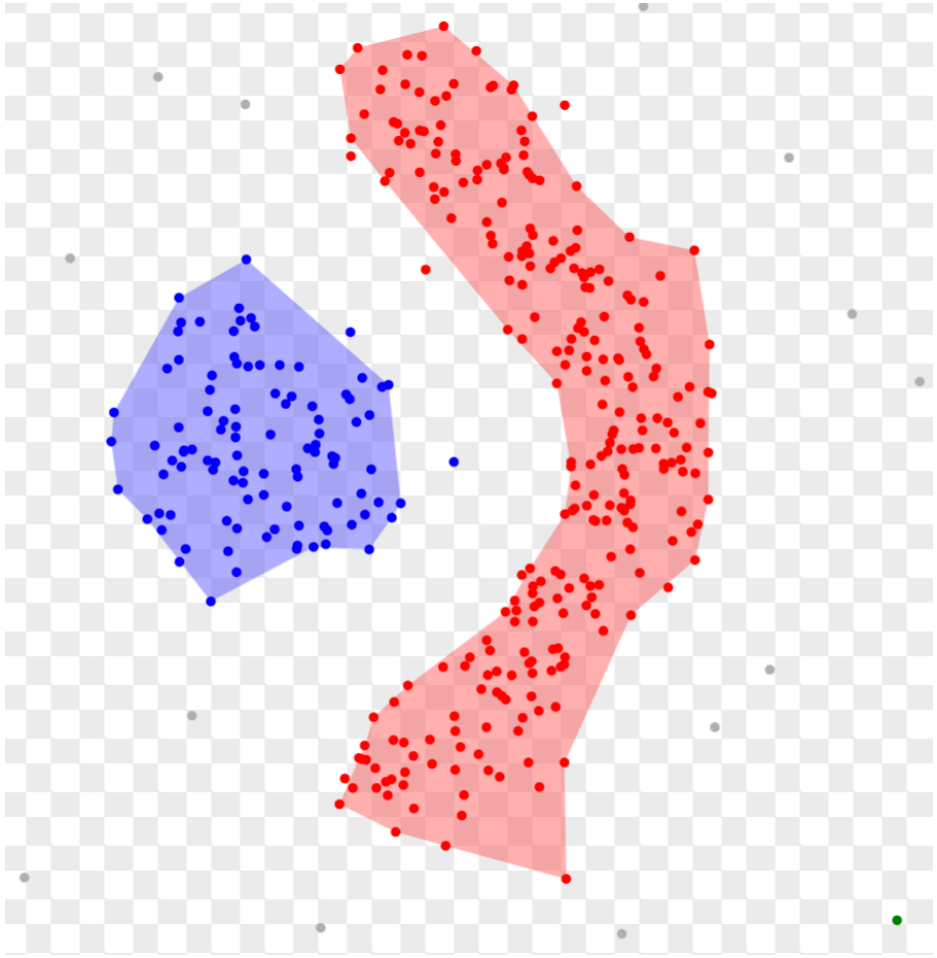


Geometry of EM algorithm

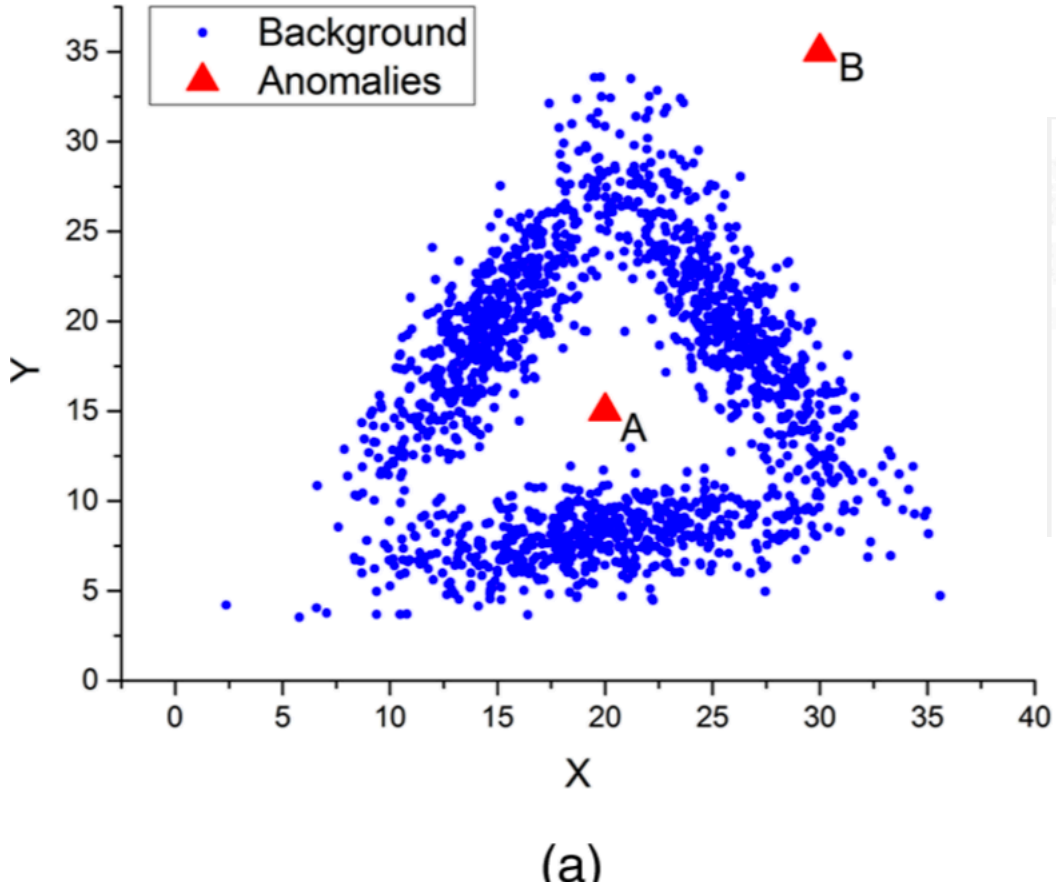


Supplementary Figure 1 Convergence of the EM algorithm. Starting from initial parameters $\theta^{(t)}$, the E-step of the EM algorithm constructs a function g_t that lower-bounds the objective function $\log P(\mathbf{x};\theta)$. In the M-step, $\theta^{(t+1)}$ is computed as the maximum of g_t. In the next E-step, a new lower-bound g_{t+1} is constructed; maximization of g_{t+1} in the next M-step gives $\theta^{(t+2)}$, etc.

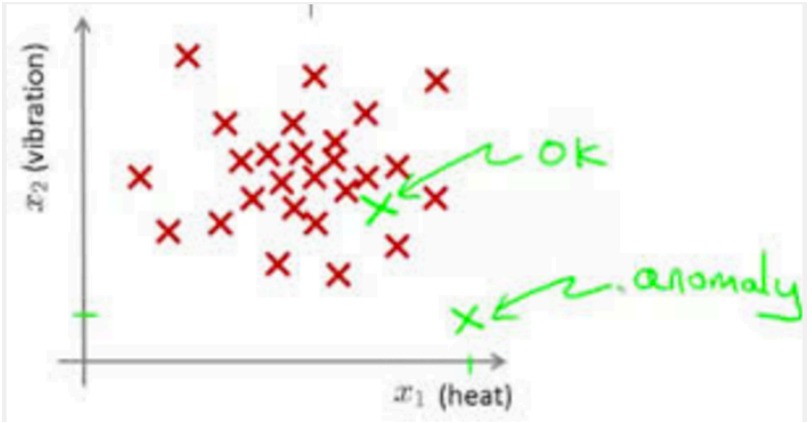
Density estimation using EM Algorithm



Anomaly Detection using Density Estimation



(a)



Recall: MLE =Maximum Likelihood Estimation

- In statistics, maximum likelihood estimation (MLE) is a method of estimating the parameters of a statistical model given observations, by finding the parameter values that maximize the likelihood of making the observations given the parameters.

Recall: MLE = Maximum Likelihood Estimate

Assume that we want to estimate an unobserved population parameter θ on the basis of observations x. Let f be the sampling distribution of x, so that $f(x \mid \theta)$ is the probability of x when the underlying population parameter is θ. Then the function:

$$\theta \mapsto f(x \mid \theta)$$

is known as the likelihood function and the estimate:

$$\hat{\theta}_{\mathrm{ML}}(x) = \arg\max_{\theta} f(x \mid \theta)$$

is the maximum likelihood estimate of θ.

Recall: MAP

- Maximum a posteriori (MAP) estimation is a model of posterior distribution.
- The MAP can be used to obtain a point estimate of an unobserved quantity on the basis of empirical data.

Now assume that a prior distribution g over θ exists. This allows us to treat θ as a random variable as in Bayesian statistics. We can calculate the posterior distribution of θ using Bayes' theorem:

$$\theta \mapsto f(\theta \mid x) = \frac{f(x \mid \theta)\, g(\theta)}{\displaystyle\int_{\vartheta \in \Theta} f(x \mid \vartheta)\, g(\vartheta)\, d\vartheta}$$

where g is density function of θ, Θ is the domain of g.

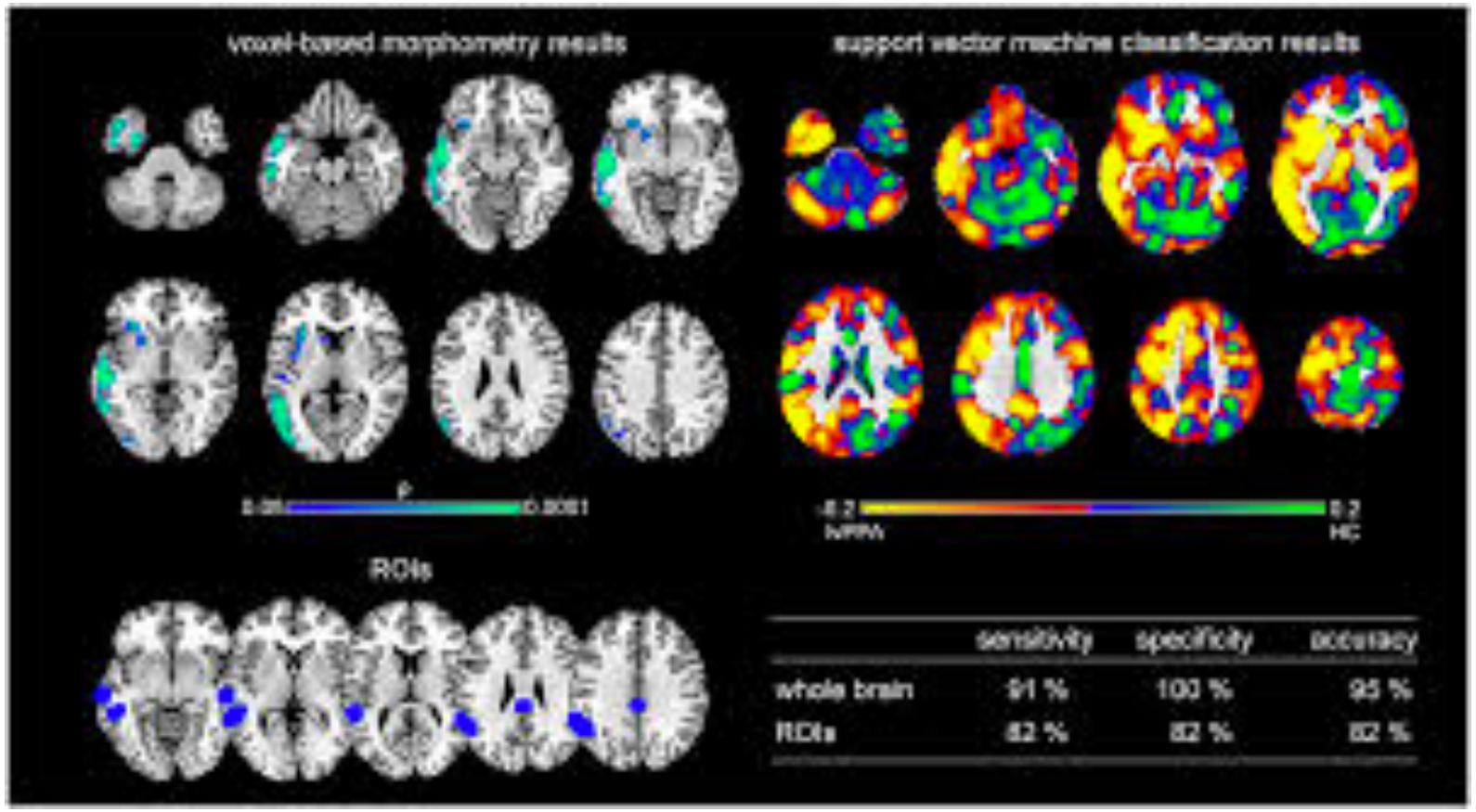
The method of maximum a posteriori estimation then estimates θ as the mode of the posterior distribution of this random variable:

$$\hat{\theta}_{\mathrm{MAP}}(x) = \arg\max_{\theta} f(\theta \mid x) = \arg\max_{\theta} \frac{f(x \mid \theta)\, g(\theta)}{\displaystyle\int_{\vartheta} f(x \mid \vartheta)\, g(\vartheta)\, d\vartheta} = \arg\max_{\theta} f(x \mid \theta)\, g(\theta).$$

The denominator of the posterior distribution (so-called marginal likelihood) does not depend on θ

Applications

[PDF]Predicting primary progressive aphasias with support vector ...



Contents lists available at ScienceDirect

NeuroImage: Clinical

journal homepage: www.elsevier.com/locate/ynicl

Predicting primary progressive aphasias with support vector machine approaches in structural MRI data

Sandrine Bisenius[a,*], Karsten Mueller[a], Janine Diehl-Schmid[b], Klaus Fassbender[c], Timo Grimmer[b], Frank Jessen[d], Jan Kassubek[e], Johannes Kornhuber[f], Bernhard Landwehrmeyer[e], Albert Ludolph[e], Anja Schneider[g], Sarah Anderl-Straub[e], Katharina Stuke[a], Adrian Danek[h], Markus Otto[e], Matthias L. Schroeter[a], &, FTLDc study group:

[a]Max Planck Institute for Human Cognitive and Brain Sciences & Clinic for Cognitive Neurology, University Hospital Leipzig, Germany
[b]Clinic and Polyclinic for Psychiatry & Psychotherapy, Technical University Munich, Germany
[c]Clinic and Polyclinic for Neurology, Saarland University Homburg, Germany
[d]Clinic and Polyclinic for Psychiatry and Psychotherapy, University of Bonn, Germany
[e]Department of Neurology, University of Ulm, Germany
[f]Clinic for Psychiatry and Psychotherapy, Friedrich-Alexander University Erlangen-Nuremberg, Germany
[g]Clinic for Psychiatry and Psychotherapy, University of Goettingen, Germany
[h]Clinic of Neurology, Ludwig Maximilian University of Munich, Germany

ARTICLE INFO

Article history:
Received 14 September 2016
Received in revised form 27 January 2017
Accepted 3 February 2017
Available online 06 February 2017

Keywords:
Grey matter
Multi-center
Primary progressive aphasia
Support vector machine classification
Whole brain approach

ABSTRACT

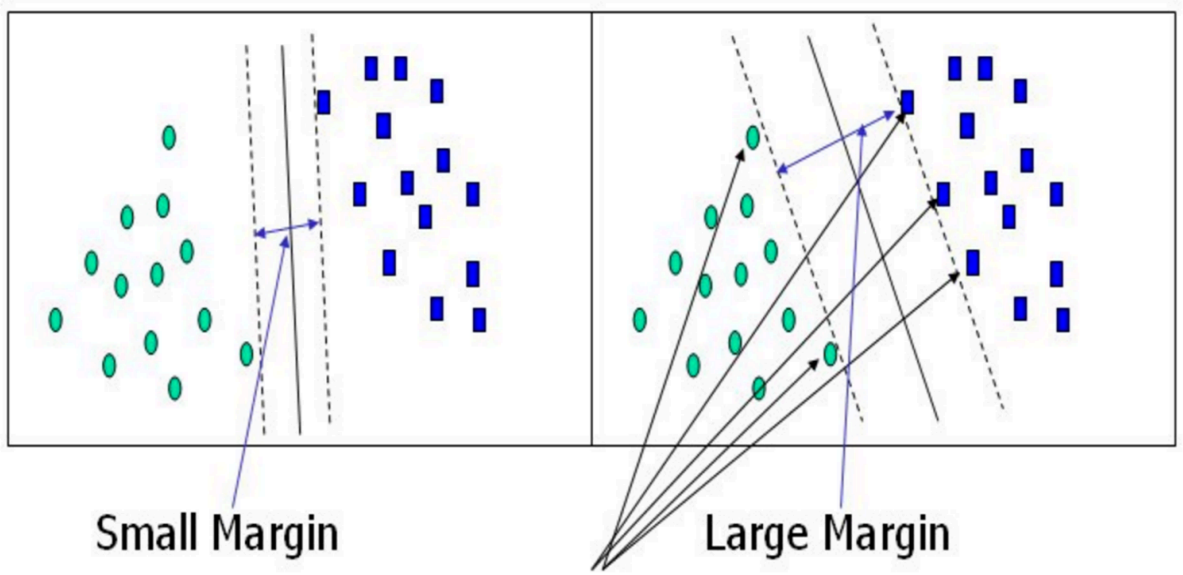
Primary progressive aphasia (PPA) encompasses the three subtypes nonfluent/agrammatic variant PPA, semantic variant PPA, and the logopenic variant PPA, which are characterized by distinct patterns of language difficulties and regional brain atrophy. To validate the potential of structural magnetic resonance imaging data for early individual diagnosis, we used support vector machine classification on grey matter density maps obtained by voxel-based morphometry analysis to discriminate PPA subtypes (44 patients: 16 nonfluent/agrammatic variant PPA, 17 semantic variant PPA, 11 logopenic variant PPA) from 20 healthy controls (matched for sample size, age, and gender) in the cohort of the multi-center study of the German consortium for frontotemporal lobar degeneration. Here, we compared a whole-brain with a meta-analysis-based disease-specific regions-of-interest approach for support vector machine classification. We also used support vector machine classification to discriminate the three PPA subtypes from each other. Whole brain support vector machine classification enabled a very high accuracy between 91 and 97% for identifying specific PPA subtypes vs. healthy controls, and 78/95% for the discrimination between semantic variant vs. nonfluent/agrammatic or logopenic PPA variants. Only for the discrimination between nonfluent/agrammatic and logopenic PPA variants accuracy was low with 55%. Interestingly, the regions that contributed the most to the support vector machine classification of patients corresponded largely to the regions that were atrophic in these patients as revealed by group comparisons. Although the whole brain approach took also into account regions that were not covered in the regions-of-interest approach, both approaches showed similar accuracies due to the disease-specificity of the selected networks. Conclusion, support vector machine classification of multi-center structural magnetic resonance imaging data enables prediction of PPA subtypes with a very high accuracy paving the road for its application in clinical settings.

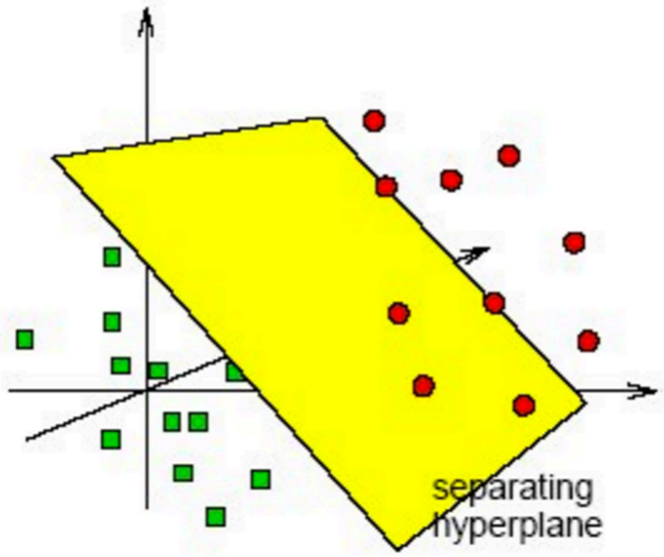
MRI Data

file:///Users/weiqinggu/Downloads/predicting-primary-progressive-aphasias-with-support-vector-machine-approaches-in-structural-mri-data%20(1).pdf

What is support vector machine?

Support Vector Machine (SVM) conti.





- Key: Gain a geometric intuition:
- ***Want to maximize the margin to increase the confidence of your prediction.***
- ***SVM is a typical example of Machine learning from of Geometric perspective.***
- There is a procedure for geometric approaches.

Summarize the Geometric Approach

- Gain a geometric intuition: For SVM--Want to maximize the margin to increase the confidence of your prediction.
- Write down your geometrical intuition Mathematically.
- Come up with a cost function: Here is the margin. Want to maximize the margin.
- Figure out what are the constraints involved.
- Make the problem into a convex problem (if possible to do so) if the optimization you formed is not convex.
- For SVM, we will make it into a quadratic program problem.
- Use exiting QP software package to solve the problem.
- Use different way to solve it. (Later use a dual method.)
- Generate to solve more complicated problems/cases.

We will deal with the case when data is not linear separable later. Just need to make a clever "transformation".