ECE M146 Introduction to Machine Learning

Lecture 13 - Spring 2021

Prof. Lara Dolecek

ECE Department, UCLA

Today's Lecture

Recap:

- Gaussian Discriminant Analysis – scalar case

New topic:

- Gaussian Discriminant Analysis – vector case
- Multivariate Gaussian RVs

Today's Lecture

Recap:

- Gaussian Discriminant Analysis – scalar case

New topic:

- Gaussian Discriminant Analysis – vector case
- Multivariate Gaussian RVs

Recap: GDA in the scalar case

- Instance of probabilistic generative modeling. $P(X, Y) = P(Y)P(X|Y)$
- Class marginals: $Y = \begin{cases} 1 & \text{w.p. } \theta \\ 0 & \text{w.p. } 1-\theta \end{cases}$
- Class conditionals:

$$X \mid Y = 0 \sim \mathcal{N}(\mu_0, \sigma_0^2)$$
$$X \mid Y = 1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$$

- Estimate the parameters by taking the derivatives of the log of the joint.

$$\theta, \mu_1, \mu_0, \sigma_1^2, \sigma_0^2$$

Parameter estimation

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

look at N points

$$\prod_{i=1}^{N} P(x_i, y_i) = \prod_{i=1}^{N} \theta^{1[y_i=1]} \cdot (1-\theta)^{1[y_i=0]} \cdot \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}}\right)^{1[y_i=1]} \cdot \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu_0)^2}{2\sigma_2^2}}\right)^{1[y_i=0]}$$

Parameter estimation

$$\hat{\theta} = \frac{N_1}{N} \qquad N_1 = \sum_{i=1}^{n} \mathbb{1}[y_i = 1]$$

$$\hat{\mu}_1 = \frac{1}{N_1} \sum_{i=1}^{n} \mathbb{1}[y_i = 1] \cdot x_i$$

$$\hat{\sigma}_1 = \frac{1}{N_1} \sum_{i=1}^{n} \mathbb{1}[y_i = 1] \, (x_i - \hat{\mu}_1)^2$$

At test time

consider a new test point x:

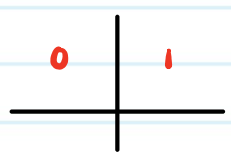
$$\frac{P(Y=1 \mid X_{TEST})}{P(Y=0 \mid X_{TEST})} \gtrless 1$$

class 1 (≥)

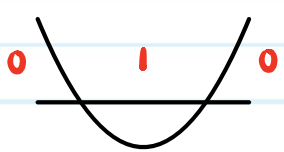
class 0 (<)

decision boundary

equal variances
σ^2



unequal variances
σ_1^2, σ_0^2



Today's Lecture

Recap:

- Gaussian Discriminant Analysis – scalar case

New topic:

- Multivariate Gaussian RVs
- Gaussian Discriminant Analysis – vector case

Now, we generalize classification to vector input data

- Notation: $X \mid Y=1 \sim \mathcal{N}(\mu_1, \Sigma_1)$
  $X \mid Y=0 \sim \mathcal{N}(\mu_0, \Sigma_0)$

Conditional pdf is modeled as jointly Gaussian.

$Z \sim \mathcal{N}(\mu, \Sigma)$

μ is vector of component-wise individual means

Σ is square matrix of pairwise covariances

Recall Gaussian distribution – scalar case

- Mean and variance

$$f_x(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\mu = E[x]$$

$$\sigma^2 = var(x) = E[(x-\mu)^2]$$

Multivariate Gaussian Distribution

- Mathematical expression

$$f_x(X|Y=1) = \frac{1}{(2\pi)^{d/2}} \cdot \frac{1}{(\det \Sigma)^{1/2}} e^{\underbrace{-\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1}(x-\mu_1)}_{\text{scalar}}}$$

for $d=1$: $f_x(X) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}$

$$\Sigma = \sigma^2$$

$$\Sigma^{-1} = \frac{1}{\sigma^2}$$

More on Jointly Gaussian RVs

- For the vector case, mean and covariance matrix represent:

$$\mu = \begin{bmatrix} E(X_1) \\ E(X_2) \\ \vdots \\ E(X_d) \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_d \end{bmatrix} \qquad X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_d \end{bmatrix}$$

$d \times 1$ $\qquad$ $d \times 1$

Recover the scalar case

$$cov(X_i, X_i) = var(X_i)$$

$$\Sigma = \begin{bmatrix} var(X_1) & cov(X_1, X_2) & \cdots & cov(X_1, X_d) \\ cov(X_2, X_1) & var(X_2) & & \\ \vdots & & \ddots & \vdots \\ cov(X_d, X_1) & & & var(X_d) \end{bmatrix}$$

$d \times d$

$$cov(X_i, X_j) = E[(X_i - m_i)(X_j - m_j)]$$

General properties of a covariance matrix (not just for jointly Gaussian!)

- Any collection of variables

$$z_1, \ldots, z_m$$

$$\Sigma_z \, \mathrm{cov}(z_1, \ldots, z_m)$$

matrix s.t.

$$\Sigma_z(i,j) = E[(z_i - E(z_i))(z_j - E(z_j))]$$
$$= \Sigma_z(j,i)$$

General properties of a covariance matrix

- Positive semi-definite
  - symmetric
  - square

$x^T A x \geq 0 \quad \forall \; x$

↑ PSD

Back to Jointly Gaussian RVs

- Properties of the covariance matrix
- Positive definite so invertible.

Σ is PSD

$\forall\ v,\ v \neq 0,\quad v^T \Sigma v \geq 0$

$\rightarrow \Sigma^{-1}$ exists

- Determinant is positive so its square root is ok.

More on the Jointly Gaussian RVS

- Special case of covariance matrix being diagonal.

$$\Sigma = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \sigma_2^2 & \\ 0 & & \ddots \\ & & & \sigma_d^2 \end{bmatrix}$$

$$\text{cov}(x_i, x_j) = 0 \quad \forall \quad i \neq j$$

$$\Sigma^{-1} = \text{diagonal}\left(\frac{1}{\sigma_i^2}\right)$$

$$\det(\Sigma) = \prod_i \sigma_i^2$$

More on the Jointly Gaussian RVS

- Special case of covariance matrix being diagonal.

$$x_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

joint PDF

$$\frac{1}{(2\pi)^{d/2}} \cdot \frac{1}{(\prod_i \sigma_i^2)^{1/2}} e^{\frac{-1}{2}\sum_i \frac{(x_i-\mu_i)^2}{\sigma_i^2}}$$

$$= \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}\cdot\sqrt{\sigma_i^2}} e^{\frac{-(x_i-\mu_i)^2}{2\sigma_i^2}}$$

$$= \prod_{i=1}^{d} f(x_i)$$

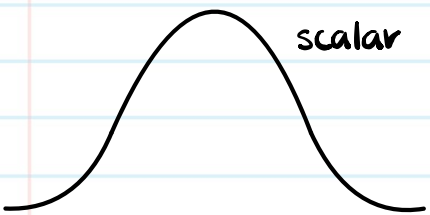
Diagonal covariance matrix, ctd.

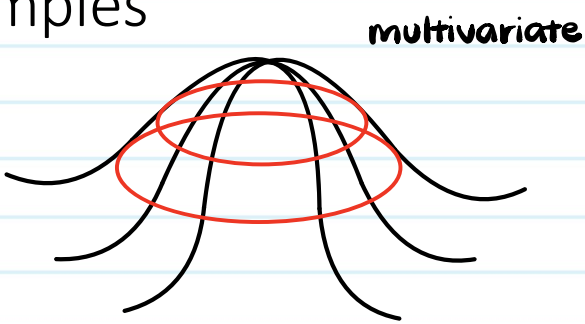
- By definition, if the covariance matrix is diagonal, then RVs are uncorrelated `for any choice of RVs`

- But, more is true for Jointly Gaussians: If RVs are uncorrelated they are also independent.

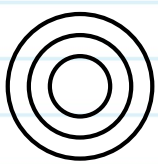
$$f_x(x) = \prod_{i=1}^{d} f(x_i)$$

Covariance matrix -- examples

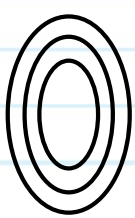




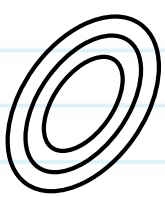
$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$



$$\Sigma = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 4 \end{pmatrix}$$



Properties

- In general, uncorrelatedness does not imply independence. Why ?

  consider z, w: $E[(z-m_z)(w-m_w)] = 0$

- But it does for Jointly Gaussians.

  in the J.G. case, distribution is completely specified through μ, Σ

- Just because two RVs are individually Gaussian, it does not mean they are Jointly Gaussian.
- Example:

  $x \sim \mathcal{N}(0, 1)$

  $z = -x \qquad z \sim \mathcal{N}(0, 1)$

  $\text{cov}(x, z) = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ → singular

Back to our set up

$$\max \prod_i P(X_i, Y_i)$$

$$y_i \sim \text{Bern}(\theta)$$

$$P(x_i \mid y_i = 1) \sim \mathcal{N}(\mu_1, \Sigma_1)$$
$$P(x_i \mid y_i = 0) \sim \mathcal{N}(\mu_0, \Sigma_0)$$

Parameter estimation

- Same idea as in the scalar case: take derivative of the log of the joint likelihood, except that we are after vectors and matrices (need to use matrix calculus)

$$\theta,\ \mu_0,\ \mu_1,\ \Sigma_0,\ \Sigma_1$$

- Start with the class prior.

$$\log \prod_i P(x_i, y_i) = \log \prod_i P(x_i | y_i)\, P(y_i)$$

$$= \sum_i \log P(y_i) + \sum_i \log P(x_i | y_i)$$

θ only appears here

Class prior parameter estimation

for the derivative w.r.t. θ, it thus suffices to only look at the first term.

$$\sum_{i=1}^{n} \log\left(\theta^{\mathbb{1}[y_i=1]} \cdot (1-\theta)^{\mathbb{1}[y_i=0]}\right)$$

$$\hat{\theta} = \frac{N_1}{N}$$

$$N_1 = \sum_{i=1}^{n} \mathbb{1}[y_i=1]$$

Mean Vector - parameter estimation

let's consider only points for which Y = 1

$$\sum_{i \in N_1} \log\left(\frac{1}{(2\pi)^{d/2}} \cdot \frac{1}{|\Sigma_1|^{1/2}} \cdot e^{-\frac{1}{2}(x_i - \mu_1)^T \Sigma_1^{-1} (x_i - \mu_1)} \right)$$

N_1 is the index set of pts in class 1

$|N_1| = N_1$

not relevant to taking the derivative with respect to μ_1

Mean Vector - parameter estimation

- Needs matrix calculus result

$$\sum_{i \in \mathcal{N}_1} \left(-\frac{1}{2}(x_i - \mu_1)^T \Sigma_1^{-1}(x_i - \mu_1)\right)$$

$$\frac{\partial}{\partial \mu_1}$$

recall: $\frac{\partial}{\partial x}(x^T A x) = (A + A^T)x$

$$-\frac{1}{2}\sum_{i \in \mathcal{N}_1}\left(2\Sigma_1(x_i - \mu_1)(-1)\right)$$

$$\sum_{i \in \mathcal{N}_1} x_i = N_1 \cdot \mu_1$$

$$\hat{\mu}_1 = \frac{1}{N_1}\sum_{i \in \mathcal{N}_1} x_i$$

$$\hat{\mu}_0 = \frac{1}{N_0}\sum_{i \in \mathcal{N}_0} x_i$$

Covariance Matrix – parameter estimation

consider Σ_1:

$$\sum_{i \in N_1} \log\left(\frac{1}{(2\pi)^{d/2}} \cdot \frac{1}{|\Sigma_1|^{1/2}} \cdot e^{-\frac{1}{2}(x_i - \mu_1)^T \Sigma_1^{-1} (x_i - \mu_1)} \right)$$

$$\sum_{i \in N_1} \left[\log \frac{1}{(2\pi)^{d/2}} - \frac{1}{2} \log|\Sigma_1| - \frac{1}{2}(x_i - \mu_1)^T \Sigma_1^{-1} (x_i - \mu_1) \right]$$

relevant:

$$-\frac{N_1}{2} \log|\Sigma_1| \qquad -\frac{1}{2} \sum_{i \in N_1} (x_i - \mu_1)^T \Sigma_1^{-1} (x_i - \mu_1)$$

Covariance Matrix – parameter estimation

- Needs matrix calculus

① $\frac{\partial |\Sigma|}{\partial \Sigma} = |\Sigma| (\Sigma^{-1})^T$

$\frac{\partial}{\partial \Sigma_i} \left[\frac{N_i}{2} \log |\Sigma_i| \right] = \frac{-N_i}{2} \frac{|\Sigma_i|}{|\Sigma_i|} \Sigma_i^{-1} = \frac{-N_i}{2} \Sigma_i^{-1}$

② $\frac{\partial}{\partial x} tr(x^{-1} A x) = -(x^{-1})^T A^T (x^{-1})^T$

$\sum_{i \in N_i} x_i^T A x_i = tr(A \cdot S)$ $\qquad S = \sum_{i \in N_i} (x_i - \mu_i)(x_i - \mu_i)^T$

Covariance Matrix – parameter estimation

use properties of S, Σ

$$\left(-\frac{1}{2}\right)(-1)\Sigma_1^{-1} S \Sigma_1^{-1}$$

$$\downarrow$$

$$-\frac{N_1}{2}\Sigma_1^{-1} + \frac{1}{2}\Sigma_1^{-1} S \Sigma_1^{-1} = 0$$

$$-N_1 + S\Sigma_1^{-1} = 0$$

$$\hat{\Sigma}_1 = \frac{S_1}{N_1}$$

Equal vs. Unequal class covariance matrices

- Same covariance matrix:

$$\hat{\Sigma} = \frac{N_1}{N}\frac{S_1}{N_1} + \frac{N_0}{N}\frac{S_0}{N_0}$$

- Different covariance matrix:

$$S_1 = \sum_{i \in N_1} (x_i - \mu_1)(x_i - \mu_1)^T \qquad \hat{\Sigma}_1 = \frac{S_1}{N_1}$$

$$S_0 = \sum_{i \in N_0} (x_i - \mu_0)(x_i - \mu_0)^T \qquad \hat{\Sigma}_0 = \frac{S_0}{N_0}$$

ECE M146 Introduction to Machine Learning

Lecture 14 - Spring 2021

Prof. Lara Dolecek

ECE Department, UCLA

Today's Lecture

Recap and continuation from last time:

- Supervised Learning
- More on GDA

New topics:

- Unsupervised learning
- K-means algorithm

Review from Last Time

- We formulated GDA as the parameter estimation problem and arrived at:

$$\theta = P(Y=1) \qquad P(X \mid Y=1) \sim \mathcal{N}(\mu_1, \Sigma_1)$$

$$\hat{\mu}_1 = \frac{1}{N_1} \sum_{i=1}^{N} x_i \cdot 1[y_i = 1] \qquad P(X \mid Y=0) \sim \mathcal{N}(\mu_0, \Sigma_0)$$

$$\hat{\Sigma}_1 = \frac{1}{N_1} S_1 \qquad \hat{\theta} = \frac{N_1}{N} \qquad N_1 = \sum_{i=1}^{N} 1[y_i = 1]$$

$$S_1 = \sum_{i=1}^{N} (x_i - \hat{\mu}_1)(x_i - \hat{\mu}_1)^T 1[y_i = 1]$$

At test time – equal covariance matrix case

$$\frac{P(Y=1|X)}{P(Y=0|X)} = \frac{\frac{P(X,Y=1)}{P(X)}}{\frac{P(X,Y=0)}{P(X)}} = \frac{P(Y=1)P(X|Y=1)}{P(Y=0)P(X|Y=0)}$$

$$= \frac{\theta}{1-\theta} \frac{\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}}{\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}} \frac{\exp\left(\frac{-1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)}{\exp\left(\frac{-1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right)}$$

$$\log\left(\frac{\theta}{1-\theta}\right) + \frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0) - \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) \underset{c0}{\overset{c1}{\gtrless}} 0$$

recall: scalar case $x \sim N(\mu, \sigma)$
we are left with linear function in x:
linear decision boundary (LDA)

At test time – unequal covariance matrix case

$$\frac{P(Y=1|X)}{P(Y=0|X)} = \frac{\theta}{1-\theta} \frac{\frac{1}{|\Sigma_1|^{1/2}}}{\frac{1}{|\Sigma_0|^{1/2}}} \frac{\exp\left(\frac{-1}{2}(x-\mu_1)^T \Sigma_1^{-1}(x-\mu_1)\right)}{\exp\left(\frac{-1}{2}(x-\mu_0)^T \Sigma_0^{-1}(x-\mu_0)\right)}$$

take log and arrive at a quadratic decision boundary (QDA)

GDA as Naïve Bayes classifier

$\dim(X) = d$

- Recall that last time we studied Naïve Bayes classifier that adopts conditional independence:

$X_i \perp X_j \mid Y \qquad 1 \le i, j \le d \qquad i \neq j$

- When this assumption is used in the current setting:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \sigma_2^2 & \\ 0 & & \ddots \\ & & & \sigma_d^2 \end{bmatrix}$$

$= \text{diagonal}(\sigma_i)$

GDA as Naïve Bayes classifier

$$\Sigma = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \sigma_2^2 & \\ 0 & & \ddots \\ & & & \sigma_d^2 \end{bmatrix}$$

$$\text{cov}(x_i, x_j) = 0 \quad \forall \quad i \neq j$$

$$\Sigma^{-1} = \text{diagonal}\left(\frac{1}{\sigma_i^2}\right)$$

$$\det(\Sigma) = \prod_{i=1}^{d} \sigma_i^2$$

in this case, we perform variance (on per coordinate basis) as in the scalar case, one

Connections with logistic regression

recall logistic regression:

a method for solving binary classification

$$P(Y=1|X) = \frac{1}{1+\exp(-w^T \cdot x)}$$

in generative Gaussian modeling:

$$P(Y=1|X) = \frac{P(X, Y=1)}{P(X)} = \frac{1}{1+\frac{P(X|Y=0)P(Y=0)}{P(X|Y=1)P(Y=1)}}$$

$$= \frac{1}{1+\exp(a)}$$

Connections with logistic regression

$$a = w^T x \qquad \tilde{w} = \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} \qquad \tilde{x} = \begin{pmatrix} x \\ 1 \end{pmatrix}$$

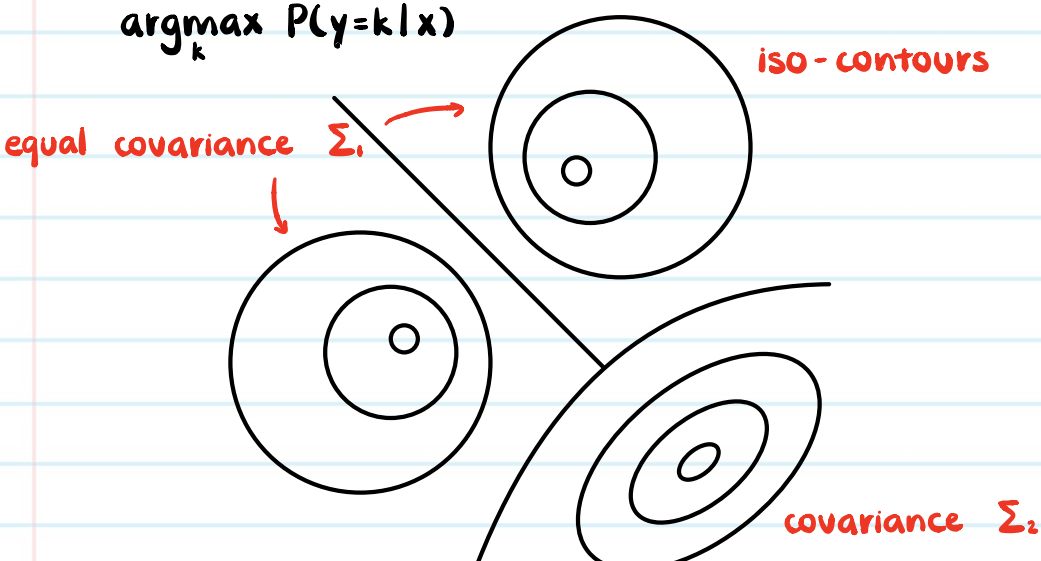
$$a = \log \frac{\theta}{1-\theta} + \frac{\mu_0^2 - \mu_1^2}{2\sigma^2} + \frac{(\mu_1 - \mu_0)x}{\sigma^2}$$

$$\Rightarrow w_1, w_0 \text{ follow}$$

GDA for multiclass classification

at test time:

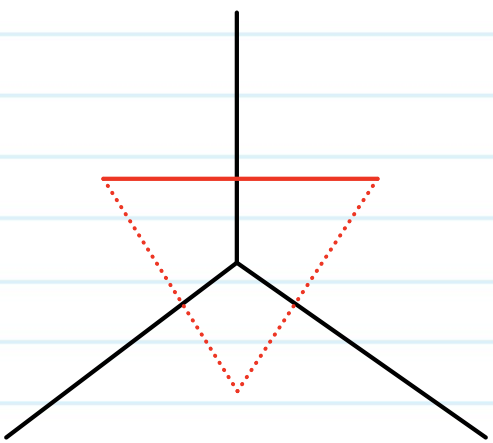
$\text{argmax}_k \; P(y=k \mid x)$



LDA with equal priors

linear decision boundary

k=3



voronoi regions

Remarks

- LDA is less likely to overfit compared to QDA (fewer parameters); linear vs. quadratic function of the dimension of the data vector.

- Can be regularized by a convex combination of the in-class and pooled covariance.

- We are computing in-class sample mean and (pooled) in-class variance. Susceptible to outliers.

contrast w/ SVM

linear regression is also susceptible to outliers

Today's Lecture

Recap and continuation from last time:

- Supervised Learning
- More on GDA

New topics:

- Unsupervised learning
- K-means algorithm

Recap: Supervised Learning

- In supervised learning, data is labeled at training time:
  - binary classification $y \in \{0, 1\}$ or $y \in \{-1, 1\}$
  - real-valued label $y \in \mathbb{R}$
- The goal is to perform classification or regression

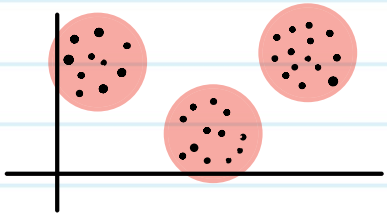
- What are some of the algorithms that you know ?

SVM, perceptron, logistic regression, linear regression, decision tree, GDA, KNN

Unsupervised learning

- Another set of problems involves having unlabeled data.
- Clustering:

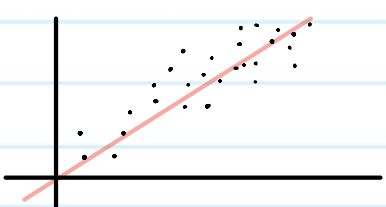
  k means



k = 3 clusters

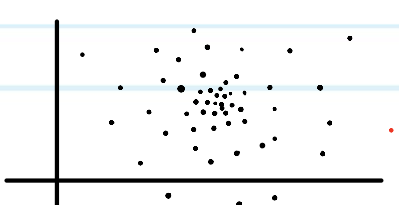
- Dimensionality reduction:

  project 2D data onto 1D line



PCA

- Density estimation:



example:
assume Gaussian distribution,
estimate μ, Σ

Unsupervised learning

- Clustering: K-means

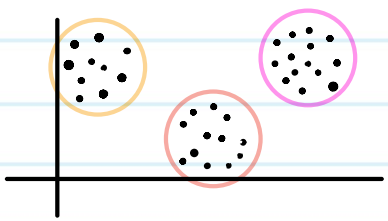
  `hard assignment: no probability`
  `soft assignment: probabilistic viewpoint`

- Dimensionality reduction: PCA

  `principal component analysis`
  `projections, linear algebra, eigenvectors and eigenvalues`

K-means algorithm for clustering

Picture:

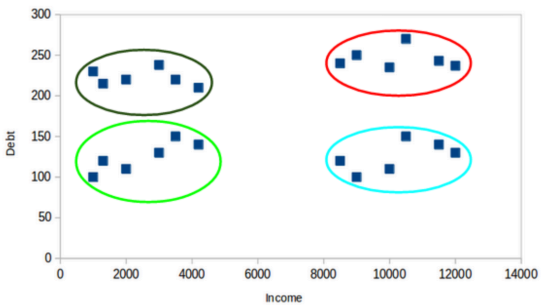


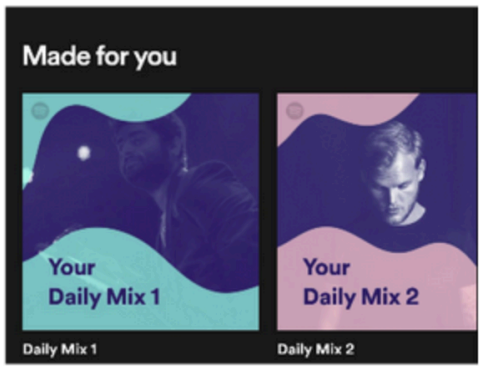
K-means is an instance of an iterative algorithm for clustering that iterates between two steps:

1. **Assignment**: for each data point assign the label of the closest prototype. cluster representation
2. **Refitting**: move each cluster center (prototype) to its center of gravity.

Clustering applications

- Targeted marketing by demographics
- Recommender systems
- Image segmentation
- Pricing of homes by location and size
- Public health services for opening hospital wards in accident prone areas





https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering

https://morioh.com/p/dcdc9957bc1c

K=6 clusters

Mathematical set-up

- Suppose we have N data points:

  $x_1, \ldots, x_n$

- Define an indicator variable:

$$r_{nk} = \begin{cases} 1 & \text{if } x_n \text{ is assigned to cluster } k \\ 0 & \text{otherwise} \end{cases}$$

$1 \leq k \leq K$

Mathematical set-up

- Distortion measure

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2$$

0 or 1

vectors

- Here μ_k is the prototype of class k.

- We want to minimize the distortion.
- We need to find both indicators as well as protypes. How ?

Procedure – overview

K is number of clusters

Step 0. Initialization.

- Start with some (random) initialization of the prototypes.

K is assumed for now μ_k $1 \le k \le K$

Step 1. Assignment.

- For each data point, find the prototype that is closest to it.
- Mathematically: $r_{nk} = \begin{cases} 1 & \text{if } k = \underset{1 \le j \le k}{\operatorname{argmin}} \|x_n - \mu_j\|^2 \\ 0 & \text{otherwise} \end{cases}$

Procedure – overview

Step 3. Refitting.

- Suppose indicators are given (or known). $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2$
- Distortion is a quadratic function of prototypes, given fixed indicators.
- How to find prototypes to minimize the distortion?
- Take a derivative: $\frac{\partial J}{\partial \mu_k} = 0 \quad \forall\, k$

$$2 \sum_{n=1}^{N} r_{nk} (x_n - \mu_k) = 0$$

$$\sum_{n=1}^{N} r_{nk} x_n = \sum_{n=1}^{N} r_{nk} \mu_k \Rightarrow \mu_k = \frac{\sum_{n=1}^{N} r_{nk} x_n}{\sum_{n=1}^{N} r_{nk}}$$

Procedure -- overview

- One evaluation of step 2 plus one evaluation of step 3 constitute one iteration.
- Iterate until a stopping criterion is satisfied.

```
for example, stop if J reaches a given threshold
or no changes occur
```

Interpretation of the prototypes

- Sample mean of the points associated with this cluster

$$\hat{\mu}_k = \frac{\sum_{n=1}^{N} r_{nk} \cdot x_n}{\sum_{n=1}^{N} r_{nk}}$$

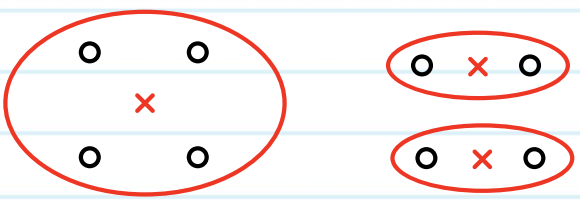
↑ # pts in cluster k

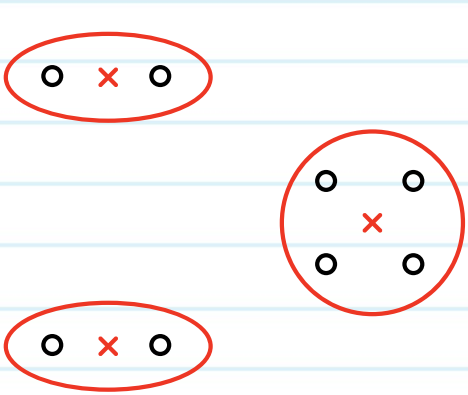
- Hence the name **K-means.**
- Where else did we see this before ?

GDA

Practical considerations

- How we start/initialize can affect the segments and prototypes that are found.
- Two examples:

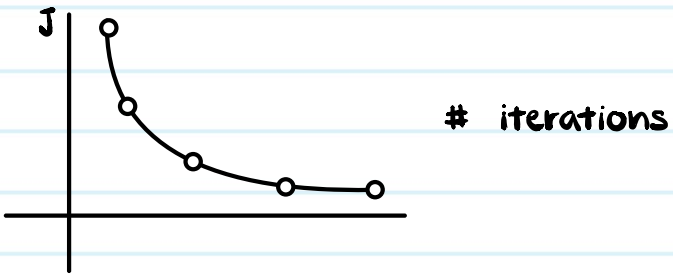




- Remedy: perform averaging.

When to stop iterating ?

- When the distortion stops decreasing.



- This will typically be when the assignments stop changing although in some peculiar cases you may see oscillations.

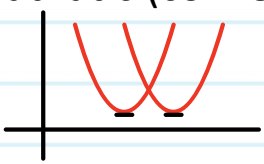
More on initialization

One strategy:

- Pick as the first prototype one of the data points. Then, pick as the second prototype the data point furthest from the first; pick as the third prototype the data point furthest from the first two, and so on.

More on the distortion minimization

- Note that K-means will always converge but not necessarily to a global optimum; it will converge to a local optimum.
- Min of quadratic (convex) functions is not convex.



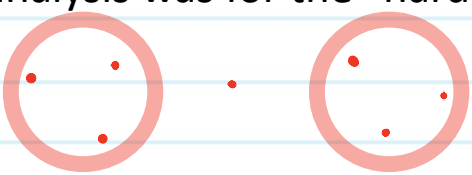
- Why doesn't distortion increase with iterations ?

because in each of the two steps in every iteration we (at least) improve on the distortion function

More on the cluster assignments

- The preceding analysis was for the "hard" 0/1 cluster assignment.
  - Issue:

$k = 2$



- There is also a "soft" cluster assignment.
  - Fractional membership.

- Math:

$$r_{nj} = \frac{\exp(-\beta \|x_n - \mu_j\|^2)}{\sum_{k=1}^{K} \exp(-\beta \|x_n - \mu_k\|^2)}$$

$$0 \leq r_{nj} \leq 1$$

$$\sum_{k=1}^{K} r_{nk} = 1$$

More on cluster assignments

- Examples of "soft" assignments:

$\beta = 1 \quad K = 3$

say, for given x_n, $\|x_n - \mu_1\| = 1$ ← closest

$\|x_n - \mu_2\| = 2$

$\|x_n - \mu_3\| = 3$

$$r_{n1} = \frac{\exp(-1^2)}{\exp(-1^2) + \exp(-2^2) + \exp(-3^2)} \approx 0.9$$

More on cluster assignments

- Examples of "soft" assignments:

$\beta = 1 \quad K = 3$

now, consider $\|x_n - \mu_1\| = 1$

$\|x_n - \mu_2\| = 1.1$

$\|x_n - \mu_3\| = 2$

$$r_{n1} = \frac{\exp(-1^2)}{\exp(-1^2) + \exp(-1.1^2) + \exp(-2^2)} \approx 0.537$$

$r_{n2} \approx 0.435$

$r_{n3} \approx 0.0273$

More on cluster assignments

- This particular mathematical expression didn't come out of nowhere. It arises in Expectation-Maximization of Gaussian Mixture Models.
- Iterative clustering with partial cluster membership and Gaussian clusters.
- Can you recall when we studied Gaussian fitting?

GDA

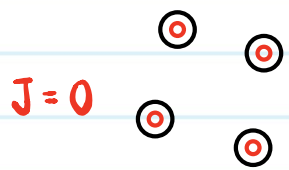
More on convergence

- Note that for K clusters and N data points, there are K^N possible cluster membership assignments.
- We do not check for all of them!
- Recall distortion: $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2$

- In the first step, update indicators to lower the current distortion i.e., re-assign to clusters with a lower cost.
- In the second step, minimize the total within-cluster squared distance and output the prototype (cluster center) that gives this minimum.

What about the choice of K ?

- The number of clusters K is a hyperparameter here.
- On one hand, can have K = N. Issue ?
- On the other hand, can have K = 1. Issue ?



J = large

- Add the penalty term to the minimization (form of regularization).

  - $\text{argmin}_k \left(J_k + k \cdot \lceil \log d \rceil\right)$

    d is the dimension of data

  - $\text{argmin}_k \left(J_k + k \cdot d\right)$

Hierarchical clustering

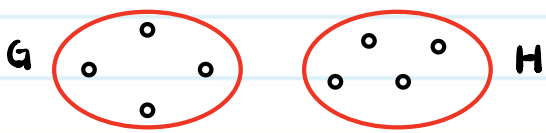
- Note that in the K-means algorithm, in going from K to K+1 clusters or K to K-1 clusters, the split/merge is often not monotonic in the clusters.

- Essentially, K-means for K+1 clusters is done without any regard for what was done for K clusters.

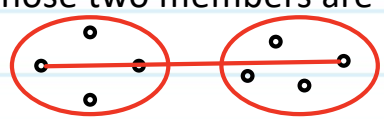
Hierarchical clustering

- An alternative is to perform clustering hierarchically with explicit subsumption, supersumption.
- For example, start with N clusters, and then keep merging as long as the overall distortion is being reduced.
  - Agglomerative clustering
- Another strategy is to start with 1 cluster, and then keep splitting as long as the overall distortion is being reduced.
  - Divisive clustering

Hierarchical clustering



- In agglomerative clustering, we can use different criteria for merging:
  - Single linkage – merge clusters G and H whose two members are closest.
  - Complete linkage – merge clusters G and H whose two members are farthest.



  - Group average – merge clusters G and H whose average dissimilarity is the smallest.

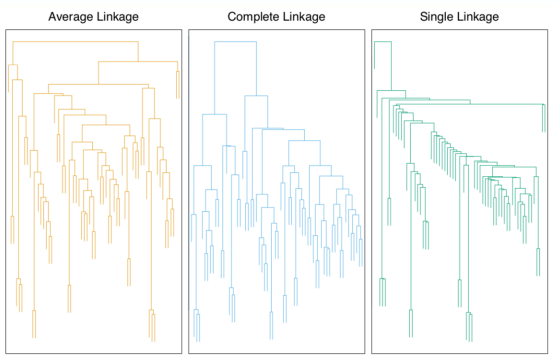


FIGURE 14.13. *Dendrograms from agglomerative hierarchical clustering of human tumor microarray data.*

Taken from *Elements of Statistical Learning*, Hastie, Tibshirani, and Friedman

Other forms of the distance measure

- The preceding analysis was for the squared Euclidean distance (L2 norm).
- When we took the derivative, we got means for the prototypes.

- Other types of distances that are of interest are Manhattan distance (L1 norm); Hamming distance, etc.
- Depending on the choice of the distance measure, we arrive at a different formula for the prototypes (it's not mean for L1!)
- Hierarchical clustering works just as well!

Stochastic update rule

- Recall stochastic gradient descent.

we took the derivative at only one data point

- We have a version of that here, too.

in k-means: $\frac{\partial J}{\partial \mu_k} = 0 \quad \forall\, k$

$$2\sum_{n=1}^{N} r_{nk}(x_n - \mu_k) = 0$$

$$\mu_k^{new} = \mu_k + \eta_n(x_n - \mu_k^{old})$$

replace the sum by just a single term

Recap

- In today's lecture, we started the discussion of unsupervised learning.
- K-means algorithm for clustering. Iterative method with "hard" membership assignment.
- Data compression (connections to information theoretic concepts we saw in decision trees).
- Next: PCA for dimensionality reduction and EM for soft means.