Unsupervised Learning

- data unlabelled and unclassified information analyzed to discover hidden knowledger
- predict outcome variable Y on the basis of feature set $X_1, X_2, \dots, X_n$ using regression and classification - supervised learning
- unsupervised learning - observe features $X_1, X_2, \dots, X_n$ - not going to predict any any outcome variable
- interest is to find out
    - the association between features or
    - their grouping to understand the nature of the data,
    - reveal correlation between features,
    - behaviour within subgroup of data
- in statistics, supervised learning - try to learn probability of outcome Y for a particular input X which is called the posterior probability

Unsupervised Learning

- unsupervised learning is related to density estimation in statistics
- input, target - a new set of inputs $\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$
- better understanding of the correlation of X and Y, this probability notation is called joint probability
- movie promotions to correct group of people - recommendation system
- blind push of same data to all demography - everyone watches the same poster or trailer
- irrespective of their choice or preference
- results into ignoring it, waste of effort and money on promotion
- database to understand what type of movie liked by segment of demography
- ML finds out the pattern or repeated behaviour of small group or cluster within this database

Unsupervised Learning

- like or dislike, relevant movie promotion or trailers pushed to the selected groups
- increase the chance of targeting the right interested person for the movie
- clustering - discovering unknown subgroups in data
- association analysis - identifies low dimensional presentation of observations
- explain variance and identify association rule for explanation
- identification of relationships among objects in a data set
- unsupervised learning works on uncategorized and unlabelled data
- segmentation of target consumer populations by consulting agency on the basis of
- demography, financial data, purchasing habit -
- advertisers can reach target consumers efficiently

Unsupervised Learning

- anomaly or fraud detection in banking sector by identifying the patterns of loan defaulters
- image processing - segmentation - face recognition, expression identification
- genetic application - grouping based on characteristics
- data scientists to reduce dimensionalities
- AI and ML - chat bots, self driven cars
- Clustering: finding subgroups or clusters in the data set based on characteristics of the objects in the data,
- objects within the group are similar - related to each other, and
- different from - unrelated to objects from other groups
- effectiveness - how similar or related objects within group and
- how different or unrelated objects in different groups from each other

Unsupervised Learning

- advertisements of new movie for promotional activity
- features: age, location, financial condition, political stability
- different type of campaign for different parts grouped accordingly to the data
- driving campaign in a targeted way
- different ways to group the set of people and arriving at different types of clusters
- applications
  - text data mining - text categorization, clustering, document summarization, concept extraction, sentiment analysis and entity relation modelling
  - customer segmentation - demographics, financial conditions, buying habits,

Unsupervised Learning

- anomaly checking - anomalous behaviour, fraudulent bank transaction,
- unauthorized computer intrusion, suspicious movements on radar scanner
- data mining
- how clustering differs from classification, how clustering defines groups
- *k*-means, *k*-medoids algorithms
- clustering - knowledge discovering - rather than prediction
- homogeneity within group
- goal is to create a model that relates features to an outcome or to other features model identifies patterns within data
- unlabelled objects given a cluster label which is inferred entirely from the relationship of attributes within data

Clustering

- induction of faculty in university on particular subject
- list of research publications of faculty members from internet
- ML to group papers and infer expertise say, Statistics, Computer Science and Machine Learning
- closeness of points to each other to form a group or cluster
- clustering techniques
  - partitioning methods
  - hierarchical methods
  - density based methods

Clustering

- partitioning methods
  - uses mean or medoid to represent cluster centre
  - distance based approach to refine cluster
  - find mutually exclusive clusters of spherical or nearly spherical shape
  - effective for small or medium data set
- hierarchical methods
  - hierarchical or tree like structure through decomposition or merger
  - distance based refinement nearest or furthest points in neighbouring clusters
  - errorneous merges or splits can not be corrected at subsequent levels

Clustering

- density based methods
  - identifying arbitrarily shaped clusters
  - cluster creation - identification of dense regions of objects in space
  - which are separated by low density regions
  - filter out outliers
- partitioning methods: k-means and k-medoid
- k-means uses centroid, mean of group of points
- centroid does not correspond to an actual data point
- k-medoid - medoid is always an actual data point

Clustering: *k*-means algorithm

- assign each of n data points to one of K clusters
- K is a user defined parameters as the number of clusters desired
- objective is to maximize homogeneity within clusters and maximize difference between clusters
- homogeneity and difference measured in terms of distance between objects or points in the dataset
    1. select K points in the data space and mark them initial centroids
       loop:
    2. assign each point in the data space in the nearest centroid to form K clusters
    3. measure the distance of each point in the cluster from the centroid
    4. calculate the sum of squared error to measure the quality of the clusters
    5. identify the new centroid of each cluster on the basis of distance between points
    6. repeat above to refine until centroid do not change

Clustering: *k*-means algorithm

- centroids are updated and points are reassigned to the updated centroids
- different numbers of starting cluster lead to completely different types of data split
- prior knowledge about number of clusters helps
- movie maker wants to cluster movies on budget high or low and casting star or non-star
- rule of thumb $K = \sqrt{\frac{n}{2}}$
- for large data set this thumb rule does not work well
- efficiency is high but random chance that may not find optimal set of cluster

Clustering: *k*-means algorithm

- Elbow method
  - this method tries to measure the homogeneity or heterogeneity within the cluster for various values of K
  - helps in arriving optimal K
  - homogeneity will increase or heterogeneity will decrease with increasing K as the number of data points inside each cluster reduces with this increase
  - more computations required
  - after a certain point, the increase in homogeneity benefit is no longer in accordance with the investment required to achieve it, this point is known as elbow point
- Choosing initial centroids
  - choose initial centroids properly
  - random points chosen and refined in iterations - may leads to higher squared error in the final clustering
  - assumption is that multiple subsequent runs will minimize the SSE and identify the optimal clusters

Clustering: *k*-means algorithm

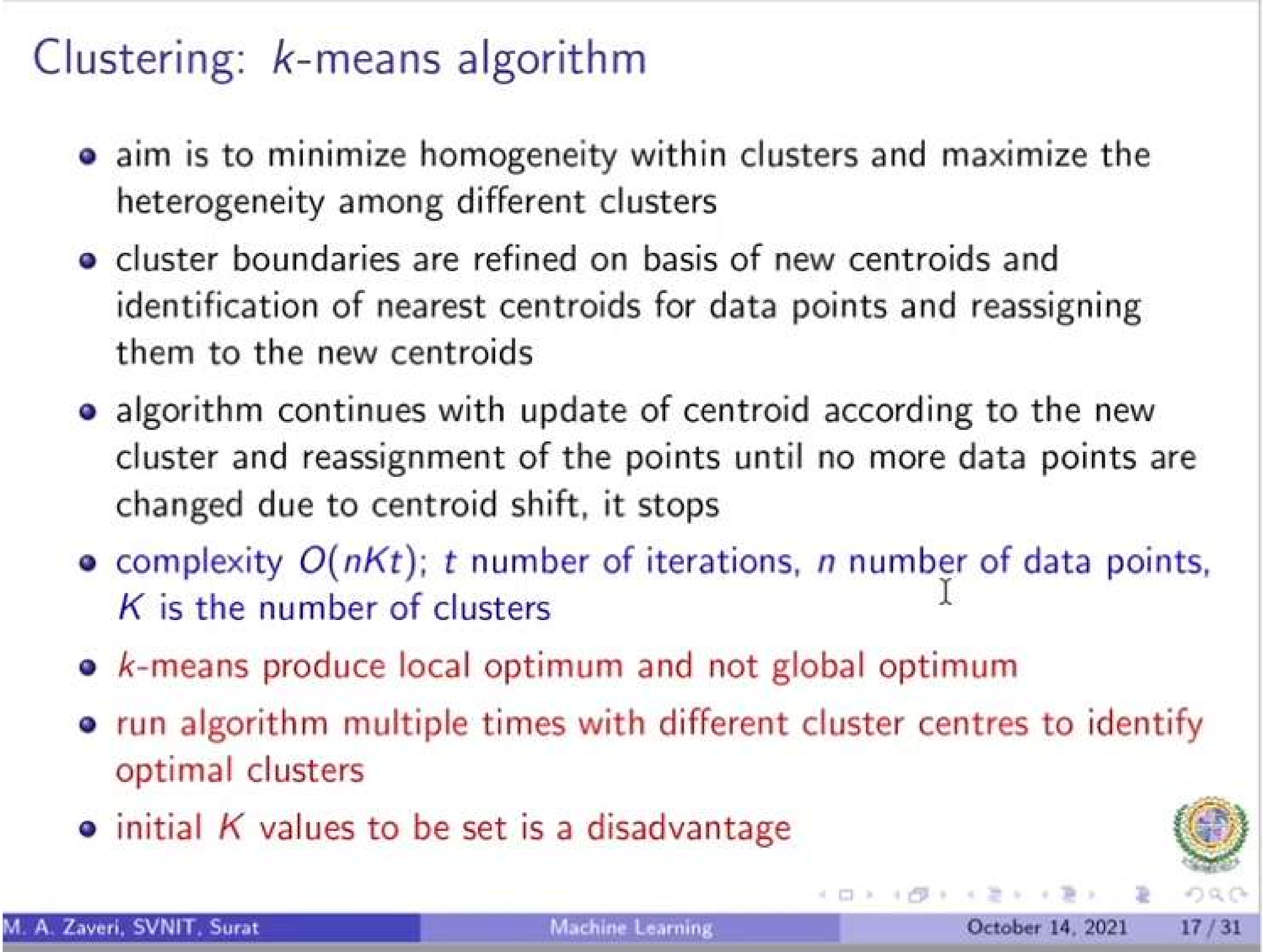
- effective approach is to employ the hierarchical clustering on sample points and then
- arrive at sample K clusters and then centroids of these initial K clusters are used as initial centroids
- recomputing cluster centroids
- proximities of data points from each other within cluster is measured to minimize the distances - Euclidean distance

$$dist(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$SSE = \sum_{k=1}^{K} \sum_{x \in C_k} dist(c_k, x)^2$$

Clustering: *k*-means algorithm

- lower SSE better is representative position of centroid
- calculate SSE of each new centroid and arriving at optimal centroid identification
- after centroids are repositioned the data points nearest to the centroids are assigned to form the refined clusters
- centroid that minimizes the SSE of the cluster is its mean
- presence of outlier in data set - it can distort the mean value of clusters
- voronoi diagram - creates boundaries of clusters
- got initial clusters created by dashed lines from vertex of the clusters which is the point with the maximal distance from the centre of the clusters



Clustering: *k*-means algorithm

- aim is to minimize homogeneity within clusters and maximize the heterogeneity among different clusters
- cluster boundaries are refined on basis of new centroids and identification of nearest centroids for data points and reassigning them to the new centroids
- algorithm continues with update of centroid according to the new cluster and reassignment of the points until no more data points are changed due to centroid shift, it stops
- complexity $O(nKt)$; t number of iterations, n number of data points, K is the number of clusters
- k-means produce local optimum and not global optimum
- run algorithm multiple times with different cluster centres to identify optimal clusters
- initial K values to be set is a disadvantage

Unsupervised Learning

- clustering is used as first step of identifying the subgroups within unlabelled set of the data then is used for classifying the new observed data
- software testing activity - identification of set of defects
- identify similar groups of defects, GUI related defects, business logic related defects,
- missing requirement defects, database related defects
- based on this grouping, item identified the developers to whom the defects should be sent for fixing
- *k*-medoids - object based technique
- *k*-means sensitive to outlier - means of data points are used as centroids

Clustering: *k*-medoids algorithm

- $1 - D$ data $1, 2, 3, 6, 9, 10, 11, 25$
- outlier is 25, $K = 2$ initial cluster
- $[1, 2, 3, 6]$ and $[9, 10, 11, 25]$; mean 3 and 14 respectively
- SSE is 179, $SSE = \sum_i (x - c_i)^2$
- if cluster $[1, 2, 3, 6, 9]$ and $[10, 11, 25]$; mean 4.2 and 15.67
- SSE is 113.84 lower, put point 9 in the cluster $1, 2, 3, 6$ though the point nearer to 10 and 11
- skewedness is introduced due to outlier point 25 which shift mean away from centre of cluster

Clustering: *k*-medoids algorithm

- *k*-medoids provides a solution to the above problem,
- instead of considering the mean of the data points in the cluster,
- *k*-medoids considers *k* representative data points from the existing points in the data set as centre of the clusters
- assign the data points according to their distance from these centres to form *k* clusters

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(o_i, x)^2$$

- o_i representative point or object of cluster C_i
- *k*-medoids groups *n* objects in *k* clusters by minimizing the *SSE*
- less influenced by the outliers in the data

Clustering: *k*-medoids algorithm

- PAM partitioning around medoids algorithm
    1. randomly choose k points in data set as initial representative points
       loop:
    2. assign each of remaining points to the cluster which has nearest representative point
    3. randomly select a non-representative point o_r in each cluster
    4. swap representative point o_j with o_r and compute the new SSE after swapping
    5. if $SSE_{new} < SSE_{old}$ swap o_j with o_r to form the new set of k representative objects
    6. refine the k clusters on the basis of nearest representative point
    7. logic continues until there is no change

Clustering: k-medoids algorithm

- in iterative process, all possible replacements are attempted until quality of clusters no longer improves
- if $o_1, o_2, \ldots, o_k$ are current set of representative objects or medoids and
- non-representative object o_r - is good replacement if SSE decreases it means that o_r
- represents the cluster better than o_j and the data points in the set are reassigned according to the nearest medoids
- k-medoids provides effective way to eliminate the noise or outliers in the data set which was the problem in k-means algorithm
- complexity of k-medoids is $O(k(n-k)^2)$

Hierarchical clustering

- data needs to be partitioned into groups at different levels such as in a hierarchy
- try to group the data into hierarchy or tree-like structure
- organizing employees of university in different departments
- group under different department
- group within each department based on role, professors, assistant professor, supervisor, lab assistant -
- creates hierarchy, eases visualization and analysis
- discover underlying hierarchy structure in data set
- agglomerative clustering and divisive clustering
- agglomerative clustering - bottom up technique which starts with individual objects as clusters and then iteratively merges them to form larger clusters
- merges the clusters according to their similarity
- terminates when a certain clustering condition imposed is achieved

Hierarchical clustering

- divisive method starts with one cluster with all given objects and then splits it iteratively to form smaller clusters
- top-down approach - end iterations when final clusters sufficiently homogeneous to each other
- split and merger should be done carefully, subsequent splits or mergers use result of previous one and
- swapping of object between clusters or rectify the decisions made in previous steps not possible, results in poor clustering quality
- dendrogram - tree structure representation of step-by-step creation of hierarchical clustering
- core measures of proximities between clusters is the distance between them
- four standard methods to measure the distance between clusters
- let, C_i and C_j two clusters n_i and n_j points respectively,
- p_i and p_j represents points in clusters C_i and C_j respectively

Hierarchical clustering

- minimum distance $D_{min}(C_i, C_j) = \min_{p_i \in C_i, p_j \in C_j} \{|p_i - p_j|\}$
- similarly, maximum distance, average distance
- $D_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p_i \in C_i, p_j \in C_j} |p_i - p_j|$
- mean distance
- these distance measures used to decide when to terminate the clustering
- nearest neighbour clustering
- user defined limit on D_{min} called single linkage algorithm
- furtherest neighbour clustering
- user defined limit on D_{max} called complete linkage algorithm
- mean or average distance avoids outlier and noisy data

Clustering: Density based method DBSCAN

- partitioning and hierarchical clustering - resulting clustering spherical or nearly spherical in nature
- other shaped cluster, S-shaped or uneven shaped clusters, density based clustering used
- identifying the dense area and sparse area within the data set and then run the clustering algorithm
- DBSCAN density based algorithm crates cluster by using connected regions with high density

Unsupervised Learning: Association Rule Mining

- association rules based analysis - set of frequent items
- market basket analysis
- retailers use for cross-selling of their products

Association Rule Mining

- low support indicates the rule has occurred by chance
- rule may not be very attractive
- support can provide the intelligence of identifying the most interesting rules for analysis
- confidence provides measurement for reliability of the inference of a rule
- higher confidence of rule $X \rightarrow Y$ denotes more likelihood of to be present in
- transactions that contain X as it is the estimate of the conditional probability of Y given X
- association rule used in context of big data and data science
- unsupervised knowledge discovery: discovering association rule
- minimum support and minimum confidence of the association rule
- $support \geq minS$ and $confidence \geq minC$

Unsupervised learning - Density estimation

- aim to represent data in some way
- data points themselves as representation of the data - not helpful when dataset is huge
- interested in representing characteristics of the data
- density estimation - represent data compactly using a density from a parametric family e.g. Gaussian or Beta distribution
- mean and variance of a dataset in order to represent the data compactly using Gaussian distribution

Density estimation

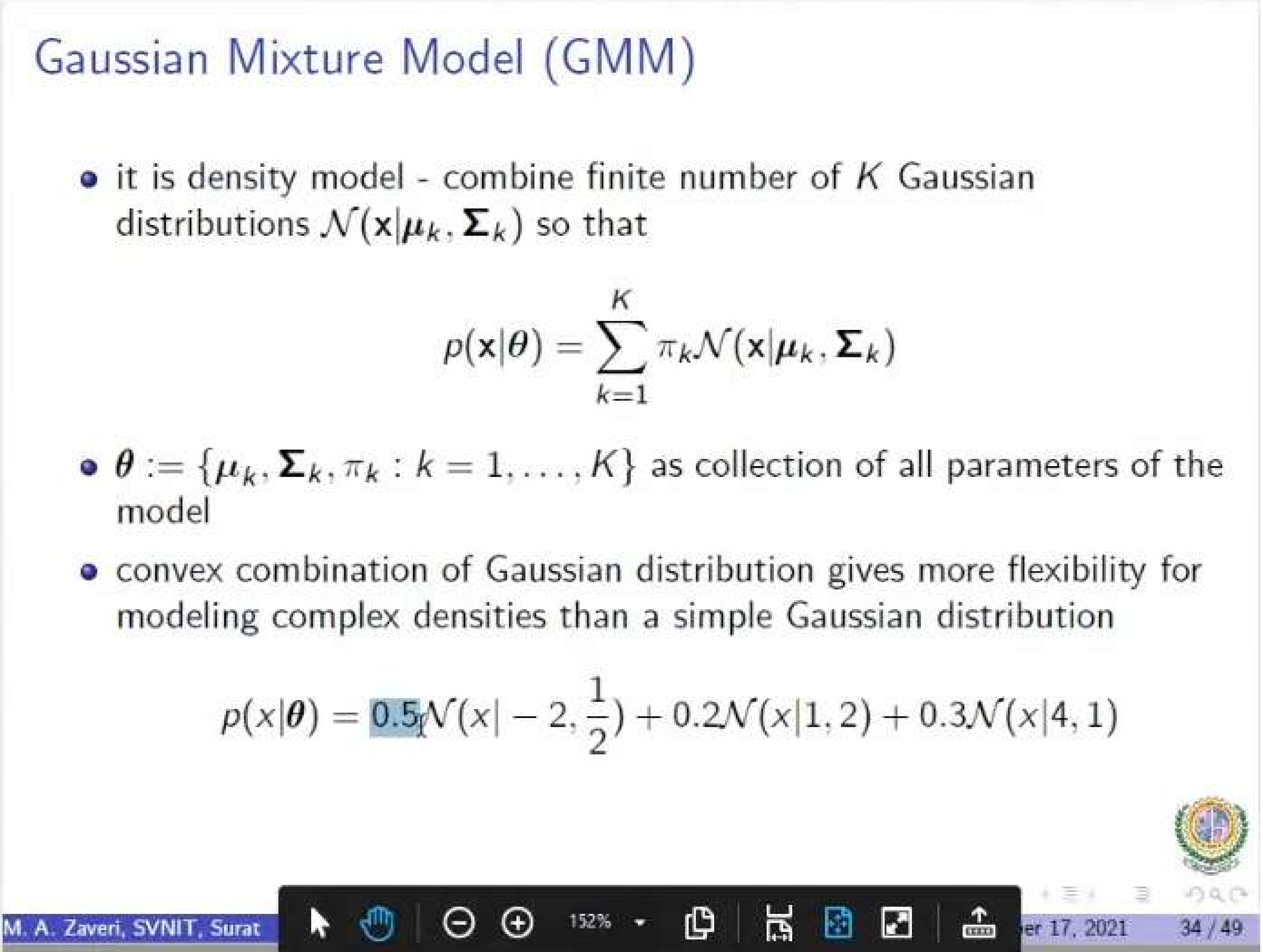
- dataset to be a typical realization from this distribution if we were to sample from it
- Gaussian have limited modeling capabilities
- Gaussian approximation of the density that generated the data may be a poor approximation
- more expressive family of distributions - can be used for density estimation: mixture models
- mixture models can be sued to describe a distribution
- $p(\mathbf{x})$ by a convex combination of K simple or base distributions

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{x})$$

$$0 \leq \pi_k \leq 1, \sum_{k=1}^{K} \pi_k = 1$$

Density estimation

- components p_k are members of a family of basic distributions e.g. Gaussians, Bernoullis, Gammas and π_k are mixture weights
- mixture models are more expressive than the corresponding base distributions
- allow for multimodal data representations
- describe datasets with multiple clusters
- Gaussian mixture models GMMs - basic distributions are Gaussians
- for a given dataset, aim to maximize the likelihood of the model parameters to train the GMM
    - will not find a closed form maximum likelihood solution
    - will arrive at a set of dependent simultaneous equations which can be solved iteratively



Gaussian Mixture Model (GMM)

- it is density model - combine finite number of K Gaussian distributions $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ so that

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- $\boldsymbol{\theta} := \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k : k = 1, \ldots, K\}$ as collection of all parameters of the model
- convex combination of Gaussian distribution gives more flexibility for modeling complex densities than a simple Gaussian distribution

$$p(x|\boldsymbol{\theta}) = 0.5\mathcal{N}(x|-2, \frac{1}{2}) + 0.2\mathcal{N}(x|1, 2) + 0.3\mathcal{N}(x|4, 1)$$

Gaussian Mixture Model (GMM)

- Parameter learning via Maximum Likelihood
- given a dataset $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ $\mathbf{x}_n; n = 1, \ldots, N$ are drawn i.i.d. from an unknown distribution $p(\mathbf{x})$
- objective is to find a good approximation / representation of this unknown distribution $p(\mathbf{x})$ by means of a GMM with K mixture components
- the parameters of the GMM are the K means $\boldsymbol{\mu}_k$, the covariances $\boldsymbol{\Sigma}_k$, mixture weights π_k
- parameters $\boldsymbol{\theta} := \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k : k = 1, \ldots, K\}$

Gaussian Mixture Model (GMM)

- say, one dimensional dataset $\mathcal{X} = \{-3, -2.5, -1, 0, 2, 4, 5\}$
- find a GMM with $K = 3$ components that models the density of the data
- initialize the mixture components as

$$\begin{aligned} p_1(x) &= \mathcal{N}(x|-4, 1) \\ p_2(x) &= \mathcal{N}(x|0, 0.2) \\ p_3(x) &= \mathcal{N}(x|8, 3) \end{aligned}$$

- assign them equal weights $\pi_1 = \pi_2 = \pi_3 = \frac{1}{3}$

Gaussian Mixture Model (GMM)

- maximum likelihood estimate $\boldsymbol{\theta}_{ML}$ of the model parameters $\boldsymbol{\theta}$
- likelihood, i.e., the predictive distribution of the training data given the parameters
- i.i.d. assumption, leads to factorized likelihood

$$p(\mathcal{X}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\boldsymbol{\theta}) \quad \text{and} \quad p(\mathbf{x}_n|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- every individual likelihood term $p(\mathbf{x}_n|\boldsymbol{\theta})$ is a Gaussian mixture density
- log likelihood

$$\log p(\mathcal{X}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\mathbf{x}_n|\boldsymbol{\theta}) = \mathcal{L}$$

$$\mathcal{L} := \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Gaussian Mixture Model (GMM)

- for all three necessary conditions, applying chain rule
- partial derivatives

$$\frac{\partial \log p(\mathbf{x}_n|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{p(\mathbf{x}_n|\boldsymbol{\theta})} \frac{\partial p(\mathbf{x}_n|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

$\boldsymbol{\theta} = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k, k = 1, \ldots, K\}$ are model parameters

$$\frac{1}{p(\mathbf{x}_n|\boldsymbol{\theta})} = \frac{1}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

define the quantity

$$r_{nk} := \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Gaussian Mixture Model (GMM)

- as responsibility of the kth mixture component for the nth data point
- the responsibility r_{nk} of the kth mixture component for data point $\mathbf{x}_n$ is proportional to the likelihood

$$p(\mathbf{x}_n|\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  of the mixture component given the data point
- r_n follows a Boltzmann/Gibbs distribution
- mixture components have a high responsibility for a data point when the data point could be a plausible sample from that mixture component
- $\mathbf{r}_n := [r_{n1}, \ldots, r_{nK}]^T \in \mathbb{R}^K$ a normalized probability vector i.e., $\sum_k r_{nk} = 1$ with $r_{nk} \geq 0$

Gaussian Mixture Model (GMM)

- this probability vector distributes probability mass among the K mixture components
- $\mathbf{r}_n$ as a soft assignment of $\mathbf{x}_n$ to the K mixture components
- responsibility r_{nk} represents the probability that $\mathbf{x}_n$ has been generated by the kth mixture component

$$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 0.057 & 0.943 & 0.0 \\ 0.001 & 0.999 & 0.0 \\ 0.0 & 0.066 & 0.934 \\ 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \in \mathbb{R}^{N \times K}$$

Gaussian Mixture Model (GMM)

- nth row tells us the responsibilities of all mixture components for x_n
- sum of all K responsibilities for a data point (sum of every row) is 1
- kth column gives us an overview of responsibility of the kth mixture component
- sum of all entries of a column gives us the values N_k, total responsibilities of the kth mixture component
  $N_1 = 2.058, N_2 = 2.008, N_3 = 2.934$
- updates of model parameters for given responsibilities
- update equations all depend on the responsibilities, which makes a closed form solution to the maximum likelihood estimation problem impossible

Gaussian Mixture Model (GMM)

- for a given responsibilities, updating one model parameter at a time, while keeping the others fixed and recompute the responsibilities,
- iterating these two steps converges to a local optimum and is a specific instantiation of EM algorithm
- update of mean parameters μ_k $k = 1, \ldots, K$ is given by

$$\boldsymbol{\mu}_k^{new} = \frac{\sum_{n=1}^{N} r_{nk} \mathbf{x}_n}{\sum_{n=1}^{N} r_{nk}}$$

- update of means $\boldsymbol{\mu}_k$ of individual mixture components depends on all means, covariance matrices $\boldsymbol{\Sigma}_k$ and mixture weights π_k via r_{nk}
- so it is not possible to obtain a closed-form solution for all $\boldsymbol{\mu}_k$ at once

Gaussian Mixture Model (GMM)

- update of covariance parameters $\mathbf{\Sigma}_k \; k = 1, \ldots, K$

$$\mathbf{\Sigma}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

$N_k := \sum_{n=1}^{N} r_{nk}$

- updating mixture weights

$$\pi_k^{new} = \frac{N_k}{N}$$

$k = 1, \ldots, K$, N number of data points

- the above updates of parameters, not possible to obtain closed form solution because
- responsibilities r_{nk} depend on those parameters in a complex way

Gaussian Mixture Model (GMM)

- EM expectation maximization algorithm proposed by Dempster et al.
- a general iterative scheme for learning parameters in mixture models, more generally, latent-variable models through maximum likelihood or MAP
- for Gaussian mixture model - choose initial values for $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$ and alternate until convergence between
  - *E-step*: evaluate the responsibilities r_{nk} (posterior probability of data points n belonging to mixture component k)
  - *M-step*: use updated responsibilities to reestimate the parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$
- every step in the EM algorithm increases the log-likelihood function
- for convergence, check log-likelihood for the parameters directly

Gaussian Mixture Model (GMM)

- EM algorithm is as follows

1. initialize $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$
2. *E-step*: evaluate responsibilities r_{nk} for every data point $\mathbf{x}_n$ using current parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$

$$r_{nk} := \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

3. *M-step*: reestimate $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$ using the current responsibilities r_{nk} (from E-step)

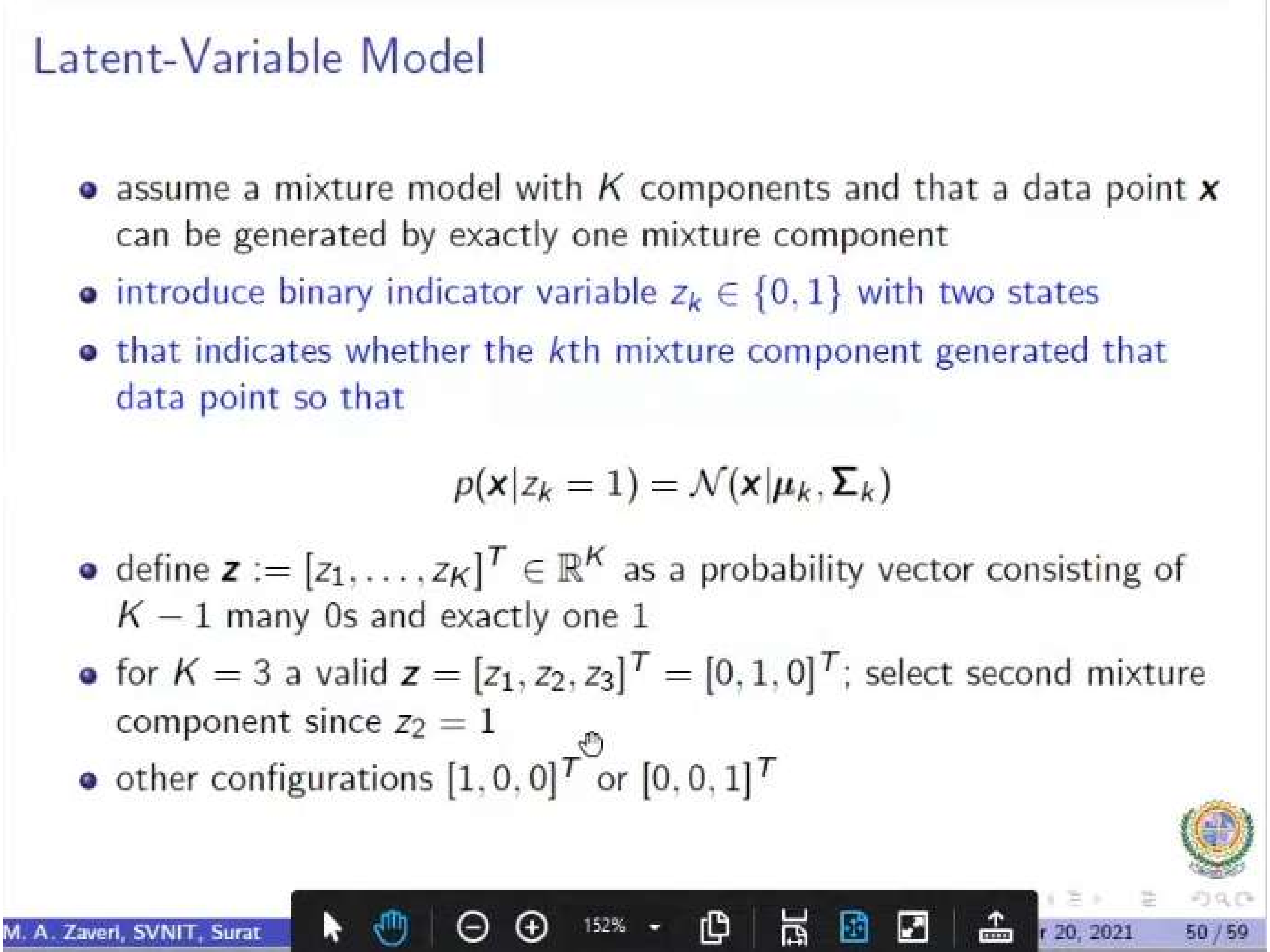
$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

Latent-Variable Model

- looking at GMM from perspective of a discrete latent-variable model
- latent variable $\boldsymbol{z}$ can attain only a finite set of values
- it is in contrast to PCA, where latent variables were continuous valued numbers in $\mathbb{R}^M$
- advantages of probabilistic perspective are
    - it allows for a concrete interpretation of responsibilities as posterior probabilities
    - iterative algorithm for updating the model parameters can be derived using the EM algorithm for maximum likelihood parameter estimation in latent variable models
- generative process and probabilistic model
- to derive the probabilistic model for GMMs, it is useful to think about the generative process, i.e.,
- the process that allows us to generate data, using a probabilistic model



Latent-Variable Model

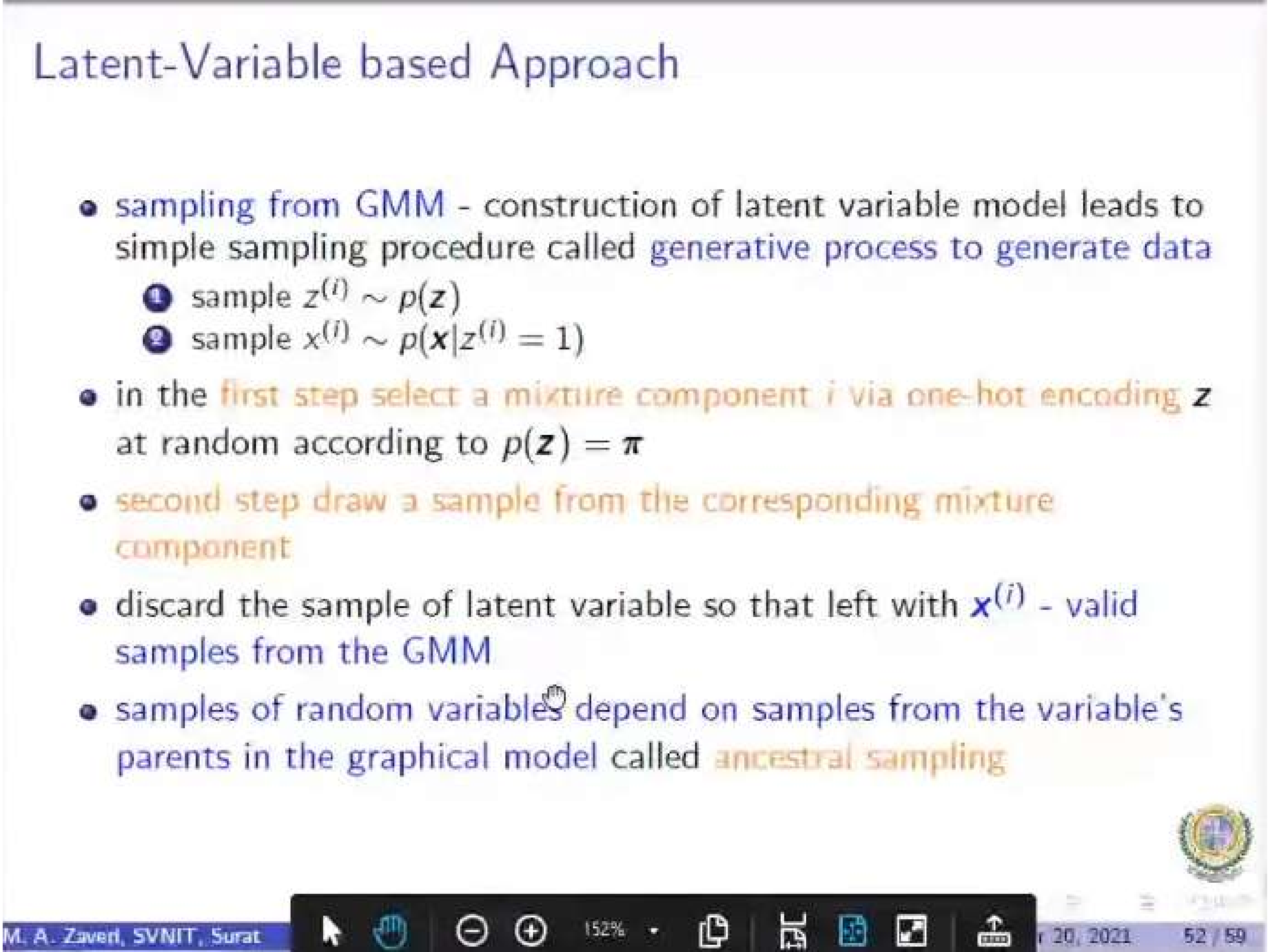
- assume a mixture model with K components and that a data point $\boldsymbol{x}$ can be generated by exactly one mixture component
- introduce binary indicator variable $z_k \in \{0, 1\}$ with two states
- that indicates whether the kth mixture component generated that data point so that

$$p(\boldsymbol{x}|z_k = 1) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- define $\boldsymbol{z} := [z_1, \ldots, z_K]^T \in \mathbb{R}^K$ as a probability vector consisting of $K - 1$ many 0s and exactly one 1
- for $K = 3$ a valid $\boldsymbol{z} = [z_1, z_2, z_3]^T = [0, 1, 0]^T$; select second mixture component since $z_2 = 1$
- other configurations $[1, 0, 0]^T$ or $[0, 0, 1]^T$

Latent-Variable Model

- this kind of probability distribution is called multinoulli, a generalization of Bernoulli distribution
- properties $\boldsymbol{z}$ imply that $\sum_{k=1}^{K} z_k = 1$ $\boldsymbol{z}$ is one-hot encoding also known as $1 - of - K$ representation
- it is assumed that indicator variables z_k are known
- in practice, this is not the case, place a prior distribution
- $p(\boldsymbol{z}) = \boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]^T$ $\sum_{k=1}^{K} \pi_k = 1$ on the latent variable $\boldsymbol{z}$ then
- the kth entry $\pi_k = p(z_k = 1)$ of this probability vector describes the probability that kth mixture component generated data point $\boldsymbol{x}$



Latent-Variable based Approach

- sampling from GMM - construction of latent variable model leads to simple sampling procedure called generative process to generate data
  1. sample $z^{(i)} \sim p(\boldsymbol{z})$
  2. sample $x^{(i)} \sim p(\boldsymbol{x}|z^{(i)} = 1)$
- in the first step select a mixture component i via one-hot encoding $\boldsymbol{z}$ at random according to $p(\boldsymbol{z}) = \boldsymbol{\pi}$
- second step draw a sample from the corresponding mixture component
- discard the sample of latent variable so that left with $\boldsymbol{x}^{(i)}$ - valid samples from the GMM
- samples of random variables depend on samples from the variable's parents in the graphical model called ancestral sampling

Latent-Variable based Approach

- probabilistic model is defined by the joint distribution of the data and latent variables
- with prior $p(\boldsymbol{z})$ and conditional $p(\boldsymbol{x}|\boldsymbol{z})$
- obtain all K components of this joint distribution via

$$p(\boldsymbol{x}, z_k = 1) = p(\boldsymbol{x}|z_k = 1)p(z_k = 1) = \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

for $k = 1, \ldots, K$

$$p(\boldsymbol{x}, \boldsymbol{z}) = \begin{bmatrix} p(\boldsymbol{x}, z_1 = 1) \\ \vdots \\ p(\boldsymbol{x}, z_K = 1) \end{bmatrix} = \begin{bmatrix} \pi_1 \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ \vdots \\ \pi_K \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K) \end{bmatrix} \tag{2}$$

Latent-Variable based Approach

- to obtain $p(\boldsymbol{x}|\boldsymbol{\theta})$ in a latent variable model

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{\theta})$$

$\boldsymbol{\theta} := \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k : k = 1, \ldots, K\}$

- sum over all K possible one-hot encodings of $\boldsymbol{z}$ by $\sum_{\boldsymbol{z}}$
- there is only a single nonzero single entry in each $\boldsymbol{z}$ there are only K possible configurations of $\boldsymbol{z}$

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{\theta}) &= \sum_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{\theta}) \\ &= \sum_{k=1}^{K} p(\boldsymbol{x}|\boldsymbol{\theta}, z_k = 1)p(z_k = 1|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \end{aligned}$$

- which is identified as GMM model

Latent-Variable based Approach

- given a dataset $\mathcal{X}$, the likelihood

$$p(\mathcal{X}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\boldsymbol{\theta}) = \prod_{n=1}^{N}\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- which is exactly the GMM likelihood
- the latent variable model with latent indicators z_k is an equivalent way of thinking about Gaussian mixture model
- posterior distribution on latent variable $\boldsymbol{z}$, according to Bayes' theorem
- the posterior of kth component having generated data point $\boldsymbol{x}$

$$p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{p(\mathbf{x})}$$

$$p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{k=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Latent-Variable based Approach

- extension to a full dataset, the concepts of prior and posterior can be extended to the case of N data points $\mathcal{N} := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
- in the probabilistic interpretation of GMM, every data point $\boldsymbol{x}_n$ possesses its own latent variable

$$\boldsymbol{z}_n = [z_{n1}, \ldots, z_{nK}]^T \in \mathbb{R}^K$$

- conditional distribution

$$p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N | \boldsymbol{z}_1, \ldots, \boldsymbol{z}_N) = \prod_{n=1}^{N} p(\boldsymbol{x}_n | \boldsymbol{z}_n)$$

- posterior distribution

$$\begin{aligned} p(z_{nk} = 1 | \boldsymbol{x}_n) &= \frac{p(\boldsymbol{x}_n | z_{nk} = 1) p(z_{nk} = 1)}{\sum_{j=1}^{K} p(\boldsymbol{x}_n | z_{nj} = 1) p(z_{nj} = 1)} \qquad (3) \\ &= \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = r_{nk} \end{aligned}$$

Latent-Variable based Approach

- $p(z_k = 1|\boldsymbol{x}_n)$ is the posterior probability that kth mixture component generated data point $\boldsymbol{x}_n$ and
- corresponds to responsibility r_{nk}; mathematically justified interpretation as posterior probabilities
- EM algorithm - iterative scheme for maximum likelihood estimation can be derived from the latent variable perspective
- given a current setting $\boldsymbol{\theta}^{(t)}$ of model parameters,
- the *E-step* calculates the expected log-likelihood

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})\mathrm{I} &= \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta}^{(t)}}[\log p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta})] \\ &= \int \log p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta}^{(t)})d\boldsymbol{z} \end{aligned} \tag{4}$$

Latent-Variable based Approach

- the expectation of $\log p(\boldsymbol{x}, \boldsymbol{z}|\theta)$ is taken with respect to the posterior $p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})$ of the latent variables
- the *M-step* selects an updated set of model parameters $\boldsymbol{\theta}^{(t+1)}$ by maximizing the above equation
- although an EM iteration does increase the log-likelihood, there are no guarantees that EM converges to the maximum likelihood solution
- it is possible that EM algorithm converges to a local maximum of the log-likelihood
- different initializations of parameters $\boldsymbol{\theta}$ can be used to reduce the risk ofending up in a bad local optimum

CREDITS SOME CS GUY!