In the absence of any prior knowledge about the shape of $p(x)$ we can model it as a Gaussian.

Many times we can identify that the given observation x has been generated through a structured process.

That is, the distribution over the features $p(x)$ has an underlying structure.

If our model for $p(x)$ can capture this structure, then the model will be closely following (approximating) the underlying data generating distribution and will likely perform well.

An underlying structure to the data generating process would imply the presence of additional hidden (not observed) properties of data that describe (define) that structure.

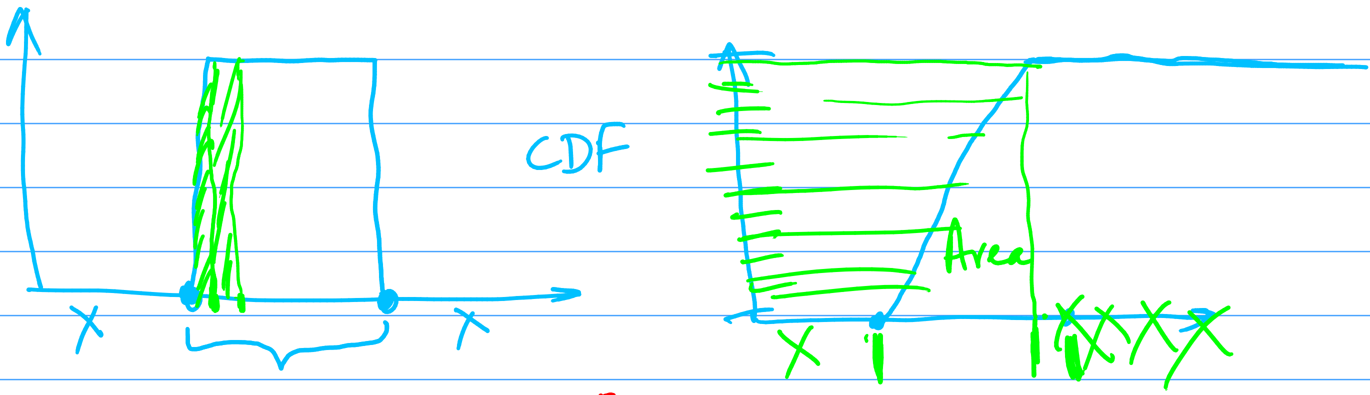
These hidden attributes (properties) are the latent attributes.

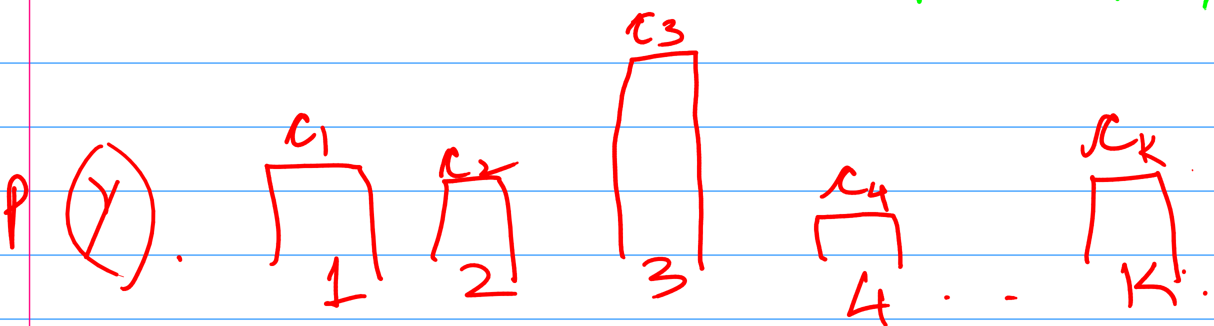
We dont know the values assigned to the latent attributes. These are unknown parameters of our model.

We treat them as uncertain (random) variables which follow an unknown distribution.

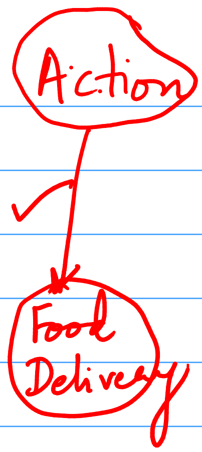
Given a $n \sim U$ uniform distribution

Cumulative distribution

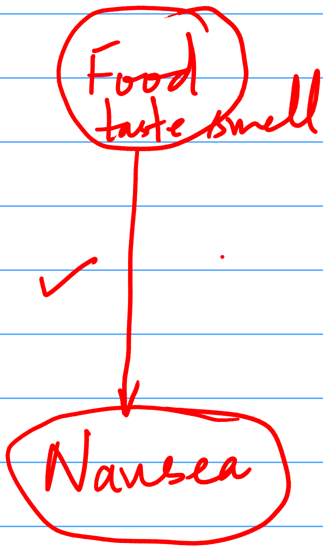


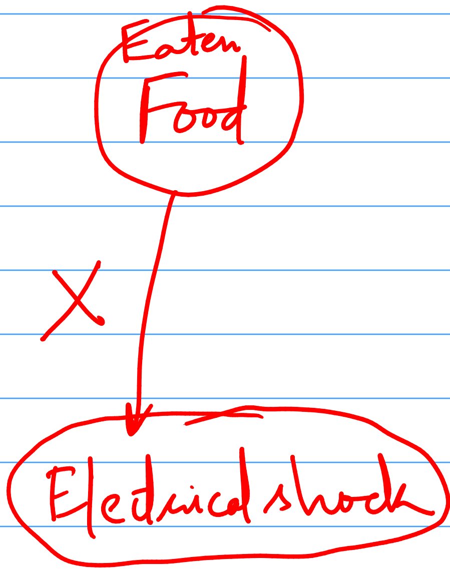


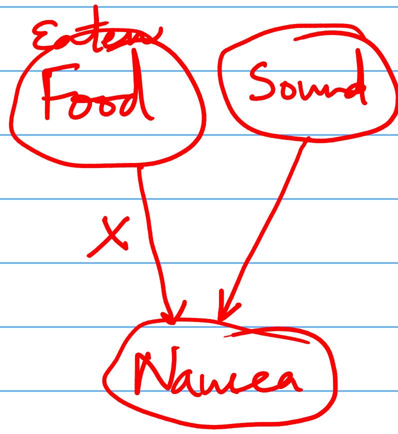
$$\sum_y c_y = 1.$$



Absurd
Useless
Hypothesis.







Inductive bias.

Inductive learning: learning from Examples
(training learning)

A simple form of structure is categorized examples (observation)

That is, the features that we observe belong to some category $Y = y_i$.

Such a structure is denoted graphically as

Y is a random variable taking discrete values. labels.

$Y \in [1, k]$ 1, 2, 3, ... k.

X is a random variable denoting the observation vector x_i

$X \in \mathbb{R}^d$.

To generate an observation, we first sample from a categorical variable $Y = y_i$ and then given the category y_i we sample an observation x_i from the conditional distribution $P(X = x_i \mid Y = y_i)$

Random variable Y has a prior distribution associated with it $P(Y = y_i)$. Since Y is discrete valued, we can have a categorical distribution to describe $P(Y = y)$.

Using such a structured model, the probability of observing example x_i can be written as $P(X = x_i, Y = y_i)$

unsupervised

$$P[X = x_i] = \sum_{y_i=1}^{k} P[Y = y_i] \, P[X = x_i \mid Y = y_i]$$

i.e. marginalizing (summing out) the joint distribution over the random variable Y.

$$P(a) = \sum_{b} P(a, b)$$

$$= \sum_{y_i=1}^{k} c_{y_i} \frac{1}{\sqrt{(2\pi)^d |\Sigma_{y_i}|}} \exp\left(-\frac{1}{2} (x - \mu_{y_i})^T \Sigma_{y_i}^{-1} (x - \mu_{y_i}) \right)$$

Multivariate Gaussian

d-dim space.

Here $P\{X = \underline{x}_i \mid Y = y_i\}$ has been modelled as a Gaussian distribution for every value of $Y = y_i$.

Since Y can take $\underline{k}$ values, there will be $\underline{k}$ such Gaussian distributions, with parameters $(\underline{\mu}_{y_i}, \Sigma_{y_i})$ ✓

$(\mu_y\ \Sigma_y)$

While referring to the parameters of these Gaussians, we drop the example specific subscript $\underline{i}$, and simply write

$(\underbrace{\underline{\mu}_y}_{d\times 1 \text{ vector}}, \underbrace{\Sigma_y}_{d\times d \text{ matrix}})$

Though we have introduced a structure for modelling the generative distribution $P(X)$, is there any advantage in assuming this structure?

incorporating prior knowledge → better learning.

And, how do we solve for the parameters

$$\underline{\Theta} = \{\underbrace{c_y}, \underbrace{\underline{\mu}_y, \Sigma_y}_{\text{Multivariate Gaussian}}\} \ \forall y \in [1,k] \quad h^*$$

Y — $P(Y)$ prior

$\downarrow$ $P(X|Y)$ likelihood of x..

X

where c_y are the parameters of the categorical distribution $P[Y=y]$ and $\underline{\mu}_y, \Sigma_y$ are the parameters of the conditional distribution $P\{X = \underline{x} \mid Y = y\}$ modeled as a multivariate Gaussian.

We formulate our solution $\underline{\Theta}$ to be the one that maximizes the likelihood of observing the examples in the training set $S = \{\underline{x}_1, \underline{x}_2, \dots \underline{x}_m\}$ MLE

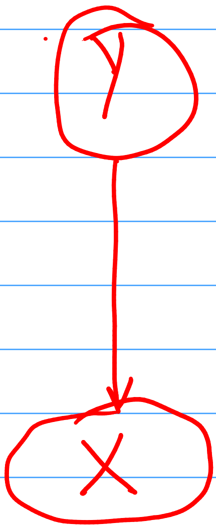
The likelihood of the training set S

independence assumption

$$\underbrace{P_{\underline{\Theta}}[S]} = \prod_{i=1}^{m} \underbrace{P_{\underline{\Theta}}[X = \underline{x}_i]}_{\underline{x}_i}$$

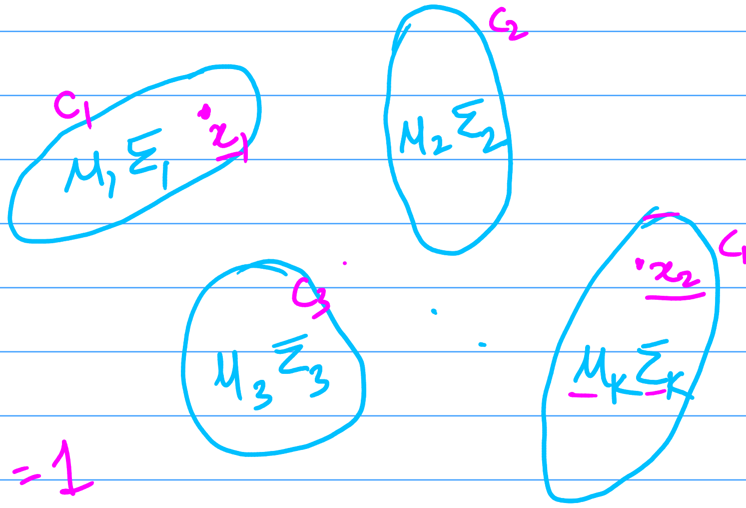
Gaussian Mixture Model (GMM)

Assume value $K=5$



$P(X)$ is maximized.

$\sum_y c_y = 1$



Hyperparameter
$K = 5$
$K = 3$

$K = 8$

$P(X)$

Try different values of values of K.
Check the fit quality

Penalty: Complexity of the model.

Assessment score. (fit quality + λ complexity)
(error) (penalty)
trade off

CV
M1 — CV
M2
M3 — CV

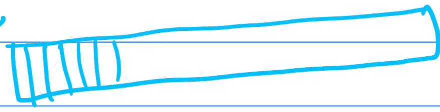
(training data)
(scarce data).

K-fold cross validation

Leave one out cross validation

$S : \{x_1 \, . \, . \, . \, . \, x_m\}$

$K = m$



K subsets

$K = 6$

Model

1.	2	3	4	5	6

M: fixed

Keep aside one subset

			Error
Session 1	Train (2, 3, 4, 5, 6)	Test (1)	e_1
2.	Train (1, 3, 4, 5, 6)	Test (2)	e_2
3	Train (1, 2, 4, 5, 6)	Test (3)	e_3
⋮			⋮
K			

Test Performance : Avg (e_i).
(Generalization performance)

$S : \begin{pmatrix} \overset{70\%}{\text{Train}} & \overset{30\%}{\text{Test}} \\ S_1 & S_2 \end{pmatrix}$ one-time partition.

Error(S_2).

For tuning hyper parameters

$K = 3$ KFCV — Test performance
$K = 4$ KFCV — ″
$K = 5$ . . -

To prevent numeric underflow, we take the log likelihood.

$$\underbrace{L(\underline{\theta})}_{\text{likelihood function.}} = \log\left(\prod_{i=1}^{m} P_{\underline{\theta}}[X = \underline{x_i}]\right)$$

$$= \sum_{i=1}^{m} \log \underbrace{P_{\underline{\theta}}[X = \underline{x_i}]}$$

$$\arg\max_{\underline{\theta}} L(\underline{\theta}) = \sum_{i=1}^{m} \log\left(\underbrace{\sum_{y_i=1}^{k} P_{\theta}[\underbrace{X = x_i}_{\text{obs}}, \underbrace{Y = y_i}_{\text{hidden}}]}\right)$$

Joint distribution. (given by the structural model).

i.e. we write $P[X = x_i]$ as a marginalization (summing out) of the latent variable Y over the joint distribution of observed and latent variables.

The optimal parameters for the model are the ones that maximize the likelihood function.

k: hyperparameter

$$\underline{\theta}^* = \arg\max_{\underline{\theta}} L(\underline{\theta}) = \arg\max_{\underline{\theta}} \sum_{i=1}^{m} \log\left(\underbrace{\sum_{y_i=1}^{k}} \overbrace{P_{\theta}[X = x_i \mid Y = y_i]}^{\text{exp Gaussian}}\right)$$

$c_y \; \mu_y \; \Sigma_y$

This maximization is hard because the log acts on the summation.

Therefore, we maximize $L(\underline{\theta})$ by formulating a surrogate function $G(Q, \underline{\theta})$ which takes in another parameter. a matrix Q of size $m \times k$

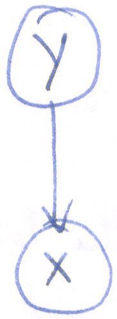
$$\underset{m \times k}{Q} = \begin{array}{c|c} & 1 \;\; 2 \;\; \cdots \;\; k \;\; (y \rightarrow) \\ \hline 1 & \\ 2 & \\ \vdots & \\ i \downarrow & \begin{bmatrix} & & & \\ & & Q_{i,y} & \\ & & & \\ p_1^m & p_2^m & p_3^m & p_k^m \end{bmatrix} \\ \vdots & \\ m & \end{array}$$

posterior $p(Y = y_i \mid X = \underline{x_i})$

Every row of the matrix Q is a categorical distribution over k discrete values

$\sum$ rowentries $= 1$

We are uncertain about the hidden variables. Y
We cannot observe them.
We capture this uncertainty by defining a posterior distribution over $Y = y_i$ given every training example $X = x_i$.



$P[Y = y_i]$ = prior

$P[Y = y_i \mid X = x_i]$: posterior

$P[X = x_i \mid Y = y_i]$: likelihood

The matrix Q captures the posterior distribution

$G(Q, \underline{\theta})$

$$Q_{i,y} = P[Y = y_i \mid X = x_i]$$

responsibility that the y_i^{th} Gaussian takes for the feature $\underline{x}_i$

The surrogate function $G(Q, \underline{\theta})$ is formulated as

$$G(Q, \underline{\theta}) = F(Q, \underline{\theta}) - \underbrace{\sum_{i=1}^{m} \sum_{y=1}^{k} Q_{i,y} \log Q_{i,y}}$$

for every row i : Entropy of the cat dist

$$F(Q, \underline{\theta}) = \sum_{i=1}^{m} \sum_{y_i=1}^{k} \underbrace{Q_{i,y_i}}_{\text{Posterior distr over } Y = y_i} \log \underbrace{\left(P_{\underline{\theta}}[X = \underline{x}_i, Y = y_i] \right)}_{\text{likelihood of Complete data: Observed + hidden random variables}}$$

$\therefore$ the function $F(Q, \underline{\theta})$ gives the expectation of the log likelihood of the complete data with respect to the posterior distribution over the latent (hidden) variable Y.