CS 461: Machine Learning Principles

Class 22: Nov 21

Gaussian **M**ixture **M**odeling (GMM)

& K-means Clustering

Instructor: Diana Kim

Outline

1. GMM (**G**aussian **M**ixture **M**odeling) and Learning its Parameter

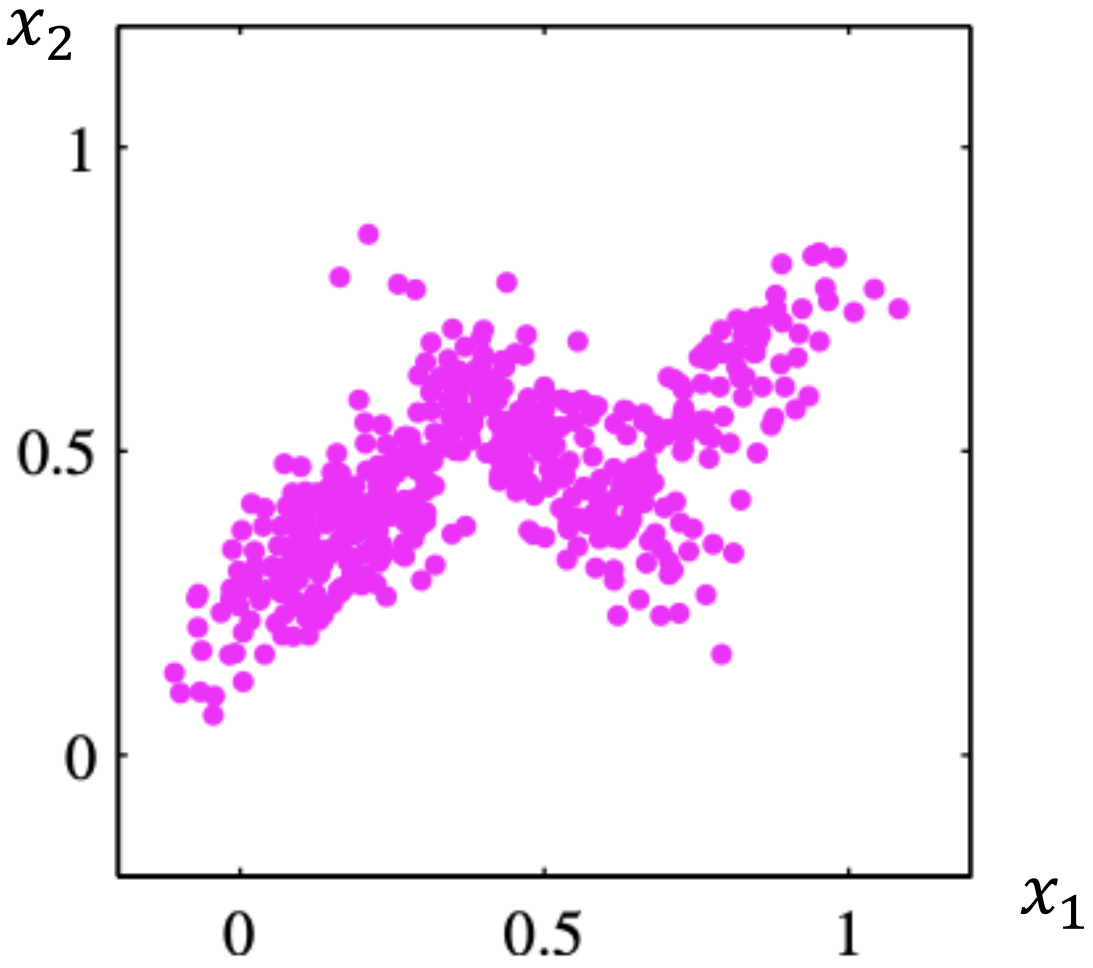
2. The Review of EM (**E**xpectation and **M**aximization) Algorithm

3. K-means Clustering Algorithm

4. Revisit Building RBF Basis Functions
   (Learning the basis using K-means)

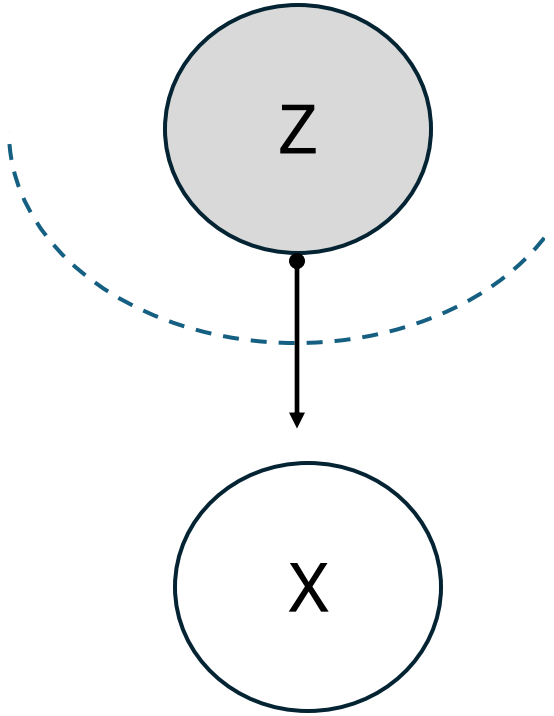
GMM models general density distributions that have multiple modes.

Data often has multiple modes.
How would you learn the density $f(x_1, x_2)$?



From textbook Bishop Fig. 9.5

Learning the density of X by adopting a hidden R.V Z
The Bayesian Network Representation



- $P(Z = k) = \pi_k$
- $P(X|Z = k) = \mathcal{N}(\mu_k, \Sigma_k)$

- $P(X)$ is marginalization of $P(Z, X)$

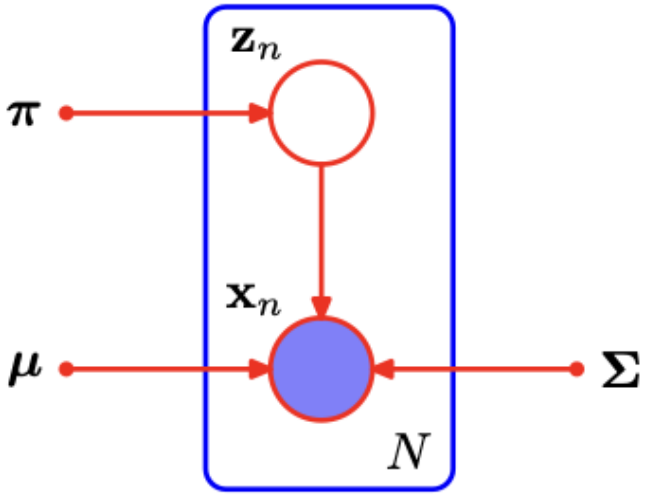
$$P(X) = \sum_Z P(Z, X) = \sum_Z P(Z)P(X|Z) = \sum_{Z=k} \pi_k \mathcal{N}(\mu_k, \Sigma_k)$$

Linear Superposition of Gaussian Components

["Plate" Notation]

Figure 9.6 Graphical representation of a Gaussian mixture model for a set of N i.i.d. data points $\{\mathbf{x}_n\}$, with corresponding latent points $\{\mathbf{z}_n\}$, where $n = 1, \ldots, N$.

From textbook Bishop Fig. 9.6



Data (X) by GMM modeling

$$P(X) = \sum_{Z} P(Z, X) = \sum_{Z} P(Z)P(X|Z) = \sum_{Z=k} \pi_k \mathcal{N}(\mu_k, \Sigma_k)$$

Learning the parameters : π_k, μ_k, Σ_k

We cannot access Z, and we can only access X.

How can we learn the parameters?

+ using EM algorithm

[About MLE]

We know MLE (Maximum Likelihood Estimation).
When $X \sim f(x \mid \theta)$ and data points are $i.i.d$,
we can estimate $\theta^* = argmax_\theta \prod_{n=1}^{N} f(x_n \mid \theta)$.

$$= argmax_\theta \sum_{n=1}^{N} \log\big(f(x_n)\big)$$

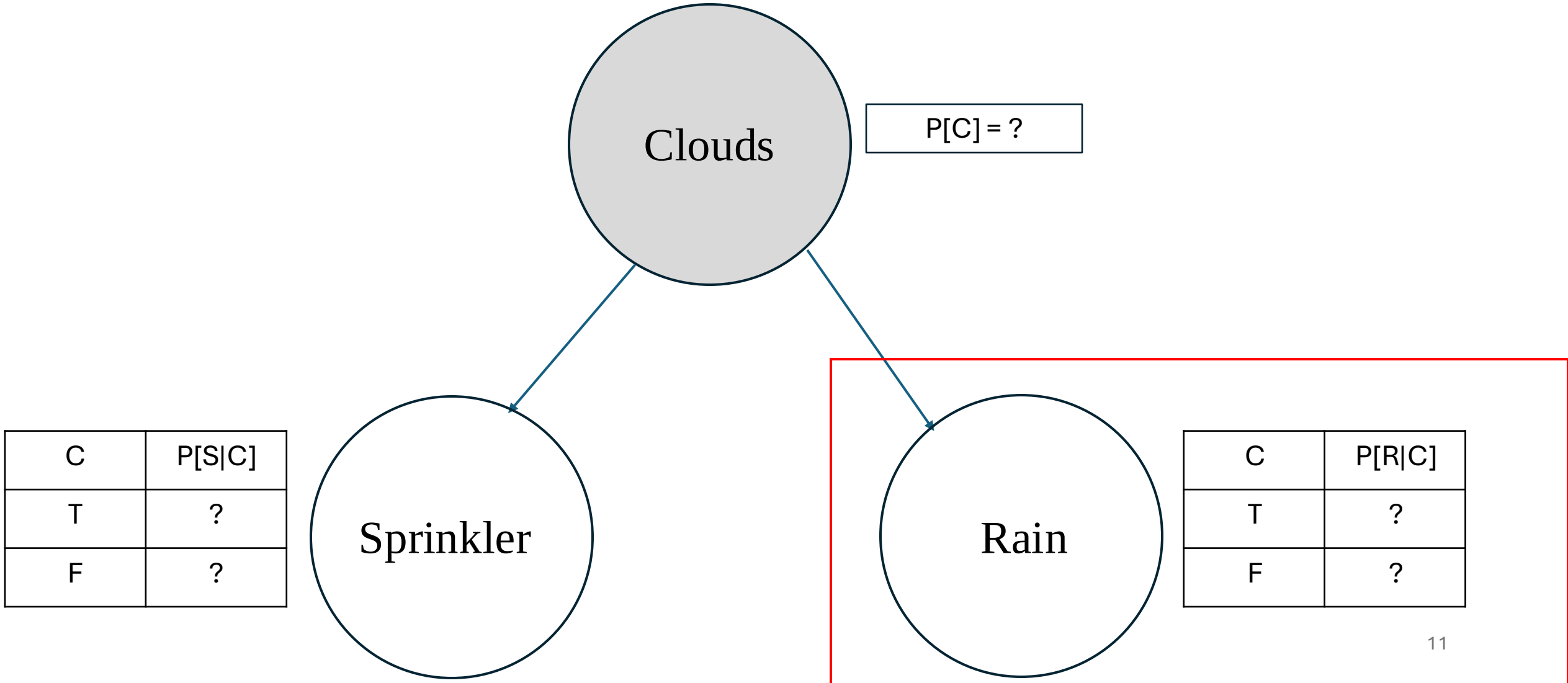
However, sometimes the optimization is NP-hard problem.
Learning the parameters for GMM is the case.

We need a heuristic method to optimize like gradient descent or EM algorithms.

Review of EM

(the goal of EM is
to conduct MLE when the optimization of a targeted likelihood is intractable for latent variable.)

When a graphical structure is known but data is partially observed,
How can we estimate CPTs (they are the parameters to estimate)?



C	P[S|C]
T	?
F	?

C	P[R|C]
T	?
F	?

Log Likelihood Comparison of Full vs. Partial Data Observation

$$\log P(D|\theta)) = \sum_{n=1}^{N} \log P(S_n, R_n, C_n|\theta)$$

$$= \sum_{n=1}^{N} \log P(C_n) + \log P(S_n|C_n) + \log P(R_n|C_n)$$

Cloud is missing

$$\log P(D|\theta) = \sum_{n=1}^{N} \log \sum_{C_n} P(S_n, R_n, C_n|\theta)$$

$$= \sum_{n=1}^{N} \log(P(C_n+)P(S_n|C_n+)P(R_n|C_n+)$$

$$+ P(C_n-)P(S_n|C_n-)P(R_n|C_n-))$$

log sum can not be simplified.

hard to optimize (parameters are interdependent)!

It’s hard to get a closed form to compute MLE,
when there are missing variables.
(log \sum is not favorable; can we change it to \sum log like the complete case?)

Jensen's Inequality: (both expectation are finite)

$E[\log X] \leq \log E[X]$

$$\log P(D|\theta) = \sum_{n=1}^{N} \log \sum_{C_n} P(S_n, R_n, C_n|\theta)$$

$$= \sum_{n=1}^{N} \log \sum_{C_n} \frac{P(S_n, R_n, C_n|\theta) q(C_n)}{q(C_n)}$$

$$= \sum_{n=1}^{N} \log E[\frac{P(S_n, R_n, C_n|\theta)}{q(C_n)}]$$

$$\geq \sum_{n=1}^{N} E[\log(\frac{P(S_n, R_n, C_n|\theta)}{q(C_n)})]$$

$$\geq \sum_{n=1}^{N} E[\log P(S_n, R_n, C_n|\theta)] + H(q(C_n))$$

This gives a lower bound of the target to optimize; the target is $\log P(D|\theta)$. The bound is tight (contact at θ_t) when the expectation is computed by $P(C_i|R_i, S_i\ \theta_t)$

EM Algorithm

to compute $\underline{argmax_\theta P(D|\theta)}$ for the example of cloud, rain, sprinkler.

1. start with arbitrary parameters: θ
2. **E step:** compute $P(C_n | S_n, R_n, \theta)$ for $\forall\, n$
3. to compute $\sum_{n=1}^{N} E[\log P(S_n, R_n, C_n|\theta)]$

4. **M step**: Update $\theta' = argmax_\theta \sum_{n=1}^{N} E[\log P(S_n, R_n, C_n|\theta)]$

5. go back to step 2 and compute new $P(C_n | S_n, R_n, \theta')$ for $\forall\, n$

Q) Why we do not consider the entropy term in M step?

+ it does not depend on \theta.

Auxiliary Function $\mathrm{Q}(\theta, \theta^t) = \sum_{n=1}^{N} E[\log P(S_n, R_n, C_n|\theta)]$

expectation over $P(C_n | S_n, R_n, \theta^t)$

This is a function of θ $defined\ at\ \theta^t$

- $\mathrm{Q}(\theta^t, \theta^t) = \sum_i \log \mathrm{P}[S_i, R_i | \theta^t]$
- $\mathrm{Q}(\theta^{t+1}, \theta^t) \geq \mathrm{Q}(\theta^t, \theta^t)$
- $\mathrm{Q}(\theta^{t+1}, \theta^{t+1}) \geq \mathrm{Q}(\theta^{t+1}, \theta^t)$
- $\mathrm{Q}(\theta^{t+1}, \theta^{t+1}) = \sum_i \log \mathrm{P}[S_i, R_i | \theta^{t+1}]$

EM monotonically increases the observed data log likelihood!

- $\sum_i \log \mathrm{P}[S_i, R_i | \theta^t] \leq (\theta^{t+1}, \theta^t) \leq \mathrm{Q}(\theta^{t+1}, \theta^{t+1}) = \sum_i \log \mathrm{P}[S_i, R_i | \theta^{t+1}]$

EM algorithm finds a local minimum.

$\log P(X|\theta)$ is often $non-convex$ or $non-concave$.

Q: how could we handle the issue of local minimum?

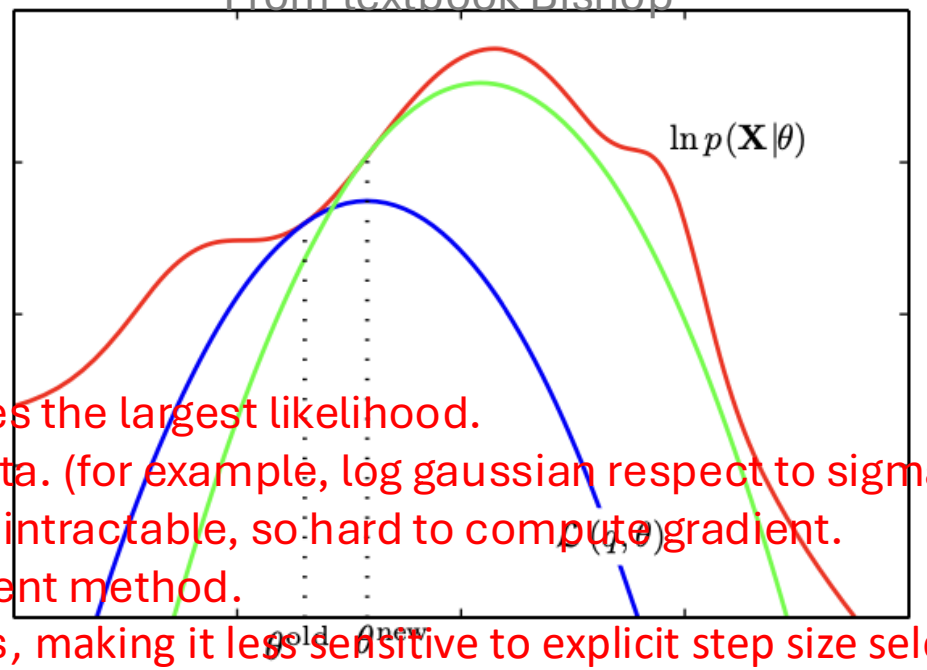
Q: is the lower bound always concave?

Q: can we use the gradient descent method to solve the target problem?

Q: how EM is different from gradient descent method?

From textbook Bishop

Figure 9.14 The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values and then maximizing this bound to obtain the new parameter values. See the text for a full discussion.



+ Q: find EM solutions for different initialization and pick the \theta gives the largest likelihood.
+ Q: no, the lower bound is not always concave function respect to theta. (for example, log gaussian respect to sigma)
+ Q: It is possible, but the marginalization over latent variables is often intractable, so hard to compute gradient.
  however, for GMM, it is possible to compute with stochastic gradient method.
+ Q: Em inherently defines a "right" step size within its iterative process, making it less sensitive to explicit step size selection because it updates parameters based on the expected complete data likelihood, leading to a more stable convergence behavior.

EM Solution for GMM

Likelihood of GMM and the Lower Bound

$$
\begin{aligned}
\log P(D|\theta) &= \sum_{n=1}^{N} \log \sum_{Z_n=k} P(Z_n, X_n|\theta) \\
&= \sum_{n=1}^{N} \log \sum_{Z_n=k} \frac{P(Z_n, X_n|\theta) q(Z_n = k)}{q(Z_n = k)} \\
&= \sum_{n=1}^{N} \log E[\frac{P(Z_n, X_n|\theta)}{q(Z_n = k)}] \\
&\geq \sum_{n=1}^{N} E[\log (\frac{P(Z_n, X_n|\theta)}{q(Z_n = k)})] \\
&= \sum_{n=1}^{N} \sum_{k} q(Z_n = k) \cdot (\log \pi_k + \log \mathcal{N}(x_n|\mu_k, \Sigma_k)) + \sum_{n=1}^{N} H(q(Z_n = k))
\end{aligned}
$$

Q What is $q(z_n = k)$?

- E step of GMM

parameters at time t

$$
\begin{aligned}
q(Z_n = k) &= P(Z_n = k | X_n = x_n, \pi_k(t), \mu_k(t), \Sigma_k(t)) \\
&= \frac{P(Z_n = k, X_n = x_n | \pi_k(t), \mu_k(t), \Sigma_k(t))}{P(X_n = x_n | \pi_t, \mu_t, \Sigma_t)} \\
&= \frac{P(Z_n = k | \pi_k(t), \mu_k(t), \Sigma_k(t)) P(X_n = x_n | Z_n = k, \pi_k(t), \mu_k(t), \Sigma_k(t))}{P(X_n = x_n | \pi_k(t), \mu_k(t), \Sigma_k(t))} \\
\gamma_{nk}(\pi_k(t), \mu_k(t), \Sigma_k(t)) &= \frac{P(Z_n = k | \pi_t, \mu_t, \Sigma_t) P(X_n = x_n | Z_n = k, \pi_k(t), \mu_k(t), \Sigma_k(t))}{P(X_n = x_n | \pi_k(t), \mu_k(t), \Sigma_k(t))}
\end{aligned}
$$

- **responsibility** that cluster K takes for data point n

- M step of GMM
  [1] update μ_k

$$\sum_{n=1}^{N}\sum_{k} q(Z_n = k)\cdot(\log \pi_k + \log \mathcal{N}(x_n|\mu_k, \Sigma_k)) + \sum_{n=1}^{N} H(q(Z_n = k))$$

+ this is the lower bound we computed in the previous slide.

$$L = \sum_{n=1}^{N}\sum_{k} q(Z_n = k)\cdot(\log \pi_k + \log \mathcal{N}(x_i|\mu_k, \Sigma_k)) + \sum_{n=1}^{N} H(q(Z_n = k))$$

$$L = \sum_{n=1}^{N}\sum_{k} r_{nk}\cdot(\log \pi_k + \log \mathcal{N}(x_n|\mu_k, \Sigma_k)) + \sum_{n=1}^{N} H(r_{nk})$$

$$\frac{\partial L}{\partial \mu_k} = \frac{\partial}{\partial \mu_k}\sum_{n=1}^{N}(-1/2\gamma_{nk}(x_n - \mu_k)^t\Sigma^{-1}(x_n - \mu_k))$$

$$\frac{\partial L}{\partial \mu_k} = \sum_{n=1}^{N}\gamma_{nk}\Sigma^{-1}(x_n - \mu_k) = 0$$

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

$\mathbf{y}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$
$\mathbf{Ax}$	$\mathbf{A}^T$
$\mathbf{x}^T\mathbf{A}$	$\mathbf{A}$
$\mathbf{x}^T\mathbf{x}$	$2\mathbf{x}$
$\mathbf{x}^T\mathbf{Ax}$	$\mathbf{Ax} + \mathbf{A}^T\mathbf{x}$

- M step of GMM
  [2] update Σ_k

$$L = \sum_{n=1}^{N} \sum_{k} r_{nk} \cdot (\log \pi_k + \log \mathcal{N}(x_n | \mu_k, \Sigma_k)) + \sum_{n=1}^{N} H(r_{nk})$$

$$\frac{\partial L}{\partial \Sigma_k^{-1}} = \frac{\partial}{\partial \Sigma_k^{-1}} \sum_{n=1}^{N} (-1/2(\gamma_{nk}(x_n - \mu_k)^t \Sigma^{-1}(x_n - \mu_k) - \log \sqrt{(2\pi)^n \cdot 1/|\Sigma^{-1}|}))$$

$$= \sum_{n=1}^{N} -1/2\gamma_{nk}(x_n - \mu_k)^t(x_n - \mu_k) + \gamma_{nk}1/2 \cdot \Sigma = 0$$

$$\Sigma = \frac{\sum_{n=1}^{N} \gamma_{nk}(x_n - \mu_k)^t(x_n - \mu_k)}{\sum_{n=1}^{N} \gamma_{nk}}$$

$$\frac{\partial}{\partial A} \log |A| = A^{-T}$$

$$\frac{\partial}{\partial A} x^T A x = \frac{\partial}{\partial A} \mathrm{tr}[x x^T A] = [x x^T]^T = x x^T$$

- M step of GMM
  [3] update π_k

$$L = \sum_{n=1}^{N}\sum_{k} r_{nk} \cdot (\log \pi_k + \log \mathcal{N}(x_n|\mu_k, \Sigma_k)) + \sum_{n=1}^{N} H(r_{nk}) + \lambda^*(\sum_k \pi_k - 1)$$

$$\frac{\partial L}{\partial \pi_k} = \frac{\partial}{\partial \pi_k}\sum_{n=1}^{N}(\gamma_{nk}\log \pi_k) + \lambda^* \pi_k$$

$$\frac{\partial L}{\partial \pi_k} = \sum_{n=1}^{N} \gamma_{nk}/\pi_k + \lambda^* = 0$$

$$\pi_k^* = \frac{\sum_n^N \gamma_{nk}}{\lambda^*}$$

$$\pi_k^* = \frac{\sum_n^N \gamma_{nk}}{N}$$

- *we know* $\sum_k \gamma_{nk} = 1$

$$\frac{\partial L}{\partial \lambda} = \sum_k \pi_k = 1$$

$$\sum_k \pi_k = \sum_k \frac{\sum_n^N \gamma_{nk}}{\lambda^*} = 1$$

$$N/\lambda^* = 1$$

$$\lambda^* = N$$

- M steps for GMM

- $\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk}}$

- $\Sigma_k = \frac{\sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)^t (x_n - \mu_k)}{\sum_{n=1}^{N} \gamma_{nk}}$

- $\pi_k = \frac{\sum_{n}^{N} \gamma_{nk}}{N}$

They are the sample means weighted by responsibility

$\gamma_{nk} = P[\, Z_k = \mathrm{k} \,|X_k = x_k, \mu_k(t), \pi_k(t), \Sigma_k\,(t)]$!!

Summary of EM for GMM

[1] E step

compute $\gamma_{nk} = P[\, Z_k = \mathrm{k} \,|X_k = x_k, \mu_k(t), \pi_k(t), \Sigma_k\,(t)]$

[2] M step

- $\mu_k = \dfrac{\sum_{n=1}^{N} \gamma_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk}}$
- $\Sigma_k = \dfrac{\sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)^t (x_n - \mu_k)}{\sum_{n=1}^{N} \gamma_{nk}}$
- $\pi_k = \dfrac{\sum_{n}^{N} \gamma_{nk}}{N}$

Let’ s go back to the problem of
Learning the parameters of Bayesian network.
We can use the concept of weighted sample mean in the Bayesian net, too!

what if data is partially observed?
Suppose we forgot to collect cloud information.
Can we still estimate *p or q : P*[rain| cloud +] or *P*[rain| cloud -]?

EM for learning a Bayesian Network (The cloud information is unknown.)

- E step : compute $\gamma_{nk}(t) = P[Cloud(n) = k \mid Rain(n) \ and \ S(n)]$

- M step: update the parameters

  - $P(C)(t+1) = \frac{\sum_{n=1}^{N} \gamma_{nk}(t)}{N}$ + weighted samples with the responsibility

  - $P(S:+, R:+|C:+)(t+1) = \frac{\sum_{n=1}^{N} \gamma_{nk} \delta(x_n = S:+, R:+)}{\sum_{n=1}^{N} \gamma_{nk}}$

Can we use GMM for Clustering?

What is clustering?

It is to assign a category / group to each data points.

How is that different from classification?

+ no predefined labels
+ clustering is unsupervised learning.

Once we complete EM for GMM
we can use responsibility for clustering.

$$\gamma_{nk} = P[\, Z_k = \mathrm{k} \,|X_k = x_k, \mu_k(t), \pi_k(t), \Sigma_k\,(t)]$$

Summary of EM for GMM
The EM performs a soft clustering.

[1] E step

compute $\gamma_{nk} = P[\, Z_k = \mathrm{k} \,|X_k = x_k, \mu_k(t), \pi_k(t), \Sigma_k\,(t)]$

[2] M step

- $\mu_k = \dfrac{\sum_{n=1}^{N} \gamma_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk}}$
- $\Sigma_k = \dfrac{\sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)^t (x_n - \mu_k)}{\sum_{n=1}^{N} \gamma_{nk}}$
- $\pi_k = \dfrac{\sum_{n}^{N} \gamma_{nk}}{N}$

Then, how could we perform hard clustering:
i.e. hard assignment of data points to clusters?

Based on the responsibility P[Zk=k|x|,
We assign a label (k) that maximizes the responsibility.

Let’s take a look at this again!

[1] E step

compute $\gamma_{nk} = P[\, Z_k = \mathrm{k} \,|X_k = x_k, \mu_k(t), \pi_k(t), \Sigma_k\,(t)]$

[2] M step

- $\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk}}$
- $\Sigma_k = \frac{\sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)^t (x_n - \mu_k)}{\sum_{n=1}^{N} \gamma_{nk}}$
- $\pi_k = \frac{\sum_{n}^{N} \gamma_{nk}}{N}$

K-means clustering even more simple!

$$= \frac{P(Z_n = k|\pi_k(t), \mu_k(t), \Sigma_k(t))P(X_n = x_n|Z_n = k, \pi_k(t), \mu_k(t), \Sigma_k(t))}{P(X_n = x_n|\pi_k(t), \mu_k(t), \Sigma_k(t))}$$

[1] E step

compute $\gamma_{nk} = 1 \ when \ k = argmin\_\{k\} \ P[\ Z_k = \mathrm{k}\ |X_k = x_k, \mu_k(t), \pi_k(t), \Sigma_k\ (t)]$

$\gamma_{nk} = 0 \ o.w$

Prob $\propto \left|\left|x_n - x_k\right|\right| ** 2$

[2] M step

- $\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk}}$

- $\Sigma_k = \frac{\sum_{n=1}^{N} \gamma_{nk}(x_n - \mu_k)^t(x_n - \mu_k)}{\sum_{n=1}^{N} \gamma_{nk}}$

- $\pi_k = \frac{\sum_{n}^{N} \gamma_{nk}}{N}$

K-means Algorithm (#k is assumed/ given)

(1) Choose some initial values for μ_k

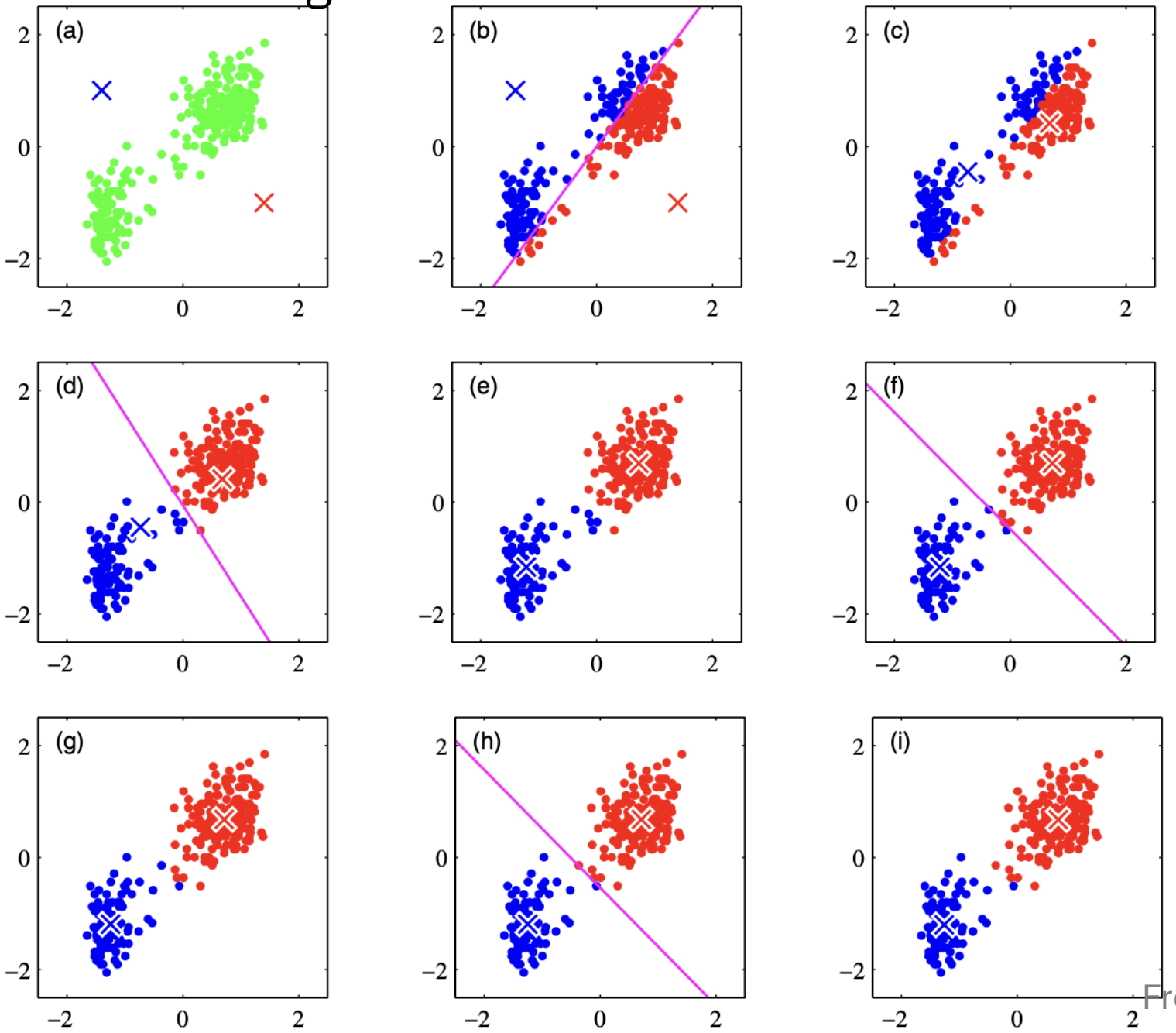
(2) Compute (E step)

- $\gamma_{nk} = 1 \ when\ k = argmin\_\{k\}||x_n - \mu_k(t)|| ** 2$
- $\gamma_{nk} = 0\ o.w$

(3) Compute (M step)

- $\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk}}$

Illustration of K means Algorithm

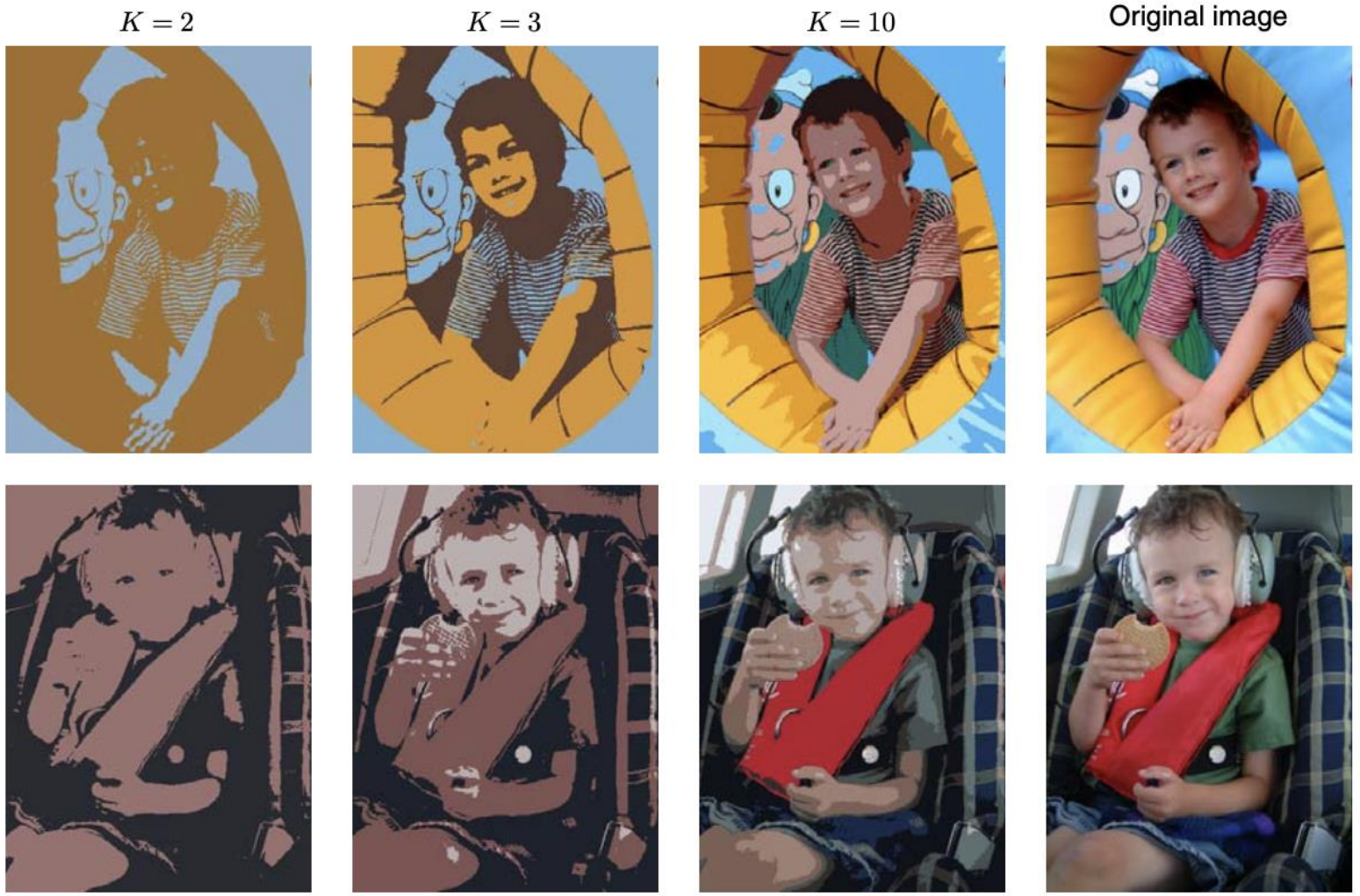


From textbook Bishop Fig. 9.1

Example of K means Algorithm

(1) Image Segmentation

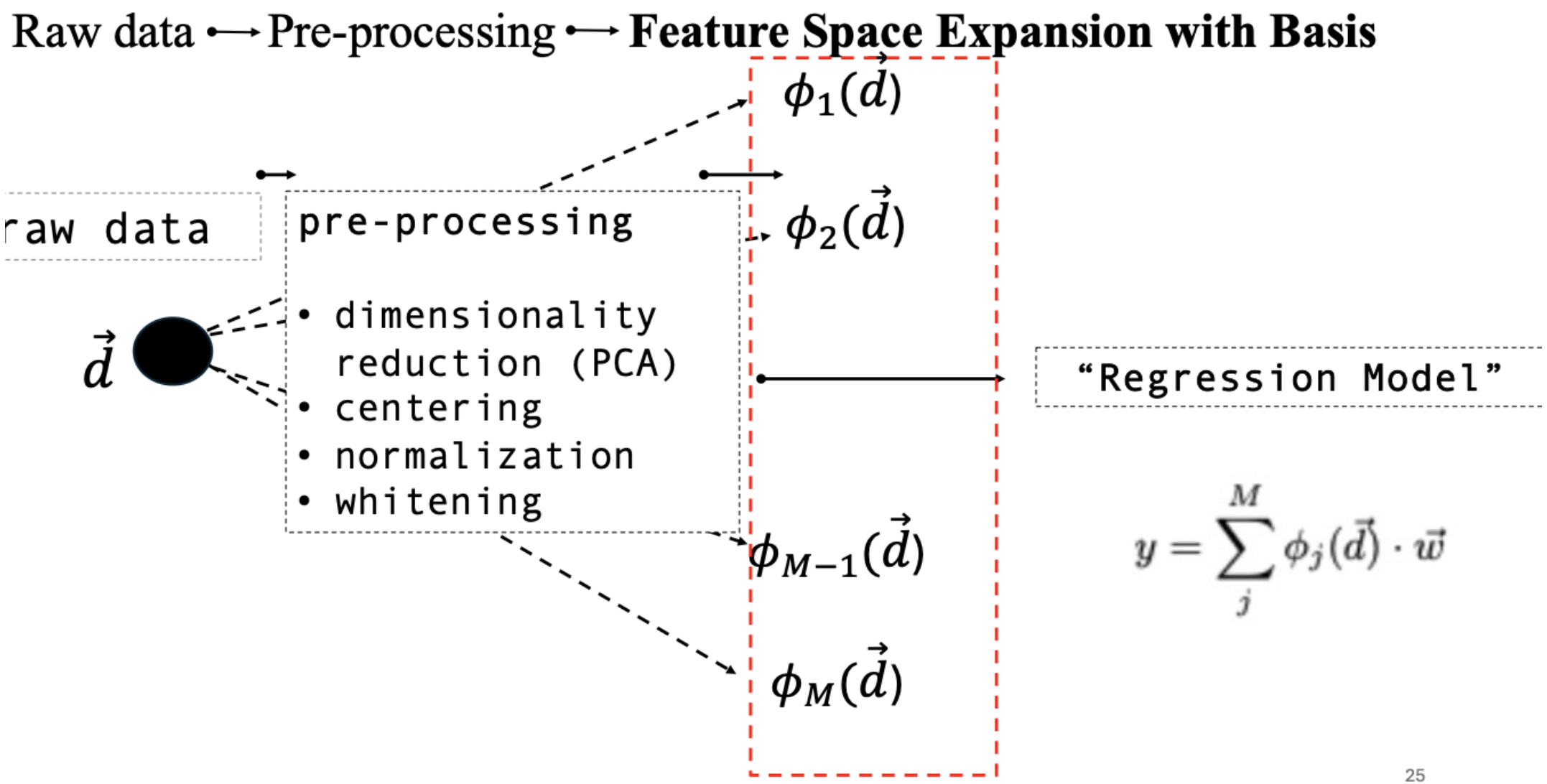
(2) Image Compression



From textbook Bishop Fig. 9.3

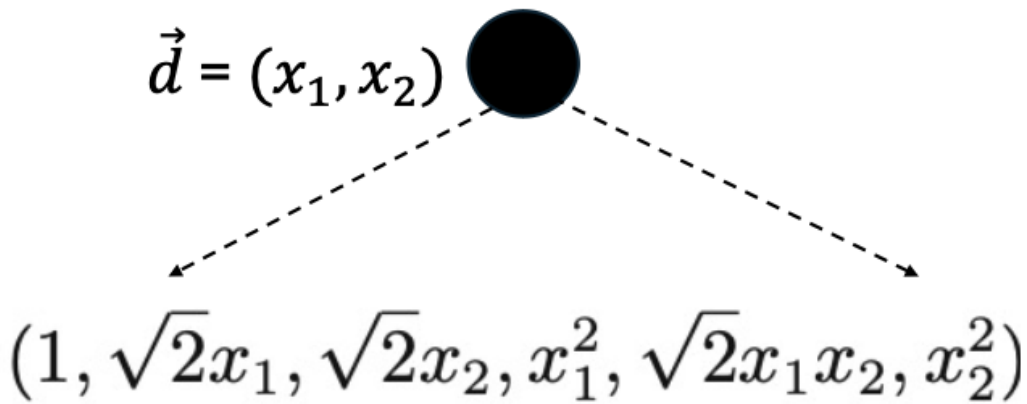
K means can be used for RBF basis formulation.
(Topic on Sept 19)

+ By using K-means, we can discover a set of \mu to set Radial basis functions for feature representation.
+ \sigma is pre-defined as a design factor.


Raw data
Pre-processing
Feature Space Expansion with Basis
raw data
pre-processing
• dimensionality reduction (PCA)
• centering
• normalization
• whitening
$\vec{d}$
$\phi_1(\vec{d})$
$\phi_2(\vec{d})$
$\phi_{M-1}(\vec{d})$
$\phi_M(\vec{d})$
"Regression Model"
$y = \sum_j^M \phi_j(\vec{d}) \cdot \vec{w}$

Basis Function (1) : Polynomial Expansion

+ Pre-processed data:
  (reduced dimensions and whitened)



	1	x_2	${x_2}^2$
1	1	x_2	${x_2}^2$
x_1	x_1	x_1x_2	×
${x_1}^2$	${x_1}^2$	×	×

Basis Function (2) : Gaussian Basis function

$$\phi_j = \exp\{-\frac{(x-\mu_j)^2}{2\sigma^2}\}$$

Q: the magnitude of σ^2?
(small: local and spiky vs. large: global and smooth)
Q: the locations of μ_j? (dense / sparse)