EE613

Machine Learning for Engineers

Generative models. Introduction to Graphical models

jean-marc odobez

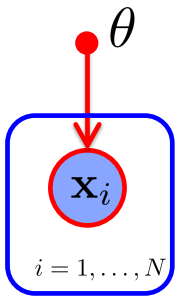
2019

overview

- Graphical models fundamentals
  - bayesian networks, representations
  - probabilities factorization
  - conditional independence
  - undirected graphical models
- Learning
  - Maximum Likelihood, Bayesian learning, Maximum a Posteriori (MAP)
  - the EM algorithm, latent variable models
    - Gaussian Mixture Model (GMM)
    - Hidden Markov Model (HMM)
- PCA, Probabilistic PCA
- Inference algorithms

learning in graphical models: maximum likelihood

- assuming
  - a parametric form $p(\mathbf{x}|\theta)$
  - identically independently distributed (i.i.d.) data drawn from the pdf
$$D = \{\mathbf{x}_n\}, \quad n = 1, \ldots, N$$



- **likelihood function** (of the parameters given the data)
$$L(\theta|D) = p(D|\theta) = \prod_{n=1}^{N} p(\mathbf{x}_n|\theta)$$
- **learning:** find parameters that **maximize**
  - **the likelihood**
$$\theta_{ML} = \arg\max_{\theta} L(\theta|D)$$
  - or the log-likelihood (analytically simpler)
$$\theta_{ML} = \arg\max_{\theta} \log\left(L(\theta|D)\right)$$

Example : categorical distribution for the data (1)

- for a discrete variable taking 1 out of K values
$$p(\mathbf{x} = k|\boldsymbol{\mu}) = \mu_k \qquad \boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)^{\mathrm{T}} \qquad \textstyle\sum_k \mu_k = 1$$
parameters to learn
- 1-of-k coding scheme
$$\mathbf{x} = (x_1, ..., x_k, ..., x_K)^T, \quad x_k \ \{0, 1\} \qquad \sum_k x_k = 1$$
$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k}$$

Example : categorical distribution for the data (2)

- Likelihood

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k}$$

$$p(D|\mu) = \prod_i p(\mathbf{x}_i|\mu) = \prod_i \prod_{k=1}^{K} \mu_k^{x_{ik}} = \prod_k \mu_k^{\sum_i x_{ik}}$$

=> $$\log p(D|\mu) = \sum_i \sum_{k=1}^{K} x_{ik} \mu_k = \sum_{k=1}^{K} N_k(D) \mu_k$$

$$N_k(D) = \sum_i x_{ik} \quad \longleftarrow$$ counts of data that have label k

- Maximizing log-likelihood + ensuring that $\sum_k \mu_k = 1$
  => lagrange multiplier

- Maximum likelihood $$\mu_k^{ML} = \frac{N_k(D)}{\sum_{k'} N_{k'}(D)}$$

Example : categorical distribution for the data (3)

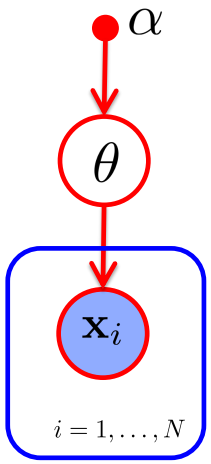
- Throw a dice N = 10 times
  => Data (K=6) = {1,1,6,2,5,3,1,6,2,1 } (not in 1-K encoding format)

k	1	2	3	4	5	6
$N_k(D)$	4	2	1	0	1	2
μ_k^{ML}	0.4	0.2	0.1	0	0.1	0.2

- p(x=4) is estimated as 0
  => if we throw the dice again, is the probability of seeing a 4 really 0 ?

- might be due to small sample size

Bayesian learning



- ML learning
  - Issue: parameters may overfit data (often depends on number of samples vs number of parameters to estimate)
- Bayesian
  - add prior on parameter $p(\theta|\alpha)$
  - compute posterior

$$p(\theta|D,\alpha) \propto p(D|\theta,\alpha)p(\theta|\alpha)$$
$$p(\theta|D,\alpha) \propto p(D|\theta)p(\theta|\alpha)$$

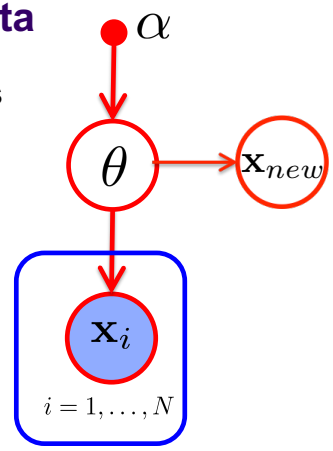
- Note
  - in Bayesian, learning is just a specific case of inference
  - the posterior can be used in several ways

    e.g Maximum a Posteriori (MAP) estimate relies on the mode

$$\theta_{MAP} = \arg\max_{\theta} \log p(\theta|D,\alpha)$$
$$\theta_{MAP} = \arg\max_{\theta} (\log p(D|\theta) + \log p(\theta|\alpha))$$

  - we could also consider the expected parameters under the distribution (i.e the mean)

Bayesian learning – handling new data



- Sometimes, we are not interested in the parameters
- The question is

  what is the likelihood of a new data point,

  given training data?

- Use ML or MAP estimates

$$p(\mathbf{x}_{new}|\boldsymbol{\theta}_{ML}) \quad p(\mathbf{x}_{new}|\boldsymbol{\theta}_{MAP})$$

- Issues
  - uses a point estimate of parameters
  - does not reflect uncertainties on parameter estimation => might be overconfident

  => full Bayesian treatment

$$p(\mathbf{x}_{new}|D,\alpha) \propto \int_{\boldsymbol{\theta}} p(\mathbf{x}_{new},\boldsymbol{\theta}|D,\alpha)d\boldsymbol{\theta} = \int_{\boldsymbol{\theta}} p(\mathbf{x}_{new}|\boldsymbol{\theta})p(\boldsymbol{\theta}|D,\alpha)d\boldsymbol{\theta}$$

  => posterior depend on training data

  => if posterior has a parametric form:

  we can use it and throw away the training data

Bayesian learning

$$p(\theta|D,\alpha) \propto p(D|\theta)p(\theta|\alpha)$$

posterior $\propto$ **likelihood** $\times$ **prior**

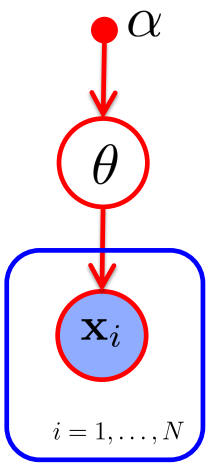
- Prior
  - favors some parameters over others
  - should reflect our knowledge about the problem
  - 'uninformative prior' : used to remove singularities and avoid spurious estimates => akin to regularization
- Interesting prior form : 'conjugate priors'

  definition: if the posterior is of the same parametric form as the prior, then we call the prior the conjugate distribution for the likelihood distribution

$$p(\theta|\alpha) \Rightarrow p(\theta|D,\alpha) = p(\theta|\alpha')$$

same parametric form
(but not the same parameters !)

Interest: we have an analytical solution, so no need for optimisation e.g. to find the max (= mode of the distribution)

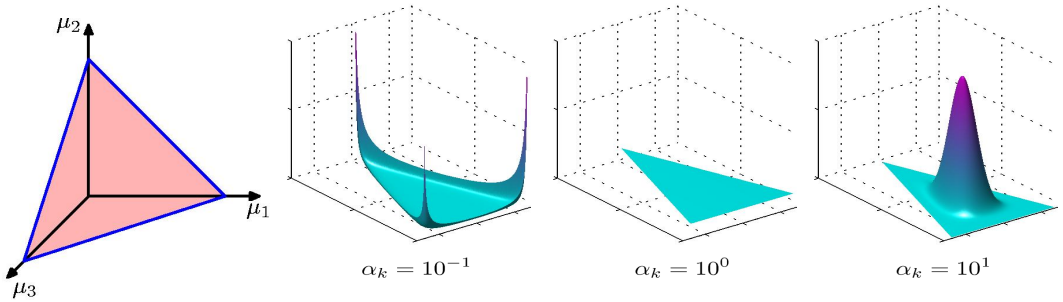


Example : categorical distribution for the data (4)

- Likelihood $\quad p(D|\mu) = \prod_k \mu_k^{N_k(D)}$
- Conjugate prior : similar expression w.r.t. parameters

  => Dirichlet distribution (defined over simplex)

$$\mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1} \qquad \alpha_0 = \sum_{k=1}^{K} \alpha_k$$



example of distribution depending on equal α_k values

Example : categorical distribution for the data (5)

- Likelihood $p(D|\mu) = \prod_k \mu_k^{N_k(D)}$
- Prior $\mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}$
- Posterior: Dirichlet distribution with updated parameters

$$p(\boldsymbol{\mu}|D,\alpha) \propto p(D|\boldsymbol{\mu})p(\boldsymbol{\mu}|\alpha) \propto \prod_k \mu_k^{N_k(D)+\alpha_k-1}$$

$$\Rightarrow p(\boldsymbol{\mu}|D,\alpha) = \mathrm{Dir}(\boldsymbol{\mu}|\alpha') \qquad \alpha'_k = \alpha_k + N_k(D)$$

$$\mu_k^{MAP} = \frac{\alpha'_k - 1}{\sum_{k'} \alpha'_k - 1} = \frac{\alpha_k + N_k(D) - 1}{\sum_{k'} (\alpha_{k'} + N_{k'}(d)) - K}$$

α_k-1 can be interpreted as a count of (virtual) observations of the class k
- can favor one class against another if have prior information
- the larger the values, the more important the prior is against real observations

Example : categorical distribution for the data (3)

- Throw a dice N = 10 times
  => Data (K=6) = {1,1,6,2,5,3,1,6,2,1 } (not in 1-K encoding format)
- Dirichlet prior with $\alpha_k = 2$

k	1	2	3	4	5	6
$N_k(D)$	4	2	1	0	1	2
μ_k^{ML}	0.4	0.2	0.1	0	0.1	0.2
μ_k^{MAP}	0.312	0.187	0.125	0.062	0.125	0.187

- 6 'virtual' observations vs 10 real observations: prior counts for 6/16 approx. 1/3 in the estimation of the parameters
- p(x=4) was 0 with ML => MAP allows to account for the potentially small sample size and gives a low probability

Example 2 : 1D - Gaussian distribution (1) - σ known

- Likelihood $p(x|\mu) = \mathcal{N}(x|\mu, \sigma)$
  - μ to be estimated
  - σ assumed to be known

$$p(\mathbf{x}|\mu) = \prod_{n=1}^{N} p(x_n|\mu) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2\right\}.$$

  - Note: it has a gaussian shape in function of μ (but is not a distribution over μ)
- Conjugate Prior $p(\mu) = \mathcal{N}\left(\mu|\mu_0, \sigma_0^2\right)$.
- Posterior: Gaussian with updated parameters

$$p(\mu|\mathbf{x}) \propto p(\mathbf{x}|\mu)p(\mu). \qquad p(\mu|\mathbf{x}) = \mathcal{N}\left(\mu|\mu_N, \sigma_N^2\right)$$

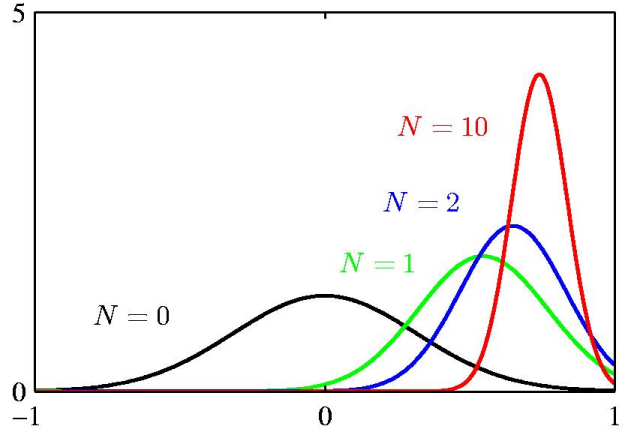
$$\mu_N = \frac{\sigma^2}{N\sigma_0^2+\sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2+\sigma^2}\mu_{\mathrm{ML}}, \qquad \mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}.$$

Example 2 : 1D - Gaussian distribution (2) - σ known

- Example of posterior with N=0,1,2,10

$$p(\mu|\mathbf{x}) = \mathcal{N}\left(\mu|\mu_N, \sigma_N^2\right)$$



- Note

	$N=0$	$N \to \infty$
μ_N	μ_0	μ_{ML}
σ_N^2	σ_0^2	0

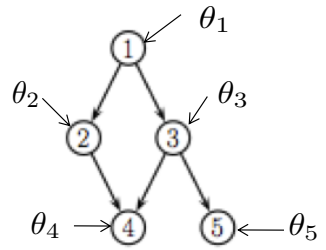
  - as N goes to infinity, we are more and more confident in our estimate of the mean (which tends to the maximum likelihood estimate)

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2+\sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2+\sigma^2}\mu_{\mathrm{ML}}, \qquad \mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}.$$

learning in a directed graphical model (DGM)

learning in DGM with fully observed data



- if for all data samples, all variables of the model are fully observed, there is no missing data, and there are no hidden variables, we say that the **data is complete**
- the likelihood is then given by

$$p(D|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}_i|\boldsymbol{\theta}) = \prod_{i=1}^{N}\prod_{t=1}^{V} p(x_{it}|\mathbf{x}_{i,pa(t)}, \boldsymbol{\theta}_t) = \prod_{t=1}^{V} p(D_t|\boldsymbol{\theta}_t)$$

where D_t is the data associated with node t and its parents

- product of terms, one per Conditional Probability Distribution (CPD)
- parameters of each CPD can be optimized separately, e.g. using ML estimation

MAP estimation in DGM – fully observed data

- The likelihood factorizes $p(D|\boldsymbol{\theta}) = \prod_{t=1}^{V} p(D_t|\boldsymbol{\theta}_t)$
- Assume the prior factorizes as well $p(\boldsymbol{\theta}) = \prod_{t=1}^{V} p(\boldsymbol{\theta}_t)$
- Then clearly the posterior factorizes

$$p(\boldsymbol{\theta}|D) \propto p(D|\boldsymbol{\theta})p(\boldsymbol{\theta}) = \prod_{t=1}^{V} p(D_t|\boldsymbol{\theta}_t)p(\boldsymbol{\theta}_t)$$

=> We can compute the posterior of each CPD independently
=> we can perform MAP estimation on each independently

- However, often, observations of a DGM rely on hidden variables
  - allows to represent complicated distribution from simple components
  - allows to do soft clustering of the data
  - e.g. Gaussian Mixture Model
  - need techniques to handle this (EM algorithm)

learning with latent variables

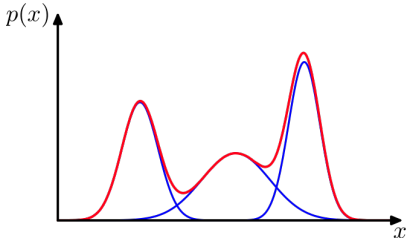
the expectation-maximization (EM) algorithm

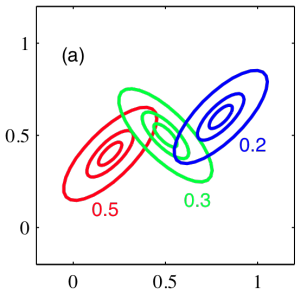
the Gaussian Mixture Model (GMM)

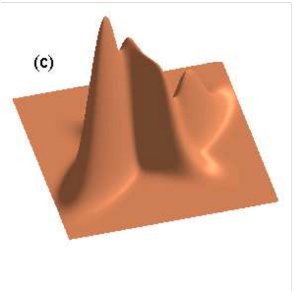
- Definition: a GMM for a multivariate r.v. **x** is defined as

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad 0 \leq \pi_k \leq 1 \qquad \sum_{k=1}^{K} \pi_k = 1$$

the π_k are called the mixing coefficients







$p(\mathbf{x}) =$

GMM as a graphical model

- assumes a hidden K-dimensional variable **categorical variable z** used to select from which Gaussian a given observation is drawn

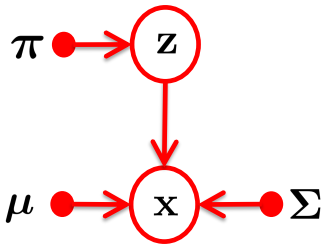
$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)^{z_k}$$

$$\mathbf{z} = (z_1, ..., z_k)$$

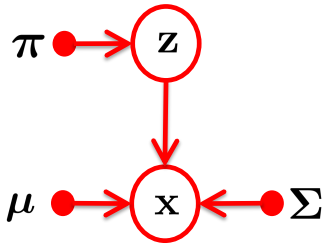
$$\sum_k z_k = 1, p(z_k = 1) = \pi_k$$



set of parameters

$$\boldsymbol{\theta} = \{\boldsymbol{\pi}, (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)_{k=1..K}\}$$

GMM as a graphical model (2)



- prior probability of $\mathbf{z}$:

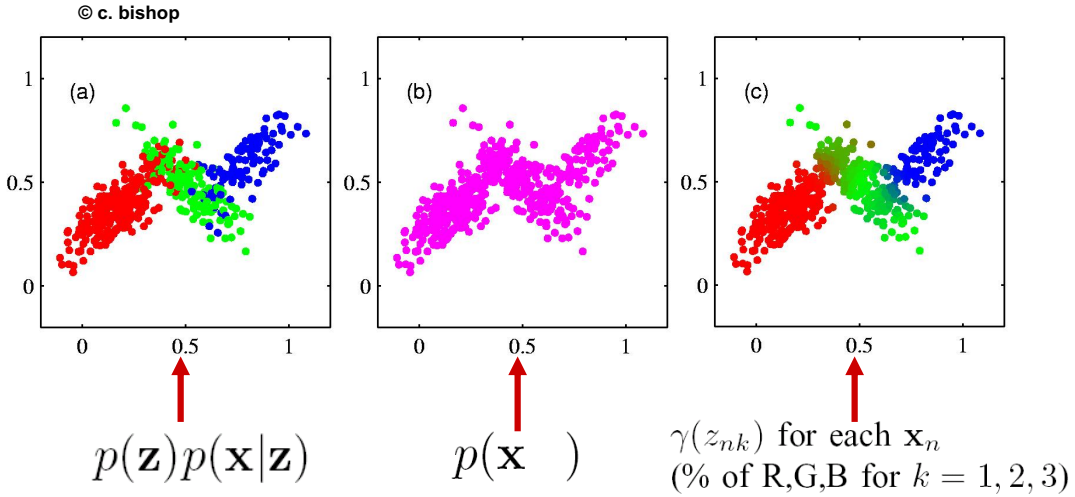
$$p(z_k = 1) = \pi_k$$

- posterior probability of $\mathbf{z}$ given $\mathbf{x}$:

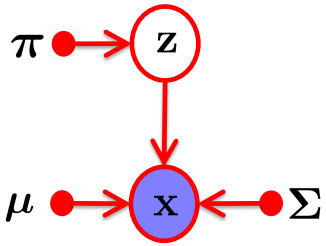
$$\gamma(z_k) = p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} = \frac{\pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\mu_j, \Sigma_j)}$$

- $\gamma(z_k)$ can also be seen as the *responsibility* of component k for explaining the observation $\mathbf{x}$.

GMM: an example



Learning the GMM parameters ?



set of parameters

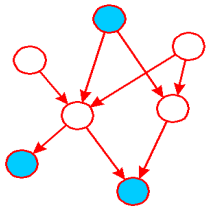
$$\theta = \{\pi, (\mu_k, \Sigma_k)_{k=1..K}\}$$

- we are given the values of X
- the z variables are not given
  - they help in obtaining a better model, e.g. in the GMM having a better model of distributions on x
- learning ?
- In the following, we describe a general procedure (EM), valid for any model. We will present its general treatment (bearing in mind the GMM case), and come back to the EM equation derivation for the GMM case afterwards

the expectation-maximization (EM) algorithm

the EM algorithm: learning with latent variables

assume an arbitrary graphical model



hidden z_i

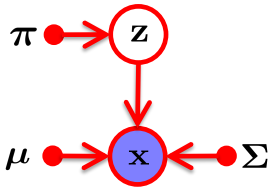
visible x_i

- **goal**: find ML solutions for latent variable models
- $\mathbf{X}$: set of all observed data variables $\mathbf{X} = \{\mathbf{x}_i, i = 1 \ldots N\}$
- $\mathbf{Z}$: set of all latent variables. $\mathbf{Z} = \{\mathbf{z}_i, i = 1 \ldots N\}$
- the log-likelihood is given by

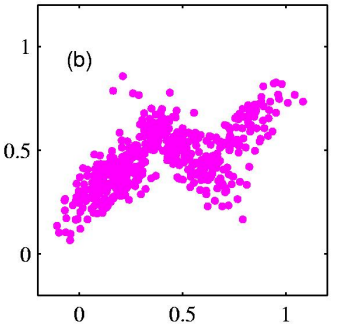
$$\ln L(\theta|\mathbf{X}) = \ln p(\mathbf{X}|\theta) = \ln\{\sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta)\}$$

- **problem**: the summation is inside the log function
- $\ln(\cdot)$ cannot act easily on the joint distribution!

EM: incomplete vs complete data (2)

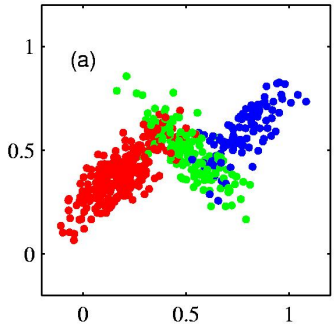


- Idea: suppose we know Z
  - X: incomplete data (previous slide)
  - X,Z : complete data
- Illustration: GMM case



incomplete data

X: 2-D observations



complete data

X: 2-D observations

Z: class labels (each point : red,green or blue)

EM algorithm : intuition

- Complete data likelihood $\ln \mathrm{L}_C(\boldsymbol{\theta}|\mathbf{X},\mathbf{Z}) = \ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})$
  - easier to maximize to obtain ML estimate $\boldsymbol{\theta}_{ML}$
  - however, we need to know **Z** ....

- If we new $\boldsymbol{\theta}_{ML}$ (but not **Z**)
  - we could estimate the posterior of Z $p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta})$
  - i.e. how probable are each values of Z for each observation

  => this would leave us with some kind of (weighted) complete dataset

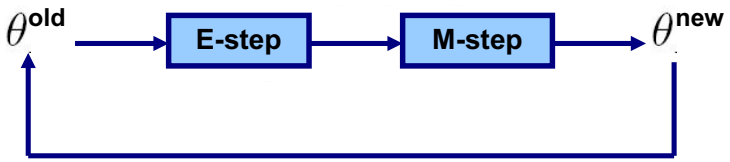
$$(\mathbf{x}_i,\mathbf{z}) \text{ with weight } p(\mathbf{z}|\mathbf{x}_i,\boldsymbol{\theta}) \qquad \forall i, \forall \mathbf{z}$$

EM : an iterative process

- Goal: estimate

$$\theta_{ML} = \arg\max_{\theta} L(\theta|D)$$

- Two steps
  - E(xpectation)
  - M(aximization)



EM : the E step

- the complete log-likelihood is not available
- its expected value under the posterior pdf of the latent variable is used

- More concretely
  - the current parameter estimate $\boldsymbol{\theta}^{old}$ is used to find the posterior

$$p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old})$$

  - the posterior is then used to find the expectation

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) &= E_{Z|X,\boldsymbol{\theta}^{old}}\left(\ln L_C(\boldsymbol{\theta}|X, Z)\right) \\ &= E_{Z|X,\boldsymbol{\theta}^{old}}\left(\ln p(X, Z|\boldsymbol{\theta})\right) \\ &= \sum_Z p(Z|X, \boldsymbol{\theta}^{old}) \ln p(X, Z|\boldsymbol{\theta})\end{aligned}$$

EM : the M step

- E-step $\quad \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_Z p(Z|X, \boldsymbol{\theta}^{old}) \ln p(X, Z|\boldsymbol{\theta})$

- M-step
  - maximize the **expected value Q**
  - update the parameters

$$\boldsymbol{\theta}^{new} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$$

  - the log function acts directly on the (factorized) joint distribution so the maximum will be tractable

the EM algorithm

- given $p(\mathbf{X}, \mathbf{Z}|\theta)$, maximize $p(\mathbf{X}\ |\theta)$ w.r.t. θ
  1) choose an initial value for the parameters θ^{old}.
  2) **E-step**: evaluate $p(\mathbf{Z}|\mathbf{X}, \theta^{old})$.
  3) **M-step**: evaluate θ^{new} given by

$$\theta^{new} = argmax_\theta \ \ Q(\theta, \theta^{old})$$

where

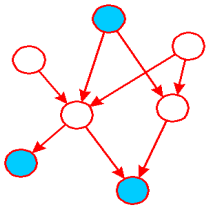
$$Q(\theta, \theta^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{old}) \ln p(\mathbf{X}, \mathbf{Z}|\theta).$$

  4) apply convergence criterion (on the log-likelihood or on the parameters). If criterion is not satisfied,

$$\theta^{old} \leftarrow \theta^{new}$$

and return to step 2.

inference, revisited



hidden z_i

visible x_i

- inference :
  - given values for the observed nodes
  - compute the posterior distribution on a subset of the hidden variables

$$p(\mathbf{Z}|\mathbf{X}, \theta)$$

- inference = the E-step
- consequence: learning (using EM) requires inference

why does EM work?
a closer look (Neal and Hinton, 1999)

- assume that direct optimization of $p(\mathbf{X}|\theta)$ is hard, but optimization of $p(\mathbf{X}, \mathbf{Z}|\theta)$ is much easier.
- for any distribution $q(\mathbf{Z})$ over latent variables

approximating distribution

$$\ln p(\mathbf{X}|\theta) = \mathcal{L}(q, \theta) + KL(q||p)$$

$$\mathcal{L}(q, \theta) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})}$$

complete likelihood

$$KL(q||p) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{Z}|\mathbf{X}, \theta)}{q(\mathbf{Z})}$$

posterior over latent variables

Kullback-Leibler divergence

EM: a closer look (2)

Proof.

$$\ln p(\mathbf{X}, \mathbf{Z}|\theta) = \ln p(\mathbf{Z}|\mathbf{X}, \theta) + \ln p(\mathbf{X}|\theta) \rightarrow$$

$$\begin{aligned}
\mathcal{L}(q, \theta) &= \sum_{\mathbf{Z}} q(\mathbf{Z})(\ln p(\mathbf{Z}|\mathbf{X}, \theta) + \ln p(\mathbf{X}|\theta) - \ln q(\mathbf{Z})) \\
&= \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{Z}|\mathbf{X}, \theta)}{q(\mathbf{Z})} + \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln p(\mathbf{X}|\theta) \\
&= -KL(q(\mathbf{Z})||p(\mathbf{Z})) + \ln p(\mathbf{X}|\theta) \rightarrow
\end{aligned}$$

$$\ln p(\mathbf{X}|\theta) = \mathcal{L}(q, \theta) + KL(q||p)$$

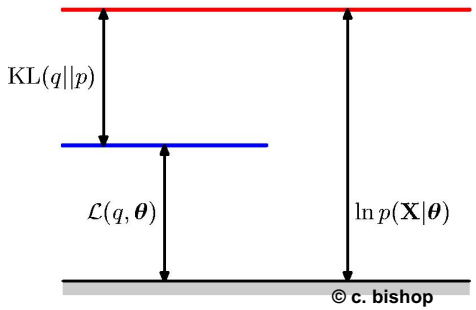
a closer look at EM (3)

$$\ln p(\mathbf{X}|\theta) = \mathcal{L}(q,\theta) + KL(q||p)$$

$$KL(q||p) \geq 0$$

$$KL(q||p) = 0 \iff q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X},\theta)$$

$$\mathcal{L}(q,\theta) \leq \ln p(\mathbf{X}|\theta)$$

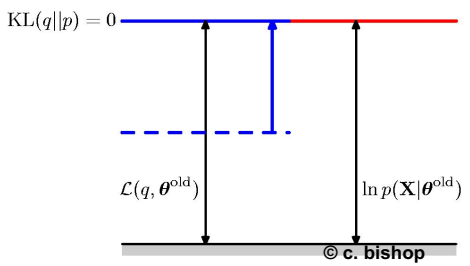


$\mathcal{L}(q,\theta)$ is a lower bound on $\ln p(\mathbf{X}|\theta)$.

EM is a coordinate ascent algorithm on $\mathcal{L}(q,\theta)$:

- **E-step**: $q^{new} = \arg\max_q \mathcal{L}(q,\theta^{old})$
- **M-step**: $\theta^{new} = \arg\max_\theta \mathcal{L}(q^{new},\theta)$

the E-step, revisited

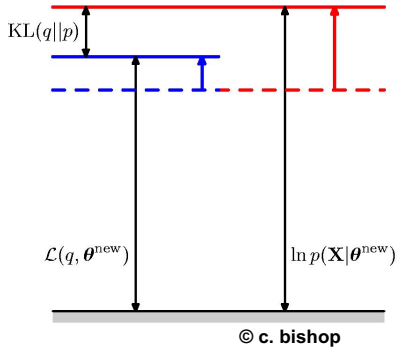


$$\ln p(\mathbf{X}|\theta) = \mathcal{L}(q,\theta) + KL(q||p)$$

- given θ^{old}, **maximize** $\mathcal{L}(q,\theta^{old})$ w.r.t. $q(\mathbf{Z})$.
- the largest $\mathcal{L}(q,\theta^{old})$ occurs when $KL(q||p) = 0$, i.e., when $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X},\theta^{old})$, the posterior distribution over latent variables.
- in this case, the lower bound equals the log-likelihood, $\mathcal{L}(q,\theta^{old}) = \ln p(\mathbf{X}|\theta^{old})$.

the M-step, revisited

$$\ln p(\mathbf{X}|\theta) = \mathcal{L}(q,\theta) + KL(q||p)$$



- given $q^{new} = p(\mathbf{Z}|\mathbf{X}, \theta^{old})$, **maximize** $\mathcal{L}(q^{new}, \theta)$ w.r.t. $\theta \rightarrow \theta^{new}$.
- $\mathcal{L}(q^{new}, \theta^{new})$ increases unless it is at a maximum
- $KL(q^{new}||p(\mathbf{Z}|\mathbf{X}, \theta^{new}))$ will be nonzero because q^{new} was computed using the old parameters
- the increase in $\ln p(\mathbf{X}|\theta^{new})$ is greater than the increase in the lower bound.

the M-step, revisited (2)

- after the *E-step*, inserting $q(\mathbf{Z}) = \mathbf{p}(\mathbf{Z}|\mathbf{X}, \theta^{\mathbf{old}})$ in the definition of $\mathcal{L}(q^{new}, \theta)$ gives

$$\begin{aligned}
\mathcal{L}(q,\theta) &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X},\mathbf{Z}|\theta)}{q(\mathbf{Z})} \\
&= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\theta^{old}) \ln \frac{p(\mathbf{X},\mathbf{Z}|\theta)}{p(\mathbf{Z}|\mathbf{X},\theta^{old})} \\
&= Q(\theta,\theta^{old}) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\theta^{old}) \ln p(\mathbf{Z}|\mathbf{X},\theta^{old}) \\
&= Q(\theta,\theta^{old}) + H(q)
\end{aligned}$$

- therefore, in the *M-step*, the quantity that is being maximized is the expected value of the complete log-likelihood $Q(\theta, \theta^{old})$.

MAP estimation with EM ?

- We want to optimize $\ln p(\boldsymbol{\theta}|X)$

$$\ln p(\boldsymbol{\theta}|X) = \ln p(X|\boldsymbol{\theta}) + \ln p(\boldsymbol{\theta}) - \ln p(X)$$
$$\ln p(\boldsymbol{\theta}|X) = \mathcal{L}(q, \boldsymbol{\theta}) + KL(q||p) + \ln p(\boldsymbol{\theta}) - \ln p(X)$$
$$\geq \mathcal{L}(q, \boldsymbol{\theta}) + \ln p(\boldsymbol{\theta}) - \ln p(X)$$

constant

- We can apply the same steps than in EM, maximizing the right hand side
  - E-step: given $\boldsymbol{\theta}^{old}$ find q the r-h.s => same E-step as in standard EM
  - M-step: maximize

$$\boldsymbol{\theta}^{new} = \arg\max_{\boldsymbol{\theta}} \left(\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) + \ln p(\boldsymbol{\theta}) \right)$$

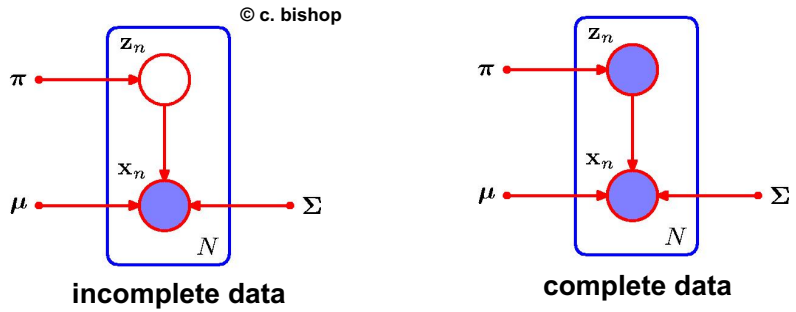
=> if factorized prior => MAP estimation for each CPD distribution in general

Limitations of EM

- EM divides a difficult problem into two steps that *might* be simpler to implement
- E-step and M-step might be intractable
  - **intractable M-step** (generalized EM): instead of maximizing $\mathcal{L}(q, \theta)$ w.r.t. θ , modify θ to increase its value (with nonlinear optimization methods)
  - **intractable E-step**: perform a partial, rather than full, optimization of $\mathcal{L}(q, \theta)$ w.r.t. $q(\mathbf{Z})$
- EM depends on the initialization values (e.g. see GMMs)
- EM can get trapped on local maxima

EM for Gaussian mixture models

EM for GMM



- observed variables: $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)$, hidden variables: $\mathbf{Z} = (\mathbf{z}_1, ..., \mathbf{z}_N)$, parameters $\theta = (\mu, \Sigma, \pi)$.
- complete likelihood

$$p(\mathbf{X}, \mathbf{Z}|\mu, \Sigma, \pi) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)^{z_{nk}}$$

EM for GMM

- recall that $Q(\theta, \theta^{old}) = E_{\mathbf{Z}|\mathbf{X},\theta^{old}} \{\ln p(\mathbf{X}, \mathbf{Z}|\theta)\}$, so

$$\begin{aligned} Q(\theta, \theta^{old}) &= E\left\{\ln\left\{\prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}}\mathcal{N}(\mathbf{x}_n|\mu_k,\Sigma_k)^{z_{nk}}\right\}\right\} \\ &= E\left\{\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\left\{\ln\pi_k + \ln\mathcal{N}(\mathbf{x}_n|\mu_k,\Sigma_k)\right\}\right\} \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} E(z_{nk})\left\{\ln\pi_k + \ln\mathcal{N}(\mathbf{x}_n|\mu_k,\Sigma_k)\right\} \end{aligned}$$

EM for GMM (2)

- where

$$\begin{aligned} E_{\mathbf{Z}|\mathbf{X},\theta^{old}}(z_{nk}) &= \sum_{\mathbf{Z}} z_{nk} p(\mathbf{Z}|\mathbf{X},\theta^{old}) \\ &= \sum_{0,1} z_{nk} p(z_{nk}|\mathbf{x}_n,\theta^{old}) \\ &= 1\cdot p(z_{nk}=1|\mathbf{x}_n,\theta^{old}) \\ &+ 0\cdot p(z_{nk}=0|\mathbf{x}_n,\theta^{old}) \\ &= p(z_{nk}=1|\mathbf{x}_n,\theta^{old}) \\ &= \gamma^{old}(z_{nk}) \end{aligned}$$

  which is the E-step.
- the M-step finds the parameters θ that maximizes $Q(\theta, \theta^{old})$, searching for

$$\frac{\partial Q}{\partial \theta} = 0$$

  for each of μ_k, Σ_k, π_k, with $\sum \pi_k = 1$.

EM for GMM (3)

$$Q(\theta, \theta^{old}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\left\{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)\right\}$$

$$\begin{aligned}\frac{\partial Q}{\partial \mu_k} &= \sum_{n=1}^{N}\gamma(z_{nk})\left\{\frac{\partial}{\partial \mu_k}(\ln \pi_k) + \frac{\partial}{\partial \mu_k}(\ln \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k))\right\} \\ &= \sum_{n=1}^{N}\gamma(z_{nk})\left\{\frac{\partial}{\partial \mu_k}(\ln C - \frac{1}{2}(\mathbf{x}_n - \mu_k)^T\Sigma_k^{-1}(\mathbf{x}_n - \mu_k))\right\} \\ &= \sum_{n=1}^{N}\gamma(z_{nk})\left\{-\Sigma_k^{-1}(\mathbf{x}_n - \mu_k)\right\} = 0\end{aligned}$$

$$\frac{\partial \mathbf{x}^T A\mathbf{x}}{\partial \mathbf{x}} = (A + A^T)\mathbf{x} = 2A\mathbf{x}$$
if A is symmetric.

Pre-multiplying both sides by Σ_k,

$$\sum_{n=1}^{N}\gamma(z_{nk})(\mathbf{x}_n - \mu_k) = 0$$

and rearranging terms,

$$\mu_k = \frac{\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n}{\sum_{n=1}^{N}\gamma(z_{nk})}$$

EM for GMM (4)

- Note the following

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\ln \pi_k + \sum_{k=1}^{K}\left(\sum_{n=1}^{N}\gamma(z_{nk})\ln \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)\right)$$

weighted log-likelihood for the categorical distribution

weighted loglikelihood for the component k

=> M optimization can be seen as standard ML estimation where the weight express the 'number of times' an observation should be counted

recipe: EM for GMMs

1) given $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)$ and K, initialize parameters: $(\mu, \Sigma, \pi) = (\mu_{1:K}, \Sigma_{1:K}, \pi_{1:K})$, and evaluate the initial value of the log-likelihood.
2) **E-step**: compute posterior of hidden variables using current parameters

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \Sigma_j)}$$

recipe: EM for GMMs (2)

3) **M-step**: re-estimate parameters using the posteriors

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \mu_k^{new})(\mathbf{x}_n - \mu_k^{new})^T$$

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

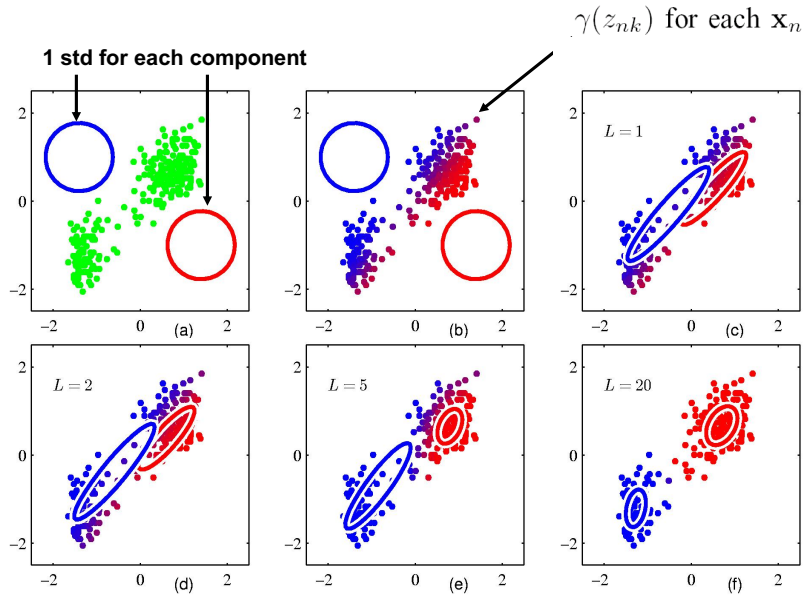
where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$

4) compute the log-likelihood

$$\ln p(\mathbf{X}|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)$$

and verify convergence of parameters or log-likelihood.
if convergence not satisfied return to step 2.

EM for GMM in action



K=2

L = EM iteration

Relation with Kmeans

$$J(\boldsymbol{\mu}, (r_{nk})) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

- K-means optimizes
  - $(\boldsymbol{\mu}_k)_{k=1,\ldots,N}$ cluster means
  - $r_{nk} = 1$ means that data point n is assigned to cluster k
  - J measures the distorsion of representing each x_i with its assigned cluster mean
- Two steps
  - E-step : compute best assignment of observation x_i to cluster k

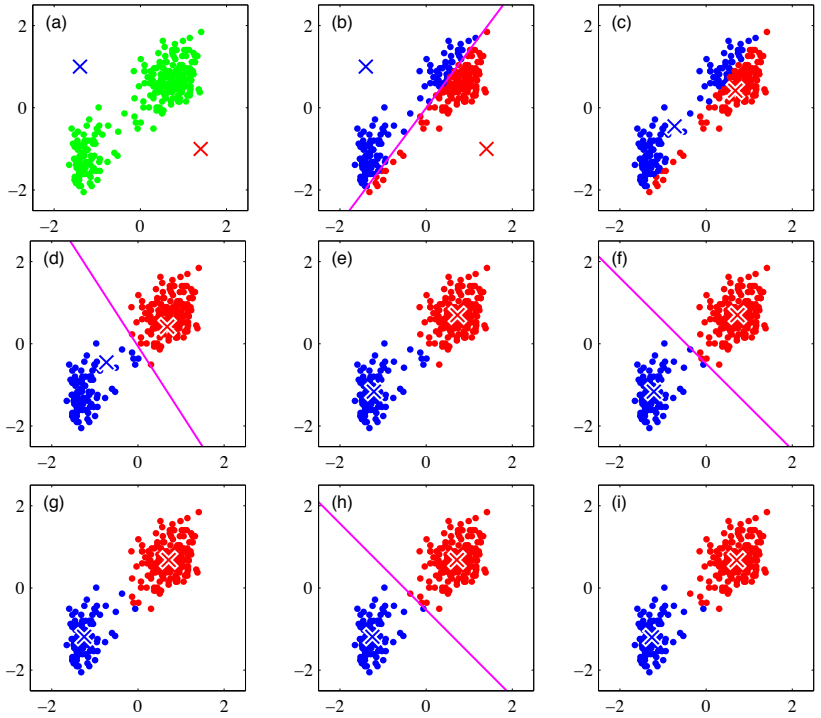
$$r_{ik} = \begin{cases} 1 & \text{if } \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 < \|\mathbf{x}_i - \boldsymbol{\mu}_{k'}\|^2 \qquad \forall k' \neq k \\ 0 & \text{otherwise} \end{cases}$$

  - M-step . given assignment, compute optimal means

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} r_{ik} \mathbf{x}_i}{\sum_i r_{ik}}$$

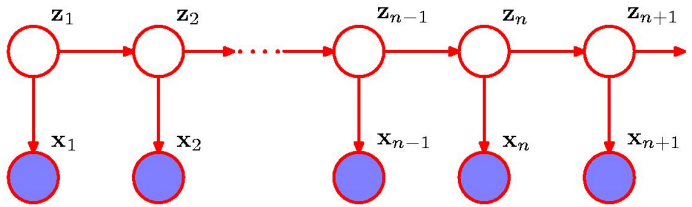
- difference with EM for GMM
  - hard assignment vs soft assignment (responsibilities in GMM)
  - covariance assumed to be diagonal, with (infinite) fixed variance
  - no mixture weights

Kmeans - illustration



hidden Markov model (HMM)

Hidden Markov model (HMM)



© c. bishop

- **dynamic Bayes net**, a.k.a. **state-space** model
- $\mathbf{x}$: observed continuous or discrete variables
- $\mathbf{z}$: latent **discrete** variables

Dynamic-model: state transitions

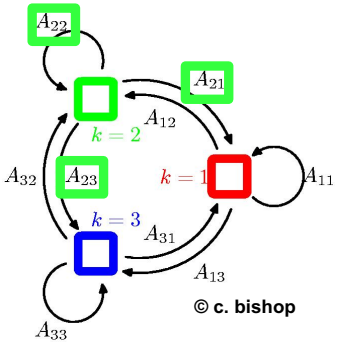
Emission model: generating the observations

$$p(\mathbf{x}_{1:N}, \mathbf{z}_{1:N}) = p(\mathbf{z}_1) \prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}) \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n)$$

HMM (2)

z can take K=3 states

caution: this is a state-transition diagram, NOT a graphical model



© c. bishop

- **transition probability matrix**

$$A_{jk} = p(z_{nk} = 1|z_{n-1,j} = 1); \quad 0 \leq A_{jk} \leq 1, \quad \sum_k A_{jk} = 1$$

$$p(\mathbf{z}_n|\mathbf{z}_{n-1}, A) = \prod_{k=1}^{K} \prod_{j=1}^{K} A_{jk}^{z_{n-1,j} z_{nk}}$$

1-K encoding trick
Note: depends on two values

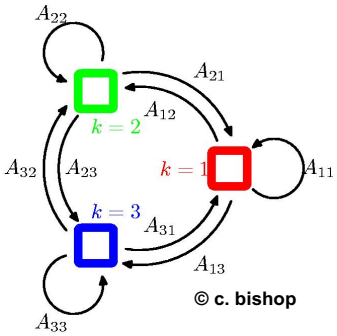
$$p(\mathbf{z}_1|\pi) = \prod_{k=1}^{K} \pi_k^{z_{1k}}, \qquad \sum_k \pi_k = 1$$

HMM (3) - examples

- Example 1
  - no state favored, self-transition high

  => favors sequences of similar states (smoothness)

state from\to	1	2	3
1	0.8	0.1	0.1
2	0.1	0.8	0.1
3	0.1	0.1	0.8



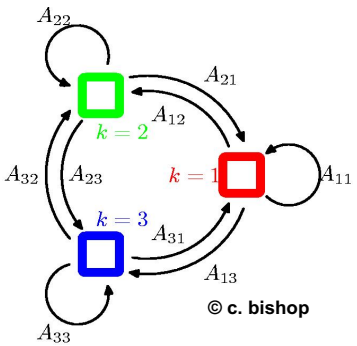
- Example 2 – left-right HMM
  - state diagram ?

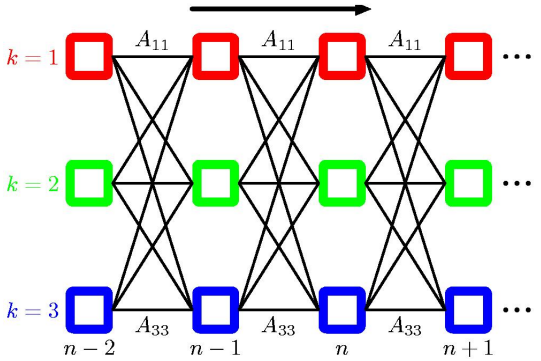
state from\to	1	2	3
1	0.6	0.3	0.1
2	0	0.7	0.3
3	0	0	1.0

- Example 3 – bouncing ball – 2 states
  - state 1 : ballistic trajectory
  - state 2 : hitting the ground – assumed to last only one time step

  transition matrix ?

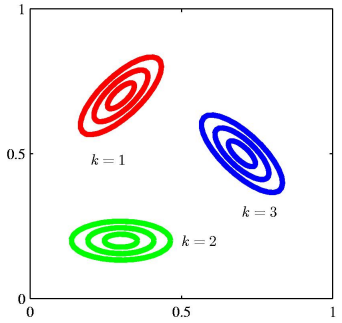
HMM (4)





the state transition diagram can be unfolded over time to show the changes between variables into a **lattice diagram**

HMM (5)



K=3

2-D Gaussian observations

© c. bishop

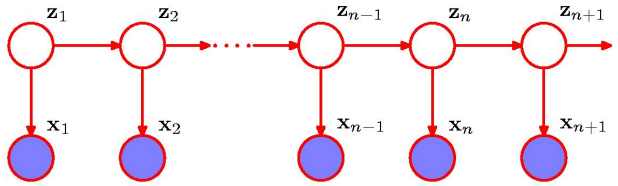
- **emission distribution (observation model)**

$$p(\mathbf{x}_n|\mathbf{z}_n,\phi) = \prod_{k=1}^{K} p(\mathbf{x}_n|\phi_k)^{z_{nk}}$$

where $\phi = \{\phi_k\}$ represent the parameters for the distribution

- emission distributions are conditional probability tables if $\mathbf{x}$ is discrete, Gaussian **or** GMMs if $\mathbf{x}$ is continuous.

homogeneous HMM



© c. bishop

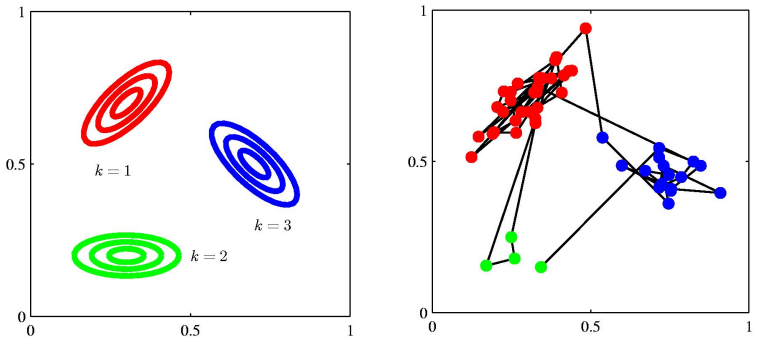
- **homogeneous model**:
  - all the conditional distributions for the latent variables share the same parameters A
  - all the emission distributions share the same parameters ϕ.
- defining $\mathbf{X} = \mathbf{x}_{1:N}$, $\mathbf{Z} = \mathbf{z}_{1:N}$, and $\theta = \{\pi, A, \phi\}$, the joint distribution can now be written as

$$p(\mathbf{X},\mathbf{Z}|\theta) = p(\mathbf{z}_1|\pi)\prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1},A)\prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n,\phi)$$

the HMM as a generative model

K=3

2-D Gaussian observations

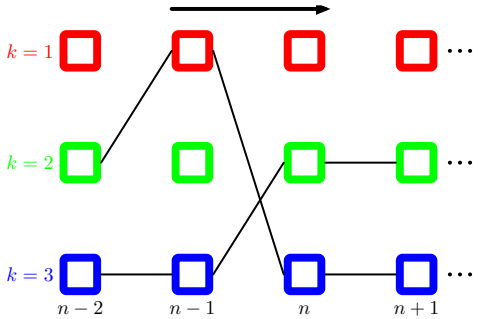


© c. bishop

ancestral sampling:

- sample $\mathbf{z}_1 \sim p(\mathbf{z}_1|\pi)$.
- sample $\mathbf{x}_1 \sim p(\mathbf{x}_1|\mathbf{z}_1, \phi)$.
- for $n = 2 : N$,
  - sample $\mathbf{z}_n \sim p(\mathbf{z}_n|\mathbf{z}_{n-1}, A)$.
  - sample $\mathbf{x}_n \sim p(\mathbf{x}_n|\mathbf{z}_n, \phi)$.

HMM "decoding" Viterbi algorithm (1)

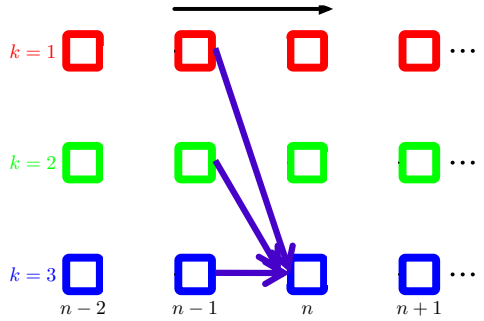


- Goal: find the most probable sequence of states that can explain the observed values

$$\arg\max_{\mathbf{z}_{1:N}} p(\mathbf{z}_{1:N}|\mathbf{x}_{1:N}) = \arg\max_{\mathbf{z}_{1:N}} p(\mathbf{z}_{1:N}, \mathbf{x}_{1:N})$$
$$\Rightarrow \arg\max_{\mathbf{z}_{1:N}} \log p(\mathbf{z}_{1:N}, \mathbf{x}_{1:N})$$

  - the number of possible paths through the lattice grows exponentially with the length of the chain
  - Viterbi algorithm:
    - efficient way to solve this problem
    - specific case of the max-sum algorithm on trees (cf last course)
    - related to/probabilistic version of: Dynamic Time Warping

HMM "decoding" Viterbi algorithm (2)



$$p(\mathbf{z}_{1:n}, \mathbf{x}_{1:n}) = p(\mathbf{z}_1) \prod_{i=2}^{n} p(\mathbf{z}_i|\mathbf{z}_{i-1}) p(\mathbf{x}_i|\mathbf{z}_i) = p(\mathbf{z}_{1:n-1}, \mathbf{x}_{1:n-1}) p(\mathbf{z}_n|\mathbf{z}_{n-1}) p(\mathbf{x}_n|\mathbf{z}_{nxt})$$

- Principle: recursively find the best path that ends up in a given state
  => define $S_n(k)$ the maximum score of the path that ends in state k at time n

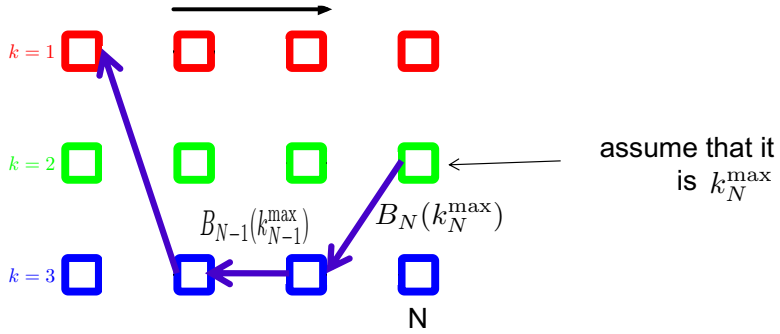
$$S_n(k) = \max_{\mathbf{z}_{1:n-1}, \mathbf{z}_n = k} \log p(\mathbf{z}_{1:n}, \mathbf{x}_{1:n})$$

- Assume that we know the best path up to time n-1, i.e. $S_{n-1}(k)$. Then

$$S_n(k) = \max_l \left(S_{n-1}(l) + \log(p(\mathbf{z}_n = k|p(\mathbf{z}_{n-1} = l) p(\mathbf{x}_n|\mathbf{z}_n = k)) \right)$$

  and keep in memory $B_n(k)$ the state l at time n-1 for which the max was achieved

HMM "decoding" Viterbi algorithm (3)



- Reaching the end of the sequence (time N) – Best score :

$$S_N(k) = \max_{\mathbf{z}_{1:N-1}, \mathbf{z}_N = k} \log p(\mathbf{z}_{1:N}, \mathbf{x}_{1:N})$$

$$\Rightarrow \max_{\mathbf{z}_{1:N}} \log p(\mathbf{z}_{1:N}, \mathbf{x}_{1:N}) = \max_k S_N(k)$$

- State of best path at time N = state k for which the max $S_N(k)$ is achieved

$$k_N^{\max} = \arg\max_k S_N(k)$$

- We can backtrack the best path using the stored $B_n(k)$

$$k_{n-1}^{\max} = B_n(k_n^{\max})$$

HMM learning: EM (1)

- the complete log-likelihood for the HMM is

$$\begin{aligned}
\ln p(\mathbf{X}, \mathbf{Z}|\theta) &= \ln \left\{ p(\mathbf{z}_1|\pi) \prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}, A) \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n, \phi) \right\} \\
&= \ln \left\{ \prod_{k=1}^{K} \pi_k^{z_{1k}} \right\} + \ln \left\{ \prod_{n=2}^{N} \prod_{k=1}^{K} \prod_{j=1}^{K} A_{jk}^{z_{n-1,j} z_{nk}} \right\} \\
&+ \ln \left\{ \prod_{n=1}^{N} \prod_{k=1}^{K} p(\mathbf{x}_n|\phi_k)^{z_{nk}} \right\} \\
&= \sum_{k=1}^{K} z_{1k} \ln \pi_k + \sum_{n=2}^{N} \sum_{k=1}^{K} \sum_{j=1}^{K} z_{n-1,j} z_{nk} \ln A_{jk} \\
&+ \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \ln p(\mathbf{x}_n|\phi_k)
\end{aligned}$$

HMM learning: EM (2)

- denote the marginal posterior of a latent variable $\mathbf{z}_n$ as

$$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{X}, \theta^{old})$$

- and the conditional probability $p(z_{nk} = 1|\mathbf{X}, \theta^{old})$ as

$$\gamma(z_{nk}) = E_{\mathbf{z}_n|\mathbf{X},\theta^{old}}(z_{nk}) = \sum_{\mathbf{z}} \gamma(\mathbf{z}_n) z_{nk}$$

- similarly, denote the joint posterior distribution of two successive latent variables $\mathbf{z}_{n-1}, \mathbf{z}_n$ as

$$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X}, \theta^{old})$$

- and similarly

$$\xi(z_{n-1,j}, z_{nk}) = E(z_{n-1,j} z_{nk})$$

HMM learning: EM (3)

- substituting the definitions for $\ln p(\mathbf{X}, \mathbf{Z}|\theta)$, γ and ξ in the definition of $Q(\theta, \theta^{old})$,

$$\begin{aligned} Q(\theta, \theta^{old}) &= \sum_{k=1}^{K} \gamma(z_{1k}) \ln \pi_k \\ &+ \sum_{n=2}^{N} \sum_{k=1}^{K} \sum_{j=1}^{K} \xi(z_{n-1,j} z_{nk}) \ln A_{jk} \\ &+ \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \ln p(\mathbf{x}_n|\phi_k) \end{aligned}$$

- **E-step**: computing $\gamma(z_{nk})$ and $\xi(z_{n-1,j} z_{nk})$ terms efficiently $\rightarrow$ *forward-backward* algorithm $\rightarrow$ *alpha-beta* recursion.
- **M-step**: maximize $Q(\theta, \theta^{old})$ w.r.t. $\theta = \{\pi, A, \phi\}$, treating γ and ξ as constant.

EM equations for HMM with Gaussian emission distributions

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})}$$

$$A_{jk} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nl})}$$

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

identical to GMM

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

what to remember

- EM algorithm
  - general method for learning in graphical models
  - complete vs. incomplete likelihood
  - EM as coordinate ascent on $\mathcal{L}(q, \theta)$
  - learning involves inference (E-step)
- mixture models
  - simple assumptions, powerful models
  - applicable to static (GMM) and sequential (HMM) data
  - EM results in "interpretable" parameter estimation algorithms
  - fundamental ideas to understand more complex models

references

- C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- Neal and Hinton, "A new view of the EM algorithm that justifies incremental and other variants," in *Learning in Graphical Models*, MIT Press, 1999.
- Jeff Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian Mixture and Hidden Markov Models," UC Berkeley Technical Report TR-97-021, Apr. 1998.