K-means, EM, Gaussian Mixture, Graph Theory **2013/2014**

Lecture 3 — October 16th

Lecturer: Francis Bach *Scribe: Marie d'Autume, Jean-Baptiste Alayrac*

To talk about estimation of "hidden" parameters, French speaking people and English speaking people use different terms which can lead to some confusions. Within a supervised framework, English people would prefer to use the term *classification* whereas the French use the term *discrimination*. Within an unsupervised context, English people would rather use the term *clustering*, whereas French people would use *classification* or *classification non-supervisée*. In the following we will only use the English terms.

3.1 K-means

K- means clustering is a method of vector quantization. K-means clustering is an algorithm of alternate minimization that aims at partitioning n observations into K clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype to the cluster (see Figure 3.1).

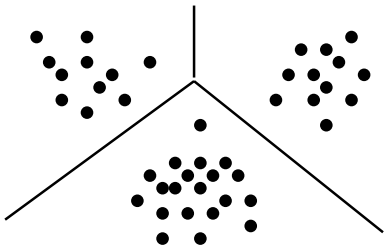

Figure 3.1. Clustering on a 2D point data set with 3 clusters.

3.1.1 Notations and notion of Distortion

We will use the following notations:

- $x_i \in \mathbb{R}^p$, $i \in \{1, ..., n\}$ are the observations we want to partition.
- $\mu_k \in \mathbb{R}^p$, $k \in \{1, ..., K\}$ are the means where μ_k is the center of the cluster k. We will denote μ the associated matrix.
- z_i^k are indicator variables associated to x_i such that $z_i^k = 1$ if x_i belongs to the cluster k, $z_i^k = 0$ otherwise. z is the matrix which components are equal to z_i^k.

Finally, we define the *distortion* $J(\mu, z)$ by:

$$J(\mu, z) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_i^k \|x_i - \mu_k\|^2.$$

3.1.2 Algorithm

The aim of the algorithm is to minimize $J(\mu, z)$. To do so we proceed with an alternating minimization :

- Step 0 : We choose a vector μ
- Step 1 : we minimize J with respect to z : $z_i^k = 1$ if $\| x_i - \mu_k \|^2 = \min_s \| x_i - \mu_s \|^2$, in other words we associate to x_i the nearest center μ_k.
- Step 2 : we minimize J with respect to μ : $\mu_k = \frac{\sum_i z_i^k x_i}{\sum_i z_i^k}$.
- Step 3 : we come back to step 1 until convergence.

Remark 3.1.1 *The step of minimization with respect to z is equivalent to allocating the x_i in the Voronoi cells which centers are the μ_k.*

Remark 3.1.2 *During the step of minimization with respect to μ, μ_k is obtained by setting to zero the k-th coordinate of the gradient of J with respect to μ. Indeed we can easily see that :*

$$\nabla_{\mu_k} J = -2 \sum_i z_i^k (x_i - \mu_k)$$

3.1.3 Convergence and Initialization

We can show that this algorithm converges in a finite number of iterations. Therefore the convergence could be local, thus it introduces the problem of initialization.

A classic method is use of random restarts. It consists in choosing several random vectors μ, computing the algorithm for each case and finally keeping the partition which minimizes the distortion. Thus we hope that at least one of the local minimum is close enough to a global minimum.

One other well known method is the K-means++ algorithm, which aims at correcting a major theoretic shortcomings of the K-means algorithm : the approximation found can be arbitrarily bad with respect to the objective function compared to the optimal clustering.

The K-means++ algorithm addresses this obstacles by specifying a procedure to initialize the cluster centers before proceeding with the standard K-means optimization iterations. With the K-means ++ initialization, the algorithm is guaranteed to find a solution that is $O(\log K)$ competitive to the optimal K-means solution.

The intuition behind this approach is that it is a clever thing to well spread out the K initial cluster centers. At each iteration of the algorithm we will build a new center. We will repeat the algorithm until we have K centers. Here are the steps of the algorithm :

- Step 0 : First initiate the algorithm by choosing the first center uniformly at random among the data points.
- Step 1: For each data point x_i of your data set, compute the distance between x_i and the nearest center that has already been chosen. We denote this distance $D_{\mu_t}(x_i)$ where μ_t is specified to recall that we are minimizing over the current chosen centers.
- Step 2: Choose one new data point at random as a new center, but now using a weighted probability distribution where a point x_i is chosen with probability proportional to $D_{\mu_t}(x_i)^2$.
- Step 3 : Repeat Step 1 and Step 2 until K centers have been chosen.

We see that we have now built K vectors with respect to our first intuition which was to well spread out the centers (because we used a well chosen weighted probability). We can now use those vectors as the initialization of our standard K-means algorithm.

More details can be found on the K-means++ algorithm in [A].

[A] Arthur, D. and Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.

3.1.4 Choice of K

It is important to point out that the choice of K is not universal. Indeed, we see that if we increase K, the distortion J decreases, until it reaches 0 when $K = n$, that is to say when each data point is the center of its own center. To address this issue one solution could be to add to J a penalty over K. Usually it takes the following form :

$$J(\mu, z, K) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_i^k \|x_i - \mu_k\|^2 + \lambda K$$

But again the choice of the penalty is arbitrary.

3.1.5 Other problems

We can also point out that K-means will work pretty well when the width of the different clusters are similar, for example if we deal with spheres. But clustering by K-means could also be disappointing in some cases such as the example given in Figure 3.2.

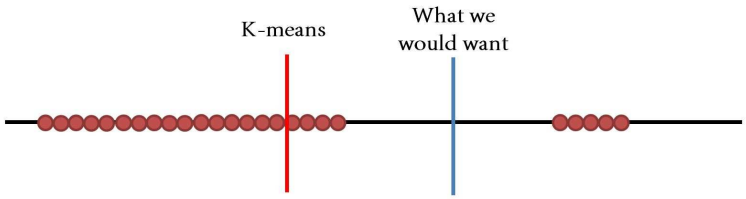

Figure 3.2. Example where K- means does not provide a satisfactory clustering result

Using Gaussian mixtures provides a way to avoid this problem (see next section).

3.2 EM : Expectation Maximization

The Expectation-maximization (EM) algorithm is an iterative method for finding maximum likelihood estimates of parameters in statistical models, where the models depend on unobserved latent or hidden variables z. Latent variables are variables that are not directly observed but are rather inferred from other variables that are observed.

Previous algorithms aimed at estimating the parameter θ that maximized the likelihood of $p_\theta(x)$, where x is the vector of observed variables.

Here it is a little bit different. Indeed we have now :

Assumption : (x, z) are random variables where x is observed (our data) and z is non observed (unknown cluster center for example).

$$p_\theta(x, z) : \text{joint density depending on a parameter } \theta \text{ (the model)}$$

The goal : to maximize the following probability :

$$\max_\theta p_\theta(x) = \sum_z p_\theta(x, z).$$

We can already infer that, because of the sum, the problem should be slightly more difficult than before. Indeed, taking the log of our probability would not lead to a simple convex problem. In the following we will see that EM is a method to solve those kind s of problems.

3.2.1 Example

Let's present a simple example to illustrate what we just said. The probability density represented on Figure 3.2.1 is akin to an average of two Gaussians. Thus, it is natural to use a mixture model and to introduce an hidden variable z, following a Bernoulli distribution defining which Gaussian the point is sampled from.

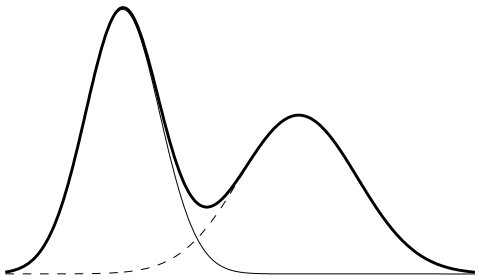

Figure 3.3. Average of two probability distributions of two Gaussian for which it is natural to introduce a mixture model

Thus we have : $z \in \{1, 2\}$ and $x|z = i \sim \mathcal{N}(\mu_i, \Sigma_i)$. The density $p(x)$ is a convex combination of normal density:

$$p(x) = p(x, z = 1) + p(x, z = 2) = p(x|z = 1)p(z = 1) + p(x|z = 2)p(z = 2)$$

It is a mixture model. It represents a simple way to model complicated phenomena.

3.2.2 Objective: maximum likelihood

Let z be the hidden variables and x be the observed data. We make the assumption that the x_i, $i \in \{1, ..., n\}$ are i.i.d..

As we mentioned it in the introduction the aim is to maximize the likelihood

$$p_\theta(x) = \sum_z p_\theta(x, z)$$

$$\log p_\theta(x) = \log \sum_z p_\theta(x, z)$$

.

Note that in practice, we often have $(x, z) = (x_1, z_1, \ldots, x_n, z_n)$ where each pair (x_i, y_i) is i.i.d. In this situation we have $\log p_\theta(x) = \sum_{i=1}^n \log \sum_{z_i} p_\theta(x_i, z_i)$.

There is at least two ways to solve this problem:

1. By a direct way, if we can, by a gradient ascent for example.
2. By using the EM algorithm.

3.2.3 Jensen's Inequality

We will use the following properties :

1. if $f : \mathbb{R} \to \mathbb{R}$ is convex and if X is an integrable random variable :
$$\mathbb{E}_X(f(X)) \geq f(\mathbb{E}_X(X))$$
2. if $f : \mathbb{R} \to \mathbb{R}$ is strictly convex, we have equality in the previous inequality if and only if $X =$ constant a.s.

3.2.4 EM algorithm

We introduce the function $q(z)$ such that $q(z) \geq 0$ and $\sum_z q(z) = 1$ in the expression of the likelihood. Thus we have :

$$\begin{aligned}
\log p_\theta(x) &= \log \sum_z p_\theta(x,z) \\
&= \log \sum_z \left(\frac{p_\theta(x,z)}{q(z)} \right) q(z) \\
&\geq \sum_z q(z) \log \frac{p_\theta(x,z)}{q(z)}, \text{ by the Jensen's inequality because log is concave} \\
&= \sum_z q(z) \log p_\theta(x,z) - \sum_z q(z) \log q(z) \\
&= \mathcal{L}(q,\theta)
\end{aligned}$$

with equality iff $q(z) = \frac{p_\theta(x,z)}{\sum_{z'} p_\theta(x,z')} = p_\theta(z|x)$ (by strict concavity of the logarithm).

Proposition 3.1 *$\forall\theta$, $\forall q$ $\log p_\theta(x) \geq \mathcal{L}(q,\theta)$ with equality if and only if $q(z) = p_\theta(z|x)$.*

Remark 3.2.1 *We have introduced an auxiliary function $\mathcal{L}(q,\theta)$ that is always below the function $\log(p_\theta(x))$*

EM algorithm is an algorithm of alternate maximization with respect to q and θ.

We initialize θ_0, then we iterate for $t > 0$, by alternating the following steps until convergence:

- $q_{t+1} \in \arg\max_q(\mathcal{L}(q,\theta_t))$
- $\theta_{t+1} \in \arg\max_\theta(\mathcal{L}(q_{t+1},\theta))$

Algorithm properties

- It is an ascent algorithm, indeed it goes up in term of likelihood (compare to before where we were descending along the distortion) :

$$\forall t \log(p_{\theta_t}) \geq \log(p_{\theta_{t-1}})$$

- The sequence of log-likelihoods converges.
- It does not converge to a global maximum but rather to a local maximum because we are dealing here with a non-convex problem. An illustration is given in Figure 3.4.

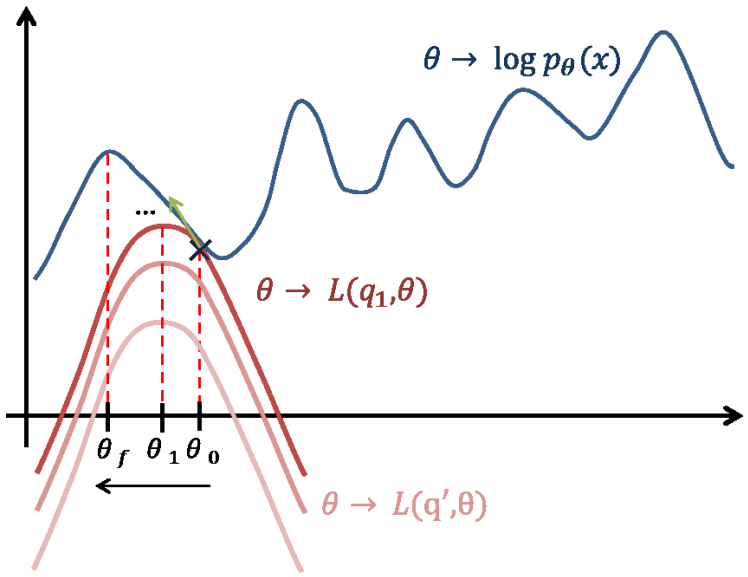

Figure 3.4. An illustration of the EM algorithm that converges to a local minimum.

- As it was already the case for K-means, we reiterate the result in order to be more confident. Then we keep the one with the highest likelihood.

Initialization Because EM gives a local maximum, it is clever to choose a θ_0 relatively close to the final solution. For Gaussian mixtures, it is quite usual to initiate EM by a K-means. The solution of K-means gives the θ_0 and a large variance is used.

The EM recipe Let's recall the initial goal of the algorithm. The goal is to maximize the *incomplete* likelihood $\log(p_\theta(x))$. To do so we want to maximize the following function which is always inferior to $\log(p_\theta(x))$:

$$\mathcal{L}(q,\theta) = \sum_z q(z)\log p_\theta(x,z) - \sum_z q(z)\log q(z).$$

1. Compute the probability of Z given X : $p_{\theta_t}(z|x)$ (Corresponding to $q_{t+1} = \arg\max_q \mathcal{L}(q,\theta_t)$)
2. Write the *complete* likelihood $l_c = \log(p_{\theta_t}(x,z))$.
3. **E-Step** : calculate the expected value of the complete log likelihood function, with respect to the conditional distribution of Z given X under the current estimate of the parameter θ_t : $\mathbb{E}_{Z|X}(l_c)$.
4. **M-Step** : find θ_{t+1} by maximizing $\mathcal{L}(q_{t+1},\theta)$ with respect to θ.

3.2.5 Gaussian Mixture

Let (x_i, z_i) be a couple, for $i \in \{1,...,n\}$ with $x_i \in \mathbb{R}^p$, $z_i \sim \mathcal{M}(1,\pi_1,...,\pi_k)$ and $(x_i|z_i = j) \sim \mathcal{N}(\mu_j,\Sigma_j)$. Here we have $\theta = (\pi,\mu,\Sigma)$.

Calculation of $p_\theta(z|x)$ We write $p_\theta(x_i)$:

$$\begin{aligned} p_\theta(x_i) &= \sum_{z_i} p_\theta(x_i,z_i) = \sum_{z_i} p_\theta(x_i|z_i)p_\theta(z_i) \\ &= \sum_{j=1}^{k} p_\theta(x_i|z_i=j)p_\theta(z_i=j) \end{aligned}$$

Then we use the Bayes formula to estimate $p_\theta(z|x)$:

$$\begin{aligned} p_\theta(z_i=j|x_i) &= \frac{p_\theta(x_i|z_i=j)p_\theta(z_i=j)}{p_\theta(x_i)} \\ &(\propto \quad p_\theta(x_i|z_i=q)p_\theta(z_i=q)) \\ &= \frac{\pi_j \mathcal{N}(x_i|\mu_j,\Sigma_j)}{\sum_{j'}\pi_{j'}\mathcal{N}(x_i|\mu'_j,\Sigma'_j)} \\ &= \tau_i^j(\theta). \end{aligned}$$

We recall that $\mathcal{N}(x_i|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}\exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$.

Suppose that we are at the t-th iteration of the algorithm.

Complete likelihood Let's write the complete likelihood of the problem.

$$\begin{aligned} l_{c,t} = \log p_{\theta_t}(x,z) &= \sum_{i=1}^{n} \log p_{\theta_t}(x_i, z_i) \\ &= \sum_{i=1}^{n} \log(p_{\theta_t}(z_i) p_{\theta_t}(x_i|z_i)) \\ &= \sum_{i=1}^{n} \log(p_{\theta_t}(z_i)) + \log(p_{\theta_t}(x_i|z_i)) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{k} z_i^j \log(\pi_{j,t}) + \sum_{i=1}^{n}\sum_{j=1}^{k} z_i^j \log(\mathcal{N}(x_i|\mu_{j,t}, \Sigma_{j,t})) \end{aligned}$$

where $z_i^j \in \{0,1\}$ with $z_i^j = 1$ if $z_i = j$ and 0 otherwise.

E-Step We can now write the expectation of the previous quantity with respect to the conditional distribution of Z given X. In fact it is equivalent to replace z_i^j by $\mathbb{E}_{Z|X}(z_i^j) = p_{\theta_t}(z = j|x_i) = \tau_i^j(\theta_t)$. Indeed, the other terms of the sum are constant from the point of view of the conditional probability of Z given X, and we finally obtain $\mathbb{E}_{Z|X}(l_{c,t})$. Since the value of θ_t will be fixed during the M-step, we drop the dependence on θ_t and write τ_i^j.

M-Step For the M-step, we this need to maximize:

$$\sum_{i=1}^{n}\sum_{j=1}^{k} \tau_i^j \log(\pi_{j,t}) + \sum_{i=1}^{n}\sum_{j=1}^{k} \tau_i^j \left[\log(\frac{1}{(2\pi)^{\frac{k}{2}}}) + \log(\frac{1}{|\Sigma_{j,t}|^{\frac{1}{2}}}) - \frac{1}{2}(x_i - \mu_{j,t})^T \Sigma_{j,t}^{-1}(x_i - \mu_{j,t}))\right]$$

We want to maximize the previous equation with respect to $\theta_t = (\Pi_t, \mu_t, \Sigma_t)$

As the sum is separated into two terms independent along the variables we can first maximize with respect to π_t :

$$\max_{\Pi} \sum_{j=1}^{k}\sum_{i=1}^{n} \tau_i^j \log \pi_j \quad \Rightarrow \quad \pi_{j,t+1} = \frac{\sum_{i=1}^{n} \tau_i^j}{\sum_{i=1}^{n}\sum_{j'=1}^{k} \tau_i^{j'}} = \frac{1}{n}\sum_{i=1}^{n} \tau_i^j$$

We can now maximize with respect to μ_t and Σ_t. By computing the gradient along the $\mu_{j,t}$ and along the $\Sigma_{j,t}$, we obtain :

$$\mu_{j,t+1} = \frac{\sum_i \tau_i^j x_i}{\sum_i \tau_i^j}$$

$$\Sigma_{j,t+1} = \frac{\sum_i \tau_i^j (x_i - \mu_{j,t+1})(x_i - \mu_{j,t+1})^T}{\sum_i \tau_i^j}$$

The M-step in the EM algorithm corresponds to the estimation of means step in K-means. Note that the value of τ_i^j in the expressions above are taken for the parameter values of the previous iterate, i.e., $\tau_i^j = \tau_i^j(\theta_t)$.

Possible forms for Σ_j

- isotropic: $\Sigma_j = \sigma_j^2 \mathrm{Id}$, 1 parameter, the cluster is a sphere.
- diagonal: Σ_j is a diagonal matrix, d parameters, the cluster is an ellipse oriented along the axis.
- general: Σ_j, $\frac{d(d+1)}{2}$ parameters, the cluster is an ellipse.

3.3 Graph theory

3.3.1 Graph

Definition 3.2 (graph) *A* graph *is a pair $G = (V, E)$ comprising a set V of vertices or nodes together with a set $E \subset V \times V$ of edges or arcs, which are 2-element subsets of V.*

Remark 3.3.1 *In this course we only consider graphs without self-loop.*

3.3.2 Undirected graphs

Definition 3.3 (undirected graph) *$G = (V, E)$ is an if $\forall (u, v) \in V \times V$ with $u \neq v$ we have:*

$$(u, v) \in E \iff (v, u) \in E$$

(Figure 3.5).

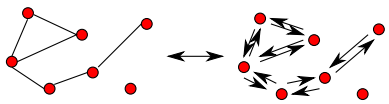

Figure 3.5. two different ways to represent an undirected graph

Definition 3.4 (neighbour) *We define $\mathcal{N}(u)$, the set of the* neighbours *of u, as*

$$\mathcal{N}(u) = \{v \in V, (v, u) \in E\}$$

(Figure 3.6).

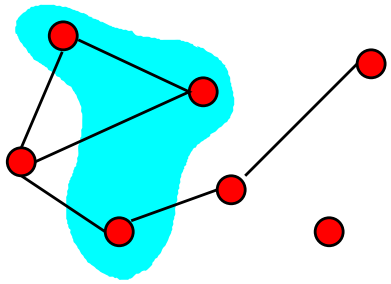

Figure 3.6. A vertex and its neighbours

Definition 3.5 (clique) *A totally connected subset of vertices or a singleton is called a* clique *(Figure 3.7).*

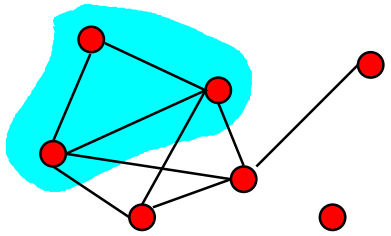

Figure 3.7. A clique.

Definition 3.6 (maximal clique) *A* maximal clique*, C, is a clique which is maximal for the inclusion order:*

$$\nexists v \in V : v \notin C \text{ and } v \cup C \text{ is a clique.}$$

(Figure 3.8).

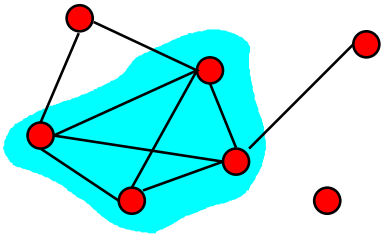

Figure 3.8. A maximal clique

Definition 3.7 (path) *A* path *is a sequence of connected vertices that are globally distinct (Figure 3.9).*

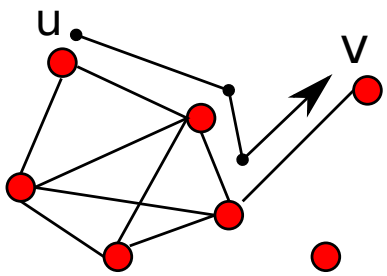

Figure 3.9. A path from u to v.

Definition 3.8 (cycle) *A* cycle *is a sequence of vertices* $(v_0, \ldots, v_k)$ *such that:*

- $v_0 = v_k$
- $\forall j, (v_j, v_{j+1}) \in E$
- $\forall i, j, v_i \neq v_j$ if $\{i, j\} \neq \{1, k\}$

Definition 3.9 *Let* A*,* B*,* C *be distinct subsets of* V*.* C separates A *and* B *if all paths from* A *to* B *go through* C *(Figure 3.10).*

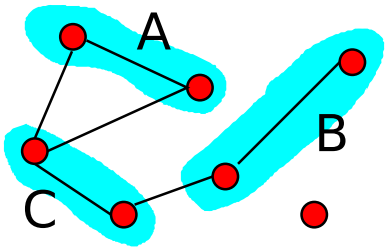

Figure 3.10. C separates A and B.

Definition 3.10 (connected component) *A* connected component *is a subgraph induced by the equivalence class of the relation* $uRv \Leftrightarrow \exists$ *path from* u *to* v *(Figure 3.11).*

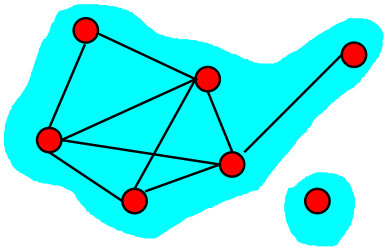

Figure 3.11. A graph with 2 connected components

In this course we will consider there is only one connected component. Otherwise we deal with them independently.

3.3.3 Oriented graphs

Definition 3.11 (parent) *v is a* parent *of u if $(v, u) \in E$*

Definition 3.12 (children) *v is a* children *of u if $(u, v) \in E$*

Definition 3.13 (ancestor) *v is an* ancestor *of u if there exists a path from u to v.*

Definition 3.14 (descendant) *v is a* descendant *of u if there exists a path from u to v*

Definition 3.15 (cycle) *A* cycle *is a sequence of vertices $(v_0, \ldots, v_k)$ (Figure 3.12) such that:*

- $v_0 = v_k$
- $\forall j, (v_j, v_{j+1}) \in E$
- $\forall i, j, v_i \neq v_j$ if $\{i, j\} \neq \{1, k\}$

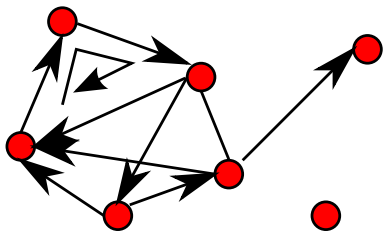

Figure 3.12. Un graphe orienté avec un cycle.

Definition 3.16 (DAG) *A* directed acyclic graph *(DAG) is a directed graph without any cycle.*

Definition 3.17 (topological order) *Let $G = (V, E)$ a graph. I is a* topological order *if*

- *I is a bijection from $\{1, \dots, \}$ to V*
- If *u is a parent of v,* then $I(u) < I(v)$

Proposition 3.18 *$G = (V, E)$ has a topological order $\Leftrightarrow$ G is a DAG.*

Proof $\Rightarrow$ easy, $\Leftarrow$ use a depth-first search ■

3.3.4 Directed graphical models

Notations n discrete random variables $X_1, \dots, X_n$.

- joint distribution:
$$p(x_1, \dots, x_n) = P(X_1 = x_1, \dots, X_n = x_n)$$
- marginal distribution: for $A \subset V$,
$$p(x_A) = p_A(x_A) = P(X_k = x_k, k \in A) = \sum_{x_{A^c}} p(x_A, x_{A^c})$$
- conditional distribution:
$$p(x_A | x_{A^c}) = p_{A|A^c}(x_A | x_{A^c}) = P(X_A = x_A | X_{A^c} = x_{A^c})$$

Review

$$\begin{array}{rcll} X \perp\!\!\!\perp Y & \Leftrightarrow & p(x,y) = p(x)p(y) & \forall x, y \\ & \Leftrightarrow & p_{XY}(x,y) = p_X(x)p_Y(y) & \\ X \perp\!\!\!\perp Y | Z & \Leftrightarrow & p(x,y|z) = p(x|z)p(y|z) & \forall x, y, z \\ & \Leftrightarrow & p(x|y,z) = p(x|z) & \end{array}$$

Definitions and first properties

Let $G = (V, E)$ a DAG with $V = \{1, \dots n\}$ and $(X_1, \dots, X_n)$ n discrete random variables. $\mathcal{L}(G)$ set of $p(x) = p(x_1, \dots, x_n)$ of the form

$$p(x) = \prod_{i=1}^{n} f_i(x_i, x_{\pi_i})$$

with

- π_i set of parents of i

- $\forall i, f_i \geqslant 0$
- $\forall i, \sum_{x_i} f_i(x_i, x_{\pi_i}) = 1$

Proposition 3.19 *If $p(x)$ factorizes in G, i.e. $(p \in \mathcal{L}(G))$, then p is a distribution and*

$$\forall i, f_i(x_i, x_{\pi_i}) = p(x_i | x_{\pi_i})$$

Proof By induction on $n = |V|$. See next class.

■