4. Mixture of Gaussians & Graph cut

Agenda for today’s lecture

Mixture of Gaussians
Multivariate Gaussian distribution
Mixture of Gaussians
Example: Mixture of three 2D Gaussians *
Parameter estimation *
Latent variables
Responsibilities *
Example: Mixture of three 2D Gaussians *
Estimation of a mixture of Gaussians
Recall the EM algorithm *
E step *
M step for $\mu$ *
M step for $\Sigma$ *
M step for $\pi$ *
The EM Algorithm for mixtures of Gaussians.
Example *
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remarks</td>
<td>20</td>
</tr>
<tr>
<td><strong>Graph Cut</strong></td>
<td>21</td>
</tr>
<tr>
<td>Graph cut</td>
<td>22</td>
</tr>
<tr>
<td><strong>Flow network</strong></td>
<td>23</td>
</tr>
<tr>
<td>Flow network and flow</td>
<td>24</td>
</tr>
<tr>
<td>The value of a flow</td>
<td>25</td>
</tr>
<tr>
<td>An equivalent definition of flows*</td>
<td>26</td>
</tr>
<tr>
<td>Working with flows*</td>
<td>27</td>
</tr>
<tr>
<td>Residual network</td>
<td>28</td>
</tr>
<tr>
<td>Max-flow–min-cut theorem</td>
<td>29</td>
</tr>
<tr>
<td>Proof of Max-flow–min-cut theorem 1) ⇒ 2)*</td>
<td>30</td>
</tr>
<tr>
<td>Proof of Max-flow–min-cut theorem 2) ⇒ 3)*</td>
<td>31</td>
</tr>
<tr>
<td>Proof of Max-flow–min-cut theorem 3) ⇒ 1)*</td>
<td>32</td>
</tr>
<tr>
<td>Ford-Fulkerson algorithm</td>
<td>33</td>
</tr>
<tr>
<td>Example: iteration 1*</td>
<td>34</td>
</tr>
<tr>
<td>Example: iteration 2*</td>
<td>35</td>
</tr>
<tr>
<td>Example: iteration 3*</td>
<td>36</td>
</tr>
<tr>
<td>Example: iteration 4*</td>
<td>37</td>
</tr>
<tr>
<td>A “bad” example*</td>
<td>38</td>
</tr>
<tr>
<td>Edmonds–Karp algorithm</td>
<td>39</td>
</tr>
<tr>
<td>Summary*</td>
<td>40</td>
</tr>
<tr>
<td>Literature*</td>
<td>41</td>
</tr>
</tbody>
</table>
4. Mixture of Gaussians & Graph cut

Agenda for today’s lecture

In the previous lecture we learnt about the energy function corresponding to the problem of binary image segmentation:

\[ E(y; x) = \sum_{i \in V} -\log \phi_i(y_i; x_i) + \sum_{(i,j) \in E} w_{ij} [y_i \neq y_j] \]

Today we are going to learn about

- the definition of the unary energies \( E_i(y_i; x_i) \). In fact we estimate mixtures of Gaussians \( f_{bg} \) and \( f_{fg} \) for the foreground and background, respectively, by making use of the EM algorithm.

\[ E_i(y_i; x_i) = \begin{cases} 
-\log f_{bg}(x_i) & \text{if } y_i = 0 \\
-\log f_{fg}(x_i) & \text{otherwise}
\end{cases} \]

- Graph cuts, which will be applied to minimize the energy function.
Mixture of Gaussians

Multivariate Gaussian distribution

Assume a $D$-dimensional random vector $\mathbf{X} = (X_1, \ldots, X_D)$, i.e. a vector whose components are random variables, with the joint density function

$$p(x_1, \ldots, x_D) = \frac{1}{\sqrt{|2\pi \Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right).$$

$\mathbf{X}$ is said to have **multivariate Gaussian (or Normal) distribution** with parameters $\mu \in \mathbb{R}^D$ and $\Sigma \in \mathbb{R}^{D\times D}$ assuming that $\Sigma$ is **positive definite**.

$\mu$ is called the **mean vector** and $\Sigma$ is called the **covariance matrix**. We often use the notation $\mathbf{X} \sim \mathcal{N}(\mathbf{x} | \mu, \Sigma)$ denoting $\mathbf{X}$ has Normal distribution.

**Reminder:** A symmetric $\mathbf{A} \in \mathbb{R}^{n\times n}$ matrix is said to be **positive definite**, if $\mathbf{u}^T \mathbf{A} \mathbf{u} > 0$ for all non-zero $\mathbf{u} \in \mathbb{R}^n$. 
**Mixture of Gaussians**

Let us consider a superposition of $K$ Gaussian densities

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k),$$

which is called a **mixture of Gaussians**.

The parameters $\pi_k$ are called **mixing coefficients**.

All the density functions are non-negative, hence $\pi_k \geq 0$ for $1 \leq k \leq K$, therefore

$$0 \leq \pi_k \leq 1 \quad \text{for all} \quad k = 1, \ldots, K.$$
Example: Mixture of three 2D Gaussians

(a) Iso-contours of each component
(b) Iso-contours of $p(x \mid \theta)$
(c) Surface plot of $p(x \mid \theta)$

Parameter estimation

We are interested in a method to find the maximum likelihood estimator of a parameter $\theta$ of a probability distribution $p(x \mid \theta)$.

Reminiscent of naming conventions:

$$p(\theta \mid x) = \frac{p(x \mid \theta)p(\theta)}{p(x)} \propto p(x \mid \theta)p(\theta).$$

Posterior probability \quad Likelihood \quad Prior probability

We are given finite amount of measurement (i.e. observed data) $x_1, x_2, \ldots$, and also know the probability distribution $p(x \mid \theta)$. The maximum likelihood estimate of $\theta$ is given by

$$\hat{\theta} \in \operatorname{argmax}_\theta p(x \mid \theta).$$

A possible solution: Expectation–maximization algorithm, which iteratively makes guesses about the data $x$, and iteratively maximizes $p(x \mid \theta)$ over $\theta$.

Latent variables

We introduce a $K$-dimensional binary random variable $z \in \mathbb{B}^K$ having a 1-of-$K$ representation, i.e. $z_k = 1$ and all other elements are equal to 0. Let us define the marginal distribution

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

which is considered as the prior probability of picking the $k^{th}$ component of a mixture of Gaussians. This distribution can be also written as a joint distribution

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

Moreover, the conditional distribution of $x$ given a particular value for $z$, i.e. the likelihood, can be written as

$$p(x \mid z_k = 1) = \mathcal{N}(x \mid \mu_k, \Sigma_k), \quad \text{thus} \quad p(x \mid z) = \prod_{k=1}^{K} \mathcal{N}(x \mid \mu_k, \Sigma_k)^{z_k}.$$
The **distribution of mixture of Gaussian**, specified by the parameter vector \( \theta = (\pi, \mu, \Sigma) \in \mathbb{R}^K \times \mathbb{R}^{D \times K} \times \mathbb{R}^{D \times D} \times K \), is given by

\[
p(x) \triangleq p(x \mid \theta) = \sum_z p(x, z \mid \theta) = \sum_z p(z \mid \theta)p(x \mid z, \theta)
\]

\[
= \sum_z \prod_{k=1}^K (\pi_k p(x \mid \mu_k, \Sigma_k))^z_k = \sum_{k=1}^K \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)
\]

The **posterior probabilities** \( p(z_k = 1 \mid x) \), denoted by \( \gamma_k(x) \), a.k.a. **responsibilities**, show the probability that a given sample \( x \) belongs to the \( k \)th component.

\[
\gamma_k(x) \triangleq p(z_k = 1 \mid x) = \frac{p(x \mid z_k = 1)p(z_k = 1)}{p(x)} = \frac{p(x \mid z_k = 1)p(z_k = 1)}{\sum_{l=1}^K p(z_l \mid x)}
\]

\[
= \frac{\sum_{l=1}^K p(z_l = 1)p(x \mid z_l = 1)}{\sum_{l=1}^K p(z_l = 1)} = \frac{\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)}{\sum_{l=1}^K \pi_l \mathcal{N}(x \mid \mu_l, \Sigma_l)}
\]
Example: Mixture of three 2D Gaussians

(a)

(b)

(c)

Samples from \( p(z | \theta)p(x | z, \theta) = \frac{1}{p(x, z | \theta)} \)

Samples from \( p(x | \theta) \)

Values of responsibilities coded by RGB colors \([\gamma_1(x), \gamma_2(x), \gamma_3(x)]\).

Estimation of a mixture of Gaussians

Suppose we have a set of \( \text{i.i.d.} \) data samples \( \{x_1, \ldots, x_N\} \) drawn from a mixture of Gaussians. The data set is represented by \( X \in \mathbb{R}^{N \times D} \).

The goal is to find the parameter vector \( \theta = (\pi, \mu, \Sigma) \), specifying the model from which the samples \( x_n \) have most likely been drawn. We may find the parameters which maximize the likelihood function \( p(x, z | \theta) \). To simplify the optimization we use the log-likelihood function \( L(\theta) \)

\[
\theta^* \in \arg\max_{\theta} L(\theta) = \arg\max_{\theta} \ln p(X, Z | \theta) \overset{\text{i.i.d.}}{=} \arg\max_{\theta} \prod_{n=1}^{N} p(x_n, z_n | \theta)
\]

\[
= \arg\max_{\theta} \ln \prod_{n=1}^{N} p(x_n | z_n, \theta)p(z_n | \theta) = \arg\max_{\theta} \ln \prod_{n=1}^{N} \prod_{k=1}^{K} (\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k))^z_{nk}
\]

\[
= \arg\max_{\theta} \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} (\ln \pi_k + \ln \mathcal{N}(x_n | \mu_k, \Sigma_k))
\]

Note that there is no closed-form solution for this model \( \Rightarrow \) iterative solution.

Recall the EM algorithm *

1. Choose an initial setting for the parameters \( \theta^{(0)} \)
2. \( t \rightarrow 0 \)
3. \textbf{repeat}
4. \( t \rightarrow t + 1 \)
5. \textbf{E step.} Evaluate \( q^{(t-1)}(Z) \overset{\Delta}{=} p(Z | X, \theta^{(t-1)}) \)
6. \textbf{M step.} Evaluate \( \theta^{(t)} \) given by

\[
\theta^{(t)} = \arg\max_{\theta} Q(\theta, \theta^{(t-1)})
\]

where \( Q(\theta, \theta^{(t-1)}) \overset{\Delta}{=} \mathbb{E}[\ln p(X, Z | \theta) | X, \theta^{(t-1)}] \)

\[
= \sum_{Z} p(Z | X, \theta^{(t-1)}) \ln p(X, Z | \theta)
\]

7. \textbf{until} convergence of either the parameters \( \theta \) or the log likelihood \( L(\theta; X) \)
**E step**

We need to calculate \( p(Z \mid X, \theta^{\text{old}}) \), which is calculated based on \( p(z_n \mid x_n, \theta^{\text{old}}) \) for all \( n = 1, \ldots, N \) as follows (see Exercise)

\[
p(z_n \mid x_n, \theta^{\text{old}}) = \frac{p(z_n \mid x_n, \theta^{\text{old}}) \ p(z_n \mid \theta^{\text{old}})}{p(x_n \mid \theta^{\text{old}})} = \frac{\pi_k N(x_n \mid \mu_k, \Sigma_k)}{\sum_{l=1}^{K} \pi_l N(x_n \mid \mu_l, \Sigma_l)} = \gamma_k(x_n).
\]

Therefore, in the **E step** we need to calculate the responsibilities \( \gamma_k(x_n) \) for all data points \( x_n \) and components \( k = 1, \ldots, K \).

---

**M step for \( \mu \)**

We have already known that \( z_{nk} = \gamma_k(x_n) \). Therefore, we may consider

\[
\theta^* \in \arg \max_{\theta} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_k(x_n) \left( \ln \pi_k + \ln N(x_n \mid \mu_k, \Sigma_k) \right) \text{ s.t. } \pi_k > 0, \sum_{k=1}^{K} \pi_k = 1.
\]

Setting the derivative of \( L(\theta) \) w.r.t. \( \mu_k \) to 0, one can obtain that (see Exercise)

\[
\frac{\sum_{n=1}^{N} \gamma_k(x_n) x_n}{\sum_{n=1}^{N} \gamma_k(x_n)} = \mu_k.
\]
M step for $\Sigma$ *

$$\theta^* \in \arg\max_{\theta} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_k(x_n) \left( \ln \pi_k + \ln \mathcal{N}(x_n | \mu_k, \Sigma_k) \right) \text{ s.t. } \pi_k > 0, \sum_{k=1}^{K} \pi_k = 1.$$ 

Setting the derivative of $\mathcal{L}(\theta)$ w.r.t. $\Sigma_k$ to 0, one can obtain (see Exercise)

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma_k(x_n)(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{m=1}^{N} \gamma_k(x_m)}.$$ 

Remark: A $\Sigma \in \mathbb{R}^{D \times D}$ matrix, calculated as

$$\Sigma = \frac{1}{N-1} \sum_{n=1}^{N} (x_n - \mu)(x_n - \mu)^T,$$

is called sample covariance matrix of data points $\{x_n \in \mathbb{R}^D\}_{n=1}^{N}$, where $\mu \in \mathbb{R}^D$ is the sample mean.
M step for $\pi^*$

To integrate the conditions on $\pi$ we use the Lagrange multiplier method

$$\theta^* \in \arg\max_{\theta} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_k(x_n) \left( \ln \pi_k + \ln N(x_n \mid \mu_k, \Sigma_k) \right) + \lambda \left( 1 - \sum_{k=1}^{K} \pi_k \right).$$

Setting the derivative w.r.t. $\pi_k$ to 0, we obtain

$$\sum_{n=1}^{N} \frac{\gamma_k(x_n)}{\pi_k} - \lambda = 0$$

$$\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_k(x_n) = \lambda \sum_{k=1}^{K} \pi_k \Rightarrow N = \lambda$$

therefore

$$\pi_k = \frac{\sum_{n=1}^{N} \gamma_k(x_n)}{N}.$$
The EM Algorithm for mixtures of Gaussians

1: Initialize the means $\mu_k$, covariances $\Sigma_k$ and mixing coefficients $\pi_k$ for all $k = 1, \ldots, K$
2: repeat
3: \hspace{1em} \textbf{E step.} Evaluate the responsibilities using the current parameter values
   \[ \gamma_k(x_n) = \frac{\pi_k N(x_n \mid \mu_k, \Sigma_k)}{\sum_{l=1}^{K} \pi_l N(x_n \mid \mu_l, \Sigma_l)} \quad \text{for } 1 \leq n \leq N \text{ and } 1 \leq k \leq K. \]
4: \hspace{1em} \textbf{M step.} Re-estimate the parameters $(\pi_k, \mu_k, \Sigma_k)$ for all $k = 1, \ldots, K$
   \[ \mu_{k}^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma_k(x_n)x_n}{\sum_{n=1}^{N} \gamma_k(x_n)} \quad \text{and} \quad \Sigma_{k}^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma_k(x_n)(x_n - \mu_{k}^{\text{new}})(x_n - \mu_{k}^{\text{new}})^T}{\sum_{n=1}^{N} \gamma_k(x_n)} \]
5: \hspace{1em} until convergence of either the parameters $\theta$ or the log likelihood $\mathcal{L}(\theta)$
Example *

(a) $L = 1$

(b) $L = 2$

(c) $L = 20$

(d) $L = 2$

(e) $L = 5$

(f) $L = 20$
Remarks

- The EM algorithm is not limited to mixture of Gaussians, but it can also be applied to other probability distributions.
- The algorithm does not necessarily yield global maxima. In practice, it is restarted with different initializations and after convergence the result with the highest log-likelihood is chosen.
- One can think the EM algorithm as an alternating minimization procedure. Considering $f(\theta, q)$ as the objective function, one iteration of the EM algorithm can be reformulated as

\[
\begin{align*}
\text{E-step:} & \quad q^{(t+1)} \in \arg\max_q f(\theta^{(t)}, q) \\
\text{M-step:} & \quad \theta^{(t+1)} \in \arg\max_\theta f(\theta, q^{(t)})
\end{align*}
\]
Graph cut

Assume a **weighted directed graph** $G = (\mathcal{V}, \mathcal{E}, c)$

- $\mathcal{V} = \{1, \ldots, n\}$ is a finite set of nodes,
- $\mathcal{E} \subseteq \{(i, j) \in \mathcal{V} \times \mathcal{V} | i \neq j\}$ is the set of edges,
- $c : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ is a weight function. (For any $(i, j) \notin \mathcal{E}$, $c(i, j) = 0$.)

A cut $(\mathcal{S}, \mathcal{T})$ of $G$ is a *disjoint* partition of $\mathcal{V}$ into $\mathcal{S}$ and $\mathcal{T} = \mathcal{V} \setminus \mathcal{S}$.

The **capacity** of the cut $(\mathcal{S}, \mathcal{T})$ is defined as

$$\text{cut}(\mathcal{S}, \mathcal{T}) = \sum_{(i, j) \in \mathcal{S} \times \mathcal{T}} c(i, j).$$

Assume distinct nodes $s, t \in \mathcal{V}$, a cut $(\mathcal{S}, \mathcal{T})$ is called *$s - t$ cut* if $s \in \mathcal{S}$ and $t \in \mathcal{T}$.

The **minimum $s - t$ cut problem** is to find an $s - t$ cut with the lowest cost.

**Example:** $\text{cut}(\mathcal{S}, \mathcal{T}) = c(v_1, v_3) + c(v_2, v_4) = 12 + 14 = 26$. 

---

IN2329 - Probabilistic Graphical Models in Computer Vision

4. Mixture of Gaussians & Graph cut – 22 / 41
Flow network and flow

Let $G = (V, E, c)$ be a directed weighted graph with non-negative edge weights. Given two distinct nodes, a source $s$ and a sink $t$, we call $(V, E, c, s, t)$ a flow network.

Let $(V, E, c, s, t)$ be a flow network. A function $f : V 	imes V \rightarrow \mathbb{R}$ is called a flow if it satisfies the following properties:

1. **Capacity constraint:** $f(i, j) \leq c(i, j)$ for all $i, j \in V$.
2. **Skew-symmetry:** $f(i, j) = -f(j, i)$ for all $i, j \in V$.
3. **Flow conservation:** $\sum_{j \in V} f(i, j) = 0$ for all $i \in V \setminus \{s, t\}$.

The edges are labeled by $f(i, j)/c(i, j)$. Only positive $f(i, j)$ are shown.
The value of a flow

The value of a flow $f$ is defined as

$$|f| \triangleq \sum_{(s,i) \in E} f(s, i) - \sum_{(i,t) \in E} f(i, t).$$

The maximum-flow problem is to find a flow $f$ with the highest cost for a given flow network $G$. 

The edges are labeled by $f(i,j)/c(i,j)$. 

$|f| = 19$. 

11/16 8/13 10 1/4 7/7 12/12 15/20 11/14 4/9 4/4
An equivalent definition of flows*

Now we give a more intuitive definition of flows. We will see that the previous definition is more helpful for the analysis of the maximum-flow algorithm.

Let \( \mathcal{F} = (V, E, c, s, t) \) be a flow network. A function \( f : E \rightarrow \mathbb{R}^+ \) is called a flow if it satisfies the following two properties:

1. \( f(i, j) \leq c(i, j) \) for all \( (i, j) \in \mathcal{E} \).
2. For all \( i \in V \setminus \{s, t\} \)
   \[
   \sum_{(i,j) \in E} f(i, j) = \sum_{(j,i) \in E} f(j, i) .
   \]

One can see that the two definitions of the flow are equivalent. (See Exercise)

Working with flows*

Let \( G = (V, E, c, s, t) \) be a flow network and let \( f \) be a flow in \( G \). We will use the following notation for \( A, B \subseteq V \)

\[
 f(A, B) = \sum_{a \in A} \sum_{b \in B} f(a, b) .
\]

It is easy to see that \( |f| = f(\{s\}, \{t\}) \), and \( f(\{i\}, V) = 0 \) for all \( i \in V \setminus \{s, t\} \) due to flow conservation.

Let \( G = (V, E, c, s, t) \) be a flow network and let \( f \) be a flow in \( G \). Then the following equalities hold:

i) For all \( A \subseteq V \), we have \( f(A, A) = 0 \).
ii) For all \( A, B \subseteq V \), we have \( f(A, B) = -f(B, A) \).
iii) For all \( A, B, C \subseteq V \) with \( A \cap B = \emptyset \), we have
   \[
   f(A \cup B, C) = f(A, C) + f(B, C) \quad \text{and} \quad f(C, A \cup B) = f(C, A) + f(C, B) .
   \]

\textit{Proof.} Exercise.
Residual network

Let $G = (\mathcal{V}, \mathcal{E}, c, s, t)$ be a flow network and let $f$ be a flow in $G$. The weighted directed graph $G_f = (\mathcal{V}, \mathcal{E}_f, c_f)$ is called residual network of $G$ induced by $f$, where

$$c_f(i, j) = c(i, j) - f(i, j),$$
$$\mathcal{E}_f = \{(i, j) \in \mathcal{V} \times \mathcal{V} : c_f(i, j) > 0\}.$$

A path $p$ from $s$ to $t$ in $G_f$ is called an augmenting path.

Max-flow–min-cut theorem

Let $f$ be a flow in a flow network $G = (\mathcal{V}, \mathcal{E}, c, s, t)$. Then the following conditions are equivalent:

1) $f$ is a maximal flow in $G$.
2) The residual graph $G_f$ contains no augmenting paths.
3) $|f| = \text{cut}(S, T)$ for some $s - t$ cut of $G$. 
Proof of Max-flow–min-cut theorem 1) $\Rightarrow$ 2) *

Suppose that $f$ is maximum flow in $G$, but and there exists an augmenting path $p$ in the residual graph $G_f$.

The maximum amount by which we can increase the flow in $p$ is the residual capacity of $p$, given by

$$c_f(p) = \min\{c_f(i, j) : (i, j) \text{ is on } p\}.$$ 

Furthermore, let us define $f_p : E \rightarrow \mathbb{R}$ as follows:

$$f_p(i, j) = \begin{cases} c_f(p) & \text{if } (i, j) \text{ is on } p \\ -c_f(p) & \text{if } (j, i) \text{ is on } p \\ 0 & \text{otherwise.} \end{cases}$$

One can see that $f_p$ is a flow in $G_f$ with value $|f_p| = c_f(p) > 0$. Therefore the flow $f + f_p$ has the value $|f| + |f_p| > |f|$, which contradicts the optimality of $f$.

Proof of Max-flow–min-cut theorem 2) $\Rightarrow$ 3) *

Suppose that $G_f$ has no augmenting path, i.e. $s$ and $t$ are disconnected in $G_f$. Define

$$S := \{v \in V : \text{there exists a path from } s \text{ to } v \text{ in } G_f\}.$$ 

Obviously, $(S, T)$ is a cut of $G$, where $T = V \setminus S$.

For each pair of $(i, j) \in S \times T$, we have $f(i, j) = c(i, j)$, otherwise $(i, j) \in E_f$ would be held, which would imply that $j \in S$.

One can see that the flow across $(S, T)$ is $|f|$:

$$f(S, T) \overset{\text{def}}{=} f(S, V) - f(S, S) \overset{\text{def}}{=} f(S, V) \overset{\text{def}}{=} f(\{s\}, V) + f(S \setminus \{s\}, V)$$

$$= f(\{s\}, V) = |f|.$$ 

Therefore $|f| = f(S, T) = \text{cut}(S, T)$. 

IN2329 - Probabilistic Graphical Models in Computer Vision

4. Mixture of Gaussians & Graph cut – 31 / 41
Proof of Max-flow–min-cut theorem 3) $\Rightarrow$ 1) *

Let $f$ be a flow in $G$ such that $|f| = \text{cut}(S, T)$. In general, for any flow $f$ in $G$ the following holds:

$$|f| = f(S, T) = \sum_{i \in S} \sum_{j \in T} f(i, j) \leq \sum_{i \in S} \sum_{j \in T} c(i, j) = \text{cut}(S, T).$$

Hence $|f| = \text{cut}(S, T)$ is maximal (equivalently $\text{cut}(S, T)$ is minimal).

---

Ford-Fulkerson algorithm *

Input: A flow network $G = (\mathcal{V}, \mathcal{E}, c, s, t)$

Output: A minimum $s - t$ cut $(S, T)$ of $G$

1: for all $(i, j) \in \mathcal{E}$ do
2: $f(i, j) \leftarrow 0$ and $f(j, i) \leftarrow 0$
3: end for
4: while there exists a path $p$ from $s$ to $t$ in the residual network $G_f$ do
5: $c_f(p) \leftarrow \min\{c_f(i, j) : (i, j) \text{ is in } p\}$
6: for all $(i, j)$ in $p$ do
7: $f(i, j) \leftarrow f(i, j) + c_f(p)$
8: $f(j, i) \leftarrow -f(i, j)$
9: end for
10: end while
11: $S \leftarrow \{v \in \mathcal{V} : \text{ there exists a path from } s \text{ to } v \text{ in } G_f\}$ and $T \leftarrow \mathcal{V} \setminus S$

The complexity of this algorithm is $O(|\mathcal{E}| \cdot |f^*|)$, where $f^*$ is the value of the maximal flow.
Example: iteration 1

Flow Residual network

IN2329 - Probabilistic Graphical Models in Computer Vision
4. Mixture of Gaussians & Graph cut – 34 / 41
Example: iteration 2

Flow Residual network

IN2329 - Probabilistic Graphical Models in Computer Vision

4. Mixture of Gaussians & Graph cut – 35 / 41
Example: iteration 3

Flow

Residual network
Example: iteration 4 *

Flow

Residual network

IN2329 - Probabilistic Graphical Models in Computer Vision

4. Mixture of Gaussians & Graph cut – 37 / 41
Note that there exists an example, where the flow, computed by the Ford–Fulkerson algorithm, does not even converge to the maximum flow.

More precisely, if a flow network has integer ($\mathbb{N}_0$) or rational ($\mathbb{Q}_0^+$) capacities, then the Ford–Fulkerson algorithm terminates and it computes a maximum flow.
Edmonds–Karp algorithm

**Input:** A flow network \( G = (V, E, c, s, t) \)

**Output:** A minimum \( s - t \) cut \((S, T)\) of \( G \)

1. for all \((i, j) \in E\) do
2. \( f(i, j) \leftarrow 0 \) and \( f(j, i) \leftarrow 0 \)
3. end for
4. while there exists a path \( p \) from \( s \) to \( t \) in the residual network \( G_f \) do
5. \( p \leftarrow \text{shortestPath}(G_f, s, t) \)
6. \( c_f(p) \leftarrow \min\{c_f(i, j) : (i, j) \text{ is in } p\} \)
7. for all \((i, j) \in p\) do
8. \( f(i, j) \leftarrow f(i, j) + c_f(p) \)
9. \( f(j, i) \leftarrow -f(i, j) \)
10. end for
11. end while
12. \( S \leftarrow \{v \in V : \text{there exists a path from } s \text{ to } v \text{ in } G_f\} \) and \( T \leftarrow V \backslash S \)

The complexity of this algorithm is \( O(|V| \cdot |E|^2) \). There exist more efficient algorithms for maximum flow calculation with complexity \( O(|V|^2 \cdot |E|) \) and \( O(|V|^3) \).
Summary *

- **Max-flow–min-cut theorem** tells us that the minimum cut problem can be solved via maximum flow. These two problems are dual to each other, moreover strong duality holds.
- **Edmonds–Karp algorithm**: The Ford-Fulkerson algorithm becomes polynomial, if the shortest path is used as augmented path.

In the next lecture we will learn about

- Exact solution for binary image segmentation via graph cut
- Boykov–Kolmogorov algorithm
- Multi-label problem (e.g., stereo matching)

Literature *

**The EM algorithm for mixture of Gaussians**


**Graph cut, Maximum flow**