Probabilistic Graphical Models in Computer Vision (IN2329)

Csaba Domokos

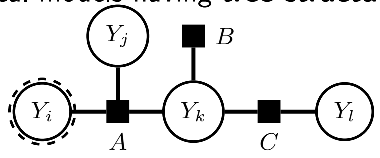
Summer Semester 2015/2016

Human pose estimation **4**

Mean Field methods **14**

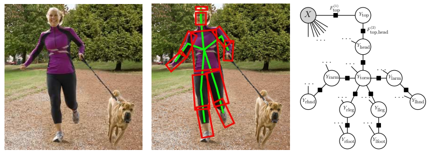
Agenda for today's lecture *

In the last lecture we learnt about inference methods on graphical models having **tree structure**.



Today we are going to learn about

- *Human-pose estimation*:



- *Mean-field approximation*: probabilistic inference via optimization (a.k.a. variational inference)

The model

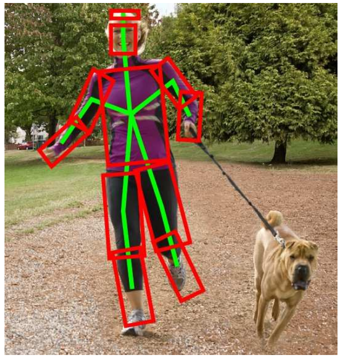
The goal is to recognize an articulated object with joints connecting different parts, here it is a human body.

An object is composed of a number of rigid parts. Each part is modeled as a rectangle parameterized by (x, y, s, θ), where

- (x, y) means the **center of the rectangle**,
- $s \in [0, 1]$ is a **scaling factor**, and
- **the orientation** is given by θ.

In overall, we have a four-dimensional pose space.

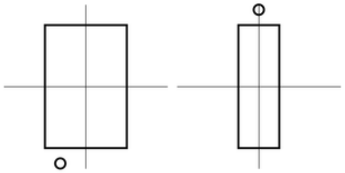
We denote the **locations** of two (connected) parts by $l_i = (x_i, y_i, s_i, \theta_i)$ and $l_j = (x_j, y_j, s_j, \theta_j)$, respectively.

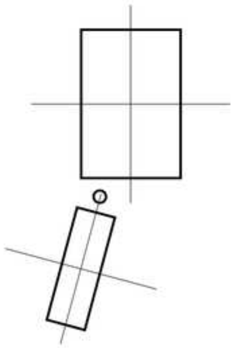


The model (cont.)

An object (e.g., human body) is given by a configuration $\mathbf{l} = (l_1, \ldots, l_n)$, where l_i specifies the location of **part** v_i. The connections encode generic relationships such as “close to”, “to the left of”, or more precise geometrical constraints such as ideal joint angles.

- The **location of a joint** between v_i and v_j is specified by two points (x_{ij}, y_{ij}) and (x_{ji}, y_{ji}).
- The **relative orientation** is given by θ_{ij}, which is the difference between the orientation of the two parts.





In principle, all parts depend on each other, however, tree structured model can be considered for an articulated pose.

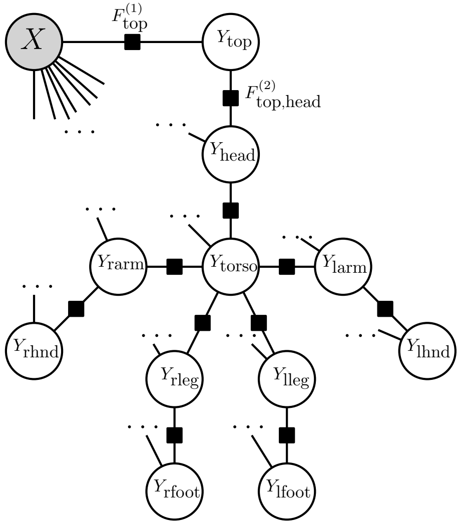
Graphical representation

The structure is encoded by a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \dots, v_n\}$ corresponds to n parts, and there is an edge $(v_i, v_j) \in \mathcal{E}$ for each pair of connected parts v_i and v_j.

We want to minimize the following *energy function*:

$$\mathbf{l}^* \in \operatorname*{argmin}_{\mathbf{l}} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right) ,$$

where $m_i(l_i)$ measures the degree of mismatch when the part v_i is placed at location l_i and $d_{ij}(l_i, l_j)$ measures the degree of deformation of the model when part v_i is placed at location l_i and part v_j is placed at location l_j.



Note that MAP inference can be efficiently done by making use of `Max-sum algorithm`.

Image filters *

The **image filtering** is a technique for modifying or enhancing an image (e.g., smoothing, edge detection, sharpening). For example, the smoothing of an input signal means of removing (or filtering out) high-frequency components.

A **digital image** can be considered as a two dimensional (discretized) signal that is $f : \mathbb{Z}^2 \to \mathbb{Z}^D$. For example $D = 3$ for color images.

Here we consider **linear filtering** in which the value of an output pixel is a linear combination of the values of the pixels in the input pixel's neighborhood. In a spatially discrete setting, a linear filter is a weighted sum:

$$g(x_0, y_0) = [f * w](x_0, y_0) = \sum_{m,n} w(m,n) f(x_0 - m, y_0 - n)$$

which is also called **discrete convolution** of f and w. In practice this summation extends over a certain neighborhood. The matrix of weights $w(m,n)$ is called a **mask**.

(For more details please refer to the course of **Computer Vision I: Variational Methods**.)

Distance transform *

Let $\mathcal{G} \subset \mathbb{Z}^2$ denote a grid. Assume a **distance function** $\rho(x,y) : \mathcal{G} \to \mathbb{R}_0^+$, that is for all $x, y, z \in \mathcal{G}$

1. $\rho(x,y) = 0 \quad \Leftrightarrow \quad x = y,$
2. $\rho(x,y) = \rho(y,x),$
3. $\rho(x,y) + \rho(y,z) \geqslant \rho(x,z).$

Given a point set $B \subset \mathcal{G}$, the **distance transform** of B specifies the distance to the closest point in the set,

$$\mathcal{D}_B(x) = \min_{y \in B} \rho(x,y) = \min_{y \in \mathcal{G}} (\rho(x,y) + \chi_B(y)) ,$$

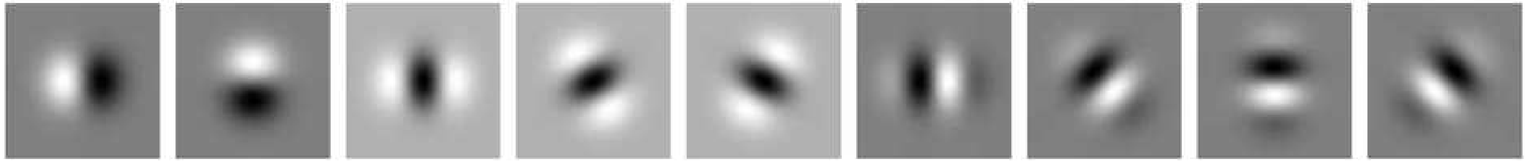
where χ_B is the characteristic function of B. The **generalized distance transform** is defined as

$$\mathcal{D}_f(x) = \min_{y \in \mathcal{G}} (\rho(x,y) + f(y)) ,$$

where $f : \mathcal{G} \to \mathbb{R}$ is an arbitrary function. There exist some algorithms to compute a generalized distance transform of B in $\mathcal{O}(|\mathcal{G}|)$ time.

Unary energies *

An image patch centered at some position is represented by a vector that collects all the responses of a set of Gaussian derivative filters of different orders, orientations and scales at that point. This vector is normalized and called the **iconic index** at that position.



The unary energies are defined as

$$m_i(l_i) = -\ln \mathcal{N}(\alpha(l_i), \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \ ,$$

where $\alpha(l_i)$ is the iconic index at location l_i in the image.

The parameters for each part (i.e. the mean vector $\boldsymbol{\mu}_i$ and the covariance matrix $\boldsymbol{\Sigma}_i$) can be obtained by maximum likelihood estimation for a given set of training samples.

Pairwise energies *

The pairwise energies have a special form as follows.

$$d_{ij}(l_i, l_j) = -\ln \mathcal{N}(T_{ji}(l_j) - T_{ij}(l_i), \mathbf{0}, \mathbf{D}_{ij}) \ ,$$

where where T_{ij}, T_{ji} and $\mathbf{D}_{ij}$ are the connection parameters

$$\begin{aligned} T_{ij}(l_i) =&(x'_i, y'_i, s_i, \cos(\theta_i + \theta_{ij}), \sin(\theta_i + \theta_{ij})), \\ T_{ji}(l_j) =&(x'_j, y'_j, s_j, \cos(\theta_j), \sin(\theta_j)), \\ \mathbf{D}_{ij} =&\text{diag}(\sigma_x^2, \sigma_y^2, \sigma_s^2, 1/k, 1/k) \ . \end{aligned}$$

$T_{ij}(l_i)$ and $T_{ji}(l_j)$ are one-to-one mappings encoding the set of possible transformed locations.

This special form for the pairwise energies allows for matching algorithms that run in $\mathcal{O}(h')$, where h' is the number of grid locations in a discretization of the space. This results in the time complexity $\mathcal{O}(h'n)$ rather than $\mathcal{O}(h^2n)$.

Pairwise energies (cont.) *

Let $\mathbf{R}$ be the matrix that performs a rotation of θ radians about the origin. Then,

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + s_i \mathbf{R}_{\theta_i} \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x_j' \\ y_j' \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \end{bmatrix} + s_j \mathbf{R}_{\theta_j} \begin{bmatrix} x_{ji} \\ y_{ji} \end{bmatrix} ,$$

where (x_i, y_i), (x_j, y_j) and (x_{ij}, y_{ij}), (x_{ji}, y_{ji}) are the positions of the joints in image and local coordinates, respectively.
We assume the following joint distributions:

- $\mathcal{N}(x_i - x_j, \mathbf{0}, \sigma_x^2)$ and $\mathcal{N}(y_i - y_j, \mathbf{0}, \sigma_y^2)$ which measures the horizontal and vertical distances, respectively, between the observed joint positions.
- $\mathcal{N}(s_i - s_j, 0, \sigma_s^2)$ measures the difference in foreshortening between the two parts.
- $\mathcal{M}(\theta_i - \theta_j, \theta_{ij}, k) \propto \exp(k \cos(\theta_i - \theta_j - \theta_{ij}))$ measures the difference between the relative angle of the two parts and the ideal relative angle.

These parameters can be also obtained by maximum likelihood estimation.

Inference

MAP inference provides a single (best) prediction of the overall pose. The factor–to–varaible messages can be written as

$$r_{F \to v_i}(l_i) = \max_{\substack{(l_i', l_j') \in \mathcal{Y}_F, \\ l_i' = l_i}} \left(\exp(-m_i(l_i') - d_{ij}(l_i', l_j')) + \sum_{k \in N(F) \setminus \{i\}} q_{v_k \to F}(l_k') \right) .$$

$\mathcal{Y}$ could be quite large ($\approx 1.5M$ possible states), hence $\mathcal{Y}_i \times \mathcal{Y}_j$ is too big. However a special form of pairwise energies is used, so that a message can be calculated in $\mathcal{O}(|\mathcal{Y}_i|)$ time.

KL divergence

Assume two discrete probability distributions P and Q. One way to measure the *difference* between P and Q is to calculate the **Kullback–Leibler (KL) divergence** (a.k.a. relative entropy) defined as

$$
\begin{aligned}
D_{\mathrm{KL}}(P \| Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} = & \sum_i P(i) \log P(i) - \sum_i P(i) \log Q(i) \\
= & \mathbb{E}_P[\log P(i)] - \mathbb{E}_P[\log Q(i)] \;.
\end{aligned}
$$

It is defined iff $Q(i) = 0$ implies $P(i) = 0$, for all i. If $P(i) = 0$, then the ith term is interpreted as 0. The KL divergence is always non-negative, moreover $D_{\mathrm{KL}}(P \| Q) = 0$ iff $P = Q$ *almost everywhere*. Nevertheless, it is neither symmetric nor does it satisfy the triangle inequality.

Interpretation (Information Theory): it is the amount of information lost when Q is used to approximate P. It measures the expected number of extra bits required to code samples from P using a code optimized for Q rather than the code optimized for P.

Motivation

For general (discrete) factor graph models, performing *probabilistic inference* is hard. Assume we are given an **intractable** distribution $p(\mathbf{y} \mid \mathbf{x})$. We consider an **approximate distribution** $q(\mathbf{y})$, which is tractable, for $p(\mathbf{y} \mid \mathbf{x})$.

One way of finding the best approximating distribution is to pose it as an **optimization problem** over probability distributions: given a distribution $p(\mathbf{y} \mid \mathbf{x})$ and a family Q of *tractable distributions* $q \in Q$ on $\mathcal{Y}$, we want to solve

$$
\begin{aligned}
q^* \in \underset{q \in Q}{\operatorname{argmin}}\, D_{\mathrm{KL}}(q(\mathbf{y}) \| p(\mathbf{y} \mid \mathbf{x})) = & \underset{q \in Q}{\operatorname{argmin}} \sum_{\mathbf{y} \in \mathcal{Y}} q(\mathbf{y}) \log \frac{q(\mathbf{y})}{p(\mathbf{y} \mid \mathbf{x})} \\
= & \underset{q \in Q}{\operatorname{argmin}} \Bigg\{ \underbrace{\sum_{\mathbf{y} \in \mathcal{Y}} q(\mathbf{y}) \log q(\mathbf{y})}_{-H(q)} - \sum_{\mathbf{y} \in \mathcal{Y}} q(y) \log p(\mathbf{y} \mid \mathbf{x}) \Bigg\} \;.
\end{aligned}
$$

The term $-\sum_{\mathbf{y} \in \mathcal{Y}} q(\mathbf{y}) \log q(\mathbf{y}) \triangleq H(q)$ is called the **entropy** of the distribution q.

Mean Field methods

$$
\begin{aligned}
D_{\mathrm{KL}}(q(\mathbf{y}) \| p(\mathbf{y} \mid \mathbf{x})) &= -\,H(q) - \sum_{\mathbf{y} \in \mathcal{Y}} q(\mathbf{y}) \log p(\mathbf{y} \mid \mathbf{x}) \\
&= -H(q) - \sum_{\mathbf{y} \in \mathcal{Y}} q(\mathbf{y}) \log \frac{1}{Z(\mathbf{x})} \prod_{F \in \mathcal{F}} \exp(-E_F(\mathbf{y}_F; \mathbf{x}_F)) \\
&= -H(q) + \sum_{\mathbf{y} \in \mathcal{Y}} q(\mathbf{y}) \sum_{F \in \mathcal{F}} E_F(\mathbf{y}_F; \mathbf{x}_F) + \log Z(\mathbf{x}) \\
&= -H(q) + \sum_{F \in \mathcal{F}} \sum_{\mathbf{y}_F \in \mathcal{Y}_F} \underbrace{\sum_{\substack{\mathbf{y}' \in \mathcal{Y}, \\ \mathbf{y}'_F = \mathbf{y}_F}} q(\mathbf{y})}_{\mu_{F,\mathbf{y}_F}(q)} E_F(\mathbf{y}_F; \mathbf{x}_F) + \log Z(\mathbf{x}) \\
&= -H(q) + \sum_{F \in \mathcal{F}} \sum_{\mathbf{y}_F \in \mathcal{Y}_F} \mu_{F,\mathbf{y}_F}(q) E_F(\mathbf{y}_F; \mathbf{x}_F) + \log Z(\mathbf{x})\ ,
\end{aligned}
$$

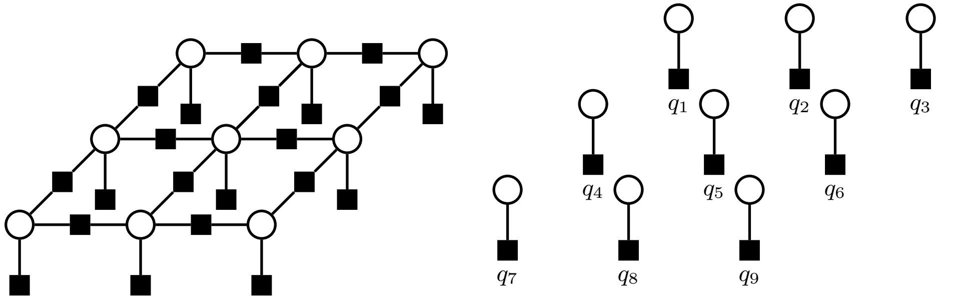
where $\mu_{F,\mathbf{y}_F}(q) = \sum_{\mathbf{y}' \in \mathcal{Y}, \mathbf{y}'_F = \mathbf{y}_F} q(\mathbf{y})$ are the marginals of q.

Naïve mean field

Take a set Q as the set of all distributions in the form:

$$q(\mathbf{y}) = \prod_{i \in \mathcal{V}} q_i(y_i) \ .$$

For example, in case of the following factor graph:



Original factor graph

Mean field approximation

Naïve mean field

Set Q consists of all distributions in the form:

$$q(\mathbf{y}) = \prod_{i \in \mathcal{V}} q_i(y_i) \ .$$

Marginals $\mu_{F,\mathbf{y}_F}$ take the form

$$\mu_{F,\mathbf{y}_F}(q) = \sum_{\substack{\mathbf{y}' \in \mathcal{Y}, \\ \mathbf{y}'_F = \mathbf{y}_F}} q(\mathbf{y}) = q_{N(F)}(\mathbf{y}_F) = \prod_{i \in N(F)} q_i(y_i) \ .$$

Entropy $H(q)$ decomposes as

$$H(q) = \sum_{i \in \mathcal{V}} H_i(q_i) = -\sum_{i \in \mathcal{V}} \sum_{y_i \in \mathcal{Y}_i} q_i(y_i) \log q_i(y_i) \ .$$

Proof. Exercise. □

Naïve Mean Field

Putting all together,

$$\begin{aligned} q^* &\in \operatorname*{argmin}_{q \in Q} D_{\mathrm{KL}}(q(\mathbf{y}) \| p(\mathbf{y} \mid \mathbf{x})) \\ &= \operatorname*{argmin}_{q \in Q} \left\{ -H(q) + \sum_{F \in \mathcal{F}} \sum_{\mathbf{y}_F \in \mathcal{Y}_F} \mu_{F,\mathbf{y}_F}(q) E_F(\mathbf{y}_F; \mathbf{x}_F) + \log Z(\mathbf{x}) \right\} \\ &= \operatorname*{argmax}_{q \in Q} \left\{ H(q) - \sum_{F \in \mathcal{F}} \sum_{\mathbf{y}_F \in \mathcal{Y}_F} \mu_{F,\mathbf{y}_F}(q) E_F(\mathbf{y}_F; \mathbf{x}_F) \right\} \\ &= \operatorname*{argmax}_{q \in Q} \left\{ -\sum_{i \in \mathcal{V}} \sum_{y_i \in \mathcal{Y}_i} q_i(y_i) \log q_i(y_i) - \sum_{F \in \mathcal{F}} \sum_{\mathbf{y}_F \in \mathcal{Y}_F} \Big(\prod_{i \in N(F)} q_i(y_i) \Big) E_F(\mathbf{y}_F; \mathbf{x}_F) \right\} . \end{aligned}$$

Optimizing over Q means to optimize over all q_i such that $q_i(y_i) \geqslant 0$ and $\sum_{y_i \in \mathcal{Y}_i} q_i(y_i) = 1$ for all $i \in \mathcal{V}$.

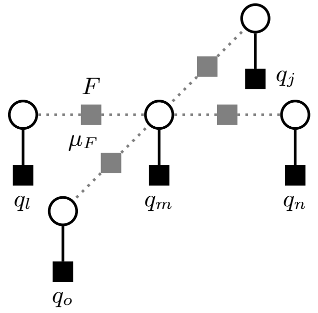
Optimization

$$\underset{q \in Q}{\operatorname{argmax}} \Bigg\{ \underbrace{- \sum_{i \in \mathcal{V}} \sum_{y_i \in \mathcal{Y}_i} q_i(y_i) \log q_i(y_i)}_{\text{entropy}} - \sum_{F \in \mathcal{F}} \sum_{\mathbf{y}_F \in \mathcal{Y}_F} \Big(\prod_{i \in N(F)} q_i(y_i) \Big) E_F(\mathbf{y}_F; \mathbf{x}_F) \Bigg\}.$$

The *entropy* term is concave and the second term is non-concave due to products of variables occurring in the expression. Therefore solving this non-concave maximization problem globally is hard in general.

Remedy: **block coordinate ascent**

We hold all variables fixed except for a single block q_m, then we obtain a tractable concave maximization problem
→ closed-form update for each q_m.



Lagrange multipliers

To obtain closed form solution, we define the *Lagrangian function*:

$$L(q_i, \lambda) = \Bigg\{ -\sum_{i \in \mathcal{V}} \sum_{y_i \in \mathcal{Y}_i} q_i(y_i) \log q_i(y_i)$$
$$- \sum_{F \in \mathcal{F}} \sum_{\mathbf{y}_F \in \mathcal{Y}_F} \Big(\prod_{i \in N(F)} q_i(y_i) \Big) E_F(\mathbf{y}_F; \mathbf{x}_F) + \lambda \Big(\sum_{y_i \in \mathcal{Y}_i} q_i(y_i) - 1 \Big) \Bigg\}.$$

Setting the derivatives of L w.r.t. q_i to 0, we obtain

$$\frac{\partial L}{\partial q_i(y_i)} = 0 = -\left(\log q_i(y_i) + 1\right) - \sum_{F \in M(i)} \sum_{\substack{\mathbf{y}'_F \in \mathcal{Y}_F, \\ y'_i = y_i}} \Big(\prod_{j \in N(F) \setminus \{i\}} \hat{q}_j(y_j) \Big) E_F(\mathbf{y}_F; \mathbf{x}_F) + \lambda$$

$$q_i^*(y_i) = \exp\left(-1 - \sum_{F \in M(i)} \sum_{\substack{\mathbf{y}'_F \in \mathcal{Y}_F, \\ y'_i = y_i}} \Big(\prod_{j \in N(F) \setminus \{i\}} \hat{q}_j(y_j) \Big) E_F(\mathbf{y}_F; \mathbf{x}_F) + \lambda \right).$$

Lagrange multipliers

λ can be calculated as follows.

$$\begin{aligned}
\sum_{y_i\in\mathcal{Y}_i} q_i(y_i) &= \sum_{y_i\in\mathcal{Y}_i} \exp\Bigg(-1-\sum_{F\in M(i)}\sum_{\substack{\mathbf{y}'_F\in\mathcal{Y}_F,\\ y'_i=y_i}}\Big(\prod_{j\in N(F)\setminus\{i\}}\hat{q}_j(y_j)\Big)E_F(\mathbf{y}_F;\mathbf{x}_F)+\lambda\Bigg)\\
\exp(1-\lambda) &= \underbrace{\sum_{y_i\in\mathcal{Y}_i} \exp\Bigg(-\sum_{F\in M(i)}\sum_{\mathbf{y}'_F\in\mathcal{Y}_F, y'_i=y_i}\Big(\prod_{j\in N(F)\setminus\{i\}}\hat{q}_j(y_j)\Big)E_F(\mathbf{y}_F;\mathbf{x}_F)\Bigg)}_{Z_i(\mathbf{x}_F)}\\
\lambda-1 &= -\log Z_i(\mathbf{x}_F)\ ,
\end{aligned}$$

where $Z_i(\mathbf{x}_F)$ is a normalizing constant for q_i.

Update equation

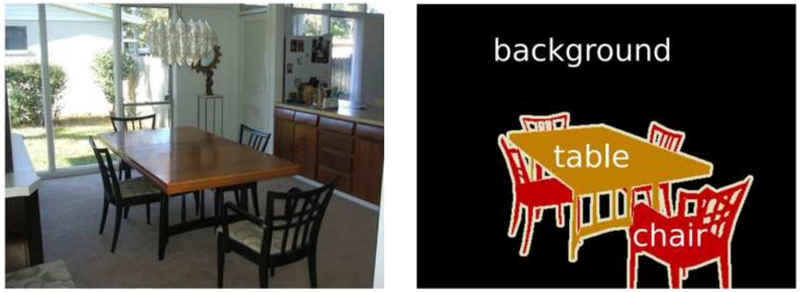
By substituting, we obtain the obtain the update equation for the *Naive Mean Field method*

$$\begin{aligned}
q_i^*(y_i) &= \exp\Bigg(-\sum_{F\in M(i)}\sum_{\substack{\mathbf{y}'_F\in\mathcal{Y}_F,\\ y'_i=y_i}}\Big(\prod_{j\in N(F)\setminus\{i\}}\hat{q}_j(y_j)\Big)E_F(\mathbf{y}_F;\mathbf{x}_F)-\log Z_i(\mathbf{x}_F)\Bigg)\\
&= \frac{1}{Z_i(\mathbf{x}_F)}\exp\Bigg(-\sum_{F\in M(i)}\sum_{\substack{\mathbf{y}'_F\in\mathcal{Y}_F,\\ y'_i=y_i}}\Big(\prod_{j\in N(F)\setminus\{i\}}\hat{q}_j(y_j)\Big)E_F(\mathbf{y}_F;\mathbf{x}_F)\Bigg)\ .
\end{aligned}$$

Semantic segmentation

Krähenbühl and Koltun proposed an efficient approximate inference in fully connected CRF model by applying *Naive Mean Field* approach.

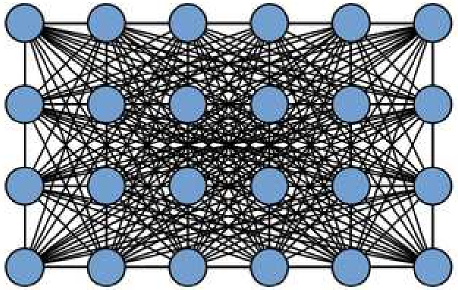
Semantic segmentation: assign a label from the set of labels $\mathcal{L} = \{l_1, l_2, \ldots, l_k\}$ for each pixel on the image regarding their semantic meaning.



For each pixel on the image a random variable is assigned taking a value from $\mathcal{L}$. A fully connected pairwise CRF model $G = (\mathcal{V}, \mathcal{E})$ is considered, where the corresponding energy function is given by

$$E(\mathbf{y}) = \sum_{i \in \mathcal{V}} E_i(y_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j) \ ,$$

where $\mathcal{E} = \{(i,j) \in \mathcal{V} \times \mathcal{V} \mid i < j\}$.



Energy functions

- **Unary energies** $E_i(y_i)$ are computed independently for each pixel as $E_i(y_i) = -\log P_i(y_i)$ measures the degree of disagreement between labelling y_i and the image at pixel i.
- **Pairwise energies** (so-called **contrast sensitive Potts-model**), measuring the extent to which the labelling y is not piecewise smooth, have the form (p_i and I_i denote the pixel coordinates and intensity, resp.)

$$\begin{aligned} E_{ij}(y_i, y_j) =& [\![y_i \neq y_j]\!] \sum_m w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) \\ =& [\![y_i \neq y_j]\!] \sum_m w^{(m)} \exp\left(-\frac{1}{2}(\mathbf{f}_i - \mathbf{f}_j)^T \mathbf{\Sigma}^{(m)} (\mathbf{f}_i - \mathbf{f}_j)\right) \\ =& [\![y_i \neq y_j]\!] \Big\{ w^{(1)} \exp\Big(- \frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\Big) \\ & + w^{(2)} \exp\Big(- \frac{|p_i - p_j|^2}{2\theta_\gamma^2}\Big)\Big\} \, . \end{aligned}$$

The parameters $\theta_\alpha, \theta_\beta$ and θ_γ are estimated on a set of training images.

Inference

The inference is based on *Naive Mean Field approximation*, where the update equation is given by

$$q_i(y_i) = \frac{1}{Z_i} \exp\left\{ -E_i(y_i) - \sum_{l' \in \mathcal{L}} [\![y_i \neq y_j]\!] \sum_{m=1}^{K} w^{(m)} \sum_{i \neq j} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) q_j(l') \right\} .$$

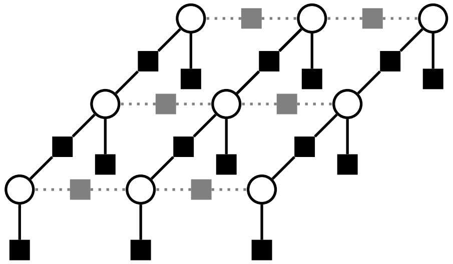
The inference is performed in average 0.2 seconds for 500.000 variables (in contrast to 36 hours).
The main idea: the message passing step can be expressed as a convolution with a Gaussian kernel $G_{\mathbf{\Sigma}^{(m)}}$ in feature space:

$$\sum_{j \in \mathcal{V}} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) q_j(l) - q_i(l) = [G_{\mathbf{\Sigma}^{(m)}} * q(l)](\mathbf{f}_i) - q_i(l) .$$

Note that the convolution sums over all variables, while message passing does not sum over q_i. This convolution can be efficiently calculated in $\mathcal{O}(|\mathcal{V}|)$ time (instead of $\mathcal{O}(|\mathcal{V}|^2)$).

Structured mean field

To improve the approximation of naive mean field one can take larger (tractable) subgraph of the original factor graph, which leads to the **structured mean field** approach.



- For each component the mean field update can be performed efficiently if inference for the component is tractable

The resulting family Q of distributions is *richer* and therefore the approximation is improved.
Compared to the *naive mean field approximation* the entropies $H(q)$ now decompose over the subgraphs instead of individual variables.

Summary *

Mean field approximation: instead of an *intractable* distribution $p(\mathbf{y} \mid \mathbf{x})$, we consider an *approximate distribution* $q(\mathbf{y})$, which minimizes the KL divergence.

In case of *naïve mean field approximation* $q(\mathbf{y})$ is defined as

$$q(\mathbf{y}) = \prod_{i \in \mathcal{V}} q_i(y_i) \ ,$$

which is tractable.

A local optimal solution can be obtained by applying the update equation:

$$q_i^*(y_i) = \frac{1}{Z_i(\mathbf{x}_F)} \exp\left(- \sum_{F \in M(i)} \sum_{\substack{\mathbf{y}'_F \in \mathcal{Y}_F, \\ y'_i = y_i}} \Big(\prod_{j \in N(F) \setminus \{i\}} \hat{q}_j(y_j) \Big) E_F(\mathbf{y}_F; \mathbf{x}_F) \right) .$$

Next lecture *

In the **next lecture** we will learn about

- Sampling of $p(\mathbf{y} \mid \mathbf{x})$: *Gibbs sampler*
- *Parameter learning*: calculate the optimal values for w_1 and w_2

$$E(\mathbf{y}) = w_1 \sum_{i \in \mathcal{V}} E_i(y_i) + w_2 \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j)$$

assuming a *set of training examples*.

Literature *

Human pose estimation

1. Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, January 2005

Mean field approximation

2. Sebastian Nowozin and Christoph H. Lampert. Structured prediction and learning in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3–4), 2010
3. Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009
4. Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Proceedings of Advances in Neural Information Processing Systems*, pages 109–117, Granada, Spain, December 2011. MIT Press