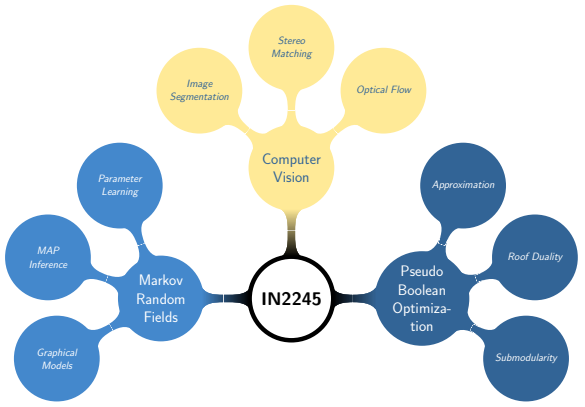
Combinatorial Optimization in Computer Vision (IN2245)

Frank R. Schmidt
Csaba Domokos

Winter Semester 2015/2016

25. Summary

Overview



Markov Random Fields

Graphical models

Probabilistic graphical models encode a joint $p(x, y)$ or conditional $p(y \mid x)$ probability distribution such that given some observations we are provided with a full probability distribution over all feasible solutions.

The graphical models allow us to encode relationships between a set of random variables using a concise language, by means of a graph. Suppose a graph such that for each node a random variable is assigned. The random variables satisfy **conditional independence assumptions** encoded in the graph.

Popular classes of graphical models:

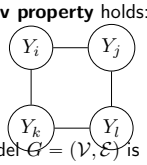
- Undirected graphical models (e.g., Markov random fields)
- Directed graphical models (e.g., Bayesian networks)
- Factor graphs

Markov Random Fields

An undirected graphical model $G = (\mathcal{V}, \mathcal{E})$ is called **Markov Random Field** (MRF) if a variable is conditionally independent of all other variables given its neighbors. In other words, for any node Y_i in the graph, the **local Markov property** holds:

$$p(Y_i \mid Y_{V \setminus \{i\}}) = p(Y_i \mid Y_N(i)) \,,$$

where $N(i)$ are the neighbors of node i in the graph.



A probability distribution $p(Y)$ on an undirected graphical model $G = (\mathcal{V}, \mathcal{E})$ is called **Gibbs distribution** if it can be factorized into potential functions $\psi_C(y_C) > 0$ defined on set of cliques $\mathcal{C}(G)$ that cover all nodes and edges of G:

$$p(Y) = \frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \psi_C(y_C) \,, \quad \text{where} \quad Z = \sum_{y \in \mathcal{Y}} \prod_{C \in \mathcal{C}(G)} \psi_C(y_C) \,.$$

Hammersley-Clifford theorem

Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graphical model. The Hammersley-Clifford theorem tells us that the following are equivalent:

- G is an MRF model
- The joint probability distribution $P(Y)$ on G has Gibbs-distribution.

An MRF defines a family of **joint probability distributions** by means of an undirected graph $G = (\mathcal{V}, \mathcal{E})$, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ (there are no self-edges), where the graph encodes conditional independence assumptions between the random variables corresponding to $\mathcal{V}$.

Since, the potential functions $\psi_c(y_c) > 0$

$$\psi_c(y_c) = \exp(-E_c(y_c)) \quad \Leftrightarrow \quad E_c(y_c) = -\log((\psi_c(y_c))) \,.$$

Inference

Inference means the procedure to estimate the probability distribution, encoded by the graphical model, for a given data (or observation).

Maximum A Posteriori (MAP) inference: Given a factor graph and the observation x, find the *state* $y^* \in \mathcal{Y}$ *of maximum probability*,

$$y^* = \operatorname*{argmax}_{y \in \mathcal{Y}} p(Y = y \mid x) = \operatorname*{argmin}_{y \in \mathcal{Y}} E(y; x) \,.$$

Probabilistic inference: Given a factor graph and the observation x, find the value of the *log partition function* and the *marginal distributions* for each factor,

$$\log Z(x) = \log \sum_{y \in \mathcal{Y}} \exp(-E(y; x)) \,,$$
$$\mu_F(y_F) = p(Y_F = y_F \mid x) \quad \forall F \in \mathcal{F}, \forall y_F \in \mathcal{Y}_F \,.$$

Both inference problems are known to be NP-hard for general graphs.

Parameter learning

Learning graphical models (from training data) is a way to find among a large class of possible models a single one that is *best* in some sense for the task at hand.

We assume a fixed underlying graphical model with **parameterized conditional probability distribution**

$$p(y \mid x, w) = \frac{1}{Z(x,w)} \exp(-E(x,y,w)) = \frac{1}{Z(x,w)} \exp(-\langle w, \varphi(x,y) \rangle) ,$$

where $Z(x,w) = \sum_{y \in \mathcal{Y}} \exp(-\langle w, \varphi(x,y) \rangle)$. The only unknown quantity is the *parameter vector* w, on which the energy $E(x,y,w)$ depends linearly.

Learning tasks

Let $d(y \mid x)$ be the (*unknown*) conditional distribution of labels for a problem to be solved. For a parameterized conditional distribution $p(y \mid x, w)$ with parameters $w \in \mathbb{R}^D$, **probabilistic parameter learning** is the task of finding a point estimate of the parameter w^* that minimizes the **expected dissimilarity** of $p(y \mid x, w^*)$ and $d(y \mid x)$:

$$\mathsf{KL}_{\mathsf{tot}}(p \| d) = \sum_{x \in \mathcal{X}} d(x) \sum_{y \in \mathcal{Y}} d(y \mid x) \log \frac{d(y \mid x)}{p(y \mid x, w)} .$$

Let $d(x,y)$ be the unknown distribution of data in labels, and let $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ be a loss function. **Loss minimizing parameter learning** is the task of finding a parameter value w^* such that the *expected prediction loss*

$$\mathbb{E}_{(x,y) \sim d(x,y)}[\Delta(y, f_p(x))]$$

is as small as possible, where $f_p(x) = \operatorname{argmax}_{y \in \mathcal{Y}} p(y \mid x, w^*)$.

Pseudo Boolean Optimization

Pseudo-Boolean Optimization

A **pseudo-Boolean function** $E : 2^\Omega \to \mathbb{R}$ assigns to every subset $A \subset \Omega$ a real value $E(A)$.

Most Computer Vision problems can be cast as the minimization of a pseudo-Boolean function $E : 2^\Omega \to \mathbb{R}$.

We are interested in the global minimum $\min_{A \subset \Omega} E(A)$ and in one of its *global minimizers* $A \in \operatorname{argmin} E$.

If the computation of a global minimizer is NP-hard, we are also satisfied with an *approximation*. A set $S \subset \Omega$ is called an $(1+\varepsilon)$-**approximation** of $\operatorname{argmin} E$, if the following holds

$$E(S) \leqslant (1+\varepsilon) \cdot \min_{A \subset \Omega} E(A) .$$

Submodularity and Supermodularity

A pseudo-Boolean function $E : 2^\Omega \to \mathbb{R}$ is called **modular** / **submodular** / **supermodular** if

$$E(A \cup B) + E(A \cap B) \begin{matrix} = \\ \leqslant \\ \geqslant \end{matrix} E(A) + E(B) \quad (\text{for all } A, B \in 2^\Omega) .$$

Minimizing an arbitrary submodular functions can be done in polynomial time. The minimization of supermodular functions is NP-hard.

Iff $E : 2^\Omega \to \mathbb{R}$ is a supermodular function, then $-E$ is submodular.

If $H : \mathbb{R} \to \mathbb{R}$ is a concave function, then $E_H : 2^\Omega \to \mathbb{R}$ is submodular with $E_H(A) := H(|A|)$.

Submodularity w.r.t. two variables

Let $E : 2^\Omega \to \mathbb{R}$ be submodular and let $S \in 2^\Omega$ and $i, j \in \Omega \backslash S$. Then

$$E(S + \{i,j\}) + E(S) \leqslant E(S + \{i\}) + E(S + \{j\}) .$$

If we define $E_2 : \mathbb{B} \times \mathbb{B} \to \mathbb{R}$ via $E_2(b_1, b_2) := E(S + b_1 \cdot \{i\} + b_2 \cdot \{j\})$, one can write

$$E_2(1,1) + E_2(0,0) \leqslant E_2(1,0) + E_2(0,1) .$$

Submodularity of MinCut

Let $G = (V, \mathcal{E}, c, s, t)$ be a network, $V_0 := V \backslash \{s,t\}$ and

$$E : \mathcal{P}(V_0) \to \mathbb{R} , \quad A \mapsto \mathsf{Cut}(A \cup \{s\}, V_o \backslash A \cup \{t\}) .$$

Then E is submodular iff $c(e) \geqslant 0$ for all $e \in \mathcal{E}$.

Let $G = (V, \mathcal{E}, c, s, t)$ be a network. Then

$$\max_{f \text{ is flow}} \mathsf{Flow}(f) = \min_{(S,T) \text{ is } s-t \text{ cut}} \mathsf{Cut}(S,T) .$$

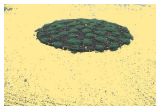
Every graph cut problem can be represented as a submodular energy that uses cliques of size 2 or smaller. The opposite is also true: If

$$E(x) = C + \sum_{i \in \Omega} C_i x_i + \sum_{i,j \in \Omega} C_{ij} x_i x_j$$

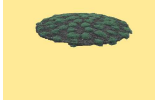
is submodular, the minimization of E can be cast as a graph cut problem.

Binary image segmentation

Given Image



Minimizing Data Term



Minimizing Data + Length Term

$$\operatorname*{argmin}_{A \subset \Omega} E(A) = \operatorname*{argmin}_{A \subset \Omega} \sum_{i \in A} f(i) + \mathsf{length}(A) .$$

Quadratic Pseudo-Boolean Optimization (QPBO)

We assume the case of quadratic pseudo-Boolean energies

$$E(x) = C_0 + \sum_{i=1}^{n} C_i x_i + \sum_{i,j=1}^{n} C_{ij} x_i x_j \ .$$

$C_{ij} < 0$ refer to submodular terms and $C_{ij} > 0$ to supermodular terms.

Let us define the sets $N := \{(i,j) \in \{1,\dots,n\}^2 \mid C_{ij} < 0\}$ and $P := \{(i,j) \in \{1,\dots,n\}^2 \mid C_{ij} > 0\}$. We know that E is submodular (supermodular) iff $|P| = 0$ ($|N| = 0$).

For submodular functions

$$E(x) = C_0 + \sum_{i=1}^{n} C_i x_i + \sum_{(i,j)\in N} C_{ij} x_i x_j$$

the optimization problem can be cast as a MaxFlow problem that can be solved efficiently.

Roof duality

The idea of the QPBO is to reformulate the minimization problem as an ILP and to find an approximative solution. Since QPBO is NP hard, we cannot expect to find the minimal energy. Instead we compute a lower bound of the minimum energy.

Let

$$\mathcal{R}(E) = \left\{ \begin{array}{c} C_0 + \sum_{i=1}^{n} C_i x_i + \\ \sum_{(i,j)\in P} C_{ij}\lambda_{ij}[x_i + x_j - 1] + \\ \sum_{(i,j)\in N} C_{ij}[\lambda_{ij} x_i + (1-\lambda_{ij}) x_j] \end{array} \middle| \ \lambda_{ij} \in [0,1] \right\} .$$

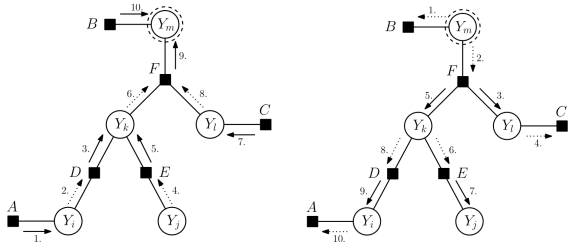
We can define a lower bound for $\min_{x\in\mathbb{B}^n} E(x)$

$$M(E) = \max_{f\in\mathcal{R}(E)} \min_{x\in\mathbb{B}^n} f(x) \leqslant \min_{x\in\mathbb{B}^n} \max_{f\in\mathcal{R}(E)} f(x) = \min_{x\in\mathbb{B}^n} E(x) \ .$$

This lower value is called the **roof dual** of E.

Belief propagation on trees

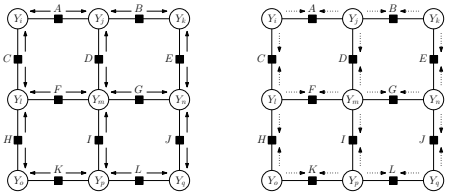
For tree-structured factor graphs there always exist at least one such message that can be computed initially, hence all the dependencies can be resolved.



1. Select one variable node as root of the tree (e.g., Y_m)
2. Compute leaf-to-root messages (e.g., by applying depth-first-search)
3. Compute root-to-leaf messages (reverse order as before)

Message passing in cyclic graphs

When the graph has cycles, then there is no well-defined *leaf–to–root* order. However, one can apply message passing on cyclic graphs, which results in **loopy belief propagation**.



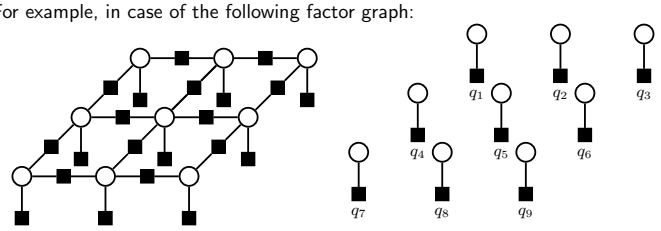
1. Initialize all messages as constant 1
2. Pass factor–to–variables and variables–to–factor messages alternately until convergence
3. Upon convergence, treat **beliefs** μ_F as approximate marginals

Naive mean field

Assume we are given an **intractable** distribution $p(y \mid x)$. We consider an **approximate distribution** $q(y)$, which is tractable, for $p(y \mid x)$.
Take a set Q as the set of all distributions in the form:

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

For example, in case of the following factor graph:



Original factor graph Mean field approximation

Move making approaches

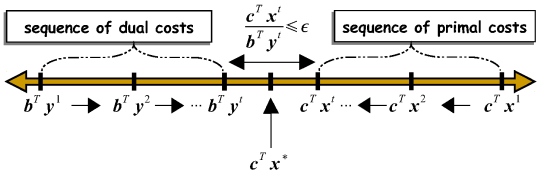
Since graph cuts will always compute the global energy of a submodular energy, one might be interested in formulating a binary submodular sub-problem that can be solved with graph cut.

Such an approach combines the main idea of mean field optimization, i.e. local improvements with the insight that graph cut optimization can change the label of multiple variables at the same time.

We discussed the following three different approaches

- α-expansion allows each variable to either keep its current label or to change it to the label $\alpha \in \mathcal{L}$. As a result, the region of α expands.
- $\alpha-\beta$-swap only changes those pixels that are labeled $\ell \in \{\alpha, \beta\}$. Each of these variables can choose between α and β.
- Fusion Move starts with two different labelings $x, y \in \mathcal{L}^n$. Each variable chooses then for itself either the label from x or y. Both, α-expansion and $\alpha-\beta$-swap can be seen as special cases of the fusion move.

Primal-dual schema



Typically, primal-dual ϵ-approximation algorithms construct a sequence $(\mathbf{x}^k, \mathbf{y}^k)_{k=1,\dots,t}$ of primal and dual solutions until the elements $\mathbf{x}^t$, $\mathbf{y}^t$ of the last pair are both **feasible** and **satisfy the relaxed primal complementary slackness conditions**, hence the condition $\langle \mathbf{c}, \mathbf{x} \rangle \leqslant \epsilon \langle \mathbf{b}, \mathbf{y} \rangle$ will be also fulfilled.

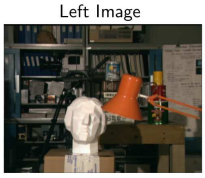
Computer Vision

Image segmentation

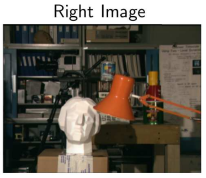
- Binary image segmentation
- Interactive segmentation
- Multi-object segmentation
- Medical image segmentation
- Semantic segmentation
- Video segmentation

Stereo matching

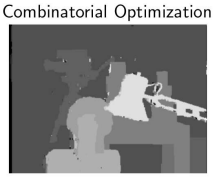
Left Image



Right Image



Combinatorial Optimization



Ground Truth

Optical flow

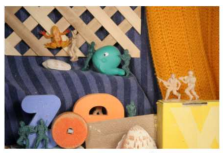
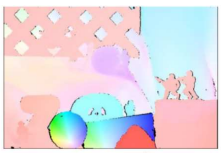


Image $I_0 \colon \Omega \to \mathbb{R}^3$

Image $I_1 \colon \Omega \to \mathbb{R}^3$



Flow $v \colon \Omega \to \mathbb{R}^2$

Given two images I_0 and I_1 of a video, we would like to detect the movements between these two images.

In other words, we are interested in a mapping $v : \Omega \to \mathbb{R}^2$ such that $I_1 \approx I_2(x + v(x))$. The vector field v is called the **optical flow**.

If we quantize $\mathbb{R}^2$, we obtain a finite label space and the optical flow v can be understood as a multilabeling of Ω.

Other applications

- Human pose estimation
- Image denoising
- Multi-camera reconstruction
- Object detection

Computer Vision *

We are always looking for master and bachelor students!

3D Reconstruction

Optical Flow

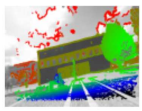
Shape Analysis

Robot Vision

RGB-D Vision

Image Segmentation

Convex Relaxation



Visual SLAM

Please fill out the application form: `https://vision.in.tum.de/application`