A tutorial on optimization with graph cuts
With applications in image and mesh processing and computer vision

Christian Wolf

LIRIS UMR 5205 CNRS
Université de Lyon
INSA-Lyon, bâtiment J. Verne
20, Avenue Albert Einstein - 69622 Villeurbanne cedex
http://liris.cnrs.fr

March 30th 2010
Plan

1. Introduction
2. 2-labels, exact : segmentation
3. M-labels, approximate : segmentation, motion, stereo, matching
4. M-labels, exact : motion, segmentation
5. QPBO : remeshing
# Plan

1. **Introduction**

2. 2-labels, exact : segmentation

3. M-labels, approximate : segmentation, motion, stero, matching

4. M-labels, exact : motion, segmentation

5. QPBO : remeshing
What can graph cuts do?

Take several classification decisions jointly, i.e.:
Given a set of a discrete valued variables (pixels, edgels, object parts, abstract variables...) :

Assign a discrete value to each of the variables taking into account dependencies between the variables
⇒ Minimize a global function on all values
Applications

<table>
<thead>
<tr>
<th>Application</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image segmentation/restoration</td>
<td>pixel labels</td>
</tr>
<tr>
<td>Background subtraction in video</td>
<td>pixel labels</td>
</tr>
<tr>
<td>Depth from stereo</td>
<td>pixel displacements</td>
</tr>
<tr>
<td>Motion/optical flow</td>
<td>motion vectors</td>
</tr>
<tr>
<td>Object detection &amp; matching</td>
<td>interest points</td>
</tr>
<tr>
<td>Mesh segmentation</td>
<td>vertex labels</td>
</tr>
<tr>
<td>Remeshing</td>
<td>vertex positions / edge labels</td>
</tr>
<tr>
<td>Viewpoint recognition</td>
<td>angles, positions etc.</td>
</tr>
<tr>
<td>Image/video indexing</td>
<td>regions/objects/interest points</td>
</tr>
</tbody>
</table>

...
A toy problem:

\[ X \] indicates a discrete variable, e.g. \( X \in \{Cat, Dog\} \)

\[ Y \] indicates an associated measure, e.g. \( Y \in (0, 100] \), measured in kg (a weight)

Problem: decide whether \( X \) is cat or a dog, based on the associated measure \( Y \)

\[ \Rightarrow \] needs a “model” of the weights of cats and dogs.
Cats & dogs: taking discrete decisions (2)

Possibility 1: thresholding

\[ X = \begin{cases} 
\text{Cat} & \text{if } Y < T \\
\text{Dog} & \text{else} 
\end{cases} \]
Possibility 2 : probabilistic model

Weights of Cats and Dogs are normally distributed, with different means and variances:

\[
p(Y|X = \text{Cat}) \sim \mathcal{N}(\mu_c, \sigma_c) \\
p(Y|X = \text{Dog}) \sim \mathcal{N}(\mu_d, \sigma_d)
\]

Maximum likelihood decision:

\[
X = \arg \max_X P(Y|X)
\]
Possibility 3: a classifier, (MLP, SVM, k-nn, decision tree etc.)

Most classifiers directly model a discriminative decision function, whose parameters (weights/support vectors etc.) are learned from training data.

Graphical notation of the problem

Independent Not independent Not independent
Generative Discriminative

Examples

Gaussian noise  ANN, SVM, k-NN

Attention! These are dependency graphs, not st-graphs used to calculate a min-cut/max-flow!
A set of independent variables

New problem: a whole bunch of cats and dogs, each associated with a weight!

\[Y_0, Y_1, Y_2, Y_3, Y_4\]

\[X_0, X_1, X_2, X_3, X_4\]

How do we take the decisions ("inference")?

\[\Rightarrow\] variablewise!
Idea: line them up into a queue with two narrow fences. Hypotheses: cats tend to cluster together and do not like to be near dogs, and vice versa.

\[
\begin{align*}
Y_0 & \quad Y_1 & \quad Y_2 & \quad Y_3 & \quad Y_4 \\
X_0 & \quad X_1 & \quad X_2 & \quad X_3 & \quad X_4
\end{align*}
\]

\[\Rightarrow\] Neighboring animals tend to be of the same species
Variables in a linear order (chain structured)

We first revisit the simple case of single node (one animal):

\[
p(Y|X = \text{Cat}) \sim \mathcal{N}(\mu_c, \sigma_c) \\
p(Y|X = \text{Dog}) \sim \mathcal{N}(\mu_d, \sigma_d)
\]

Maximizing \(P(Y|X)\) over \(X \in \{\text{Cat, Dog}\}\) can be rewritten as:

\[
E_d(X, Y) = \begin{cases} 
\frac{(Y-\mu_c)^2}{\sigma_c} & \text{if } X = \text{Cat} \\
\frac{(Y-\mu_d)^2}{\sigma_d} & \text{if } X = \text{Dog}
\end{cases}
\]

and minimize \(E_d(X, Y)\) over \(X \in \{\text{Cat, Dog}\}\) \(\implies\) **Energy function**
Variables in a linear order (chain structured)

Idea

Create an energy function over all the $X$ and all the $Y$ which handles the dependence of the weights on the animals, as well as the spatial relationships.

$$E(X, Y) = E(X_0, X_1, \ldots, X_N, Y_0, Y_1, \ldots, Y_N)$$
An “unnecessary” global energy function

Revisiting the case of independent measurements:

\[ E(X, Y) = \sum_i E_d(X_i, Y_i) \]

Minimizing the energy function means minimizing each term separately!

\[ \hat{X} = \arg \min_X E(X, Y) = \bigcup_i \arg \min_{X_i} E_d(X_i, Y_i) \]
Variables in a linear order (chain structured)

Combine a data attached term and a regularizing term:

\[ E(X, Y) = \alpha \sum_{i=0}^{N} E_d(X_i, Y_i) + \beta \sum_{i=1}^{N} E_r(X_i, X_{i-1}) \]

where

\[ E_r(a, b) = 1 - \delta_{a,b} = \begin{cases} 
0 & \text{if } a = b \\
1 & \text{else}
\end{cases} \]
Variables in a linear order (chain structured)

Minimization?

No variablewise minimization!

⇒ Dynamic programming

\[
\min_{X_0, \ldots, X_4} E_d(X_0, Y_0) + E_r(X_0, X_1) + E_d(X_1, Y_1) + E_r(X_1, X_2) + E_d(X_2, Y_2)
\]

\[
= \min_{X_0} \left[ E_d(X_0, Y_0) + \min_{X_1} \left[ E_r(X_0, X_1) + E_d(X_1, Y_1) + \right. \right. \\
\left. \left. E_r(X_1, X_2) + E_d(X_2, Y_2) \right] \right] \\
+ \min_{X_2} \left[ R(X_1) \right]
\]
Plan

1. Introduction

2. 2-labels, exact : segmentation

3. M-labels, approximate : segmentation, motion, stereo, matching

4. M-labels, exact : motion, segmentation

5. QPBO : remeshing
What can graph cuts do?

Dynamic programming can be extended to trees, but not to arbitrary graphs (DAGs or graphs with cycles)

In general NP-Complete!

$$E(X, Y) = \alpha \sum_i E_1(X_i, Y_i) + \beta \sum_{i \sim j}^N E_2(X_i, X_j) + \ldots$$
Example: image binarization

\[ E(X, Y) = \alpha \sum_i E_d(X_i, Y_i) + \beta \sum_{i \sim j} E_r(X_i, X_j) \]

\((i \sim j \implies i \text{ and } j \text{ are neighbors})\)

Minimizing \(E(X, Y)\) generally beats pixelwise binarization + post processing!
Application: background subtraction in video (1)

[Howe and Deschamps, 2004]

Input image
Groundtruth
Raw difference
Morphology-1
Graph cuts
Morphology-2
Application: background subtraction in video (2)

The decisions are taken globally for a whole section of the spatio-temporal cube  [Under review]
Application: background subtraction in video

The decisions are taken globally for a whole section of the spatio-temporal cube:

$$E(X, Y) = \alpha_d \sum_i E_d(X_i, Y_i) + \alpha_s \sum_{i \sim j} \delta(X_i, X_j) + \alpha_t \sum_i \delta(X_i, X_{i \rightarrow U_i})$$

- $E_d(X_i, Y_i)$ represents the GMM background model.
- $\delta(X_i, X_j)$ represents the spatial regularizer.
- $\delta(X_i, X_{i \rightarrow U_i})$ represents the temporal regularizer.

$U_i$ is a motion vector calculated at position $i$ with optical flow.
1 Variable, 2 labels, first order term

Toy example: construction of an \(st\)-graph for a single 1\(st\)-order term (two labels, i.e. \(X \in \{0, 1\}\)):

\[
E = E(X, Y)
\]

Interpretation of the cut

- Node \(X\) is connected with \(s \implies X = 1\)
- Node \(X\) is connected with \(t \implies X = 0\)
Two independent hidden variables

\[ E = E(X_0, Y_0) + E(X_1, Y_1) \]

\[ \Rightarrow \] Independent minimization!
Two independent hidden variables  [Boykov et al., 2001]

\[ E = \alpha \left[ E_1(X_0, Y_0) + E_1(X_1, Y_1) \right] + \beta E_2(X_1, X_2) \]

\[ E_2(a, b) = 1 - \delta_{a,b} = \begin{cases} 
0 & \text{if } a = b \\
1 & \text{else}
\end{cases} \]

\[ \alpha E_2(X_0=0, Y_0) \quad \alpha E_2(X_0=1, Y_0) \]
\[ \alpha E_2(X_1=0, Y_1) \quad \alpha E_2(X_1=1, Y_1) \]

\[ \beta \]

\[ t \]

\[ s \]

\[ X_0 \quad X_1 \]

\[ Y_0 \quad Y_1 \]

\[ \Rightarrow \] Having the variables \( X_i \) in different terminal costs \( \beta \)!

Calculates the exact solution!
Submodular functions

\[ \alpha E_2(X_0=0, Y_0) \]
\[ \alpha E_2(X_1=0, Y_1) \]
\[ \alpha E_2(X_0=1, Y_0) \]
\[ \alpha E_2(X_1=1, Y_1) \]

Caveat
Not all second order terms can be minimized!!

Submodularity criterion

\[ E_2(0, 0) + E_2(1, 1) \leq E_2(0, 1) + E_2(1, 0) \]

[Kolmogorov and Zabih, 2004]
More efficient graph construction

[Boykov et al. 2001]

[Kolmogorov et al., 2004]
More efficient graph construction

[First name Last name et al., Year]

![Graph and table]

\[ E^{i,j} = \begin{bmatrix} E^{i,j}(0, 0) & E^{i,j}(0, 1) \\ E^{i,j}(1, 0) & E^{i,j}(1, 1) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \]

**TABLE 2**

\[
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix} = A + \begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix} + \begin{bmatrix}
0 & D - C \\
0 & D - C
\end{bmatrix} + \begin{bmatrix}
0 & B + C - A - D \\
0 & 0
\end{bmatrix}
\]

Fewer edges \(\rightarrow\) more efficient!
Plan

1. Introduction

2. 2-labels, exact : segmentation

3. M-labels, approximate : segmentation, motion, stereo, matching

4. M-labels, exact : motion, segmentation

5. QPBO : remeshing
Most problems involve multiple labels, $X \in \{0, 1, \ldots, L-1\}$:

$$E(X, Y) = \alpha \sum_i E_1(X_i, Y_i) + \beta \sum_{i \sim j} E_2(X_i, X_j) + \ldots$$

Solving with graph cuts needs a new way to interpret a cut, since only 2 terminal nodes $(s, t)$ are possible.
Example: document restoration

Top to bottom: k-means, graph cuts, separate graph cuts par document side [Wolf 2010]
Mesh segmentation (No graph cuts $\Rightarrow$ sim. annealing)

\[ E(X, Y) = \sum_i E_0(X_i) + \alpha \sum_i E(X_i, Y_i) + \beta \sum_{i \sim j} \delta_{X_i, X_j} \]

+ Unsupervised parameter estimation

[Lavoué and Wolf, EUROGRAPHICS-W3DOR 2008]
Concept

- Each pixel can either keep its value (→ terminal $s$) or change to new value $\alpha$ (→ terminal $t$).
- Iterate and change $\alpha$
- Allow many pixels to change their labels simultaneously
Application: depth from stereo

Disparity maps $D$ are calculated from image pairs $I, J$ [Zhang and Seitz, 2007]

$$E(D, I, J) = \sum_{i} \min(|I(x_i, y_i) - J(x_i - D_i, y_i)|, \sigma) + \sum_{i \sim j} \min(|D_i - D_j|, \tau)$$

First order terms alone are pretty much useless in this application!
Application: optical flow (1)

[Boykov et al., 2001]

Approximate solution!
Application requirements: objects can be slightly deformed

- Distances are preserved (or not too different) in a local neighborhood
- Rotation angles are coherent (not too different) in a local neighborhood
Graphmatching

MODEL GRAPH

SCENE GRAPH

Christian Wolf - http://liris.cnrs.fr/christian.wolf
A tutorial on optimization with graph cuts
Graphmatching

\[ E(X) = \lambda_1 \sum_i \text{Feature distance } i \leftrightarrow X_i \]

\[ + \lambda_2 \sum_{i \sim j} \text{Coherence in Euclidean distance } i \leftrightarrow j, X_i \leftrightarrow X_j \]

\[ + \lambda_3 \sum_{i \sim j} \text{Path length } X_i \leftrightarrow X_j \]

\[ + \lambda_4 \sum_{i \sim j} \text{Coherence in rotation angle } i \leftrightarrow X_i, j \leftrightarrow X_j \]
Graphmatching

[Torresani, Kolmogorov and Rother, 2008]

\[
E(X) = \lambda_1 \sum_i \text{Feature distance } i \leftrightarrow X_i \\
+ \lambda_2 \sum_{i \sim j} \text{Coherence in Euclidean distance } i \leftrightarrow j, X_i \leftrightarrow X_j
\]

Subject to unicity constraints on the destination labels.
Plan

1. Introduction
2. 2-labels, exact : segmentation
3. M-labels, approximate : segmentation, motion, stereo, matching
4. M-labels, exact : motion, segmentation
5. QPBO : remeshing
Application: optical flow (2)

Try to calculate an exact solution instead of approximate solution

[Under review] :

\[
E(U, Y) = \alpha_t \sum_i \min_{a \in [1,T]} |Y_i - Y_{i \rightarrow [U_{i,x}]^a}|
+ \alpha_t \sum_i \min_{a \in [1,T]} |Y_i - Y_{i \rightarrow [U_{i,y}]^a}|
+ \alpha_0 \sum_i |U_{i,x}| + \alpha_0 \sum_i |U_{i,y}|
+ \alpha_m \sum_{i \sim j} |U_{i,x} - U_{j,x}| + |U_{i,y} - U_{j,y}|
\]

Properties of \( E \):

- Multiple Labels with linear order
- Terms are convex in label differences
Graph cuts minimization of convex functions

Convexity in label differences

\[ E_2(x_i, x_j) = g(x_i - x_j), \quad g(\cdot) \text{ convex} \]

[Ishikawa, 2003]
M-label, submodular terms

[Darbon, 2009] (Similar development as the convex case)
M-label, submodular terms

[Darbon, 2009]

Input image
Degraded image
Exact solution
Median filtered
Expansion move

115sec on a 3 GHz CPU...
Plan

1. Introduction
2. 2-labels, exact: segmentation
3. M-labels, approximate: segmentation, motion, stereo, matching
4. M-labels, exact: motion, segmentation
5. QPBO: remeshing
Objectives & Challenges

- Objective: to change the mesh geometry in order to create more regular and “neat” triangles
- Change as little as possible the approximated surface

Idea: sample candidate positions and take a global decision over the whole mesh
Remeshing: energy function

\[ E(X, Y) = \alpha \sum_i E_d(X_i, Y) + \beta \sum_{i,j,k \in \nabla} E_s(X_i, X_j, X_k) \]

\[ E_d(X_i, Y) = \text{distance of } X_i \text{ to the original surface } (Y) \]

\[ E_s(X_i, X_j, X_k) = \frac{\text{circumradius}(X_i, X_j, X_k)}{\min(||X_i - X_j||, ||X_i - X_k||, ||X_j - X_k||)} \]

Non-submodular 3\textsuperscript{rd} order terms!
QPBO: quadratic pseudo boolean optimization

[Hammer et al., 1984] [Boros et al. 1991]
[Kolmogorov and Rother, 2007]

Properties

- Expansion move: never decreases the energy
- Invariant to label orderings (submodular criterion!)
- May leave nodes unlabelled
Remeshing results

[Vidal et al., 2009] + [Under review]
## Introduction

- **2-labels, exact**: segmentation
- **M-labels, approx.**: motion, stereo, matching
- **M-labels, exact**: motion, segmentation
- **QPBO**: remeshing

## Conclusion

<table>
<thead>
<tr>
<th>Clique size (\Lambda)</th>
<th>2(^{nd}) order</th>
<th>3(^{rd}) order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Lambda = 2, \text{ subm.})</td>
<td>Exact: [Kolmogorov and Zabih, 2004], PAMI</td>
<td></td>
</tr>
<tr>
<td>(\Lambda &gt; 2, \text{ subm.})</td>
<td>Approximative via (\alpha)-expansion move: [Boykov et al., 2001], PAMI; More efficiently: [Kolmogorov and Zabih, 2004], PAMI</td>
<td>Exact (but non-polynomial in the number of labels): [Darbon, 2009], DAM</td>
</tr>
<tr>
<td>(\Lambda = 2, \text{ general})</td>
<td>Exact on a <strong>subset</strong> of labels: [Kolmogorov and Rother, 2007], PAMI; earlier: [Borros et al., 1991] [Hammer et al., 1984]</td>
<td></td>
</tr>
<tr>
<td>(\Lambda &gt; 2, \text{ conv.lab.}) (\Delta)</td>
<td>[Ishikawa, 2003], PAMI</td>
<td></td>
</tr>
<tr>
<td>Cont.+subm.</td>
<td>Appr.exp.merge: [Lempitsky et al., 2008], ECCV</td>
<td></td>
</tr>
</tbody>
</table>
Network flows and minimization of quadratic pseudo-boolean functions. 

Fast approximate energy minimization via graph cuts. 

Global optimization for first order markov random fields with submodular priors. 
*Discrete Applied Mathematics (to appear).*
References II


