Recap: How to Set the Potentials?

**Unary potentials**
- E.g. color model, modeled with a Mixture of Gaussians
  \[ \phi(x_i, y_i; \theta) = \log \sum_k \theta_k \phi(x_i, k) p(k | x_i) N(y_i; \mu_k, \Sigma_k) \]
  
  \( \Rightarrow \) Learn color distributions for each label

- Pairwise potentials
  - **Potts Model**
    \[ \psi(x_i, x_j; \theta_{ij}) = \theta_{ij} \delta(x_i \neq x_j) \]
    - Simplest discontinuity preserving model.
    - Discontinuities between any pair of labels are penalized equally.
    - Useful when labels are unordered or number of labels is small.
  - Extension: "contrast sensitive Potts model"
    \[ \psi(x_i, x_j, g_{ij}(y); \theta_{ij}) = \theta_{ij} g_{ij}(y) \delta(x_i \neq x_j) \]
    where
    \[ g_{ij}(y) = e^{A \cdot y} \cdot \beta = 2 \cdot \text{avg} \left[ |y_i - y_j| \right] \]
    - Discourages label changes except in places where there is also a large change in the observations.

Recap: MRF Structure for Images

- Basic structure
  - Noisy observations
  - “True” image content

- Two components
  - Observation model
    - How likely is that node \( x_i \) has label \( L_i \), given observation \( y_i \)?
    - This relationship is usually learned from training data.
  - Neighborhood relations
    - Simplest case: 4-neighborhood
    - Serve as smoothing terms.
    \( \Rightarrow \) Discourage neighboring pixels to have different labels.
    - This can either be learned or be set to fixed “penalties”.

Recap: Energy Formulation

- Energy function
  \[ E(x, y) = \sum_i \phi(x_i, y_i) + \sum_{i,j} \psi(x_i, x_j) \]
  
  - Single-node (unary) potentials \( \phi \)
    - Encode local information about the given pixel/patch.
    - How likely is a pixel/patch to belong to a certain class (e.g. foreground/background)?
  
  - Pairwise potentials \( \psi \)
    - Encode neighborhood information.
    - How different is a pixel/patch’s label from that of its neighbor? (e.g. based on intensity/color/texture difference, edges)

Course Outline

- Regression Techniques
  - Linear Regression
  - Regularization (Ridge, Lasso)
  - Kernels (Kernel Ridge Regression)
- Deep Reinforcement Learning
- Probabilistic Graphical Models
  - Bayesian Networks
  - Markov Random Fields
  - Inference (exact & approximate)
- Deep Generative Models
  - Generative Adversarial Networks
  - Variational Autoencoders
**Extension: Conditional Random Fields (CRF)**

- Idea: Model conditional instead of joint probability

  - Pairwise potential \(\phi(D|x_i, x_j)\)
  - Unary potential \(\phi(D|x_i)\)

- Energy formulation

  \[
  E(x) = \sum_{i \in I} \phi(D|x_i) + \sum_{j \in N} \phi(D|x_i, x_j) + \phi(x_i, x_j) + \text{const}
  \]

- Pixels
- Labels
- Prior Potts model

**Energy Minimization**

- Goal:
  - Infer the optimal labeling of the MRF.

- Many inference algorithms are available, e.g.
  - Simulated annealing
  - Iterated conditional modes (ICM)
  - Belief propagation
  - Graph cuts
  - Variational methods
  - Monte Carlo sampling

- Recently, Graph Cuts have become a popular tool
  - Only suitable for a certain class of energy functions.
  - But the solution can be obtained very fast for typical vision problems (~1 MPixel/sec).

**Graph Cuts – Basic Idea**

- Construct a graph such that:
  1. Any st-cut corresponds to an assignment of \(x\)
  2. The cost of the cut is equal to the energy of \(x\): \(E(x)\)

\[
E(x) \rightarrow \text{st-mincut} \rightarrow \text{Solution}
\]

**Graph Cuts for Binary Problems**

- Idea: convert MRF into source-sink graph

- Minimum cost cut can be computed in polynomial time (max-flow/min-cut algorithms)

**Example: CRF for Image Segmentation**

- CRF structure
  - Unary potential \(\phi(D|x_i)\)
  - Pairwise terms

- Pixels
- Labels
- Prior Potts model

**Topics of This Lecture**

- Solving MRFs with Graph Cuts
  - Graph cuts for image segmentation
  - s-t mincut algorithm
  - Graph construction
  - Extension to non-binary case
  - Applications
Simple Example of Energy

\[
E(L) = \sum_{p} D_p(L_p) + \sum_{pq \in N} w_{pq} \delta(L_p \neq L_q)
\]

pairwise potentials

t-links

unary potentials

\( L_p \in \{s,t\} \)

(binary object segmentation)

Adding Regional Properties

Regional bias example

Suppose \( I' \) and \( I'' \) are given "expected" intensities of object and background

\[
D_p(s) \propto \exp \left( -\frac{\|I_p - I'\|^2}{2\sigma^2} \right)
\]

\[
D_p(t) \propto \exp \left( -\frac{\|I_p - I''\|^2}{2\sigma^2} \right)
\]

NOTE: hard constrains are not required, in general.

EM-style optimization

Topics of This Lecture

• Solving MRFs with Graph Cuts
  - Graph cuts for image segmentation
  - s-t mincut algorithm
  - Graph construction
  - Extension to non-binary case
  - Applications

How Does it Work? The s-t-Mincut Problem

Graph \( (V, E, C) \)

Vertices \( V = \{v_1, v_2, ..., v_n\} \)

Edges \( E = \{(v_i, v_j) \} \)

Costs \( C = \{c_{ij} \} \)
The s-t-Mincut Problem

What is an st-cut?

An st-cut (S, T) divides the nodes between source and sink.

What is the cost of a st-cut?

Sum of cost of all edges going from S to T

What is the st-mincut?

st-cut with the minimum cost

Source

2

Sink

9

V1

V2

5

2

4

9 + 2 + 9 = 16

Slide credit: Pushmeet Kohli

How to Compute the st-Mincut?

Solve the dual maximum flow problem

Compute the maximum flow between Source and Sink

Constraints

Edges: Flow < Capacity

Nodes: Flow in = Flow out

Min-cut/Max-flow Theorem

In every network, the maximum flow equals the cost of the st-mincut

Source

2

Sink

9

V1

V2

5

2

4

Slide credit: Pushmeet Kohli

Maxflow Algorithms

Flow = 0

Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity

2. Push maximum possible flow through this path

3. Adjust the capacity of the used edges

4. Repeat until no path can be found

Algorithms assume non-negative capacity

Source

2

Sink

9

V1

V2

5

2

4

Slide credit: Pushmeet Kohli
Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Adjust the capacity of the used edges
4. Repeat until no path can be found

Algorithms assume non-negative capacity
Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Adjust the capacity of the used edges
4. Repeat until no path can be found

Algorithms assume non-negative capacity

Flow = 6

Flow = 6 + 1

Flow = 7

Applications: Maxflow in Computer Vision

• Specialized algorithms for vision problems
  – Grid graphs
  – Low connectivity (m \sim O(n))

• Dual search tree augmenting path algorithm
  [Boykov and Kolmogorov PAMI 2004]
  – Finds approximate shortest augmenting paths efficiently.
  – High worst-case time complexity.
  – Empirically outperforms other algorithms on vision problems.
  – Efficient code available on the web
    http://pub.ist.ac.at/~vnk/software.html

• s-t graph cuts can only globally minimize binary energies that are submodular.

\[ E(L) = \sum_{p \in \text{t-links}} E_p(L_p) + \sum_{n \in \text{n-links}} E(L_p, L_q) \quad L_p \in \{s,t\} \]

\[ E(L) \text{ can be minimized by } s-t \text{ graph cuts } \iff E(s,t) + E(t,s) \leq E(s,t) + \min E(t,s) \]

Submodularity ("convexity")

• Submodularity is the discrete equivalent to convexity.
  – Implies that every local energy minimum is a global minimum.
  ⇒ Solution will be globally optimal.

\[ E(L) \text{ can be minimized by } s-t \text{ graph cuts } \iff E(s,t) + E(t,s) \leq E(s,t) + \min E(t,s) \]

\[ \text{Submodularity ("convexity")} \]
Topics of This Lecture

• Solving MRFs with Graph Cuts
  − Graph cuts for image segmentation
  − s-t mincut algorithm
  − Graph construction
  − Extension to non-binary case
  − Applications

Example: Graph Construction

\[ E(a_1, a_2) = 2a_1 \]

Source (0)

\[ a_1 \quad a_2 \]

Sink (1)

Example: Graph Construction

\[ E(a_1, a_2) = 2a_1 + 5B_1 \]

Source (0)

\[ a_1 \quad a_2 \]

Sink (1)

Example: Graph Construction

\[ E(a_1, a_2) = 2a_1 + 5a_1 + 9a_2 + 4a_2 \]

Source (0)

\[ a_1 \quad a_2 \]

Sink (1)

Example: Graph Construction

\[ E(a_1, a_2) = 2a_1 + 5a_1 + 9a_2 + 4a_2 + a_1a_2 \]

Source (0)

\[ a_1 \quad a_2 \]

Sink (1)
**Example: Graph Construction**

\[ E(a_1, a_2) = 2a_1 + 5a_1 + 9a_2 + 4a_2 + a_1a_2 + 2a_1a_2 \]

**How Does the Code Look Like?**

```c
Graph *g;

for all pixels p
    /* Add a node to the graph */
    nodeID(p) = g->add_node();
    /* Set cost of terminal edges */
    set_weights(nodeID(p), fgCost(p), bgCost(p));
end

for all adjacent pixels p,q
    add_weights(nodeID(p), nodeID(q), cost);
end

g->compute_maxflow();

label_p = g->is_connected_to_source(nodeID(p));
// is the label of pixel p (0 or 1)
```

**Graph Construction**

\[ E(a_1, a_2) = 2a_1 + 5a_1 + 9a_2 + 4a_2 + a_1a_2 + 2a_1a_2 \]

**How Does the Code Look Like?**

```c
Graph *g;

for all pixels p
    /* Add a node to the graph */
    nodeID(p) = g->add_node();
    /* Set cost of terminal edges */
    set_weights(nodeID(p), fgCost(p), bgCost(p));
end

for all adjacent pixels p,q
    add_weights(nodeID(p), nodeID(q), cost);
end

g->compute_maxflow();

label_p = g->is_connected_to_source(nodeID(p));
// is the label of pixel p (0 or 1)
```
Topics of This Lecture

- Solving MRFs with Graph Cuts
  - Graph cuts for image segmentation
  - s-t mincut algorithm
  - Graph construction
  - Extension to non-binary case
  - Applications

Dealing with Non-Binary Cases

- Limitation to binary energies is often a nuisance.
  - E.g. binary segmentation only...
  - We would like to solve also multi-label problems.
    - The bad news: Problem is NP-hard with 3 or more labels!
  - There exist some approximation algorithms which extend graph cuts to the multi-label case:
    - $\alpha$-Expansion
    - $\alpha\beta$-Swap
    - They are no longer guaranteed to return the globally optimal result.
      - But $\alpha$-Expansion has a guaranteed approximation quality (2-approx) and converges in a few iterations.

$\alpha$-Expansion Algorithm

1. Start with any initial solution
2. For each label "$\alpha$" in any (e.g. random) order:
   1. Compute optimal $\alpha$-expansion move (s-t graph cuts).
   2. Decline the move if there is no energy decrease.
3. Stop when no expansion move would decrease energy.
Example: Stereo Vision

Original pair of “stereo” images

Depth map

\(\alpha\)-Expansion Moves

- In each \(\alpha\)-expansion a given label “\(\alpha\)” grabs space from other labels

For each move, we choose the expansion that gives the largest decrease in the energy: \(\Rightarrow\) binary optimization problem

Topics of This Lecture

- Solving MRFs with Graph Cuts
  - Graph cuts for image segmentation
  - \(s-t\) mincut algorithm
  - Extension to non-binary case
  - Applications

GraphCut Applications: "GrabCut"

- Interactive Image Segmentation [Boykov & Jolly, ICCV’01]
  - Rough region cues sufficient
  - Segmentation boundary can be extracted from edges
- Procedure
  - User marks foreground and background regions with a brush.
  - This is used to create an initial segmentation which can then be corrected by additional brush strokes.

GrabCut: Data Model

- Obtained from interactive user input
  - User marks foreground and background regions with a brush
  - Alternatively, user can specify a bounding box

Iterated Graph Cuts

- Color model (Mixture of Gaussians)

Result
GrabCut: Example Results

This is included in all MS Office versions since 2010!

Applications: Interactive 3D Segmentation

References and Further Reading

• A gentle introduction to Graph Cuts can be found in the following paper:

• Try the GraphCut implementation at
  http://pub.ist.ac.at/~vnk/software.html