

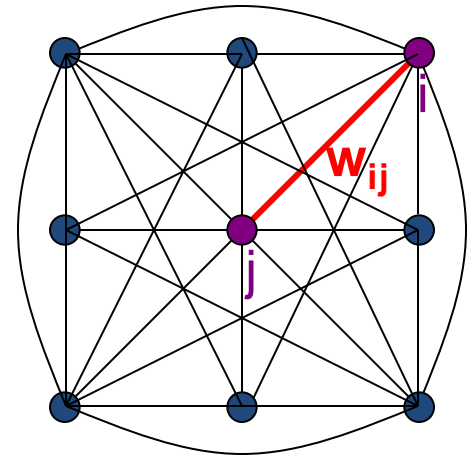
MRFs and Segmentation with Graph Cuts

Computer Vision
CS 543 / ECE 549
University of Illinois

Derek Hoiem

Today's class

- Finish up EM
- MRFs
- Segmentation with Graph Cuts



EM Algorithm: Recap

1. E-step: compute

$$E_{z|x, \theta^{(t)}} [\log(p(\mathbf{x}, \mathbf{z} | \theta))] = \sum_{\mathbf{z}} \log(p(\mathbf{x}, \mathbf{z} | \theta)) p(\mathbf{z} | \mathbf{x}, \theta^{(t)})$$

2. M-step: solve

$$\theta^{(t+1)} = \operatorname{argmax}_{\theta} \sum_{\mathbf{z}} \log(p(\mathbf{x}, \mathbf{z} | \theta)) p(\mathbf{z} | \mathbf{x}, \theta^{(t)})$$

- Determines hidden variable assignments and parameters that maximize likelihood of observed data
- Improves likelihood at each step (reaches local maximum)
- Derivation is tricky but implementation is easy

EM Demos

- Mixture of Gaussian demo
- Simple segmentation demo

“Hard EM”

- Same as EM except compute z^* as most likely values for hidden variables
- K-means is an example
- Advantages
 - Simpler: can be applied when cannot derive EM
 - Sometimes works better if you want to make hard predictions at the end
- But
 - Generally, pdf parameters are not as accurate as EM

Missing Data Problems: Outliers

You want to train an algorithm to predict whether a photograph is attractive. You collect annotations from Mechanical Turk. Some annotators try to give accurate ratings, but others answer randomly.

Challenge: Determine which people to trust and the average rating by accurate annotators.



Annotator
Ratings

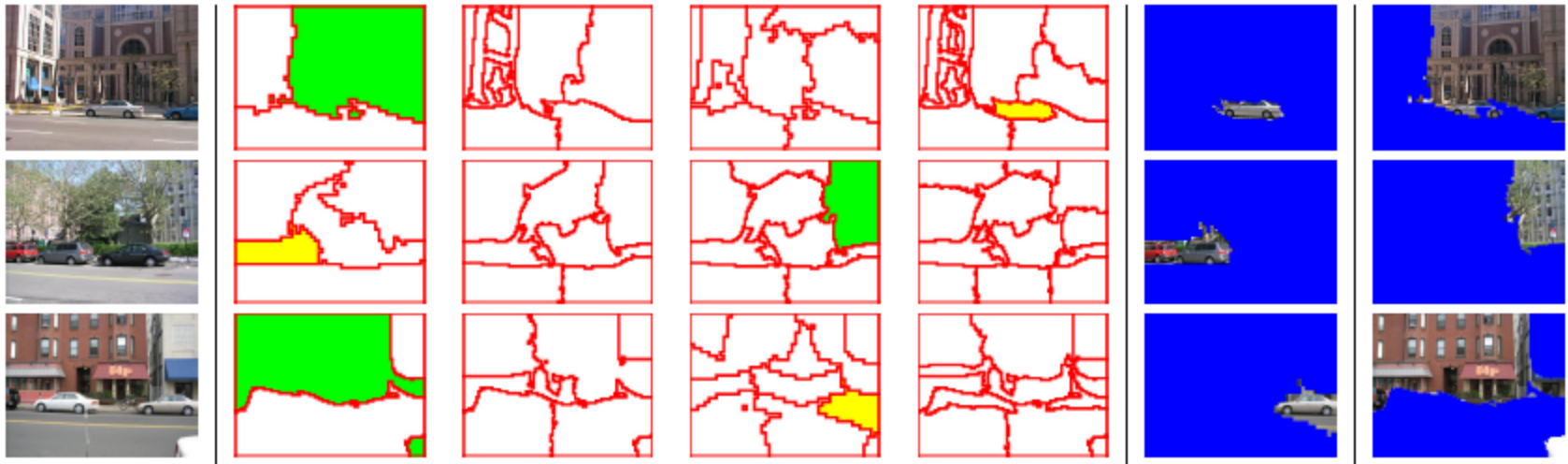
10
8
9
2
8

Photo: Jam343 (Flickr)

Missing Data Problems: Object Discovery

You have a collection of images and have extracted regions from them. Each is represented by a histogram of “visual words”.

Challenge: Discover frequently occurring object categories, without pre-trained appearance models.



3 EM - Mixture of Multinomials (15%)

Probabilistic mixture models are useful in a variety of applications, such as gaussian mixture models for segmentation (see problem 4). Multinomial distributions are another useful distribution for mixture models, and can be used to model the bag-of-words representation seen in the previous problem: for a given texture i , codeword j occurs with probability θ_{ij} .

A mixture of multinomials would allow modeling images that are composed of multiple textures, each defined by Θ_i , where texture i occurs with probability π_i . More sophisticated methods such as pLSA and LDA replace τ with a per-image distribution over textures classes, which must be inferred. Again, note that this was originally introduced for representing documents composed of multiple “topics.”

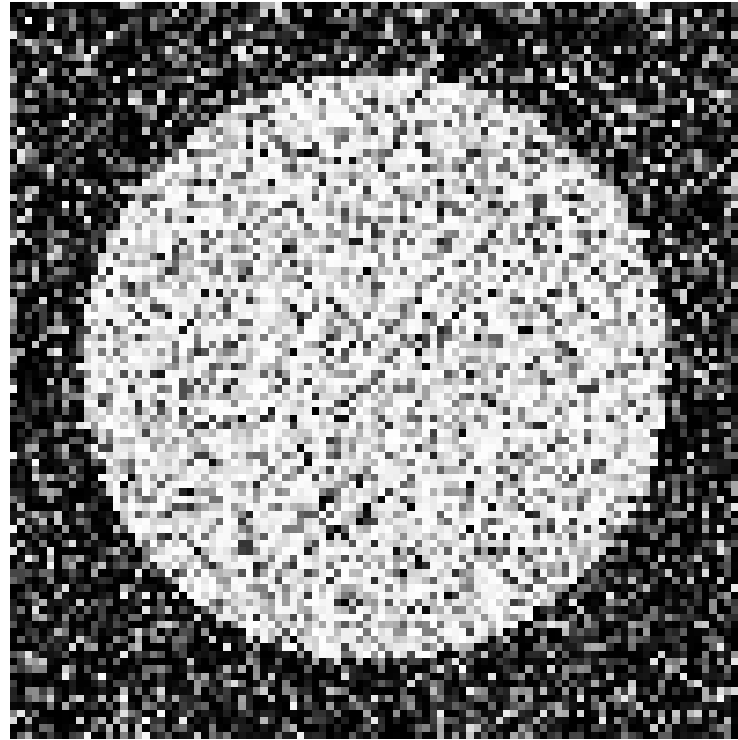
Derive the EM algorithm for the following multinomial mixture model for n examples $\{x_i\}$:

$$P(\mathbf{x}|\{\Theta_i\}, \{\pi_i\}) = \sum_i \pi_i P(\mathbf{x}|\Theta_i), \text{ s.t. } \sum_i \pi_i = 1, 0 \leq \pi_i \leq 1$$
$$P(\mathbf{x}|\Theta_i) = \frac{n!}{\prod_j x_j!} \prod_j \theta_{ij}^{x_j}, \text{ s.t. } \sum_j \theta_{ij} = 1, 0 \leq \theta_{ij} \leq 1$$

Show the Expectation step (7 pts) and give the EM update formulae for π_i (3 pts) and Θ_i (5 pts). Show all steps including application of Bayes rule and computation of derivatives. Lagrange multipliers can be helpful for keeping π_i and Θ_i on the probability simplex.

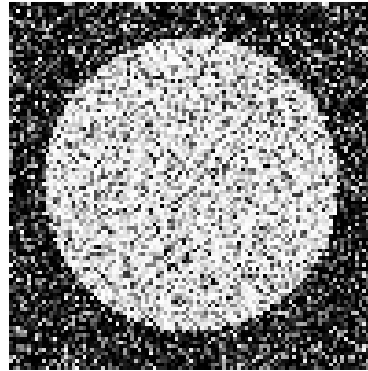
n is the total count of the histogram

What's wrong with this prediction?



$P(\text{foreground} \mid \text{image})$

Solution



$P(\text{foreground} \mid \text{image})$

Encode dependencies between pixels

Normalizing constant

$$P(\mathbf{y}; \theta, data) = \frac{1}{Z} \prod_{i=1..N} f_1(y_i; \theta, data) \prod_{i,j \in edges} f_2(y_i, y_j; \theta, data)$$

Labels to be predicted

Individual predictions

Pairwise predictions

Writing Likelihood as an “Energy”

$$P(\mathbf{y}; \theta, data) = \frac{1}{Z} \prod_{i=1..N} p_1(y_i; \theta, data) \prod_{i,j \in edges} p_2(y_i, y_j; \theta, data)$$

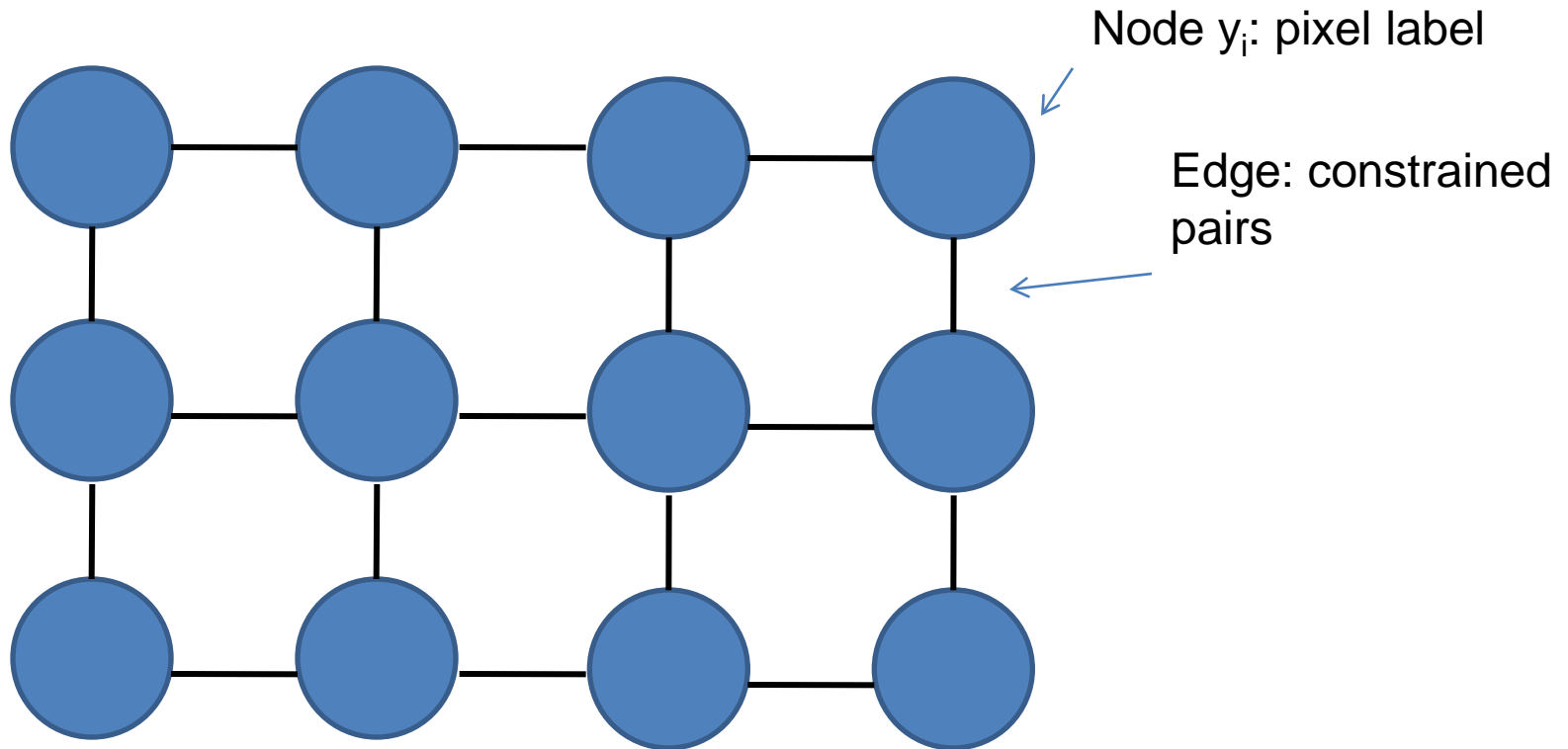


$$Energy(\mathbf{y}; \theta, data) = \sum_i \psi_1(y_i; \theta, data) + \sum_{i,j \in edges} \psi_2(y_i, y_j; \theta, data)$$

“Cost” of assignment y_i

“Cost” of pairwise assignment y_i, y_j

Markov Random Fields



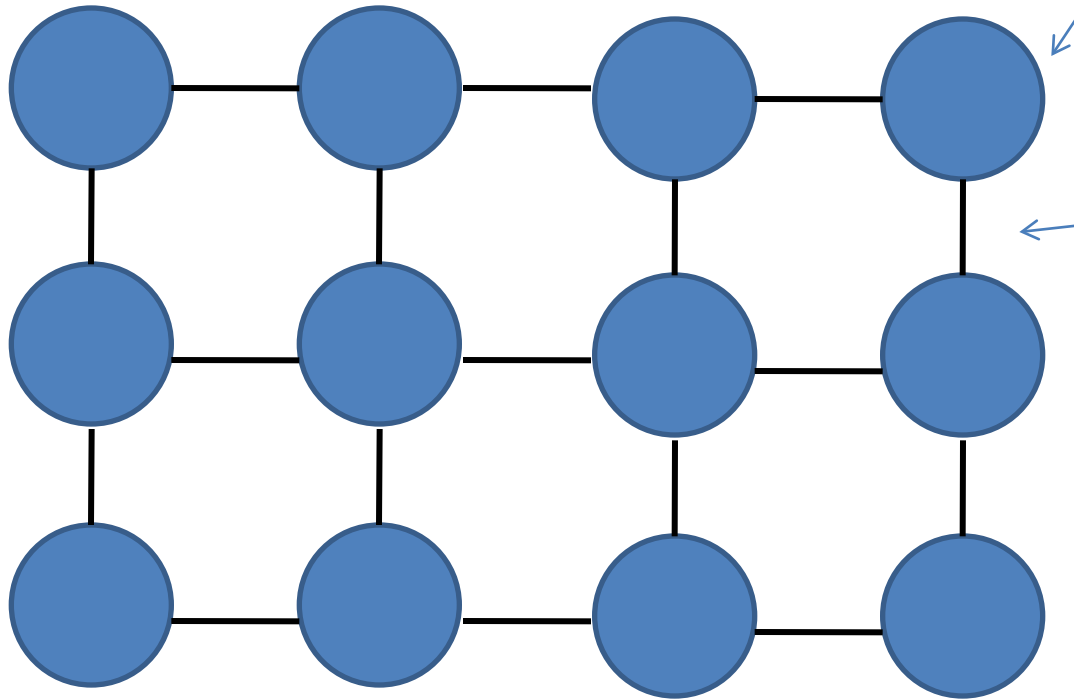
Cost to assign a label to
each pixel

Cost to assign a pair of labels to
connected pixels

$$Energy(\mathbf{y}; \theta, data) = \sum_i \psi_1(y_i; \theta, data) + \sum_{i,j \in edges} \psi_2(y_i, y_j; \theta, data)$$

Markov Random Fields

- Example: “label smoothing” grid



Unary potential

0: $-\log P(y_i = 0 ; \text{data})$

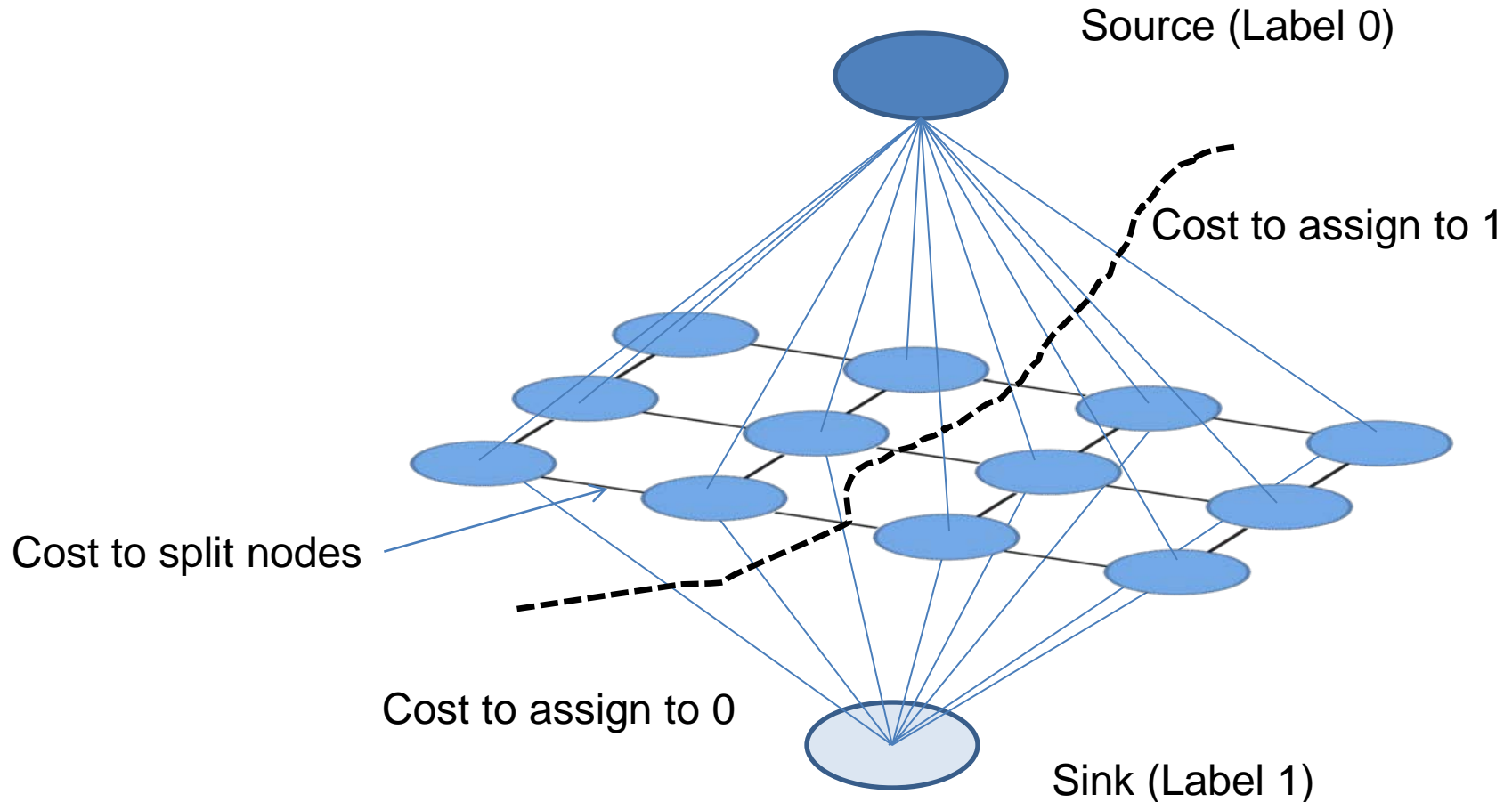
1: $-\log P(y_i = 1 ; \text{data})$

Pairwise Potential

	0	1
0	0	K
1	K	0

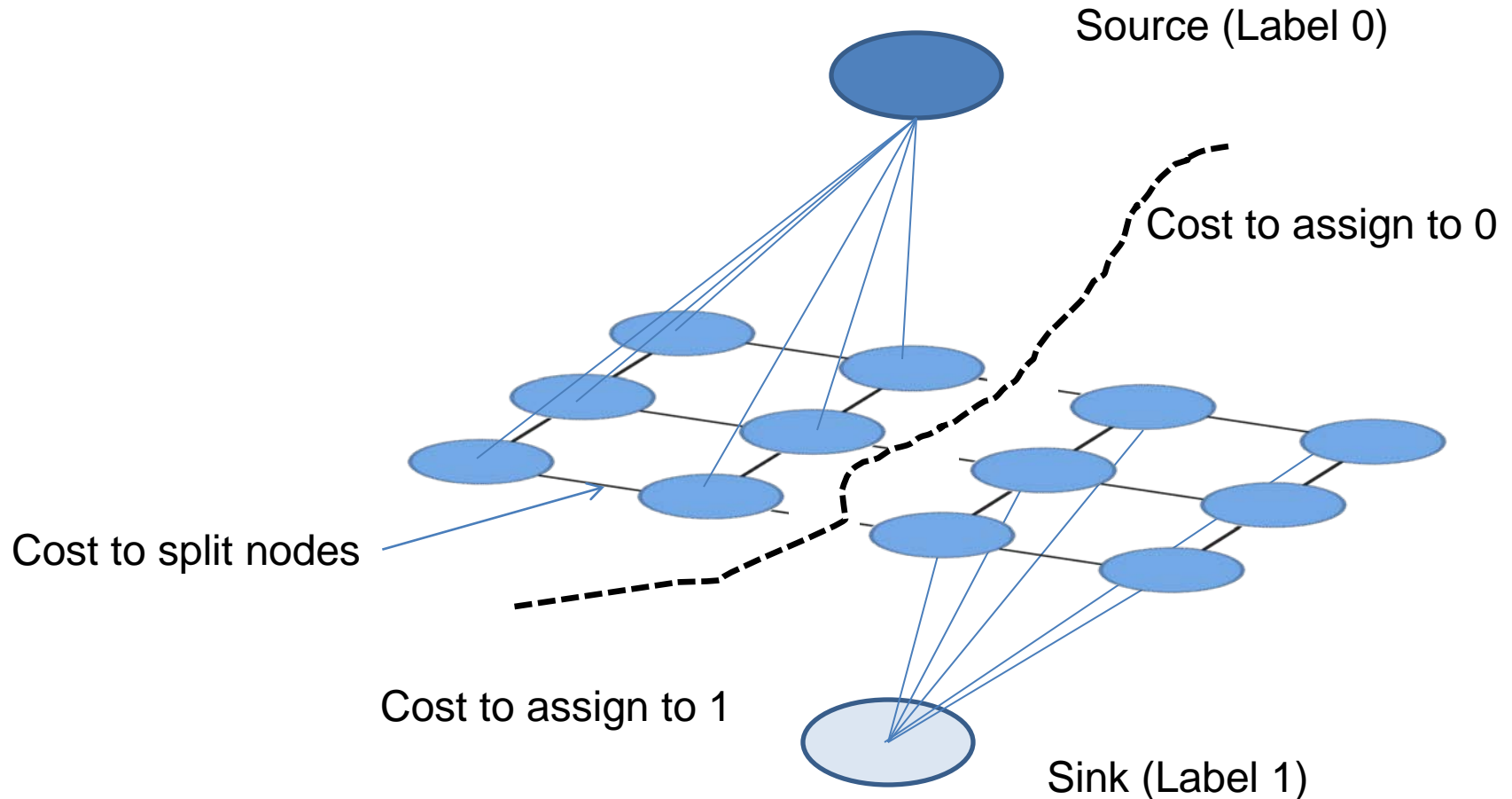
$$\text{Energy}(\mathbf{y}; \theta, \text{data}) = \sum_i \psi_1(y_i; \theta, \text{data}) + \sum_{i,j \in \text{edges}} \psi_2(y_i, y_j; \theta, \text{data})$$

Solving MRFs with graph cuts



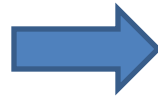
$$Energy(\mathbf{y}; \theta, data) = \sum_i \psi_1(y_i; \theta, data) + \sum_{i,j \in edges} \psi_2(y_i, y_j; \theta, data)$$

Solving MRFs with graph cuts



$$Energy(\mathbf{y}; \theta, data) = \sum_i \psi_1(y_i; \theta, data) + \sum_{i,j \in edges} \psi_2(y_i, y_j; \theta, data)$$

GrabCut segmentation

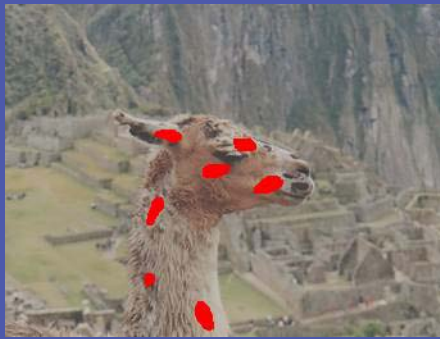


User provides rough indication of foreground region.

Goal: Automatically provide a pixel-level segmentation.

Grab cuts and graph cuts

Magic Wand
(198?)



User
Input



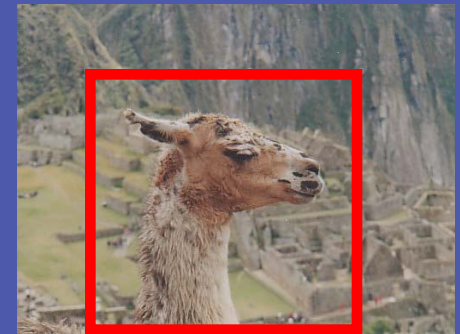
Regions

Intelligent Scissors
Mortensen and Barrett (1995)



Boundary

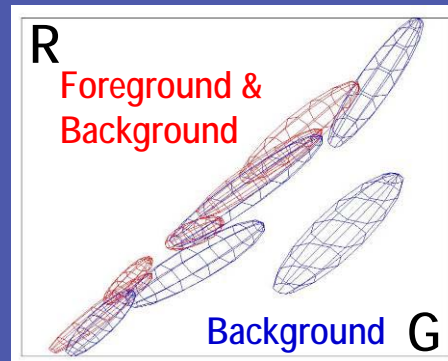
GrabCut



Regions & Boundary

Result

Colour Model

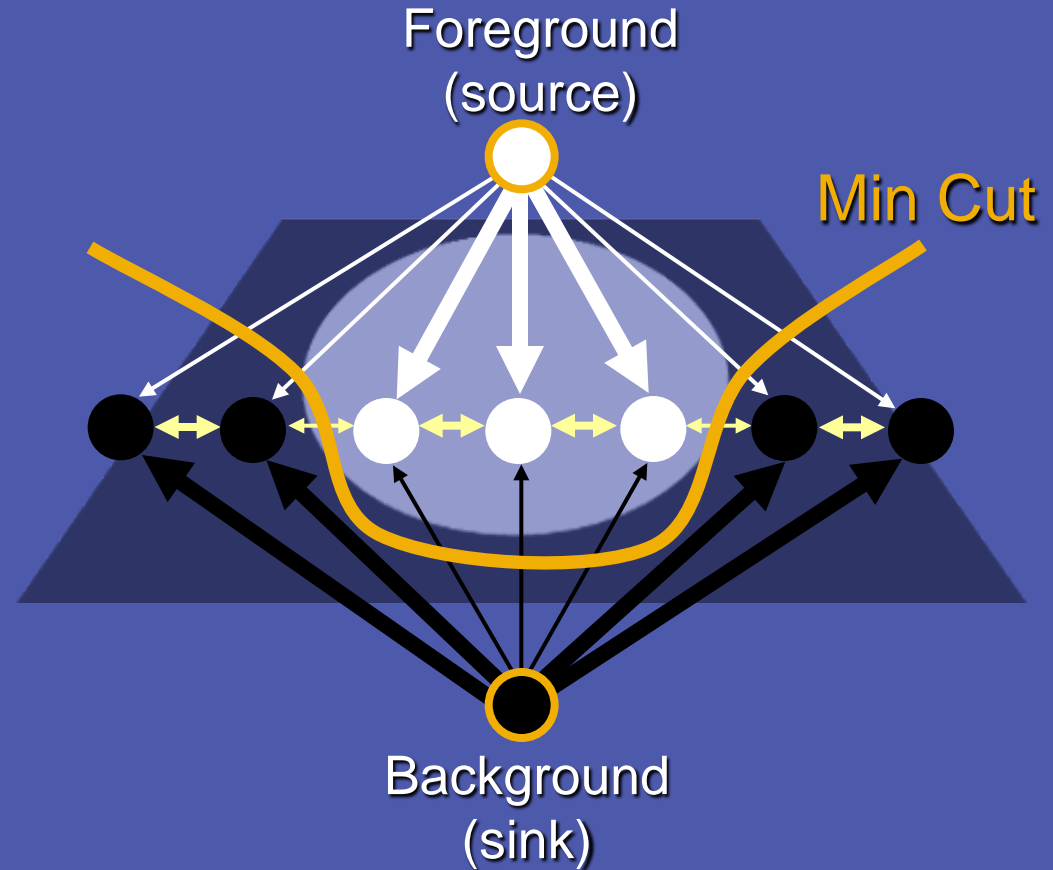
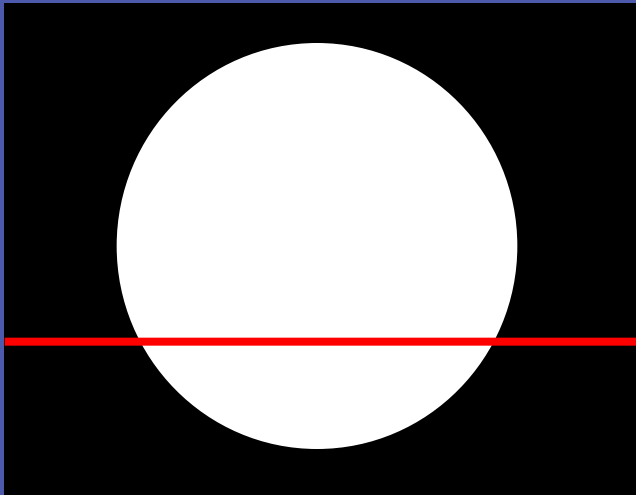


Gaussian Mixture Model (typically 5-8 components)

Graph cuts

Boykov and Jolly (2001)

Image



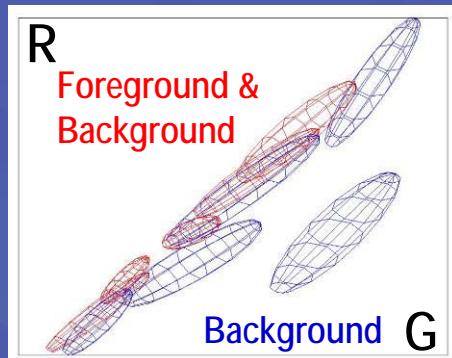
Cut: separating source and sink; Energy: collection of edges

Min Cut: Global minimal energy in polynomial time

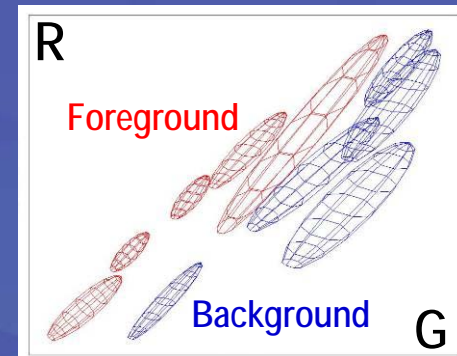
Colour Model



SIGGRAPH2004



Iterated
graph cut



Gaussian Mixture Model (typically 5-8 components)

GrabCut segmentation

1. Define graph

- usually 4-connected or 8-connected

2. Define unary potentials

- Color histogram or mixture of Gaussians for background and foreground

$$\text{unary_potential}(x) = -\log \left(\frac{P(c(x); \theta_{\text{foreground}})}{P(c(x); \theta_{\text{background}})} \right)$$

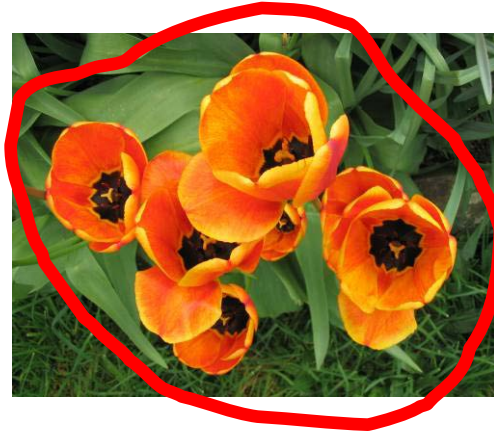
3. Define pairwise potentials

$$\text{edge_potential}(x, y) = k_1 + k_2 \exp \left\{ \frac{-\|c(x) - c(y)\|^2}{2\sigma^2} \right\}$$

4. Apply graph cuts

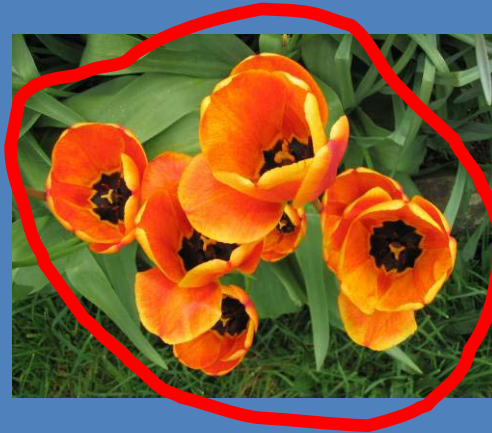
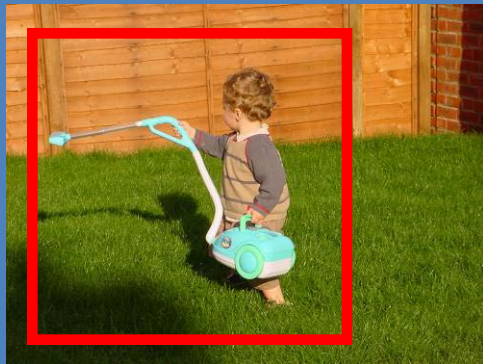
5. Return to 2, using current labels to compute foreground, background models

What is easy or hard about these cases for graphcut-based segmentation?





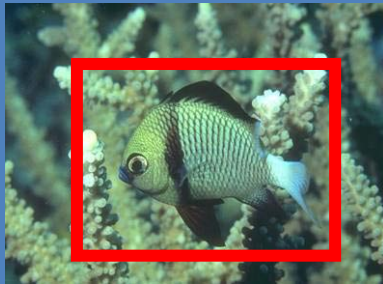
Easier examples





More difficult Examples

Camouflage &
Low Contrast



Fine structure



Harder Case



Initial
Rectangle

Initial
Result

4 Segmentation (25%)

a) (15 pts) Implement a Gaussian mixture model with diagonal covariance to perform foreground/background segmentation on the butterfly RGB image provided. Initialize the foreground by the pixels inside the box shown in red and background by the rest. The top left and bottom right corners of this box are $[(29; 104)(475; 248)]$. Generate separate GMMs for foreground and background pixels ($P(x|\Theta_{fg})$, $P(x|\Theta_{bg})$ respectively). Use an appropriate number of clusters and estimate the model parameters using EM. You may use K-means for initializing EM. Show the probability that each pixel belongs to the foreground with an intensity image and the final pixel segmentation. Explain your choice of the number of mixture components for foreground and background and the choice of the threshold for generating the final segmentation.

b) (10 pts) Refine the segmentation using graph cuts. Use the estimated FG and BG log probabilities for the unary term and a contrast sensitive model (e.g. $\exp(\frac{-\|c(x)-c(y)\|^2}{2\sigma^2})$, where $c(x)$ indicates a pixel's color) for the pairwise term.

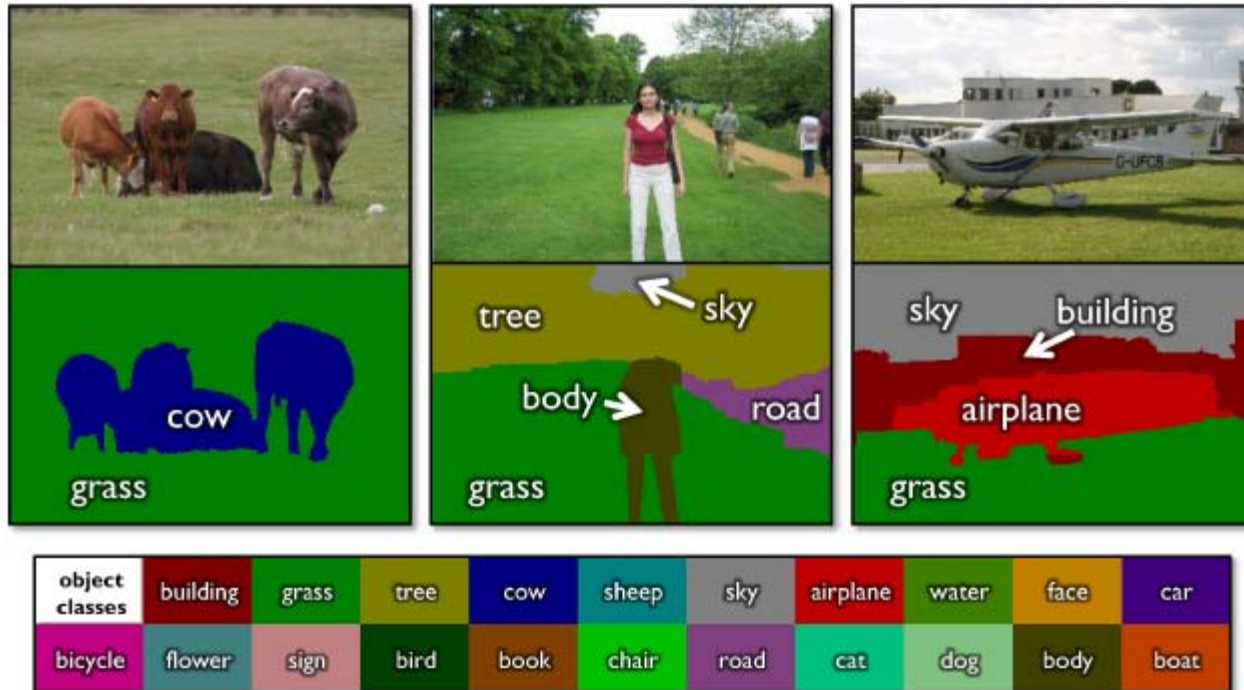
Code for computing the graph cuts can be found in the included GCmex1.5/ directory. OS-X users must first run compile_gc.m. Please contact Ian if there are any issues executing these functions.



Lazy Snapping (Li et al. SG 2004)

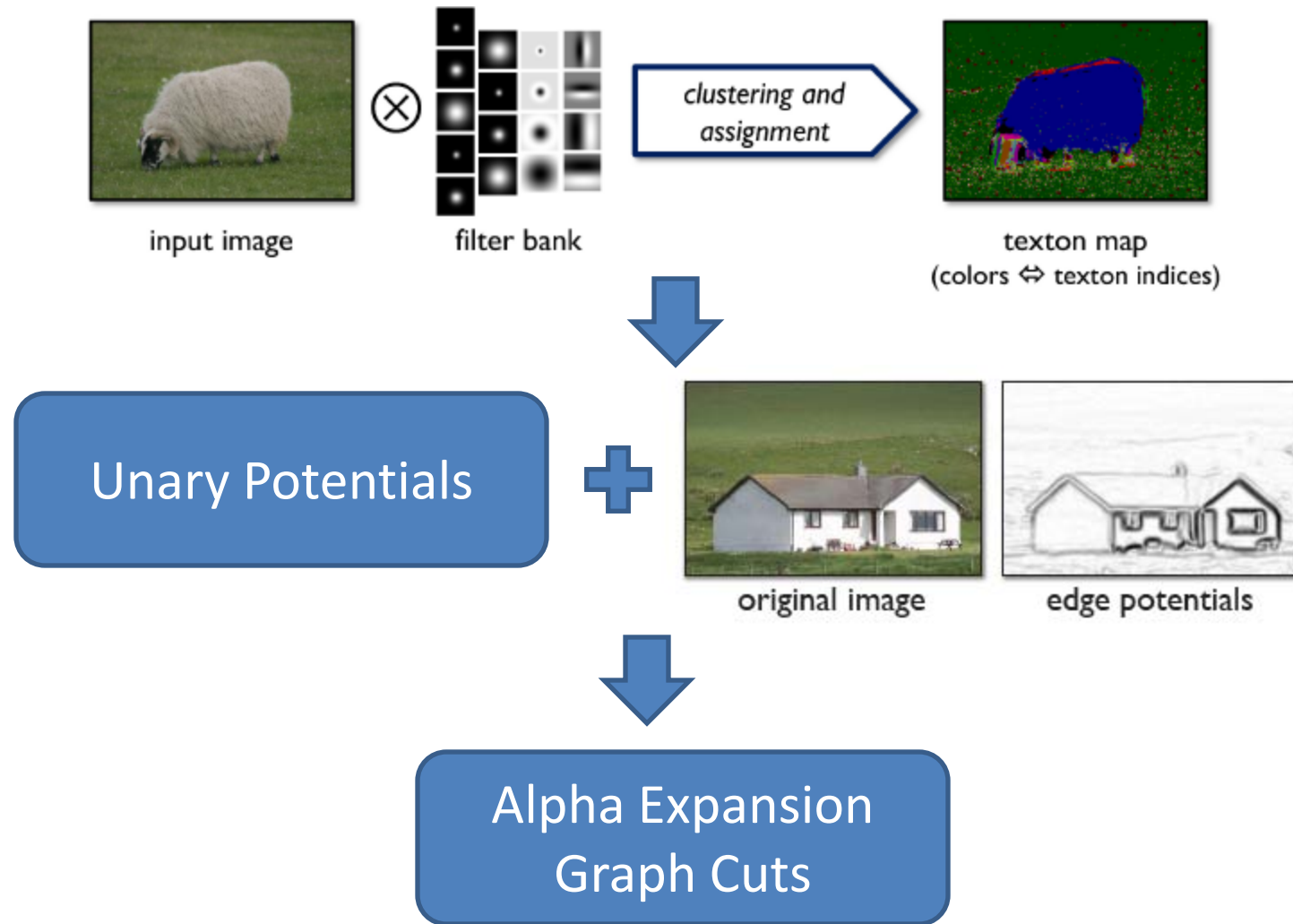


Using graph cuts for recognition



TextonBoost (Shotton et al. 2009 IJCV)

Using graph cuts for recognition



Limitations of graph cuts

- Associative: edge potentials penalize different labels

Must satisfy

$$E^{i,j}(0,0) + E^{i,j}(1,1) \leq E^{i,j}(0,1) + E^{i,j}(1,0)$$

- If not associative, can sometimes clip potentials
- Approximate for multilabel
 - Alpha-expansion or alpha-beta swaps

Graph cuts: Pros and Cons

- Pros
 - Very fast inference
 - Can incorporate data likelihoods and priors
 - Applies to a wide range of problems (stereo, image labeling, recognition)
- Cons
 - Not always applicable (associative only)
 - Need unary terms (not used for generic segmentation)
- Use whenever applicable

More about MRFs/CRFs

- Other common uses
 - Graph structure on regions
 - Encoding relations between multiple scene elements
- Inference methods
 - Loopy BP or BP-TRW: approximate, slower, but works for more general graphs

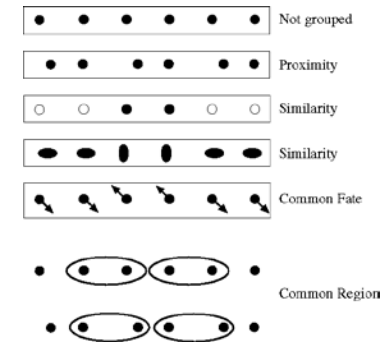
Further reading and resources

- Graph cuts
 - <http://www.cs.cornell.edu/~rdz/graphcuts.html>
 - Classic paper: [What Energy Functions can be Minimized via Graph Cuts?](#) (Kolmogorov and Zabih, ECCV '02/PAMI '04)
- Belief propagation

Yedidia, J.S.; Freeman, W.T.; Weiss, Y., "Understanding Belief Propagation and Its Generalizations", Technical Report, 2001:
<http://www.merl.com/publications/TR2001-022/>
- Normalized cuts and image segmentation (Shi and Malik)
<http://www.cs.berkeley.edu/~malik/papers/SM-ncut.pdf>
- N-cut implementation
<http://www.seas.upenn.edu/~timothee/software/ncut/ncut.html>

Next Class

- Gestalt grouping



- More segmentation methods

