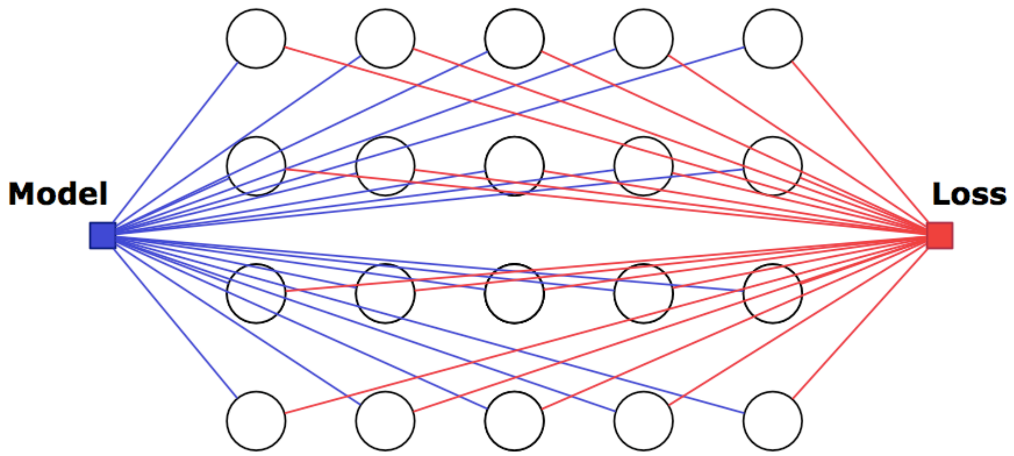
# Learning and Inference to Exploit High Order Potentials





Richard Zemel


CVPR Workshop

June 20, 2011

# Collaborators

Danny Tarlow

Inmar Givoni

Nikola Karamanov

Maks Volkovs

Hugo Larochelle

# Framework for Inference and Learning

Strategy: define a common representation and interface via which components communicate

- Representation: Factor graph – potentials define energy

$$-E(\mathbf{y}) = \underbrace{\sum_{i \in \nu} \phi_i(y_i) + \sum_{i,j \in \varepsilon} \phi_{ij}(y_i, y_j)}_{\text{Low order (standard)}} \quad \underbrace{+ \sum_{c \in C} \phi_c(\mathbf{y}_c)}_{\text{High order (challenging)}}$$

- Inference: Message-passing, e.g., max-product BP

Factor to variable message:

$$m_{\phi_c \to y_i}(y_i) = \max_{\mathbf{y}_c \setminus \{y_i\}} \left[ \phi_c(\mathbf{y}_c) + \sum_{y_{i'} \in \mathbf{y}_c \setminus \{y_i\}} m_{y_{i'} \to \phi_c}(y_{i'}) \right]$$

# Learning: Loss-Augmented MAP

- Scaled margin constraint

$$E(\mathbf{y}) - E(\mathbf{y}^{(n)}) \geq loss(\mathbf{y}, \mathbf{y}^{(n)})$$

$$\underbrace{\sum_c w_c \psi_c(\mathbf{y}_c^{(n)}; \mathbf{x})}_{\text{Fixed}} \geq \underbrace{\sum_c w_c \psi_c(\mathbf{y}_c; \mathbf{x})}_{\text{MAP objective}} + \underbrace{loss(\mathbf{y}, \mathbf{y}^{(n)})}_{\text{loss}}$$

To find margin violations

$$\arg\max_{\mathbf{y}} \left[ \sum_c w_c \psi_c(\mathbf{y}_c; \mathbf{x}_c) + loss(\mathbf{y}, \mathbf{y}^{(n)}) \right]$$

# Expressive models incorporate high-order constraints

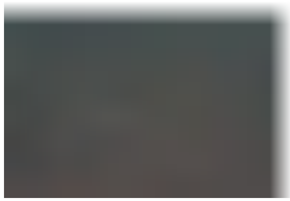
- Problem: map input **x** to output vector **y**, where elements of **y** are inter-dependent

- Can ignore dependencies and build unary model: independent influence of **x** on each element of **y**
- Or can assume some structure on y, such as simple pairwise dependencies (e.g., local smoothness)

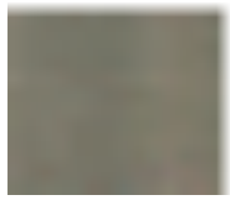
- Yet these often insufficient to capture constraints – many are naturally expressed as higher order

- Example: image labeling

# Image Labeling: Local Information is Weak

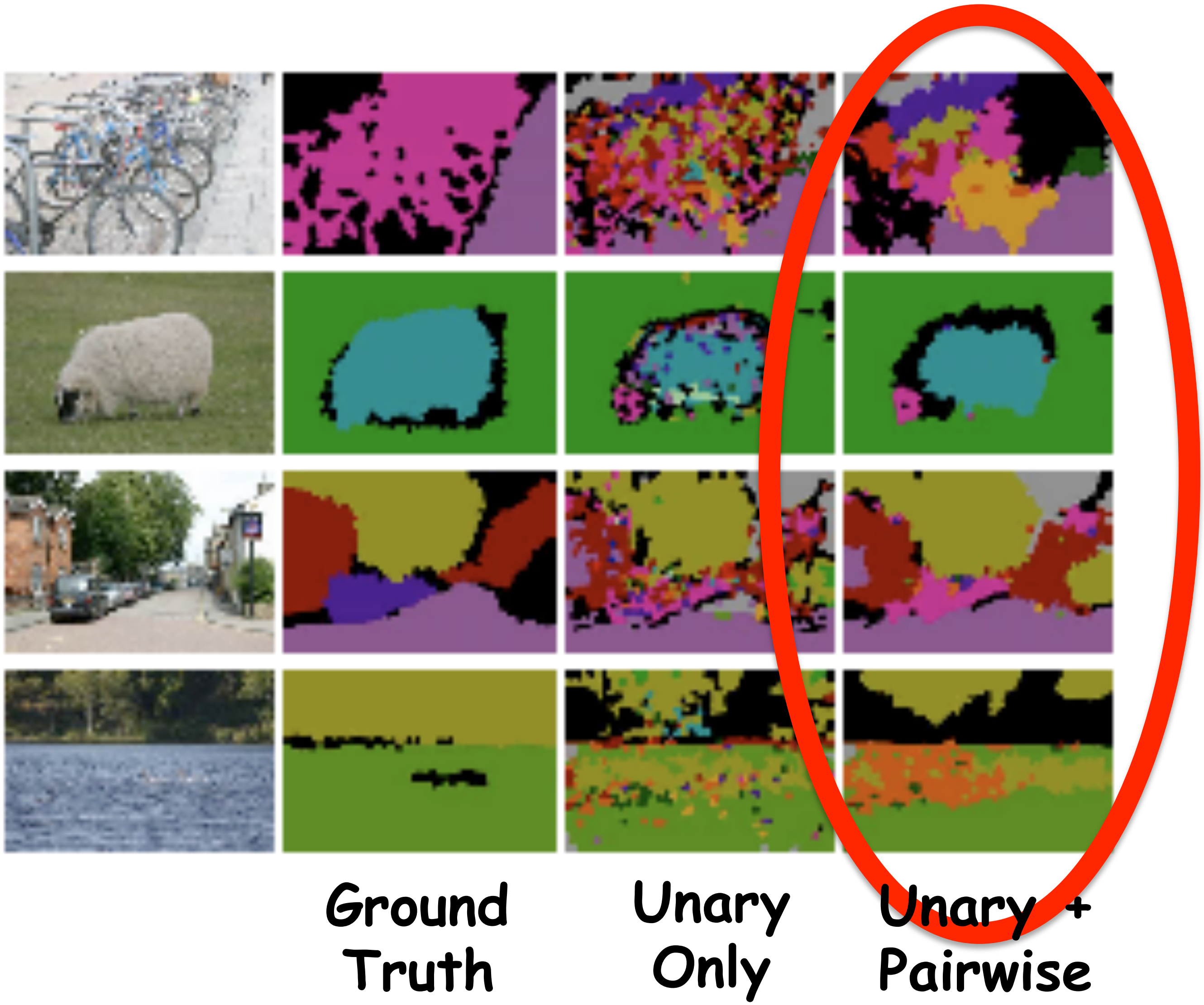


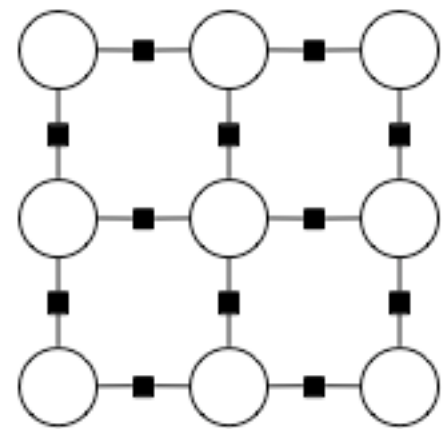
Hippo



Water

# Add Pair-wise Terms: Smoother, but no magic



Pairwise CRF

# Summary of Contributions

Aim: more expressive high-order models (clique-size > 2)

Previous work on HOPs

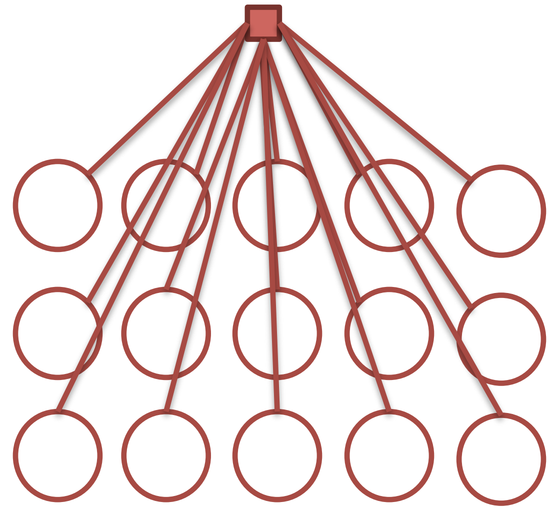
- Pattern potentials (Rother/Kohli/Torr; Komodakis/Paragios)
- Cardinality potentials: (Potetz; Gupta/Sarawagi);
  b-of-N (Huang/Jebara; Givoni/Frey)
- Connectivity (Nowozin/Lampert)
- Label co-occurrence (Ladicky et al)

Our chief contributions:

- Extend vocabulary, unifying framework for HOPs
- Introduce idea of incorporating high-order potentials into loss function for learning
- Novel applications: extend range of problems on which MAP inference/learning useful

# Cardinality Potentials

$$\phi(\mathbf{y}) = f\left(\sum_{y_i \in \mathbf{y}} y_i\right)$$



Assume: binary y; potential defined over all variables

Potential: arbitrary function value based on number of on variables

# Cardinality Potentials: Illustration

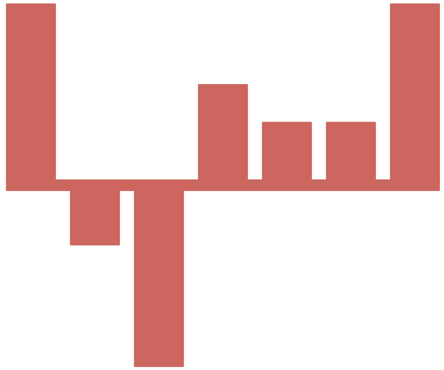
$$\phi(\mathbf{y}) = f(\sum_{y_i \in \mathbf{y}} y_i)$$

$$\tilde{m}_{f \to y_j}(y_j) = \max_{\mathbf{y}_{-j}} \left[ f(\sum_j y_j) + \sum_{j':j' \neq j} m_{y_{j'} \to f}(y_{j'}) \right]$$

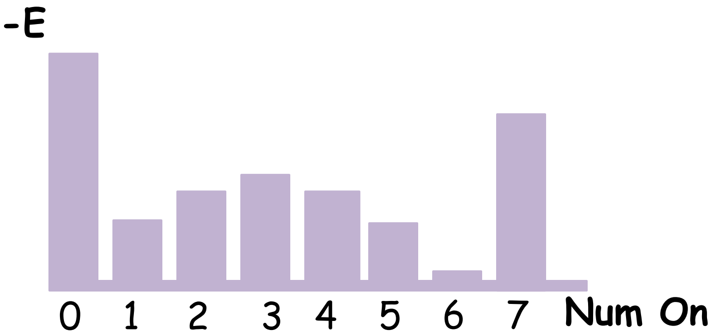
Variable to factor messages: values represent how much that variable wants to be on

Factor to variable message: must consider all combination of values for other variables in clique?

Key insight: conditioned on sufficient statistic of **y**, joint problem splits into two easy pieces

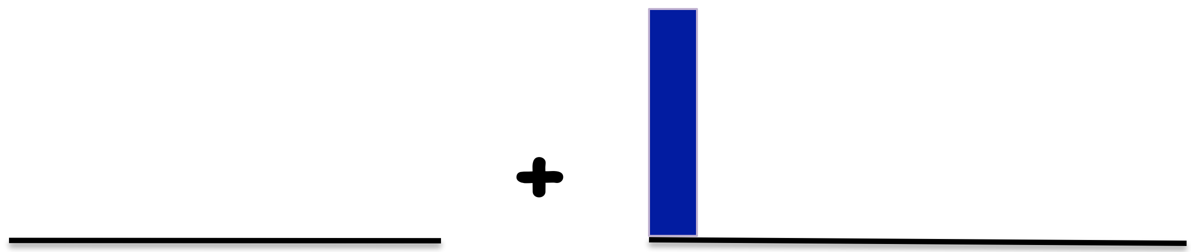


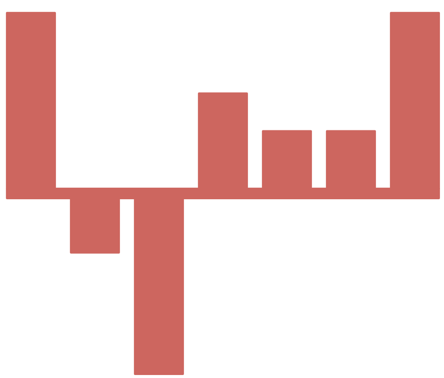
Incoming messages
(preferences for y=1)



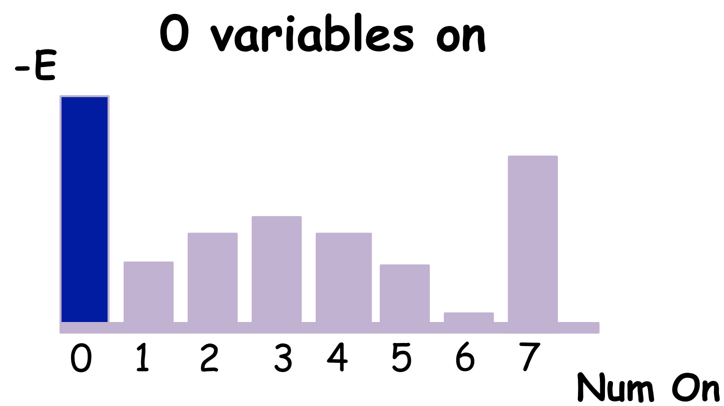

Cardinality Potential

# Total Objective (Factor + Messages):



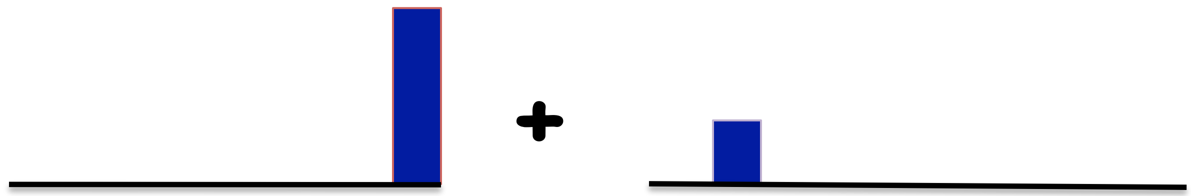


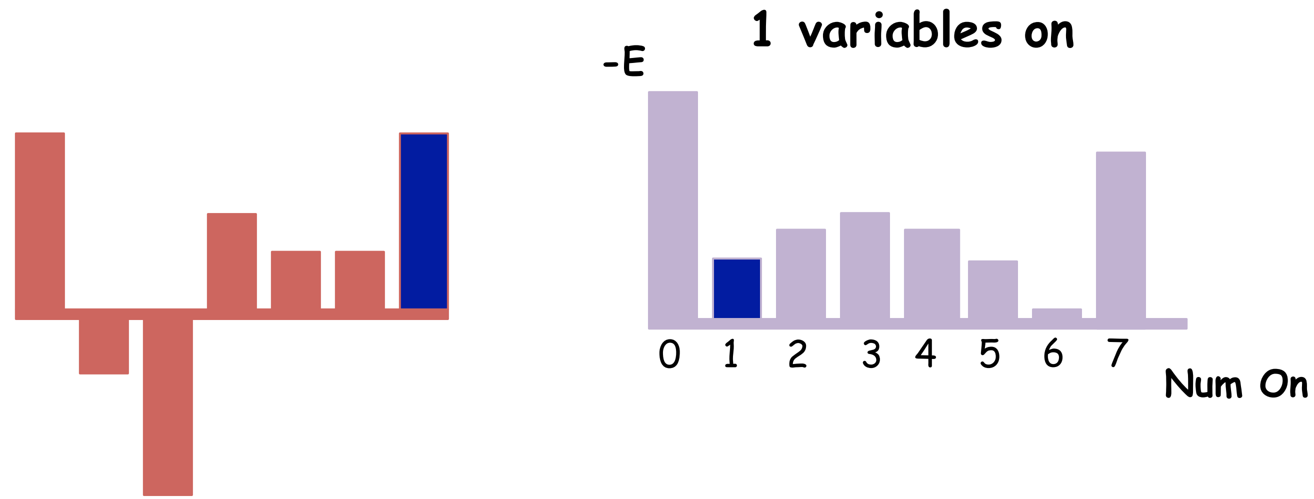
Incoming messages (preferences for y=1)




Cardinality Potential

# Total Objective (Factor + Messages):

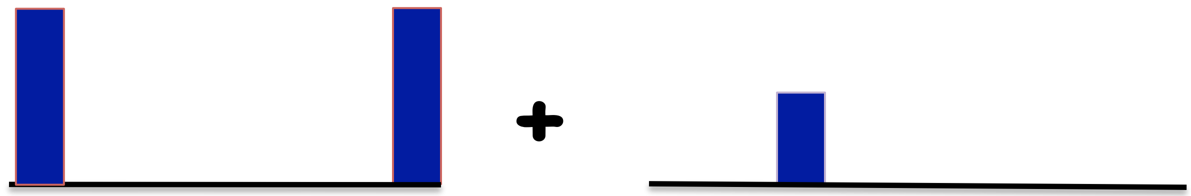


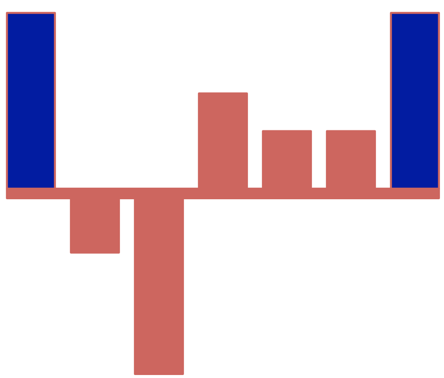



Incoming messages (preferences for y=1)

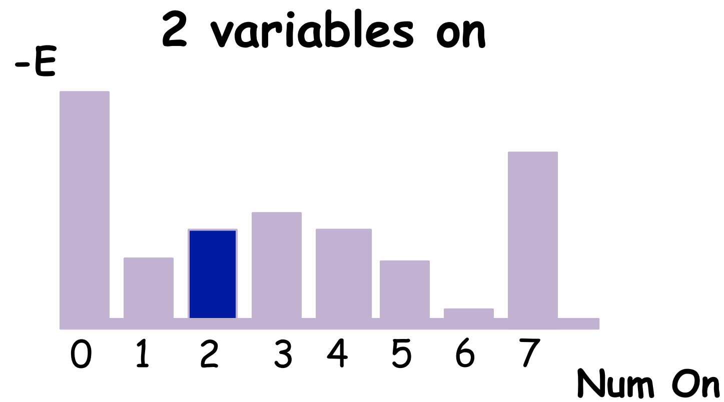
Cardinality Potential

# Total Objective (Factor + Messages):



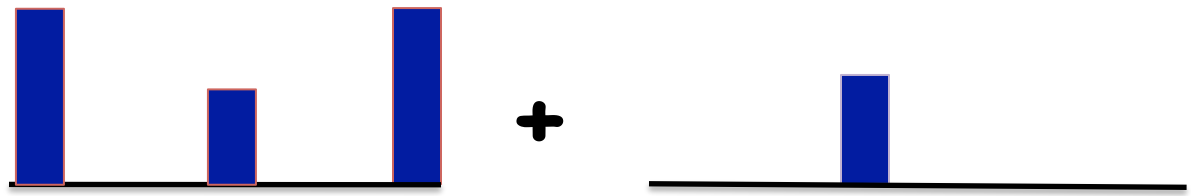


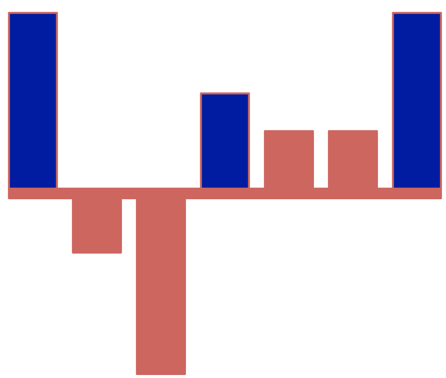
Incoming messages
(preferences for y=1)



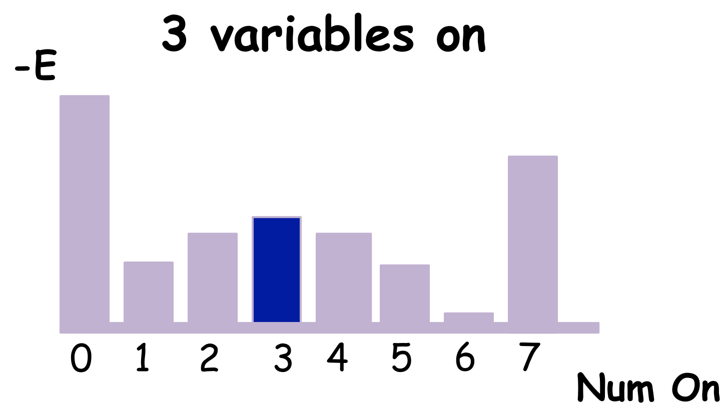

Cardinality Potential

# Total Objective (Factor + Messages):



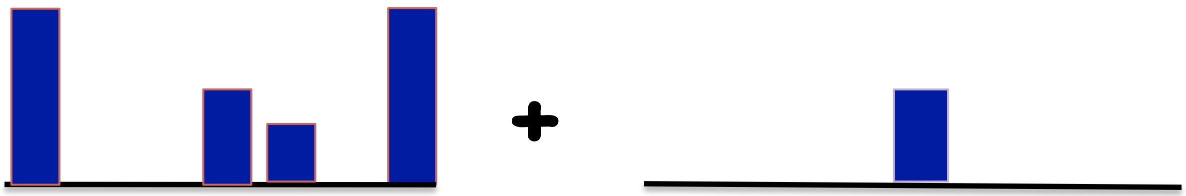


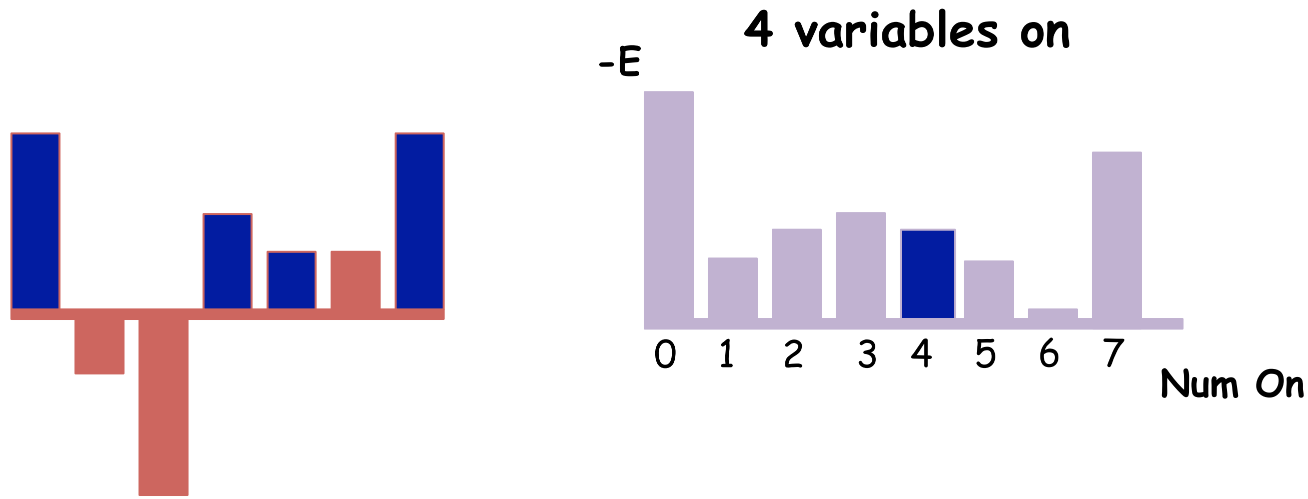
Incoming messages
(preferences for y=1)




Cardinality Potential

# Total Objective (Factor + Messages):

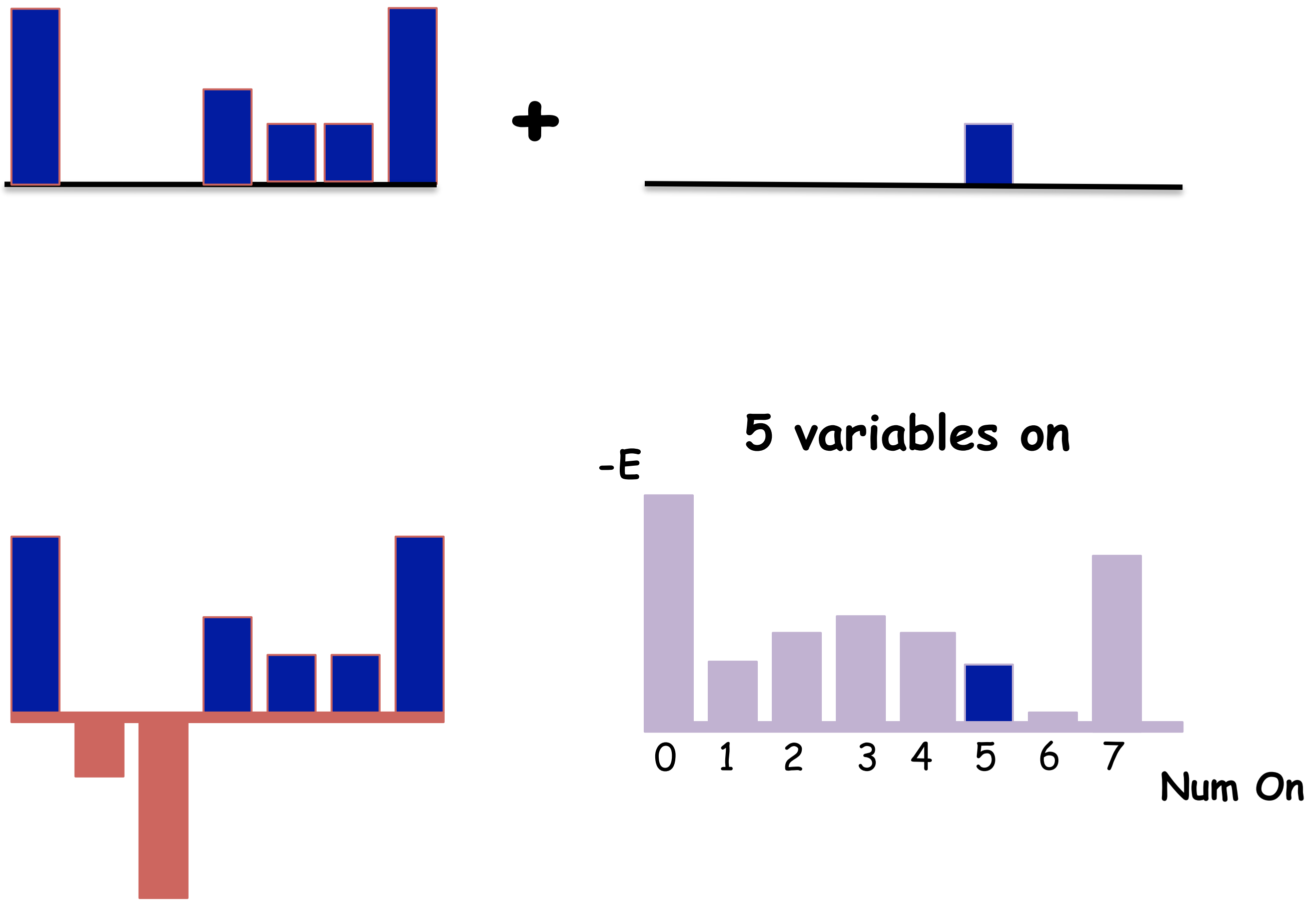





Incoming messages (preferences for y=1)

Cardinality Potential

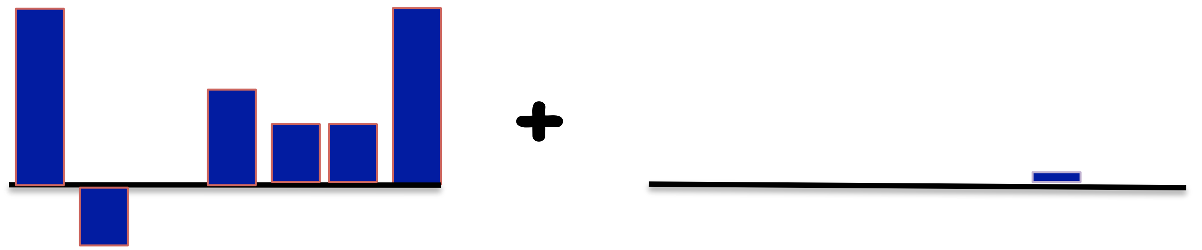
# Total Objective (Factor + Messages):

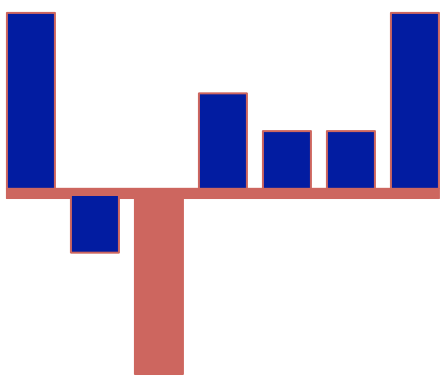



Incoming messages (preferences for y=1)

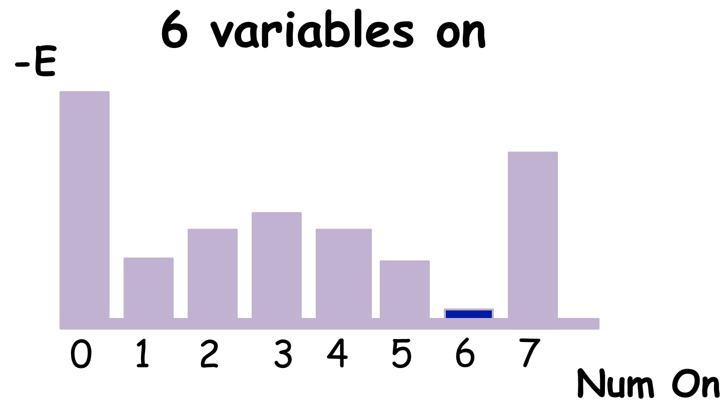
Cardinality Potential

# Total Objective (Factor + Messages):



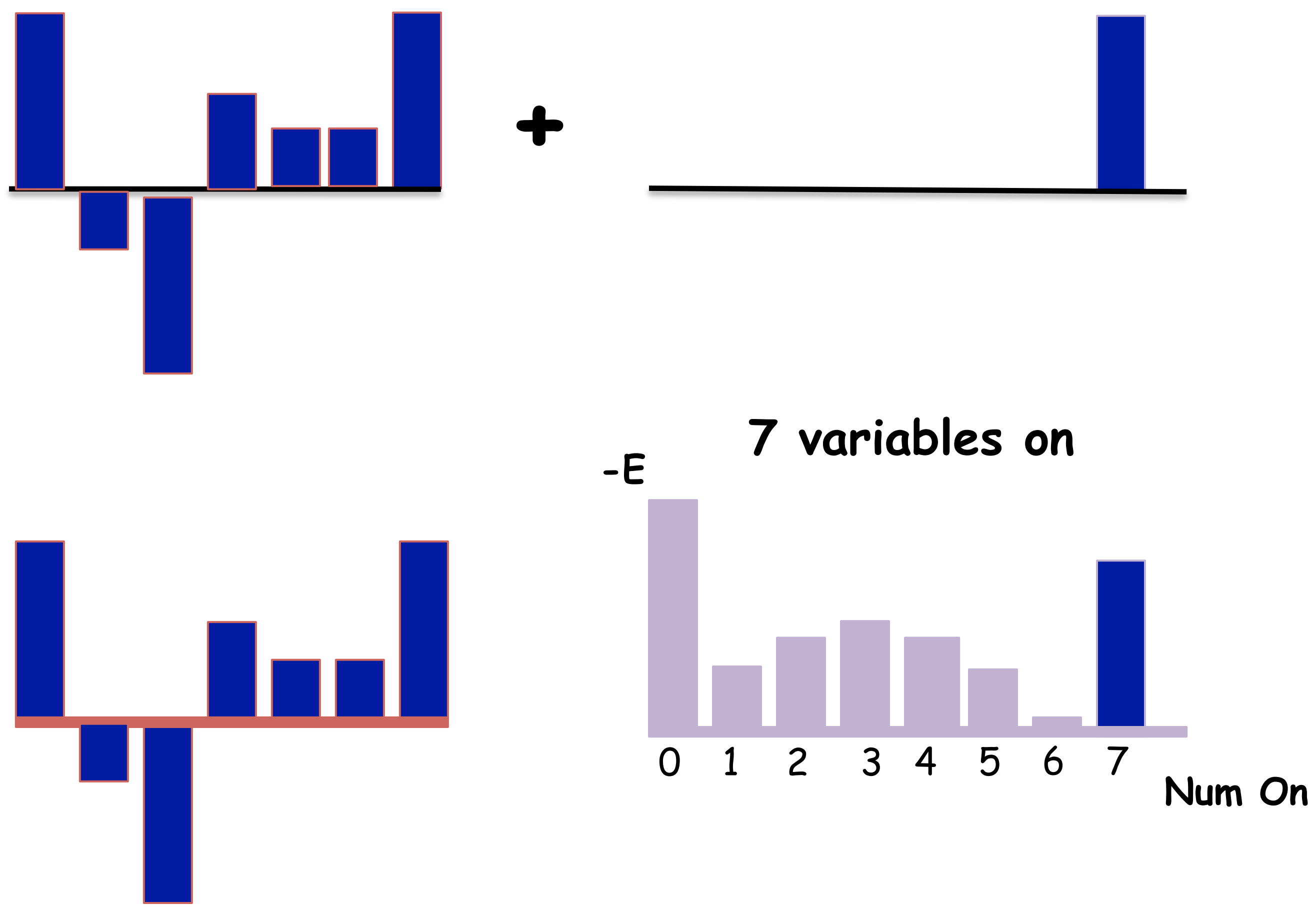


Incoming messages
(preferences for y=1)




Cardinality Potential

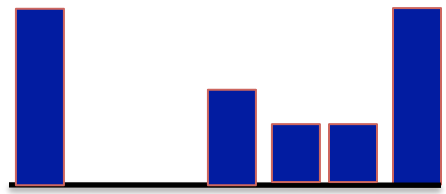
# Total Objective (Factor + Messages):



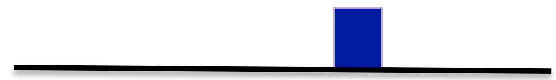

Incoming messages
(preferences for y=1)

Cardinality Potential

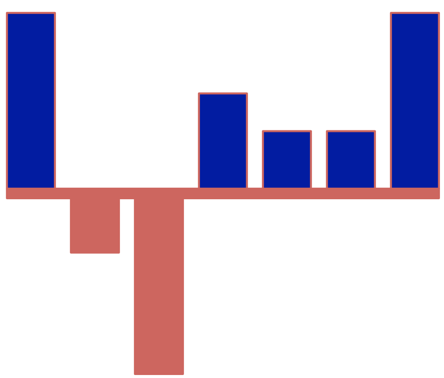
Total Objective (Factor + Messages):

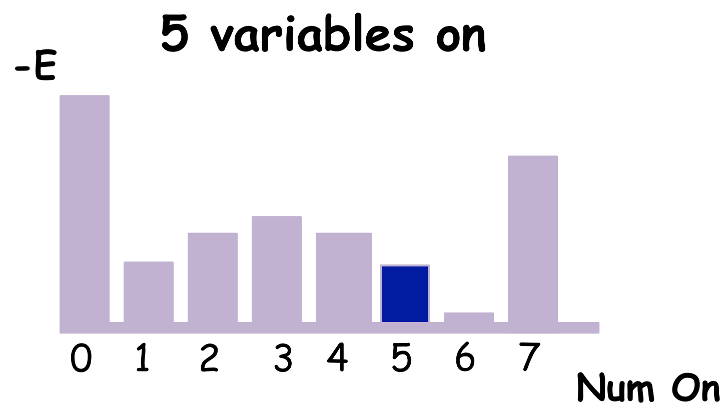


+



**Maximum Sum**



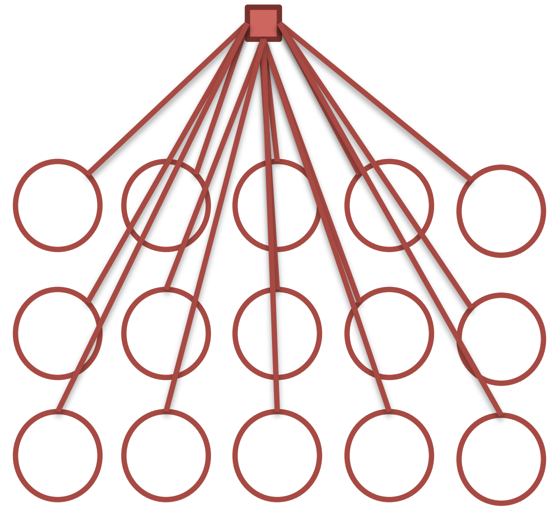



Incoming messages (preferences for y=1)

Cardinality Potential

# Cardinality Potentials

$$\phi(\mathbf{y}) = f\Big(\sum_{y_i \in \mathbf{y}} y_i\Big)$$

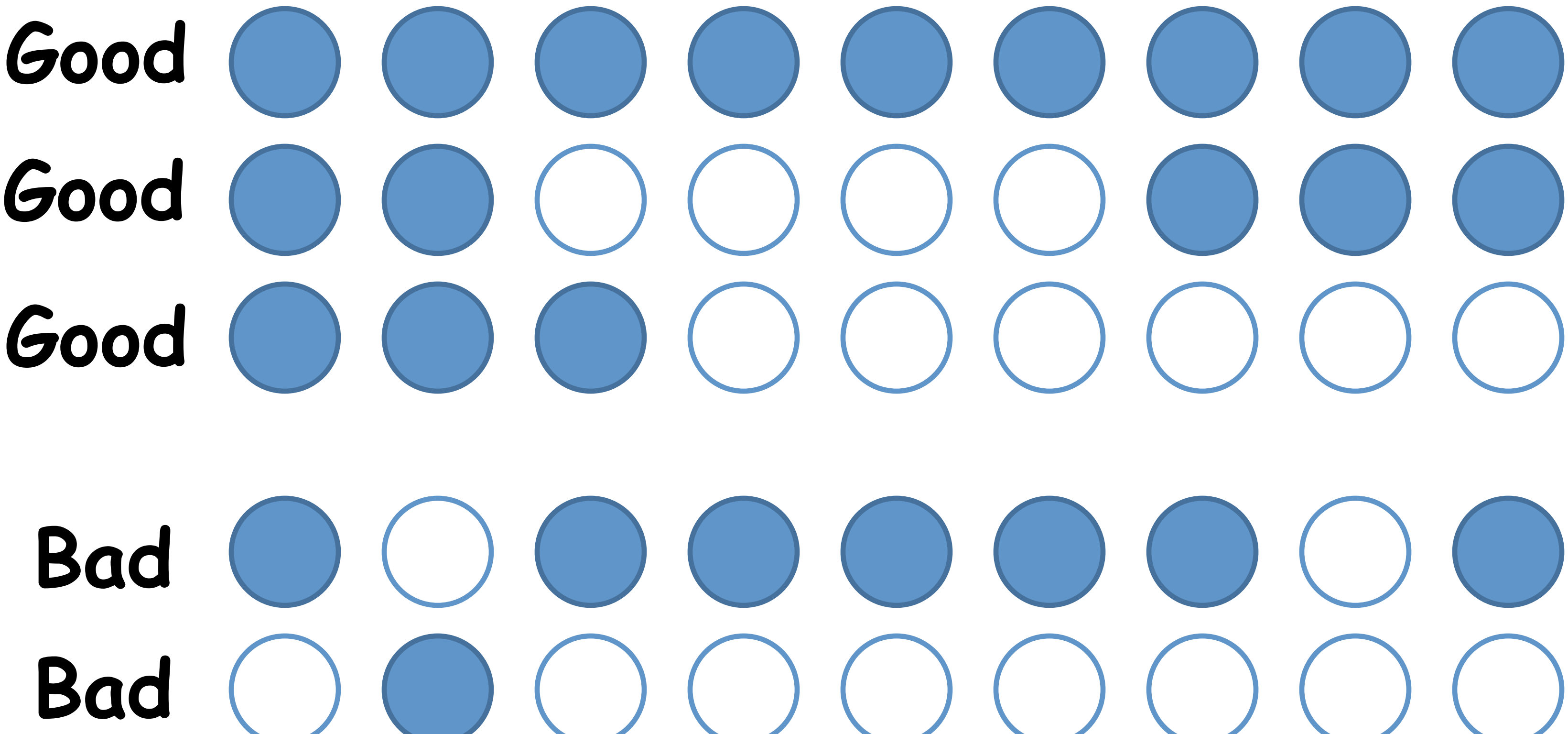


Applications:

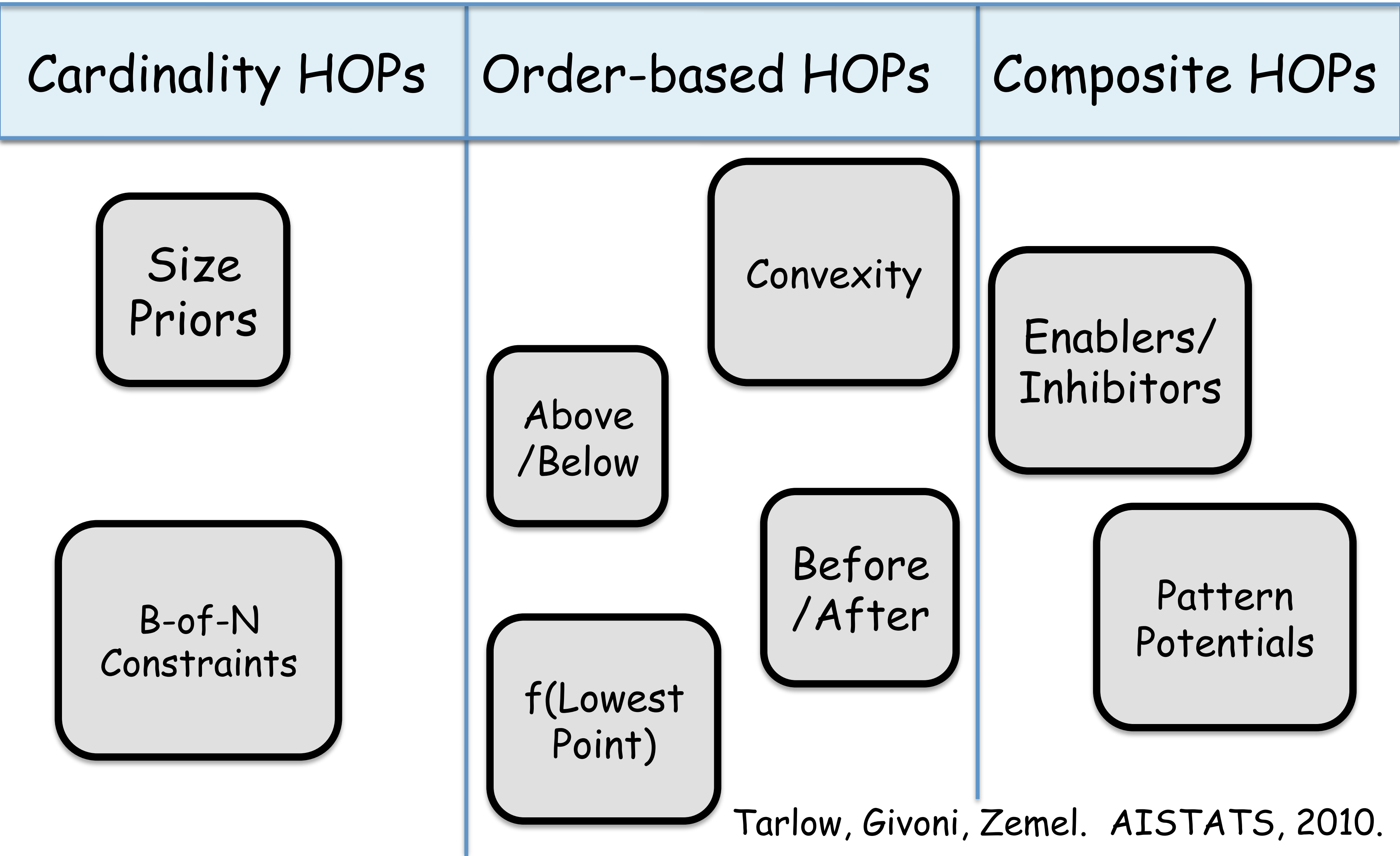
- b-of-N constraints – paper matching
- segmentation: approximate number of pixels per label
- also can specify in image-dependent way → Danny's poster

# Order-based: 1D Convex Sets

$$f(y_1,\ldots,y_N)=\begin{cases} 0 & \text{if } y_i=1 \wedge y_k=1 \Rightarrow y_j=1 \;\; \forall i<j<k \\ -\alpha & \text{otherwise} \end{cases}$$



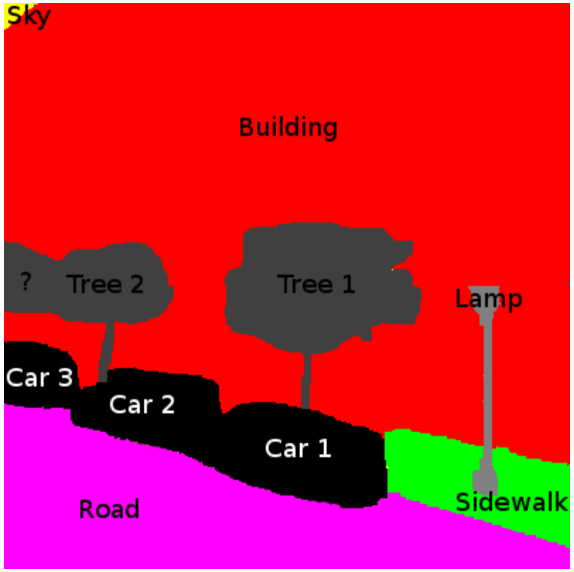
# High Order Potentials

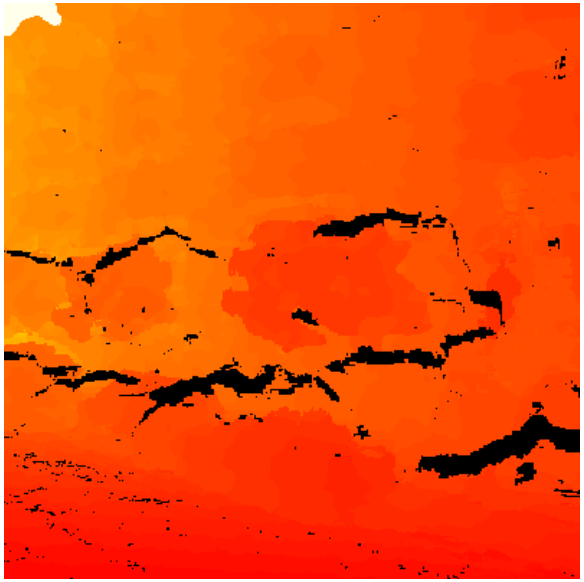



Tarlow, Givoni, Zemel. AISTATS, 2010.

# Joint Depth-Object Class Labeling

- If we know where and what the objects are in a scene we can better estimate their depth
- Knowing the depth in a scene can also aid our semantic understanding
- Some success in estimating depth given image labels (Gould et al)
- Joint inference – easier to reason about occlusion



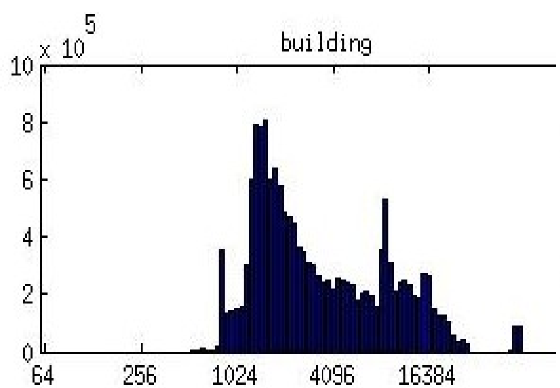
# Potentials Based on Visual Cues

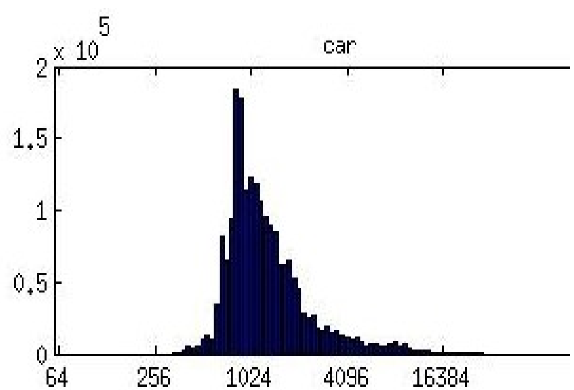
Aim: infer depth & labels from static single images

Represent y: position+depth voxels, w/multi-class labels

Several visual cues, each with corresponding potential:

- Object-specific class, depth unaries
- Standard pairwise smoothness
- Object-object occlusion regularities
- Object-specific size-depth counts
- Object-specific convexity constraints



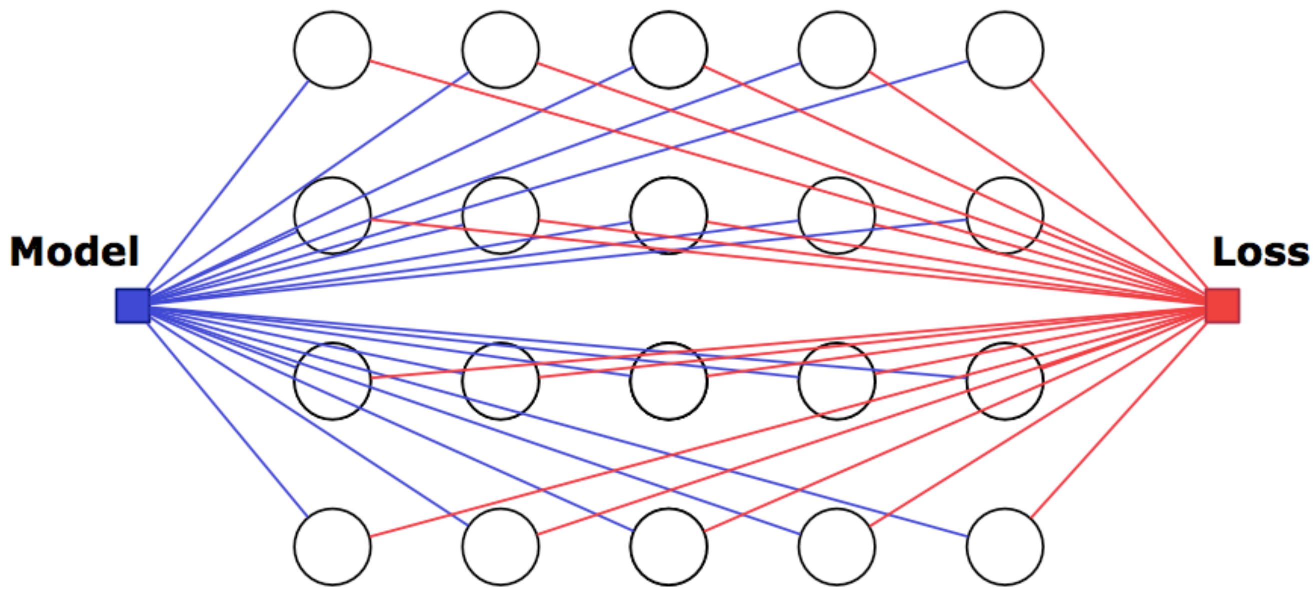

# High-Order Loss Augmented MAP

- Finding margin violations is tractable if loss is decomposable (e.g., sum of per-pixel losses)

$$\arg\max_{\mathbf{y}} \left[ \sum_c w_c \psi_c(\mathbf{y}_c; \mathbf{x}) + loss(\mathbf{y}, \mathbf{y}^{(n)}) \right]$$

- High-order losses not as simple
- But…we can apply same mechanisms used in HOPs!

➢ Same structured factors apply to losses



# Learning with High Order Losses

Introducing HOPs into learning → High-Order Losses (HOLs)

Motivation:

1. **Tailor to target loss:** often non-decomposable

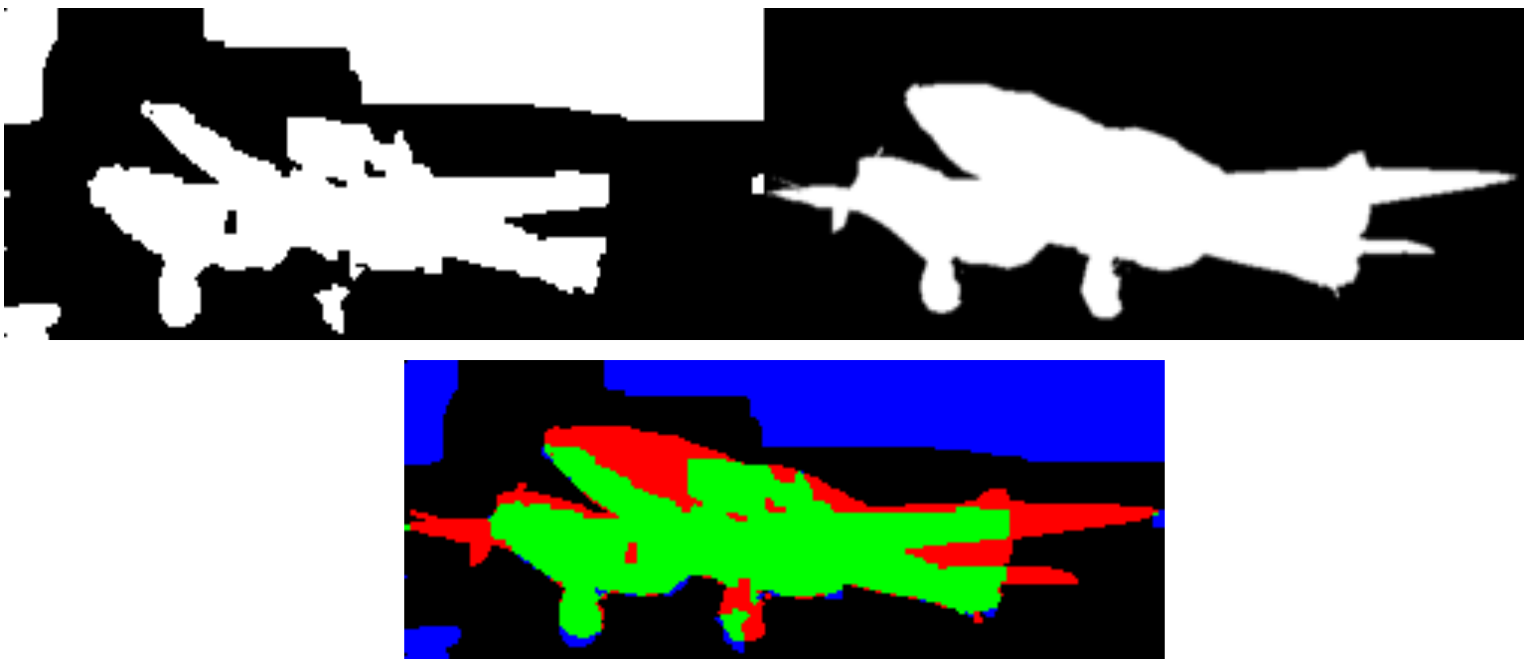
2. **May facilitate fast test-time inference:** keep potentials in model low-order; utilize high-order information only during learning

# HOL 1: PASCAL segmentation challenge

Loss function used to evaluate entries is:

|intersection|/|union|

- Intersection: True Positives (Green) [Hits]
- Union: Hits + False Positives (Blue) + Misses (Red)



- Effect: not all pixels weighted equally; not all images equal; score of all ground is zero

# HOL 1: Pascal loss

Define Pascal loss: quotient of counts

Key: like a cardinality potential – factorizes once condition on number on (but now in two sets) → recognizing structure type provides hint of algorithm strategy

# Pascal VOC Aeroplanes

Images

Pixel Labels

- 110 images (55 train, 55 test)
- At least 100 pixels per side
- 13.6% foreground pixels

# HOL 1: Models & Losses

- Model
  - 84 unary features per pixel (color and texture)
  - 13 pairwise features over 4 neighbors
    - Constant
    - Berkeley PB boundary detector-based

- Losses
  - 0-1 Loss (constant margin)
  - Pixel-wise accuracy Loss
  - HOL 1: Pascal Loss: |intersection|/|union|

- Efficiency: loss-augmented MAP takes <1 minute for 150x100 pixel image; factors: unary+pairwise model + Pascal loss

# Test Accuracy

| Train \ Evaluate | Pixel Acc. | PASCAL Acc. |
|---|---|---|
| 0-1 Loss | 82.1% | 28.6 |
| Pixel Loss | **91.2%** | 47.5 |
| PASCAL Loss | 88.5% | **51.6** |

(a) Unary only model

| Train \ Evaluate | Pixel Acc. | PASCAL Acc. |
|---|---|---|
| 0-1 Loss | 79.0% | 28.8 |
| Pixel Loss | **92.7%** | 54.1 |
| PASCAL Loss | 90.0% | **58.4** |

(b) Unary + pairwise model

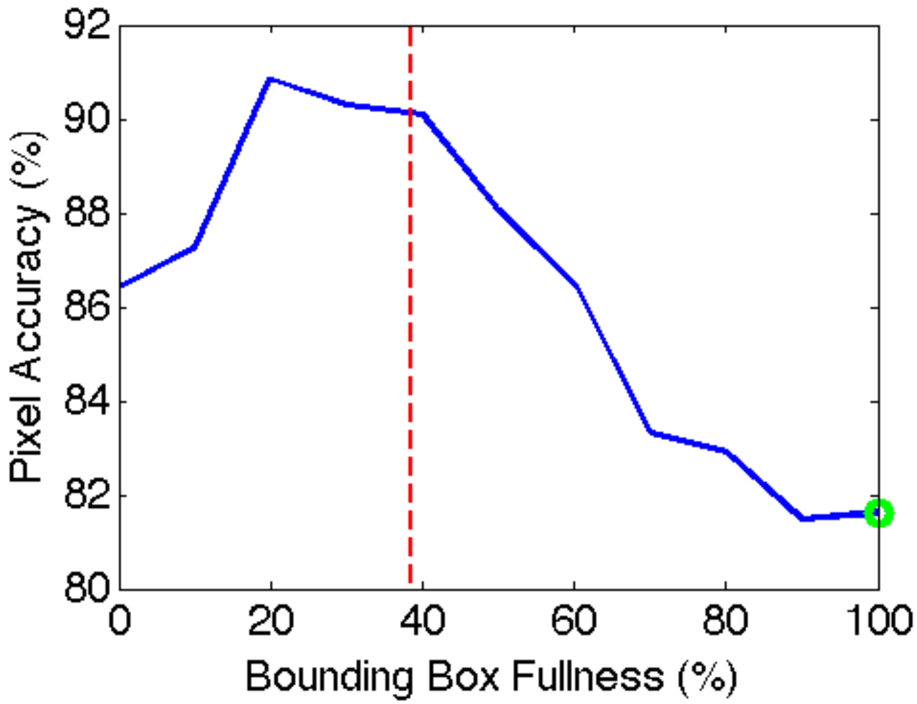
SVM trained independently on pixels does similar to Pixel Loss

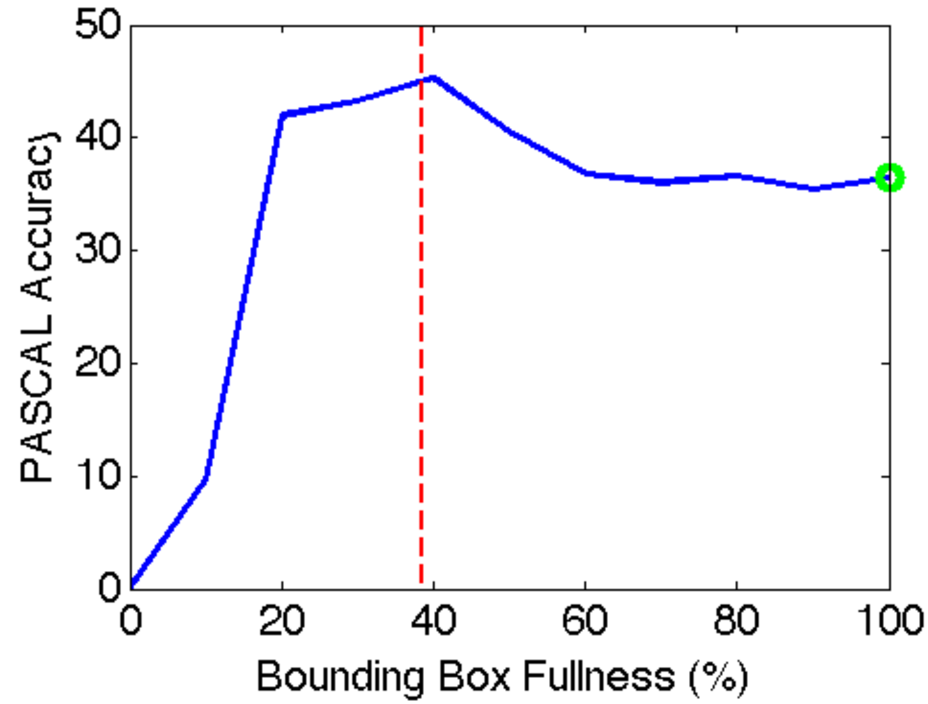
# HOL 2: Learning with BBox Labels

- Same training and testing images; bounding boxes rather than per-pixel labels
- Evaluate w.r.t. per-pixel labels - see if learning is robust to weak label information

- HOL 2: Partial Full Bounding Box
  - 0 loss when K% of pixels inside bounding box and 0% of pixels outside
  - Penalize equally for false positives and #pixel deviations from target K%

# HOL 2: Experimental Results




Like treating bounding box as noiseless foreground label

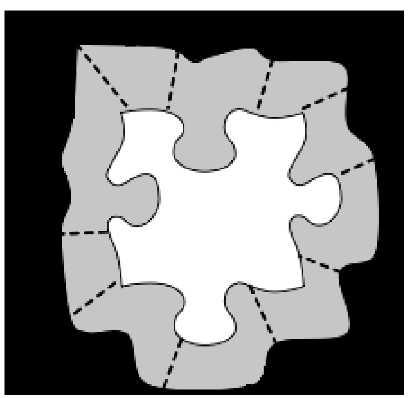
Average bounding box fullness of true segmentations

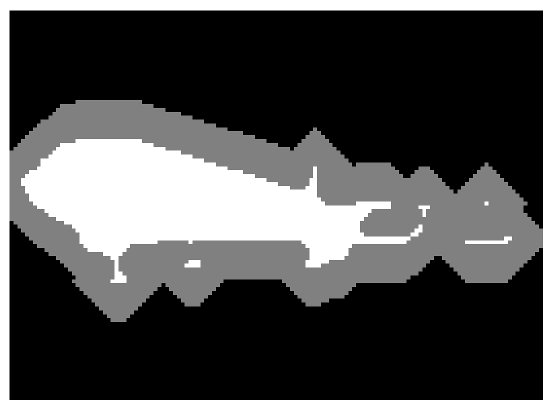
# HOL 3: Local Border Convexity

Other form of weak labeling: rough inner-bound + outline
example: Strokes mark internal object skeleton; coarse circular stroke around outer boundary

→ assume monotonic labeling of any ray from interior passing thru border ($1^m0^n$)

HOL 3: LBC – gray takes on any label, penalty of *a* for each outward path that changes from background to foreground





Training data obtained by eroding labeled images

# HOL 3: Results

| Train \ Evaluate | | Pixel Acc. | PASCAL Acc. |
|---|---|---|---|
| **Aero** | Mod. Loss SVM | 90.2% | 36.4 |
| | LBC Loss | **90.6%** | **38.1** |
| **Car** | Mod. Loss SVM | 79.8% | 0 |
| | LBC Loss | **80.2%** | **5.3** |
| **Cow** | Mod. Loss SVM | **78.4%** | 15.6 |
| | LBC Loss | 76.8% | **32.3** |
| **Dog** | Mod. Loss SVM | 80.2% | 0 |
| | LBC Loss | **82.4%** | **24.2** |

# Wrap Up

- If we're spending so much time working on optimizing objectives -- make sure they're the right objectives
  - Developing toolbox for richer models and objectives with high order models and high order loss functions

- High-order information in energy, or loss?
  - Some HO constraints depend on ground truth: must go in loss (e.g., translation-invariance, assign zero loss to few pixel shifts of object)
  - Adding HO structure only to loss creates variational-like scenario: model must learn to use restricted d.o.f. to optimize loss

- Extensions:
  - Multi-label
  - HOLs not just wrt outputs of one image, but across multiple images (e.g., smoothness of patterns thru frames)

# Learning CRFs

- Conditional Random Fields (CRF): model label **y** conditionally given input **x**

$$P(\mathbf{y} \mid \mathbf{x}, \theta) = \exp(-E(\mathbf{y}, \mathbf{x}; \theta)) / \sum_{\mathbf{y}' \in Y(\mathbf{x})} \exp(-E(\mathbf{y}', \mathbf{x}; \theta))$$

- Include various structures in **y**, like trees, chains, 2D grids, permutations
- Considerable work on developing potentials, energy fcns, and approximate inference in CRFs, but little on loss function
- Typically trained by ML – ignores task's loss

1. Can methods used by SSVMs to adapt training to loss be utilized in CRFs?
2. Develop other loss-sensitive training objectives that rely on probabilistic nature of CRFs?

# Loss Functions for CRFs

- Standard CRF learning: shape energy (learn **θ**) to max. conditional likelihood (MCL) of ground truth **y**, conditioned on its corresponding **x** – ignores loss

$$\ell_{ML}(\mathbf{D};\theta) = -\sum_{(\mathbf{x},\mathbf{y})\in\mathbf{D}} \log p(\mathbf{y}_t \mid \mathbf{x}_t) = E(\mathbf{y}_t,\mathbf{x}_t;\theta) + \log\left(\sum_{\mathbf{y}\in Y(\mathbf{x})} \exp(-E(\mathbf{y},\mathbf{x}_t;\theta))\right)$$

- In well-specified case, with sufficient data, ignoring loss probably not a problem – asymptotic consistency, efficiency of ML
- Assume given loss (evaluate performance of CRF), aim of learning: obtain low average

$$\frac{1}{|\mathbf{D}|}\sum_{(\mathbf{x}_t,\mathbf{y}_t)\in\mathbf{D}} \ell(\widehat{\mathbf{y}}(\mathbf{x}_t))$$

- Hard to optimize: loss not smooth fcn of parameters, loss not smooth fcn of prediction, prediction not smooth fcn of parameters → indirectly optimize avg loss

# New CRF Loss Functions

(1). Loss-augmented

$$E_t^{LA}(\mathbf{y},\mathbf{x}_t;\theta) = E(\mathbf{y},\mathbf{x}_t;\theta) - \ell_t(\mathbf{y})$$

$$\ell_{LA}(\mathbf{D};\theta) = \frac{1}{|\mathbf{D}|}\sum_{(\mathbf{x}_t,\mathbf{y}_t)\in\mathbf{D}} E_t^{LA}(\mathbf{y}_t,\mathbf{x}_t;\theta) + \log\left(\sum_{\mathbf{y}\in Y(\mathbf{x})}\exp(-E_t^{LA}(\mathbf{y},\mathbf{x}_t;\theta))\right)$$

- high loss cases important, increase energy
- analog of margin scaling
- upper bound on avg loss

(2). Loss-scaled

$$E_t^{LS}(\mathbf{y},\mathbf{x}_t;\theta) = \ell_t(\mathbf{y})[E(\mathbf{y},\mathbf{x}_t;\theta) - E(\mathbf{y}_t,\mathbf{x}_t;\theta)] - \ell_t(\mathbf{y})$$

$$\ell_{LS}(\mathbf{D};\theta) = \frac{1}{|\mathbf{D}|}\sum_{(\mathbf{x}_t,\mathbf{y}_t)\in\mathbf{D}} E_t^{LS}(\mathbf{y}_t,\mathbf{x}_t;\theta) + \log\left(\sum_{\mathbf{y}\in Y(\mathbf{x})}\exp(-E_t^{LS}(\mathbf{y},\mathbf{x}_t;\theta))\right)$$

- only focus on high loss cases whose energy is low
- analog of slack scaling
- also upper bound on avg loss

# More New CRF Loss Functions

## (3). Expected-loss

$$\ell_{EL}(\mathbf{D};\theta) = \frac{1}{|\mathbf{D}|}\sum_{(\mathbf{x}_t,\mathbf{y}_t)\in\mathbf{D}} \mathbf{E}_{\mathbf{y}|\mathbf{x}_t}\left[\ell_t(\mathbf{y})\right] = \frac{1}{|\mathbf{D}|}\sum_{(\mathbf{x}_t,\mathbf{y}_t)\in\mathbf{D}}\sum_{\mathbf{y}\in Y(\mathbf{x})} \ell_t(\mathbf{y})p(\mathbf{y}\,|\,\mathbf{x}_t)$$

- not an upper bound on avg loss, but approaches it as learning puts all mass on MAP $y(x_t)$

## (4). KL

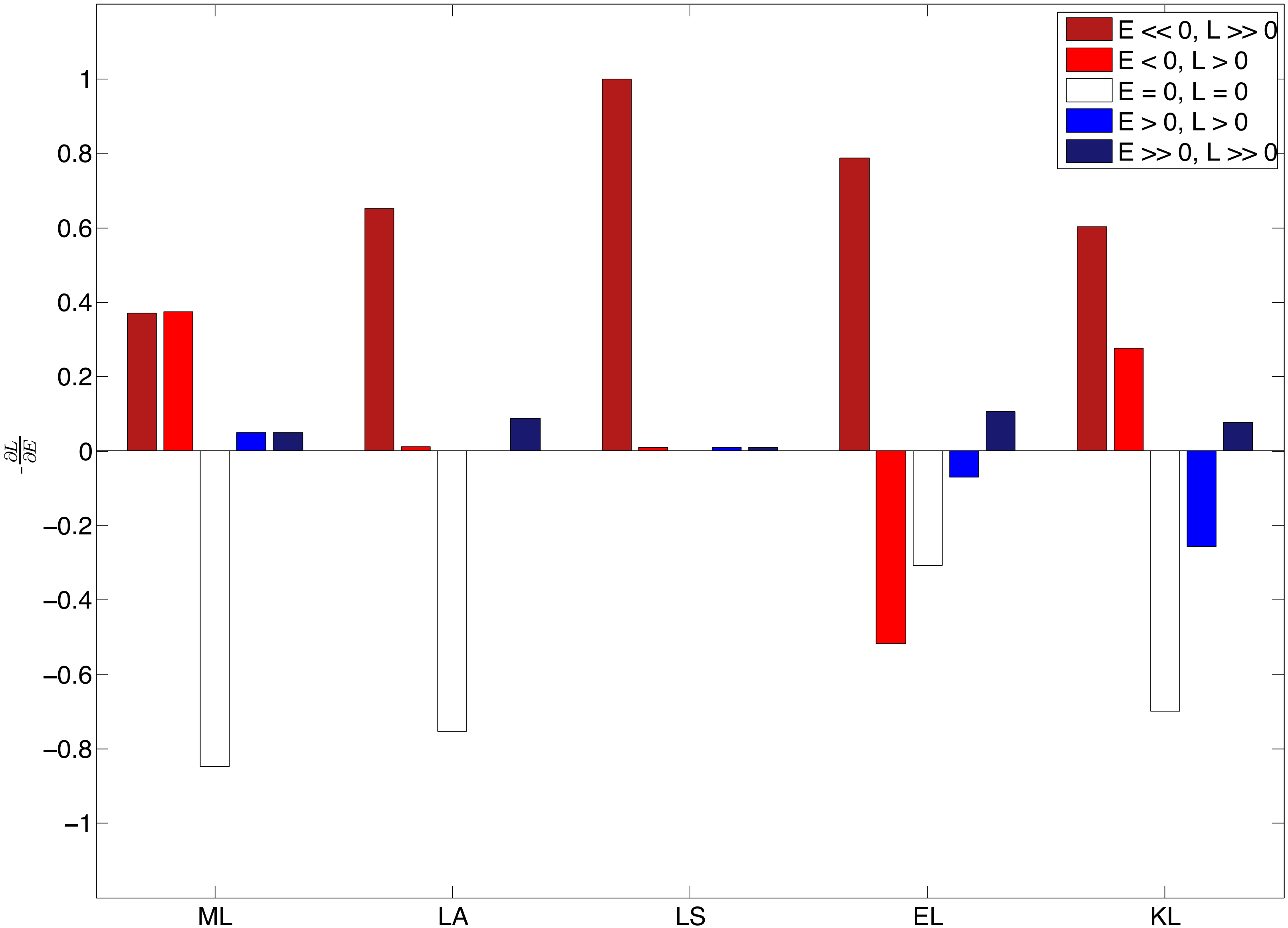
$$\ell_{KL}(\mathbf{D};\theta) = \frac{1}{|\mathbf{D}|}\sum_{(\mathbf{x}_t,\mathbf{y}_t)\in\mathbf{D}} D_{KL}\left[q(\cdot\,|\,t)\,\|\,p(\cdot\,|\,\mathbf{x}_t)\right]$$

$$= -\frac{1}{|\mathbf{D}|}\sum_{(\mathbf{x}_t,\mathbf{y}_t)\in\mathbf{D}}\sum_{\mathbf{y}\in Y(\mathbf{x})} q(\mathbf{y}\,|\,t)p(\mathbf{y}\,|\,\mathbf{x}_t) - C$$

$$q(\mathbf{y}\,|\,t) = \exp(-\ell_t(y)/T)/Z_t$$

- use loss to regularize CRF
- think of loss as ranking all predictions
- if not putting all mass on $p(y_t|x_t)$, use loss to decide how to distribute excess mass on other configurations

# Behavior of CRF Loss Functions

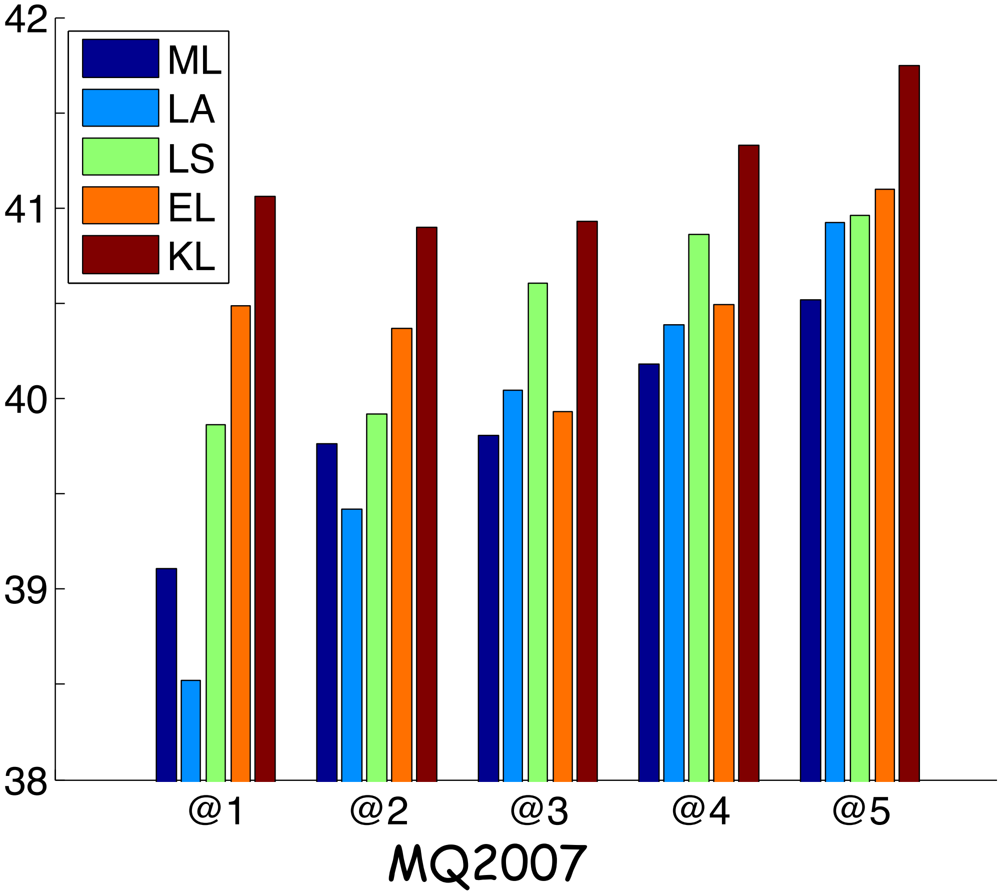


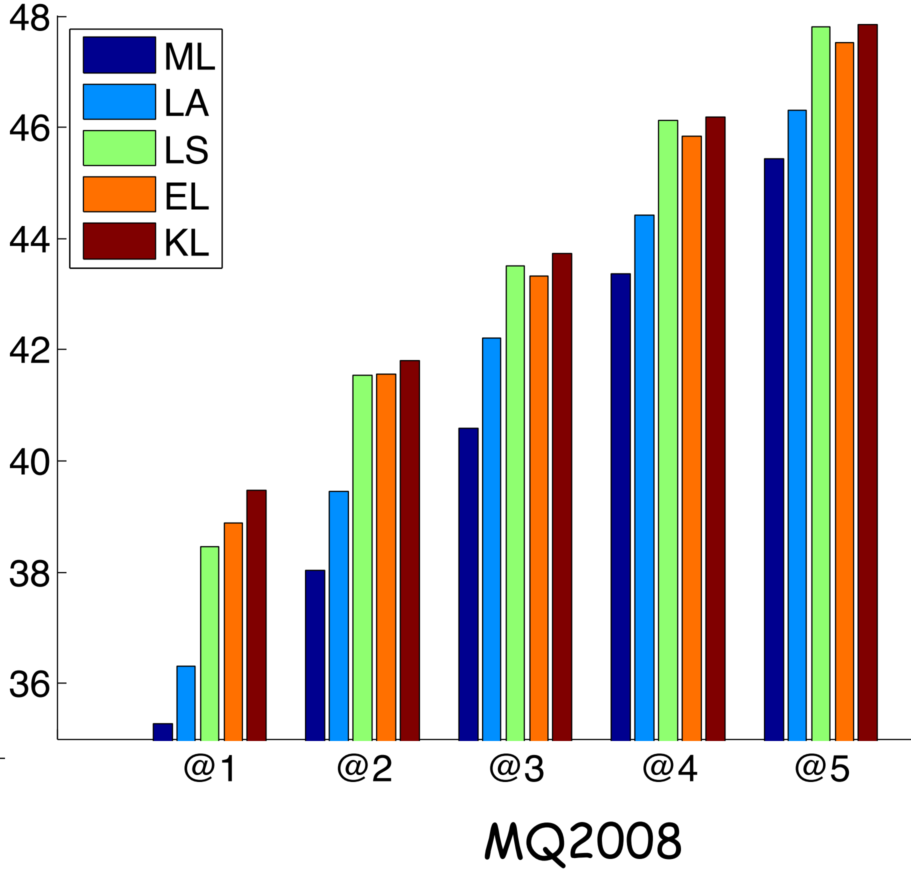
# Ranking Experiments: LETOR 4.0

Ranking problem: **x** = features of documents relevant to query;
**y** = permutation of the documents

- Interesting: complex output space; multiple ground truths
- Performance metric

$$NDCG@K(\mathbf{y},\mathbf{r}_t) = N\sum_{i=1}^{K}\frac{r_{ti}\log(2)}{\log(1+y_i)}$$



# Final Wrap Up

- CRFs benefit from loss-sensitive training
- Tractable to incorporate variety of losses, including slack-scaling
- Analog of KL for SSVMs?