Discrete Inference and Learning
Lecture 5

MVA

2018 – 2019

http://thoth.inrialpes.fr/~alahari/disinflearn

Slides based on material from Nikos Komodakis, M. Pawan Kumar

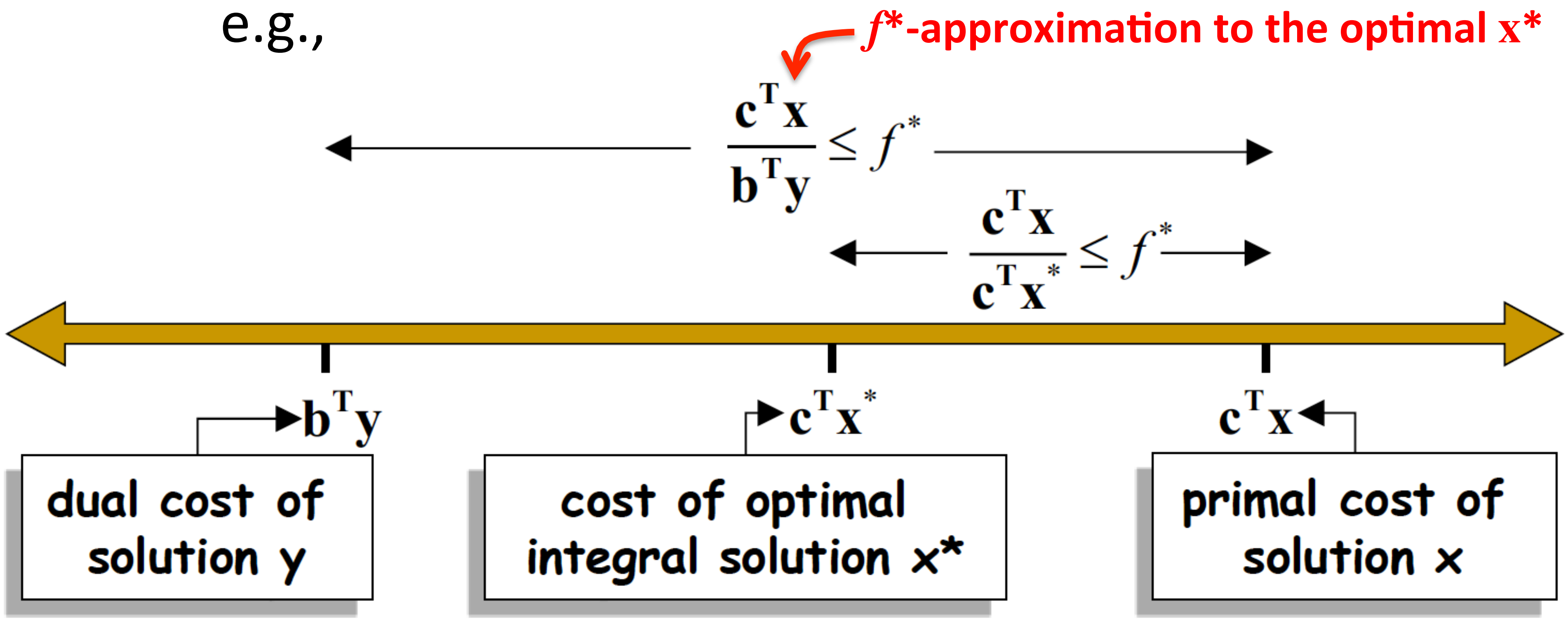
Outline

- Previous class (Dec 2018)
  - Primal-dual schema
  - Dual decomposition

- Today
  - Recap of the course
  - Learning parameters

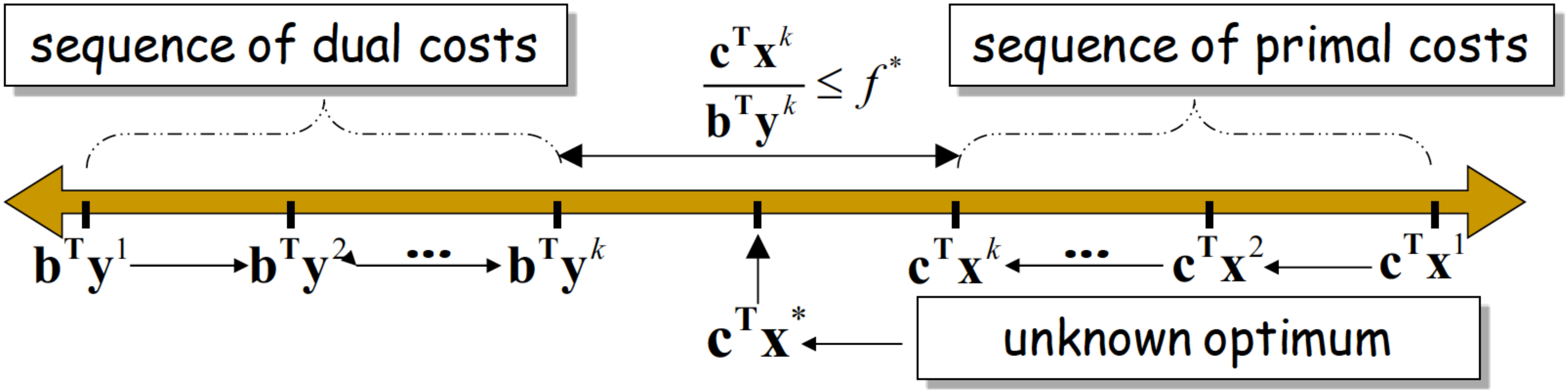
Primal-dual schema

- Goal: Find integral-primal solution $\mathbf{x}$, feasible dual solution $\mathbf{y}$,
  - such that their primal-dual costs are “close enough”, e.g.,



Primal-dual schema

- Works iteratively



- Easier to use relaxed complementary slackness, instead of working directly with costs

Primal-dual schema

- Relaxed complementary slackness

primal LP: $\min \mathbf{c}^T\mathbf{x}$

s.t. $\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$

dual LP: $\max \mathbf{b}^T\mathbf{y}$

s.t. $\mathbf{A}^T\mathbf{y} \leq \mathbf{c}$

Exact CS: $\forall 1 \leq j \leq n: \quad x_j > 0 \Rightarrow \sum_{i=1}^{m} a_{ij} y_i = c_j$

Relaxed CS: $\forall 1 \leq j \leq n: \quad x_j > 0 \Rightarrow \sum_{i=1}^{m} a_{ij} y_i \geq c_j / f_j$

Dual decomposition

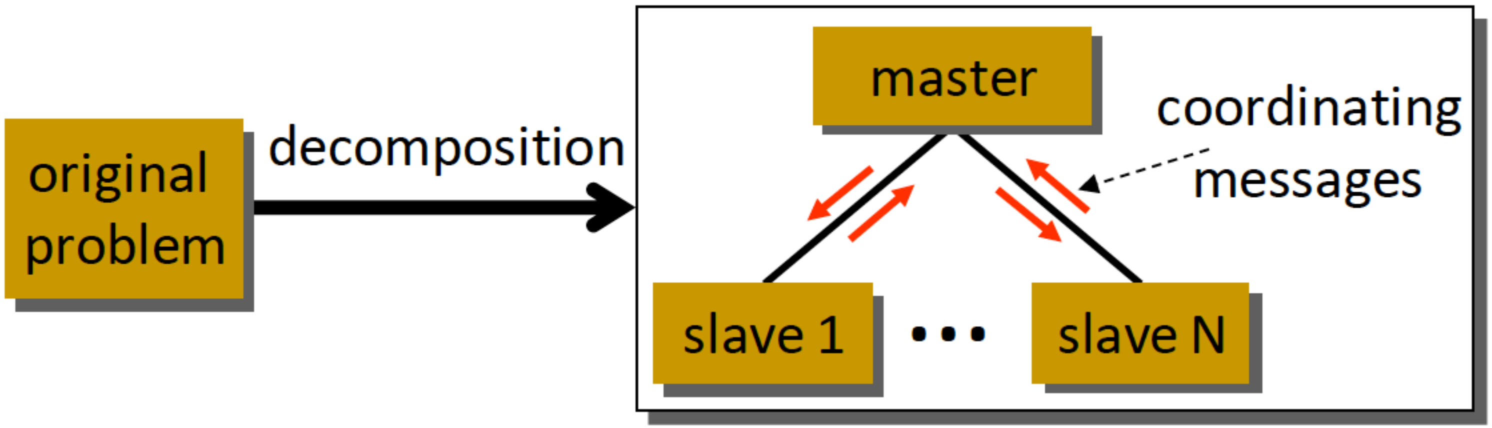
- Reduces MRF optimization to a simple projected subgradient method

- Combines solutions from sub-problems in a principled and optimal manner

- Applies to a wide variety of cases

Dual decomposition

- Decomposition into subproblems (slaves)
- Coordination of slaves by a master process



Dual decomposition

- Master
  - updates the parameters of the slave-MRFs by "averaging" the solutions returned by the slaves
  - tries to achieve consensus among all slave-MRFs
  - e.g., if a certain node is already assigned the same label by all minimizers, the master does not touch the MRF potentials of that node.

Comparison: TRW and DD

TRW	**DD**
Fast	Slow
Local Maximum	Global Maximum
Requires Min-Marginals	Requires MAP Estimate

Outline

- Recap of the course
- Learning parameters

Conditional Random Fields (CRFs)

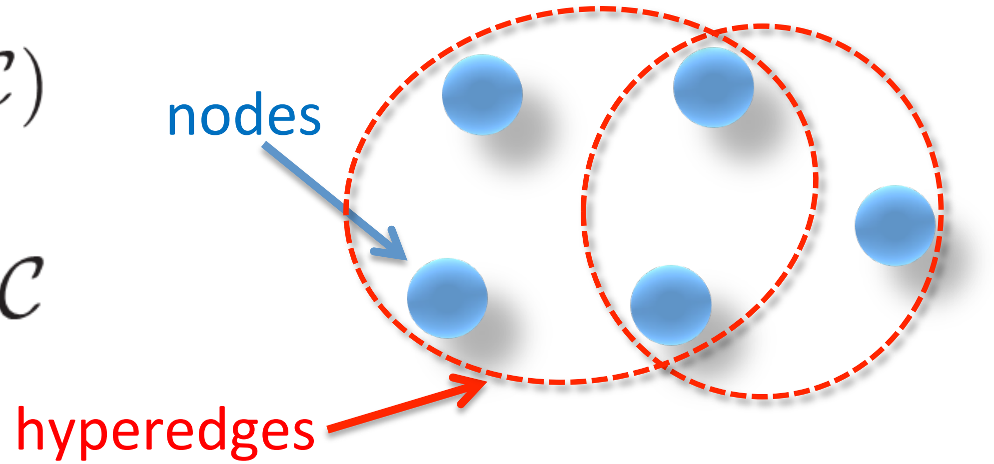
- Ubiquitous in computer vision
  - segmentation | stereo matching
    optical flow | image restoration
    image completion | object detection/localization
    ...
- and beyond
  - medical imaging, computer graphics, digital communications, physics…
- Really powerful formulation

Conditional Random Fields (CRFs)

- Key task: inference/optimization for CRFs/MRFs
- Extensive research for more than 20 years
- Lots of progress
- Many state-of-the-art methods:
  - Graph-cut based algorithms
  - Message-passing methods
  - LP relaxations
  - Dual Decomposition
  - ….

MAP inference for CRFs/MRFs

- Hypergraph $G = (\mathcal{V}, \mathcal{C})$
  - Nodes $\mathcal{V}$
  - Hyperedges/cliques $\mathcal{C}$



- High-order MRF energy minimization problem

$$\mathrm{MRF}_G(\mathbf{U}, \mathbf{H}) \equiv \min_{\mathbf{x}} \sum_{q \in \mathcal{V}} \underbrace{U_q(x_q)}_{\substack{\text{unary potential} \\ \text{(one per node)}}} + \sum_{c \in \mathcal{C}} \underbrace{H_c(\mathbf{x}_c)}_{\substack{\text{high-order potential} \\ \text{(one per clique)}}}$$

CRF training

- But how do we choose the CRF potentials?

- Through training
  - Parameterize potentials by **w**
  - Use training data to <u>learn</u> correct **w**

- Characteristic example of structured output learning [Taskar], [Tsochantaridis, Joachims]

- Equally, if not more, important than MAP inference
  - Better optimize correct energy (even approximately)
  - Than optimize wrong energy exactly

Outline

- Supervised Learning
- Probabilistic Methods
- Loss-based Methods
- Results

Image Classification

Is this an urban or rural area?

Input: **d**

Output: $\mathbf{x} \in \{-1,+1\}$

Image Classification

Is this scan healthy or unhealthy?

Input: **d**

Output: $\mathbf{x} \in \{-1,+1\}$

Image Classification

Labeling **X** = **x**

Label set **L** = {-1,+1}

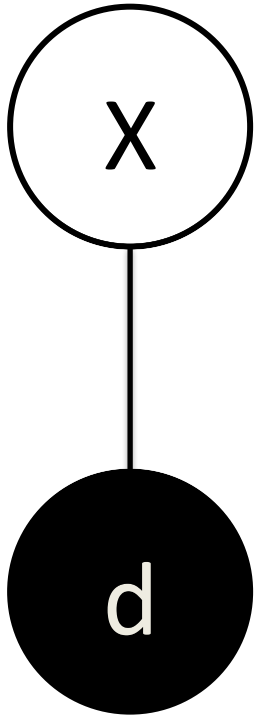


Image Classification

Which city is this?

Input: **d** Output: $\mathbf{x} \in \{1,2,\ldots,h\}$

Image Classification

What type of tumor does this scan contain?

Input: **d** Output: $\mathbf{x} \in \{1,2,\ldots,h\}$

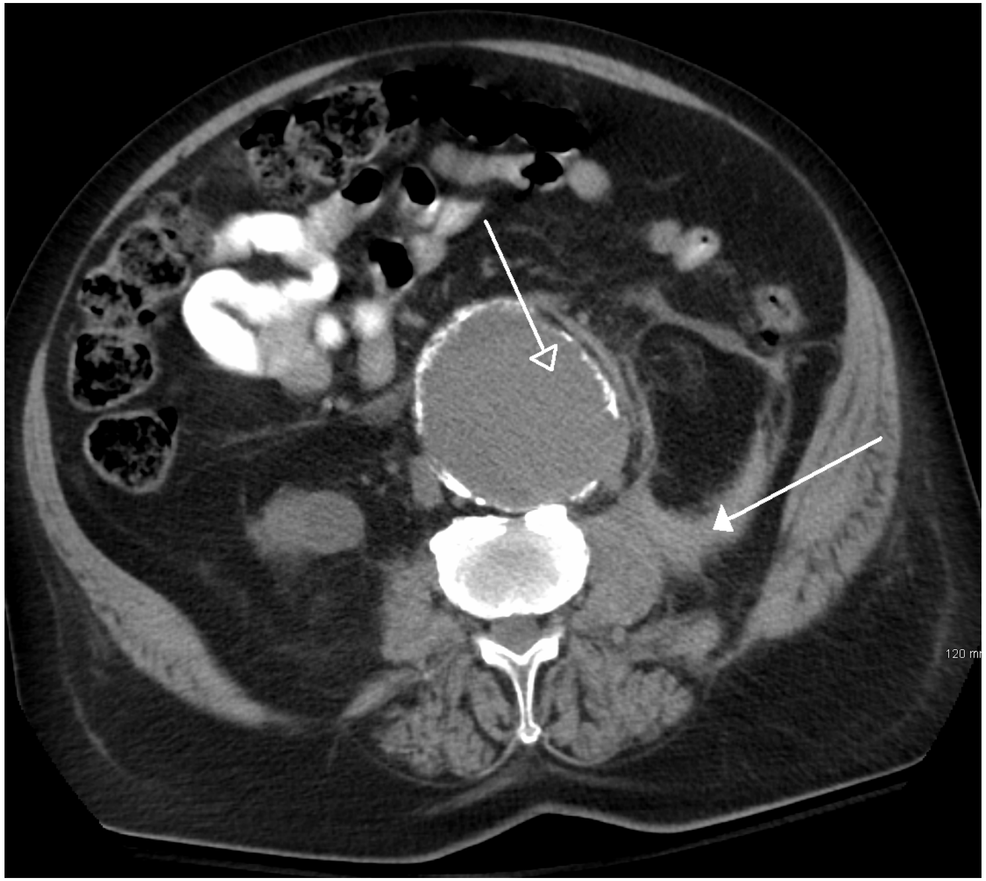
Object Detection

Where is the object in the image?

Input: **d**

Output: $\mathbf{x} \in \{\text{Pixels}\}$

Object Detection



Where is the rupture in the scan?

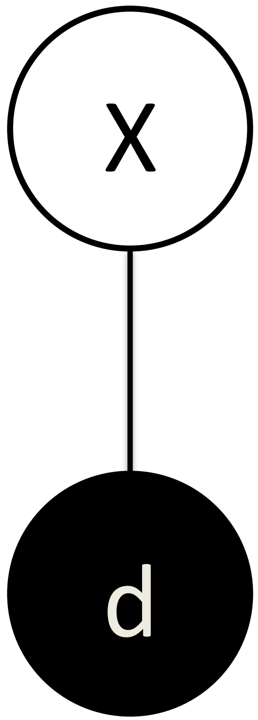
Input: **d**

Output: $\mathbf{x} \in$ {Pixels}

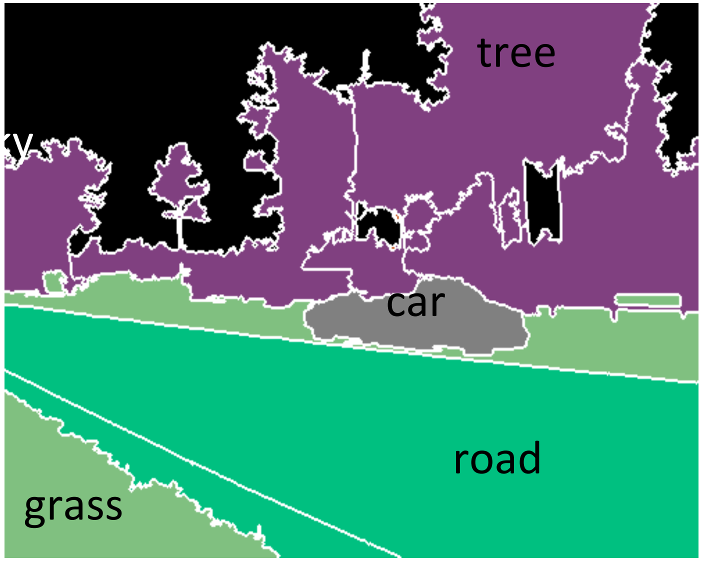
Object Detection

Labeling **X** = **x**

Label set **L** = {1, 2, ..., h}



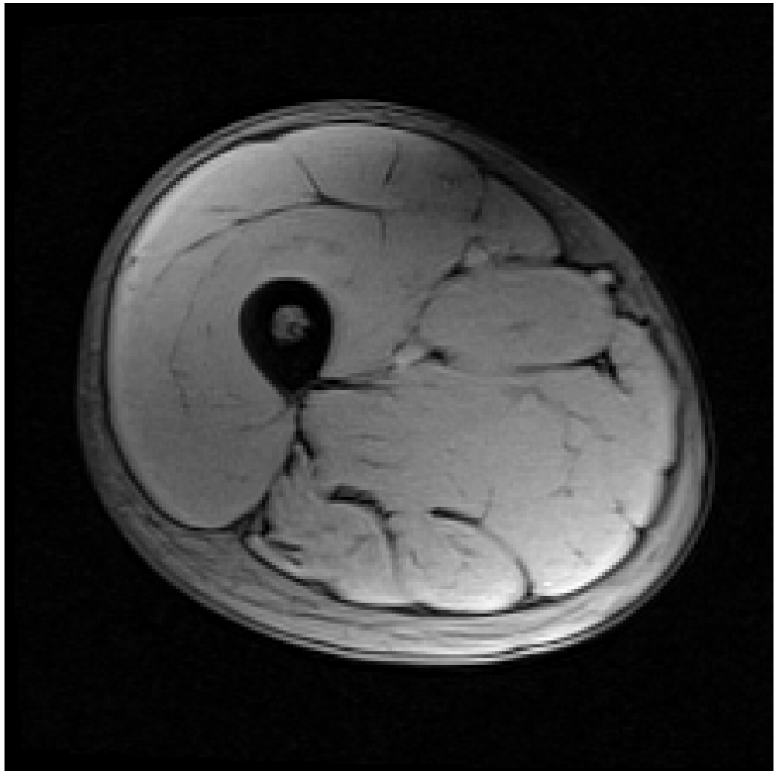
Segmentation

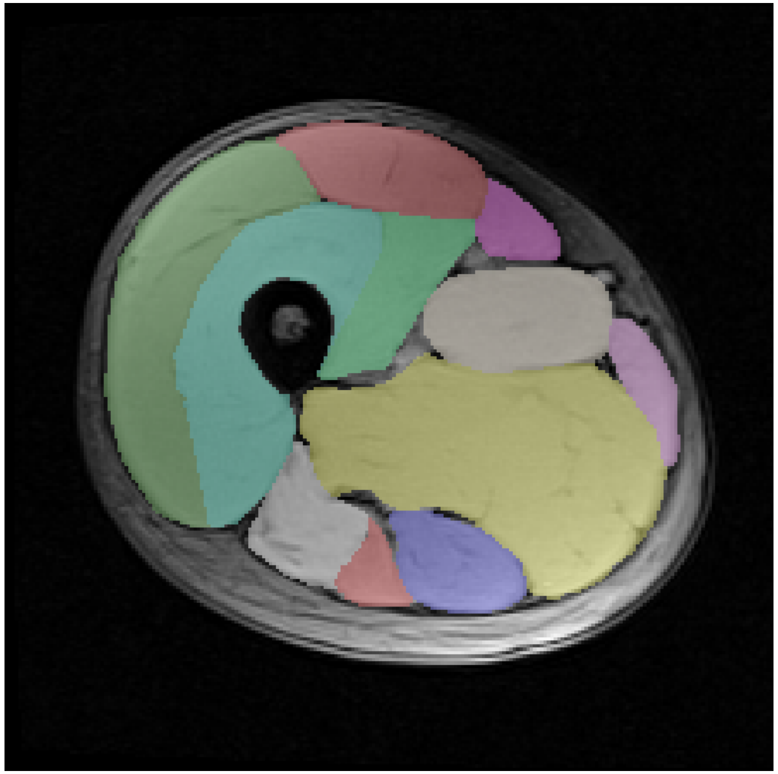


What is the semantic class of each pixel?

Input: $\mathbf{d}$ Output: $\mathbf{x} \in \{1,2,\ldots,h\}^{|\text{Pixels}|}$

Segmentation





What is the muscle group of each pixel?

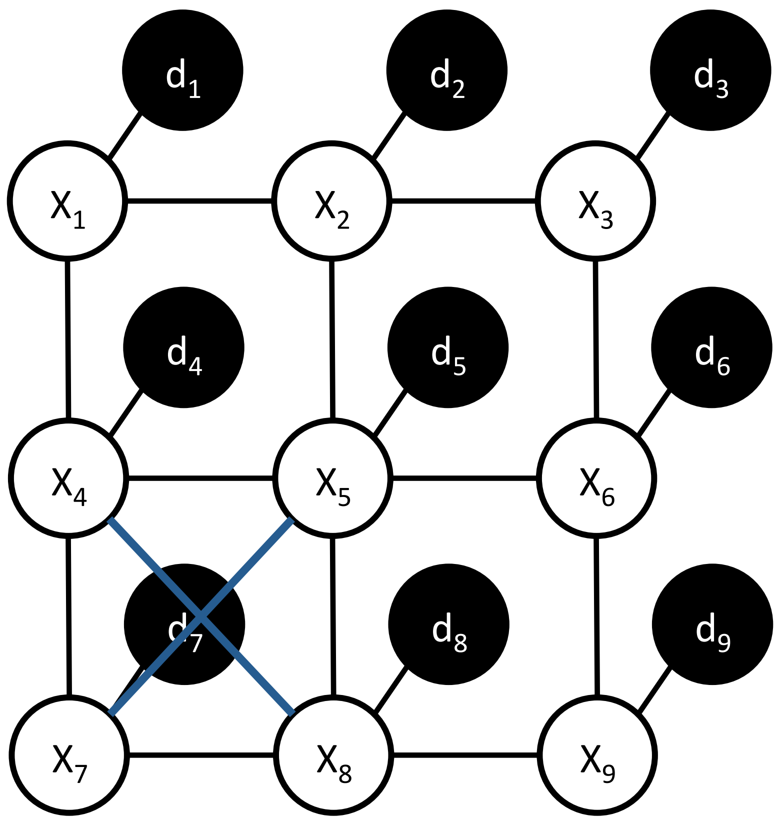
Input: **d**

Output: $\mathbf{x} \in \{1,2,\ldots,h\}^{|\text{Pixels}|}$

Segmentation

Labeling $\mathbf{X} = \mathbf{x}$

Label set $\mathbf{L} = \{1, 2, \ldots, h\}$



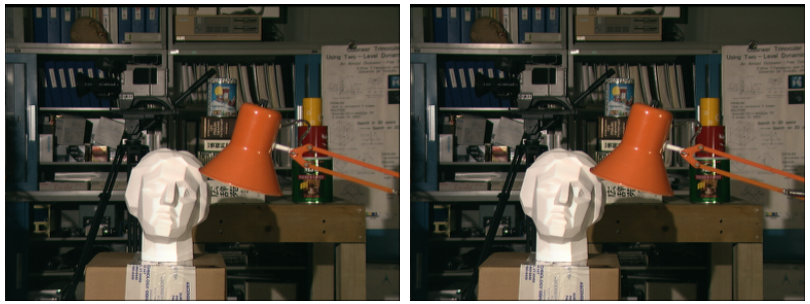
CRF training

- Stereo matching:
  - Z: left, right image
  - X: disparity map

Goal of training:

estimate proper **w**

$f:$



Z

X

$$f = \arg\min_{\mathbf{x}} \mathrm{MRF}_G(\mathbf{x}; \mathbf{u}, \mathbf{h})$$

parameterized by **w**

CRF training

- Denoising:
  - Z: noisy input image
  - X: denoised output image

Goal of training:

estimate proper **w**

f :

Z

X

$$f = \arg\min_{\mathbf{x}} \mathrm{MRF}_G(\mathbf{x}; \mathbf{u}, \mathbf{h})$$

parameterized by **w**

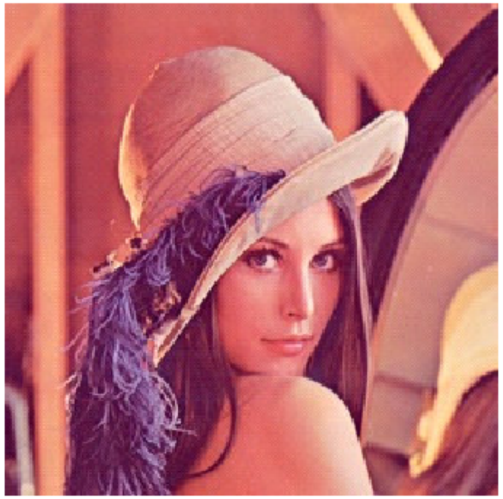
CRF training

- Object detection:
  - Z: input image
  - X: position of object parts

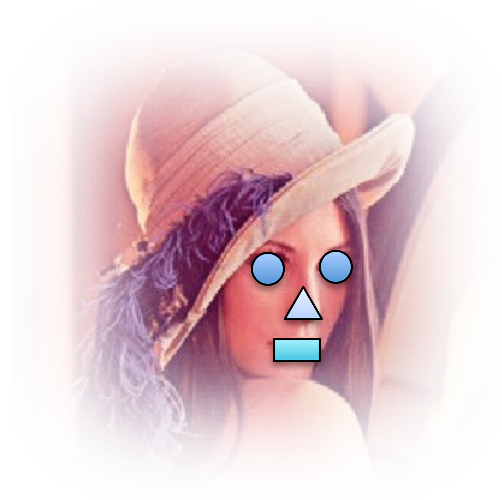
Goal of training:

estimate proper **w**

$f:$



Z



X

$$f = \arg\min_{\mathbf{x}} \mathrm{MRF}_G(\mathbf{x}; \mathbf{u}, \mathbf{h})$$

parameterized by **w**

CRF training (some further notation)

$$\mathrm{MRF}_G(\mathbf{x};\mathbf{u}^k,\mathbf{h}^k)=\sum_p u_p^k(x_p)+\sum_c h_c^k(\mathbf{x}_c)$$

$$u_p^k(x_p)=\mathbf{w}^T g_p(x_p,\mathbf{z}^k),\quad h_c^k(\mathbf{x}_c)=\mathbf{w}^T g_c(\mathbf{x}_c,\mathbf{z}^k)$$

vector valued feature functions

$$\mathrm{MRF}_G(\mathbf{x};\mathbf{w},\mathbf{z}^k)=\mathbf{w}^T\left(\sum_p g_p(x_p,\mathbf{z}^k)+\sum_c g_c(\mathbf{x}_c,\mathbf{z}^k)\right)=\mathbf{w}^T g(\mathbf{x},\mathbf{z}^k)$$

Learning formulations

Risk minimization

$$\hat{\mathbf{x}}^k = \arg\min_{\mathbf{x}} \mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k)$$

$$\min_{\mathbf{w}} \sum_{k=1}^{K} \Delta\left(\mathbf{x}^k, \hat{\mathbf{x}}^k\right)$$

K training samples $\left\{(\mathbf{x}^k, \mathbf{z}^k)\right\}_{k=1}^{K}$

Regularized Risk minimization

$$\hat{\mathbf{x}}^k = \arg\min_{\mathbf{x}} \mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k)$$

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} \Delta\left(\mathbf{x}^k, \hat{\mathbf{x}}^k\right)$$

$$R(\mathbf{w}) = ||\mathbf{w}||^2, \;\; ||\mathbf{w}||_1, \;\; \text{etc.}$$

Regularized Risk minimization

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right)$$

Replace Δ(.) with easier to handle upper bound L_G (e.g., convex w.r.t. **w**)

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} \Delta\left(\mathbf{x}^k, \hat{\mathbf{x}}^k\right)$$

Choice 1: Hinge loss

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right)$$

$$\boxed{L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) - \min_{\mathbf{x}}\left(\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k)\right)}$$

- Upper bounds Δ(.)
- Leads to **max-margin learning**

Max-margin learning

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_k \xi_k$$

subject to the constraints:

$$\mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) \leq \mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k) + \xi_k$$

energy of ground truth

any other energy

desired margin

slack

Max-margin learning

CONSTRAINED

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_k \xi_k$$

subject to the constraints:

$$\mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) \leq \mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k) + \xi_k$$

or equivalently

UNCONSTRAINED

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_k \xi_k$$

$$\xi_k = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) - \min_{\mathbf{x}} \left(\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k)\right)$$

Choice 2: logistic loss

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right)$$

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) + \underbrace{\log \sum_{\mathbf{x}} e^{-\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k)}}_{\text{partition function}}$$

- Can be shown to lead to **maximum likelihood learning**

Max-margin vs Maximum-likelihood

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \overbrace{\boxed{\mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k)} \boxed{- \min_{\mathbf{x}}} \left(\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k)\right)}^{\text{max-margin}}$$

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \underbrace{\boxed{\mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k)} + \boxed{\log \sum_{\mathbf{x}} e} ^{-\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k)}}_{\text{maximum likelihood}}$$

Max-margin vs Maximum-likelihood

$$L_G\left(\mathbf{x}^k,\mathbf{z}^k;\mathbf{w}\right) = \overbrace{\boxed{\mathrm{MRF}_G(\mathbf{x}^k;\mathbf{w},\mathbf{z}^k)} + \boxed{\max_{\mathbf{x}}}\left(-\mathrm{MRF}_G(\mathbf{x};\mathbf{w},\mathbf{z}^k) + \Delta(\mathbf{x},\mathbf{x}^k)\right)}^{\text{max-margin}}$$

soft-max

$$L_G\left(\mathbf{x}^k,\mathbf{z}^k;\mathbf{w}\right) = \underbrace{\boxed{\mathrm{MRF}_G(\mathbf{x}^k;\mathbf{w},\mathbf{z}^k)} + \boxed{\log\sum_{\mathbf{x}} e}^{-\mathrm{MRF}_G(\mathbf{x};\mathbf{w},\mathbf{z}^k)}}_{\text{maximum likelihood}}$$

Solving the learning formulations

Maximum-likelihood learning

$$\min_{\mathbf{w}} \frac{\mu}{2}||\mathbf{w}||^2 + \sum_{k=1}^{K} L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right)$$

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) + \underbrace{\log \sum_{\mathbf{x}} e^{-\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k)}}_{\text{partition function}}$$

- Differentiable & convex
- Global optimum via gradient descent, for example

Maximum-likelihood learning

$$\min_{\mathbf{w}} \frac{\mu}{2}||\mathbf{w}||^2 + \sum_{k=1}^{K} L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right)$$

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) + \log \sum_{\mathbf{x}} e^{-\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k)}$$

gradient

$$\nabla_{\mathbf{w}} = \mathbf{w} + \sum_k \left(g(\mathbf{x}^k, \mathbf{z}^k) - \sum_{\mathbf{x}} p(\mathbf{x}|w, \mathbf{z}^k) g(\mathbf{x}, \mathbf{z}^k) \right)$$

Recall that: $\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) = \mathbf{w}^T g(\mathbf{x}, \mathbf{z}^k)$

Maximum-likelihood learning

$$\min_{\mathbf{w}} \frac{\mu}{2}||\mathbf{w}||^2 + \sum_{k=1}^{K} L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right)$$

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) + \log \sum_{\mathbf{x}} e^{-\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k)}$$

gradient $\longrightarrow \nabla_{\mathbf{w}} = \mathbf{w} + \sum_k \left(g(\mathbf{x}^k, \mathbf{z}^k) - \underbrace{\sum_{\mathbf{x}} p(\mathbf{x}|w, \mathbf{z}^k) g(\mathbf{x}, \mathbf{z}^k)} \right)$

- Requires MRF probabilistic inference
- **NP-hard** (exponentially many $\mathbf{x}$): approximation via loopy-BP ?

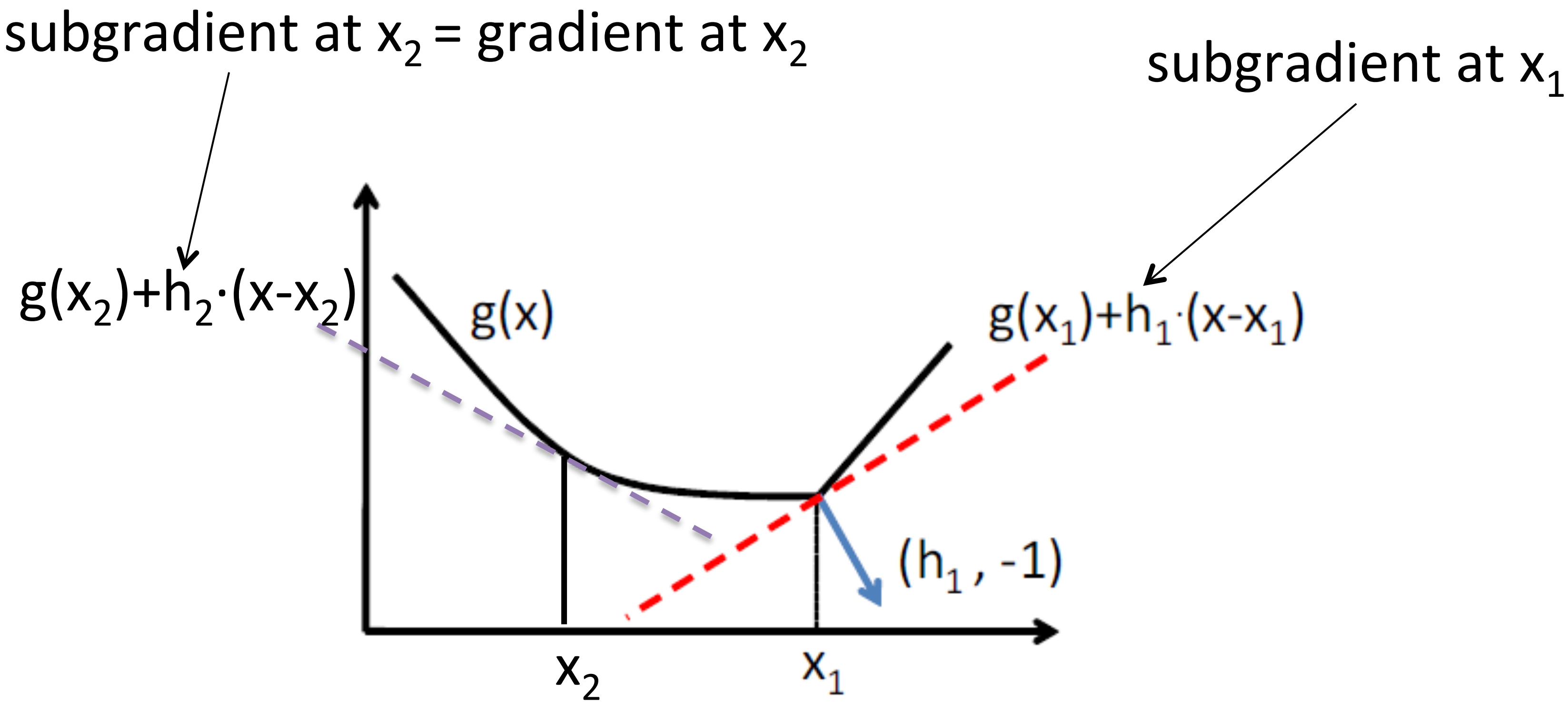
Max-margin learning (UNCONSTRAINED)

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right)$$

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) - \min_{\mathbf{x}}\left(\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k)\right)$$

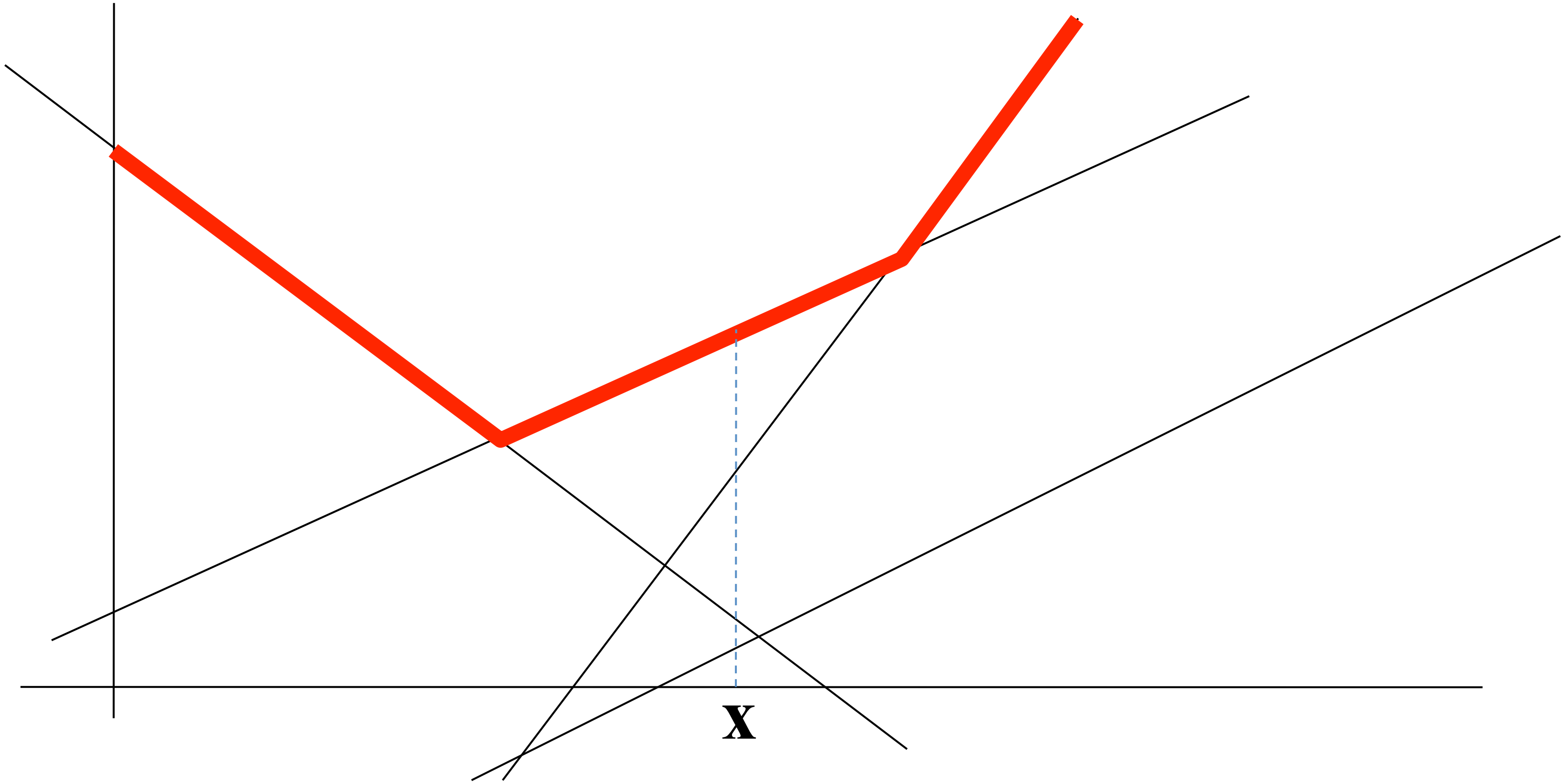
- Convex but non-differentiable
- Global optimum via **subgradient method**

Subgradient



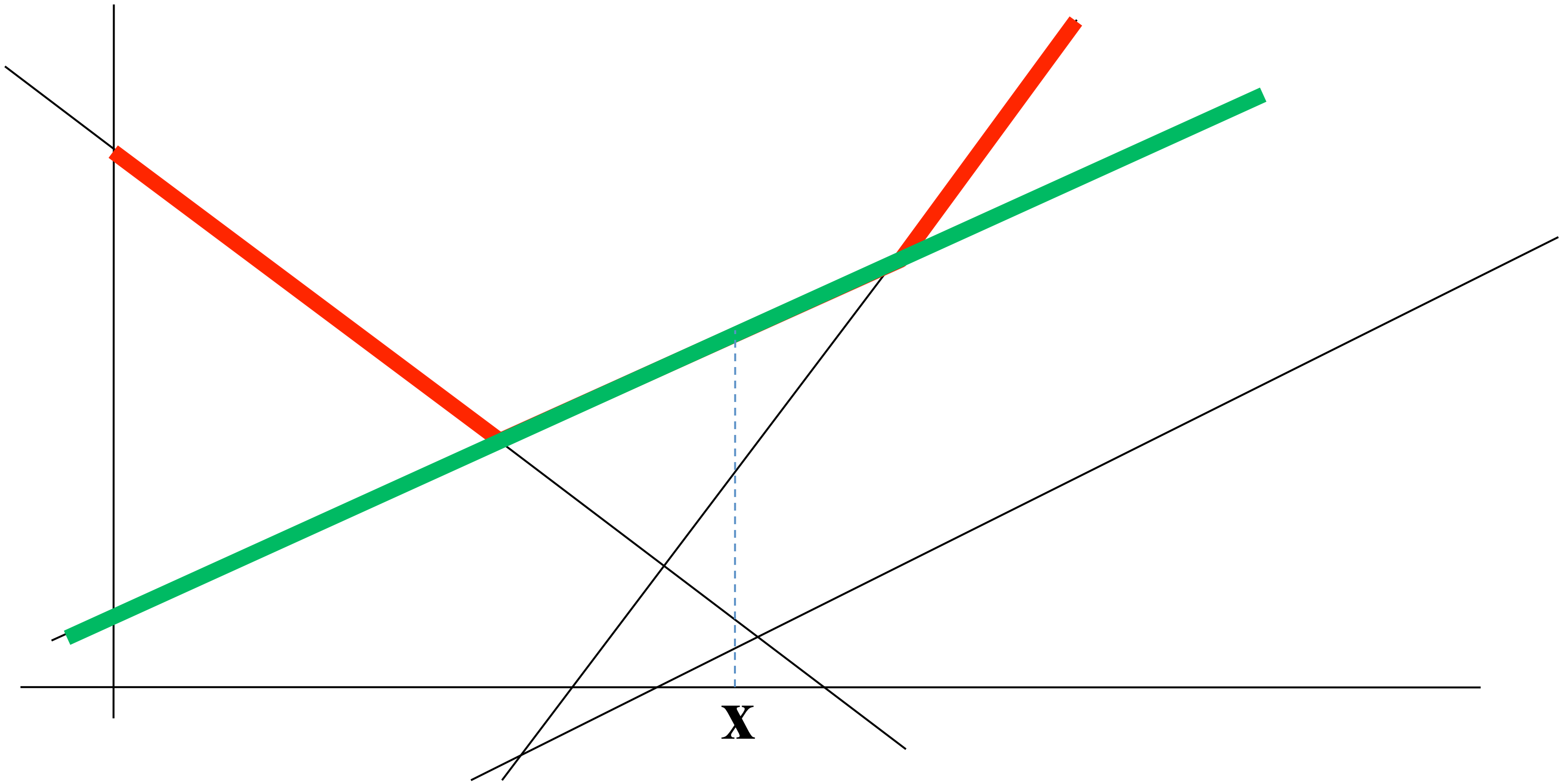
Subgradient

Lemma. *Let* $f(\cdot) = \max_{m=1,\ldots,M} f_m(\cdot)$, *with* $f_m(\cdot)$ *convex and differentiable. A subgradient of* f *at* $\mathbf{y}$ *is given by* $\nabla f_{\hat{m}}(\mathbf{y})$, *where* $\hat{m}$ *is any index for which* $f(\mathbf{y}) = f_{\hat{m}}(\mathbf{y})$.



Subgradient

Lemma. *Let* $f(\cdot) = \max_{m=1,\ldots,M} f_m(\cdot)$, *with* $f_m(\cdot)$ *convex and differentiable. A subgradient of* f *at* $\mathbf{y}$ *is given by* $\nabla f_{\hat{m}}(\mathbf{y})$, *where* $\hat{m}$ *is any index for which* $f(\mathbf{y}) = f_{\hat{m}}(\mathbf{y})$.



Subgradient

Lemma. *Let* $f(\cdot) = \max_{m=1,\dots,M} f_m(\cdot)$, *with* $f_m(\cdot)$ *convex and differentiable. A subgradient of* f *at* $\mathbf{y}$ *is given by* $\nabla f_{\hat{m}}(\mathbf{y})$, *where* $\hat{m}$ *is any index for which* $f(\mathbf{y}) = f_{\hat{m}}(\mathbf{y})$.

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) - \min_{\mathbf{x}} \left(\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k)\right)$$

$$\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) = \mathbf{w}^T g(\mathbf{x}, \mathbf{z}^k)$$

subgradient of L_G = $g(\mathbf{x}^k, \mathbf{z}^k) - g(\hat{\mathbf{x}}^k, \mathbf{z}^k)$

$$\hat{\mathbf{x}}^k = \arg\min_{\mathbf{x}} \left(\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k)\right)$$

Max-margin learning (UNCONSTRAINED)

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right)$$

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) - \min_{\mathbf{x}}\left(\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k)\right)$$

Subgradient algorithm
Repeat 1. compute global minimizers $\hat{\mathbf{x}}^k$ at current $\mathbf{w}$ 2. compute **total subgradient** at current $\mathbf{w}$ 3. update $\mathbf{w}$ by taking a step in the negative total subgradient direction **until convergence**

total subgr. $= \mathrm{subgradient}_{\mathbf{w}}[R(\mathbf{w})] + \sum_k \left(g(\mathbf{x}^k, \mathbf{z}^k) - g(\hat{\mathbf{x}}^k, \mathbf{z}^k)\right)$

Max-margin learning (UNCONSTRAINED)

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right)$$

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) - \boxed{\min_{\mathbf{x}}\left(\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k)\right)}$$

Stochastic subgradient algorithm

Repeat

1. pick k at random
2. compute global minimizer $\hat{\mathbf{x}}^k$ at current $\mathbf{w}$
3. compute **partial subgradient** at current $\mathbf{w}$
4. update $\mathbf{w}$ by taking a step in the negative partial subgradient direction

until convergence

MRF-MAP estimation per iteration
(unfortunately NP-hard)

partial subgradient = $\mathrm{subgradient}_{\mathbf{w}}[R(\mathbf{w})] + g(\mathbf{x}^k, \mathbf{z}^k) - g(\hat{\mathbf{x}}^k, \mathbf{z}^k)$

Max-margin learning (CONSTRAINED)

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_k \xi_k$$

subject to the constraints:

$$\mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) \leq \mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k) + \xi_k$$

Max-margin learning (CONSTRAINED)

$$\min_{\mathbf{w}} \frac{\mu}{2}||\mathbf{w}||^2 + \sum_k \xi_k$$

subject to the constraints:

$$\mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) \leq \mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k) + \xi_k$$

linear in $\mathbf{w}$

- Quadratic program (great!)
- But exponentially many constraints (not so great)

Max-margin learning (CONSTRAINED)

- What if we use only a small number of constraints?

  - Resulting QP can be solved
  - But solution may be infeasible

- **Constraint generation** to the rescue
  - only few constraints **active** at optimal solution !! (variables much fewer than constraints)
  - Given the active constraints, rest can be ignored
  - Then let us try to find them!

Constraint generation

1. Start with some constraints
2. Solve QP
3. Check if solution is feasible w.r.t. to **all** constraints
4. If yes, we are done!
5. If not, pick a violated constraint and add it to the current set of constraints. Repeat from step 2.

   (optionally, we can also remove inactive constraints)

Constraint generation

- **Key issue:** we must always be able to find a violated constraint if one exists

- Recall the constraints for max-margin learning

$$\mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) \leq \mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k) + \xi_k$$

- To find violated constraint, we therefore need to compute:

$$\hat{\mathbf{x}}^k = \arg\min_{\mathbf{x}} \left(\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k) \right)$$

(just like subgradient method!)

Constraint generation

1. Initialize set of constraints C to empty
2. Solve QP using current constraints C and obtain new $(\mathbf{w},\xi)$
3. Compute global minimizers $\hat{\mathbf{x}}^k$ at current $\mathbf{w}$
4. For each k, if the following constraint is violated then add it to set C:
$$\mathrm{MRF}_G(\mathbf{x}^k;\mathbf{w},\mathbf{z}^k) \leq \mathrm{MRF}_G(\hat{\mathbf{x}}^k;\mathbf{w},\mathbf{z}^k) - \Delta(\hat{\mathbf{x}}^k,\mathbf{x}^k) + \xi_k$$
5. If no new constraint was added then terminate. Otherwise go to step 2.

MRF-MAP estimation **per sample** (unfortunately **NP-hard**)

Max-margin learning (CONSTRAINED)

$$\min_{\mathbf{w}} \frac{\mu}{2}||\mathbf{w}||^2 + \sum_k \xi_k$$

subject to the constraints:

$$\mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) \leq \mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k) + \xi_k$$

- Alternatively, we can solve above QP in the **dual domain**
- dual variables ↔ primal constraints
- Too many variables, but most of them zero at optimal solution
- Use a **working-set** method
  (essentially dual to constraint generation)

CRF Training via Dual Decomposition

Komodakis, CVPR 2011

CRF training

- Existing max-margin (maximum likelihood) methods:
  - ~~use MAP inference (probabilistic inference) w.r.t. an **equally complex CRF** as subroutine~~
  - ~~have to call subroutine **many times** during learning~~

- Suboptimal
  - computational efficiency ?
  - accuracy ?
  - theoretical guarantees/properties ?

- **Key issue**: can we exploit the CRF structure more aptly during training?

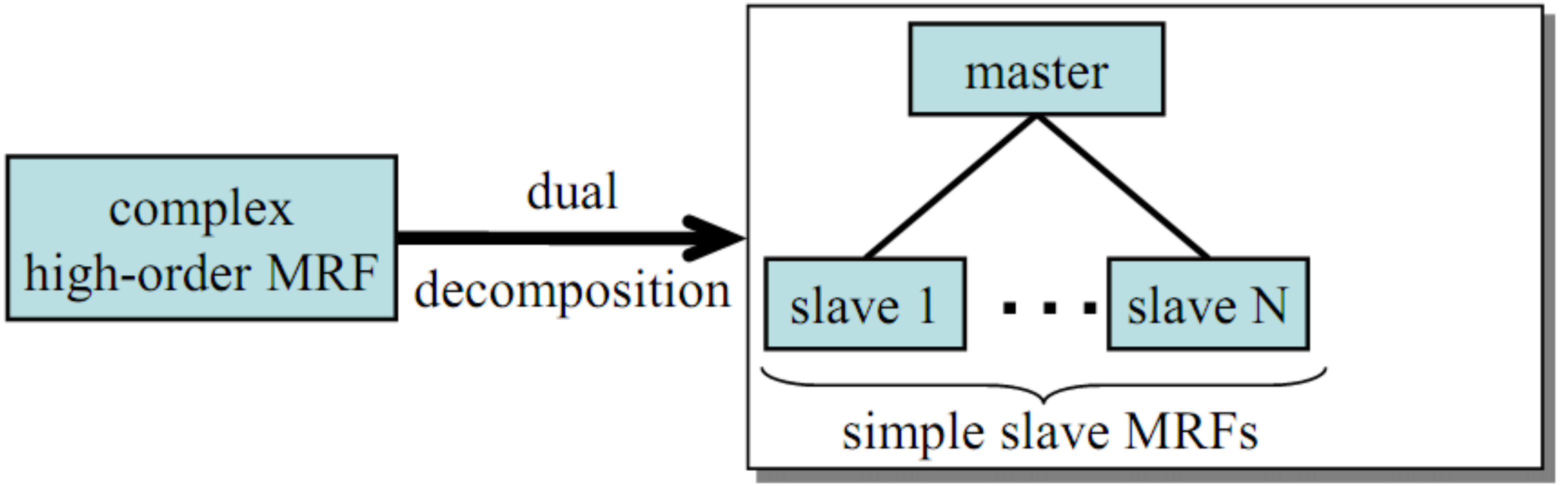
CRF Training via Dual Decomposition

- Efficient max-margin training method
- Reduces training of complex CRF to **parallel training of a series of easy-to-handle slave CRFs**
- Handles arbitrary **pairwise or higher-order** CRFs
- Uses **very efficient** projected subgradient learning scheme
- Allows hierarchy of structured prediction learning algorithms of **increasing accuracy**

Dual Decomposition for MRF Optimization (another recap)

MRF Optimization via Dual Decomposition

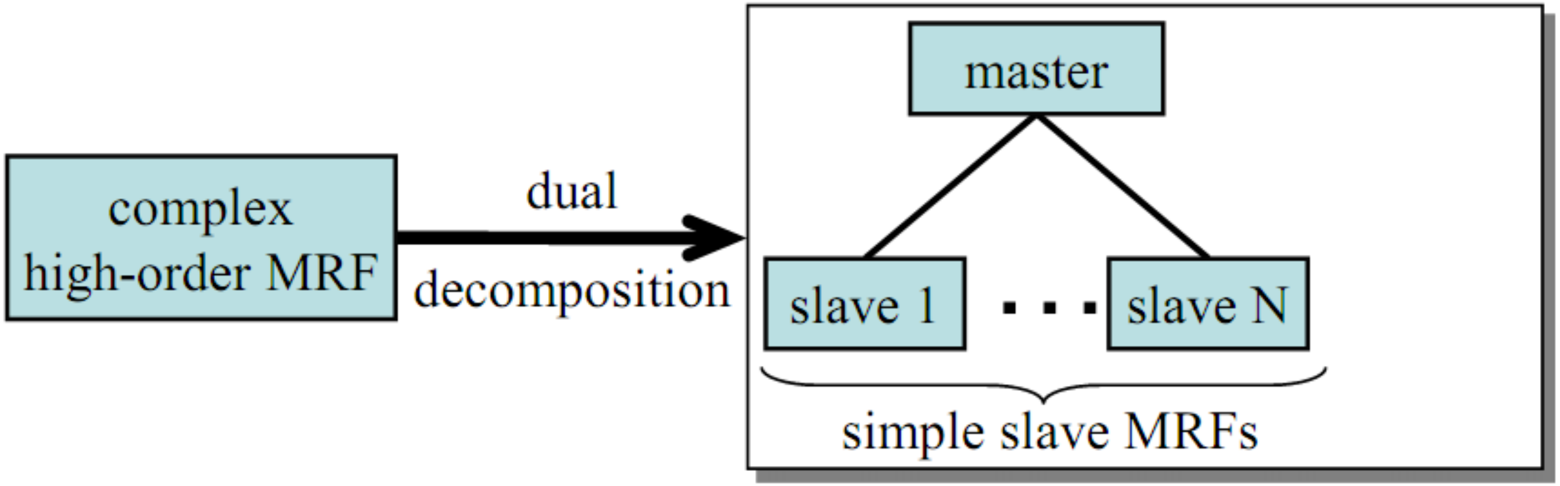
- Very general framework for MAP inference [Komodakis et al. ICCV07, PAMI11]



- Master = coordinator (has global view)
  Slaves = subproblems (have only local view)

MRF Optimization via Dual Decomposition

- Very general framework for MAP inference [Komodakis et al. ICCV07, PAMI11]

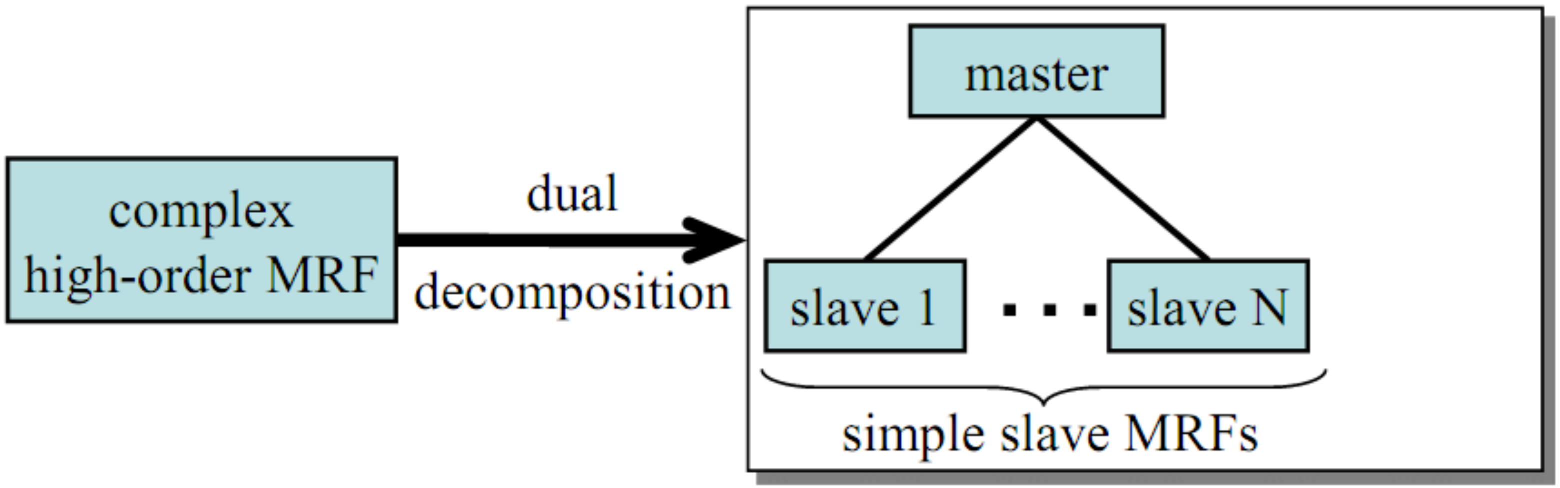


- Master = $\text{MRF}_G(\mathbf{u}, \mathbf{h})$ ⟵ (MAP-MRF on hypergraph *G*)

  = min $\text{MRF}_G(\mathbf{x}; \mathbf{u}, \mathbf{h}) := \sum_{p \in \mathcal{V}} u_p(x_p) + \sum_{c \in \mathcal{C}} h_c(\mathbf{x}_c)$

MRF Optimization via Dual Decomposition

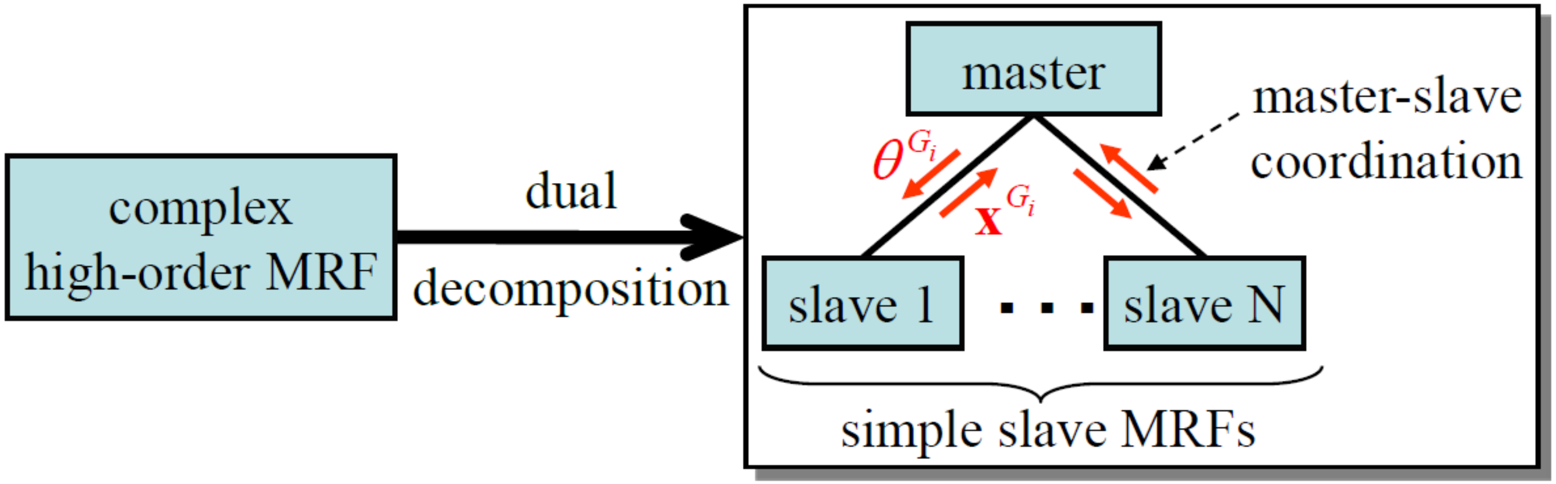
- Very general framework for MAP inference [Komodakis et al. ICCV07, PAMI11]



- Set of slaves = $\{\mathrm{MRF}_{G_i}(\boldsymbol{\theta}^i, \mathbf{h})\}$
  (MRFs on sub-hypergraphs G_i whose union covers G)
- Many other choices possible as well

MRF Optimization via Dual Decomposition

- Very general framework for MAP inference [Komodakis et al. ICCV07, PAMI11]



- Optimization proceeds in an iterative fashion via **master-slave coordination**

MRF Optimization via Dual Decomposition

Set of slave MRFs
$\{\mathrm{MRF}_{G_i}(\boldsymbol{\theta}^i, \mathbf{h})\}$

convex dual relaxation

$$\mathrm{DUAL}_{\{G_i\}}(\mathbf{u}, \mathbf{h}) = \max_{\{\boldsymbol{\theta}^i\}} \sum\nolimits_i \mathrm{MRF}_{G_i}(\boldsymbol{\theta}^i, \mathbf{h})$$
$$\text{s.t.} \sum_{i \in \mathcal{I}_p} \theta_p^i(\cdot) = u_p(\cdot)$$

For each choice of slaves, master solves (possibly different) dual relaxation

- Sum of slave energies = lower bound on MRF optimum
- Dual relaxation = maximum such bound

MRF Optimization via Dual Decomposition

Set of slave MRFs

$$\{\mathrm{MRF}_{G_i}(\boldsymbol{\theta}^i, \mathbf{h})\}$$

convex dual relaxation

$$\mathrm{DUAL}_{\{G_i\}}(\mathbf{u}, \mathbf{h}) = \max_{\{\boldsymbol{\theta}^i\}} \sum_i \mathrm{MRF}_{G_i}(\boldsymbol{\theta}^i, \mathbf{h})$$

$$\text{s.t.} \sum_{i \in \mathcal{I}_p} \theta_p^i(\cdot) = u_p(\cdot)$$

Choosing more difficult slaves $\Rightarrow$ tighter lower bounds

$\Rightarrow$ tighter dual relaxations

CRF training via Dual Decomposition

Max-margin learning via dual decomposition

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right)$$

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{w}, \mathbf{z}^k) - \min_{\mathbf{x}}\left(\mathrm{MRF}_G(\mathbf{x}; \mathbf{w}, \mathbf{z}^k) - \Delta(\mathbf{x}, \mathbf{x}^k)\right)$$

Max-margin learning via dual decomposition

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right)$$

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{u}^k, \mathbf{h}^k) - \min_{\mathbf{x}}\left(\mathrm{MRF}_G(\mathbf{x}; \mathbf{u}^k, \mathbf{h}^k) - \Delta(\mathbf{x}, \mathbf{x}^k)\right)$$

Max-margin learning via dual decomposition

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right)$$

$$L_G\left(\mathbf{x}^k, \mathbf{z}^k; \mathbf{w}\right) = \mathrm{MRF}_G(\mathbf{x}^k; \mathbf{u}^k, \mathbf{h}^k) - \min_{\mathbf{x}}\left(\mathrm{MRF}_G(\mathbf{x}; \mathbf{u}^k, \mathbf{h}^k) - \Delta(\mathbf{x}, \mathbf{x}^k)\right)$$

$$\Delta(\mathbf{x}, \mathbf{x}^k) = \sum_p \delta_p(x_p, x_p^k) + \sum_c \delta_c(\mathbf{x}_c, \mathbf{x}_c^k) \qquad \boxed{\Delta(\mathbf{x}, \mathbf{x}) = 0}$$

$$\boxed{\begin{aligned} \bar{u}_p^k(\cdot) &= u_p^k(\cdot) - \delta_p(\cdot, x_p^k) \\ \bar{h}_c^k(\cdot) &= h_c^k(\cdot) - \delta_c(\cdot, \mathbf{x}_c^k) \end{aligned}}$$

loss-augmented potentials

Max-margin learning via dual decomposition

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G(\mathbf{x}^k, \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k; \mathbf{w})$$

$$L_G(\mathbf{x}^k, \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k; \mathbf{w}) = \mathrm{MRF}_G(\mathbf{x}^k; \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k) - \min_{\mathbf{x}} \mathrm{MRF}_G(\mathbf{x}; \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k)$$

$$\Delta(\mathbf{x}, \mathbf{x}^k) = \sum_p \delta_p(x_p, x_p^k) + \sum_c \delta_c(\mathbf{x}_c, \mathbf{x}_c^k) \qquad \boxed{\Delta(\mathbf{x}, \mathbf{x}) = 0}$$

$$\boxed{\begin{aligned} \bar{u}_p^k(\cdot) &= u_p^k(\cdot) - \delta_p(\cdot, x_p^k) \\ \bar{h}_c^k(\cdot) &= h_c^k(\cdot) - \delta_c(\cdot, \mathbf{x}_c^k) \end{aligned}}$$

$$\delta_p(x_p, x_p) = 0$$
$$\delta_c(\mathbf{x}_c, \mathbf{x}_c) = 0$$

loss-augmented potentials

Max-margin learning via dual decomposition

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G(\mathbf{x}^k, \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k; \mathbf{w})$$

$$L_G(\mathbf{x}^k, \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k; \mathbf{w}) = \mathrm{MRF}_G(\mathbf{x}^k; \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k) - \min_{\mathbf{x}} \mathrm{MRF}_G(\mathbf{x}; \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k)$$

Problem

Learning objective intractable due to this term

Max-margin learning via dual decomposition

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G(\mathbf{x}^k, \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k; \mathbf{w})$$

$$L_G(\mathbf{x}^k, \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k; \mathbf{w}) = \mathrm{MRF}_G(\mathbf{x}^k; \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k) - \min_{\mathbf{x}} \mathrm{MRF}_G(\mathbf{x}; \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k)$$

Solution: approximate this term with dual relaxation from decomposition $\{G_i = (\mathcal{V}_i, \mathcal{C}_i)\}_{i=1}^{N}$

$$\min_{\mathbf{x}} \mathrm{MRF}_G(\mathbf{x}; \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k) \approx \mathrm{DUAL}_{\{G_i\}}(\bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k)$$

$$\mathrm{DUAL}_{\{G_i\}}(\bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k) = \max_{\{\boldsymbol{\theta}^{(i,k)}\}} \sum_i \mathrm{MRF}_{G_i}(\boldsymbol{\theta}^{(i,k)}, \bar{\mathbf{h}}^k)$$

$$\text{s.t.} \sum_{i \in \mathcal{I}_p} \theta_p^{(i,k)}(\cdot) = \bar{u}_p^k(\cdot)$$

Max-margin learning via dual decomposition

now

$$\min_{\mathbf{w},\{\boldsymbol{\theta}^{(i,k)}\}} R(\mathbf{w}) + \sum_k \sum_i L_{G_i}(\mathbf{x}^k, \boldsymbol{\theta}^{(i,k)}, \bar{\mathbf{h}}^k; \mathbf{w})$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{I}_p} \theta_p^{(i,k)}(\cdot) = \bar{u}_p^k(\cdot) \quad .$$

before

$$\min_{\mathbf{w}} R(\mathbf{w}) + \sum_{k=1}^{K} L_G(\mathbf{x}^k, \bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k; \mathbf{w})$$

Essentially, training of complex CRF decomposed to parallel training of easy-to-handle slave CRFs !!!

Max-margin learning via dual decomposition

- Global optimum via projected subgradient method
(slight variation of subgradient method)

Projected subgradient
Repeat 1. compute subgradient at current $\mathbf{w}$ 2. update $\mathbf{w}$ by taking a step in the negative subgradient direction 3. project into feasible set **until convergence**

Projected subgradient learning algorithm

- **Input:**
  - K training samples $\{(\mathbf{x}^k, \mathbf{z}^k)\}_{k=1}^K$
  - Hypergraph $G = (\mathcal{V}, \mathcal{C})$
    (in general hypergraphs can vary per sample)
  - Vector valued feature functions $\{g_p(\cdot,\cdot)\}$, $\{g_c(\cdot,\cdot)\}$

Projected subgradient learning algorithm

$\forall k,$ choose decomposition $\{G_i = (\mathcal{V}_i, \mathcal{C}_i)\}_{i=1}^N$ of hypergraph G

$\forall k, i,$ initialize $\boldsymbol{\theta}^{(i,k)}$ so as to satisfy $\sum_{i \in \mathcal{I}_p} \theta_p^{(i,k)}(\cdot) = \bar{u}_p^k(\cdot)$

repeat

// ***optimize slave MRFs***

$\forall k, i,$ compute minimizer $\hat{\mathbf{x}}^{(i,k)} = \arg\min_{\mathbf{x}} \mathrm{MRF}_{G_i}(\mathbf{x}; \boldsymbol{\theta}^{(i,k)}, \bar{\mathbf{h}}^k)$

// ***update*** $\mathbf{w}$

$\mathbf{w} \leftarrow \mathbf{w} - \alpha_t \cdot d\mathbf{w}$ fully specified from $\hat{\mathbf{x}}^{(i,k)}$

// ***update*** $\boldsymbol{\theta}^{(i,k)}$

$$\theta_p^{(i,k)}(\cdot) \mathrel{+}= \alpha_t \left(\left[\hat{x}_p^{(i,k)} = \cdot \right] - \frac{\sum_{j \in \mathcal{I}_p} \left[\hat{x}_p^{(j,k)} = \cdot \right]}{|\mathcal{I}_p|} \right)$$

until convergence

(we only need to know how to optimize slave MRFs !!)

Projected subgradient learning algorithm

- Resulting learning scheme:
  - ✓ Very efficient and very flexible
  - ✓ Requires from the user only to provide an optimizer for the slave MRFs
  - ✓ Slave problems freely chosen by the user
  - ✓ Easily adaptable to further exploit special structure of any class of CRFs

Choice of decompositions $\{G_i\}$

$\mathcal{F}_0$ = true loss (intractable)

$\mathcal{F}_{\{G_i\}}$ = loss when using decomposition $\{G_i\}$

- $\mathcal{F}_0 \leq \mathcal{F}_{\{G_i\}}$

  (upper bound property)

- $\{G_i\} < \{\tilde{G}_j\}$

  (hierarchy of learning algorithms)

Choice of decompositions $\{G_i\}$

- $G_{\text{single}} = \{G_c\}_{c \in \mathcal{C}}$ denotes following decomposition:
  - One slave per clique $c \in \mathcal{C}$
  - Corresponding sub-hypergraph $G_c = (\mathcal{V}_c, \mathcal{C}_c)$:

    $\mathcal{V}_c = \{p | p \in c\}, \mathcal{C}_c = \{c\}$

- Resulting slaves often easy (or even trivial) to solve even if global problem is complex and NP-hard
  - leads to widely applicable learning algorithm

- Corresponding dual relaxation is an LP
  - Generalizes well known LP relaxation for pairwise MRFs (at the core of most state-of-the-art methods)

Choice of decompositions $\{G_i\}$

- But we can do better if CRFs have special structure…
- Structure means:
  - More efficient optimizer for slaves **(speed)**
  - Optimizer that handles more complex slaves **(accuracy)**

  (Almost all known examples fall in one of above two cases)

- We are essentially adapting decomposition to exploit the structure of the problem at hand

Choice of decompositions $\{G_i\}$

- But we can do better if CRFs have special structure…
- e.g., **pattern-based** high-order potentials (for a clique c) [Komodakis & Paragios CVPR09]

$$H_c(\mathbf{x}) = \begin{cases} \psi_c(\mathbf{x}) & \text{if } \mathbf{x} \in \mathcal{P} \\ \psi_c^{\max} & \text{otherwise} \end{cases}$$

$\mathcal{P}$ subset of $\mathcal{L}^{|c|}$ (its vectors called **patterns**)

Choice of decompositions $\{G_i\}$

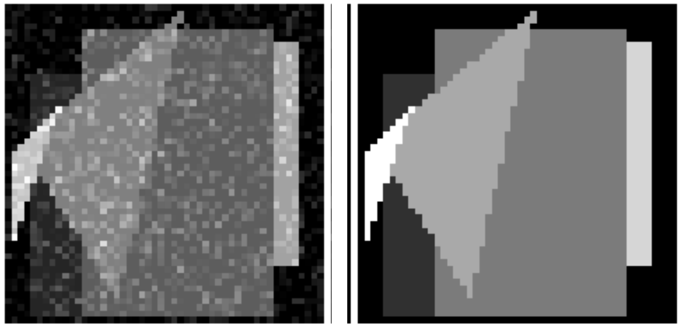
- Tree decomposition $G_{\text{tree}} = \{T_i\}_{i=1}^{N}$
  (T_i are spanning trees that cover the graph)

- No improvement in accuracy
  $\mathrm{DUAL}_{G_{\text{tree}}} = \mathrm{DUAL}_{G_{\text{single}}} \Rightarrow \mathcal{F}_{G_{\text{tree}}} = \mathcal{F}_{G_{\text{single}}}$

- But improvement in speed
  ($\mathrm{DUAL}_{G_{\text{tree}}}$ converges faster than $\mathrm{DUAL}_{G_{\text{single}}}$)

Image denoising

- Piecewise constant images

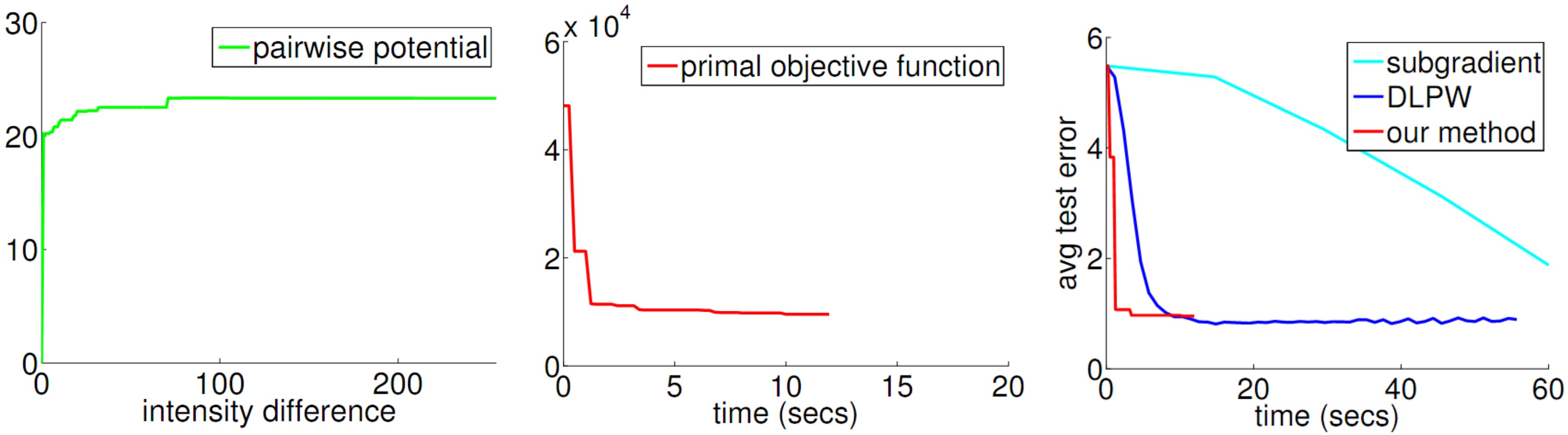


Z X

- Potentials: $u_p^k\left(x_p\right) = \left|x_p - z_p\right| \qquad h_{pq}^k\left(x_p, x_q\right) = V\left(\left|x_p - x_q\right|\right)$

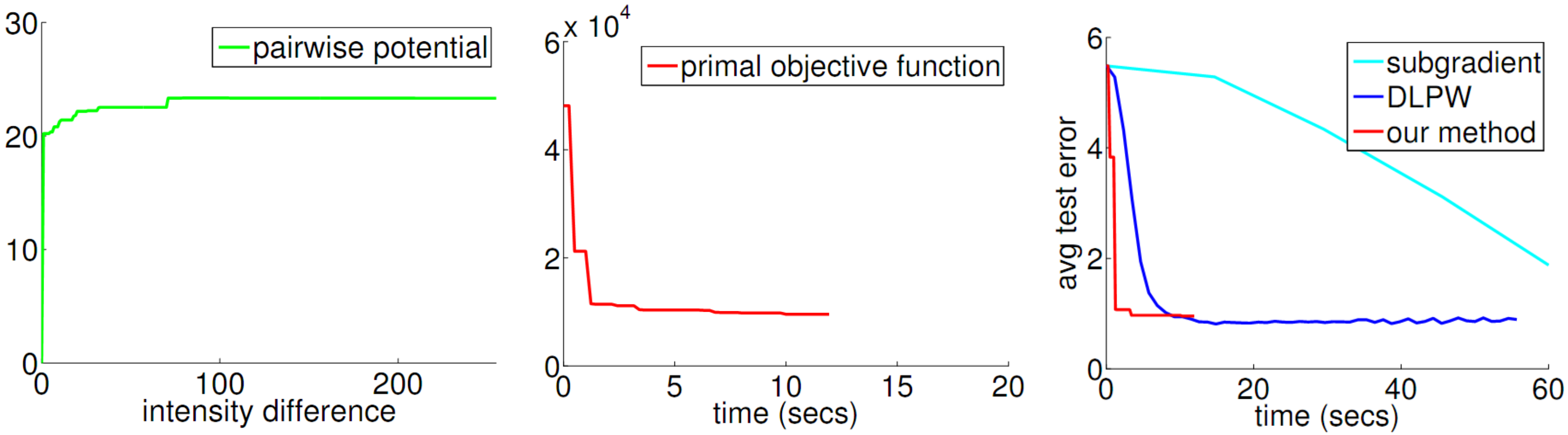
- Goal: learn pairwise potential $V(\cdot)$

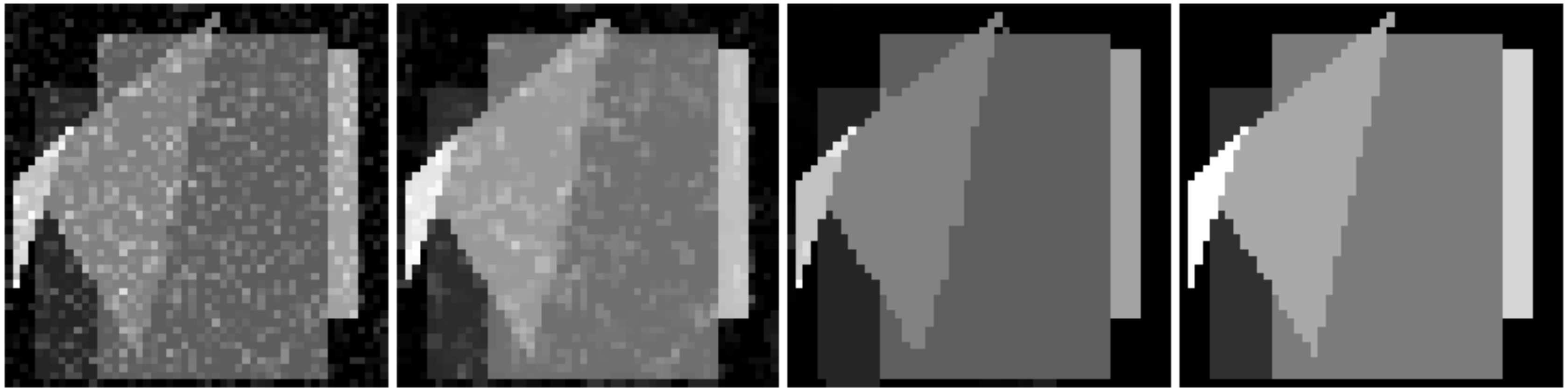
Image denoising



learnt potential

Image denoising



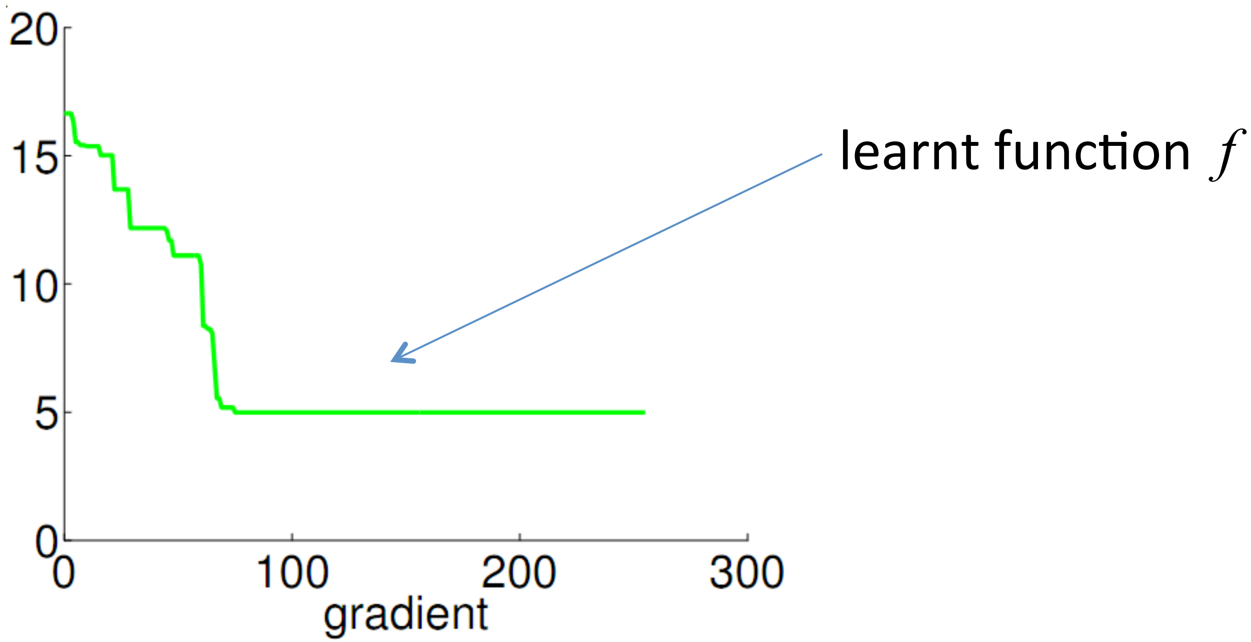


Stereo matching

- Potentials: $u_p^k\left(x_p\right) = \left|I^{left}\left(p\right) - I^{right}\left(p - x_p\right)\right|$

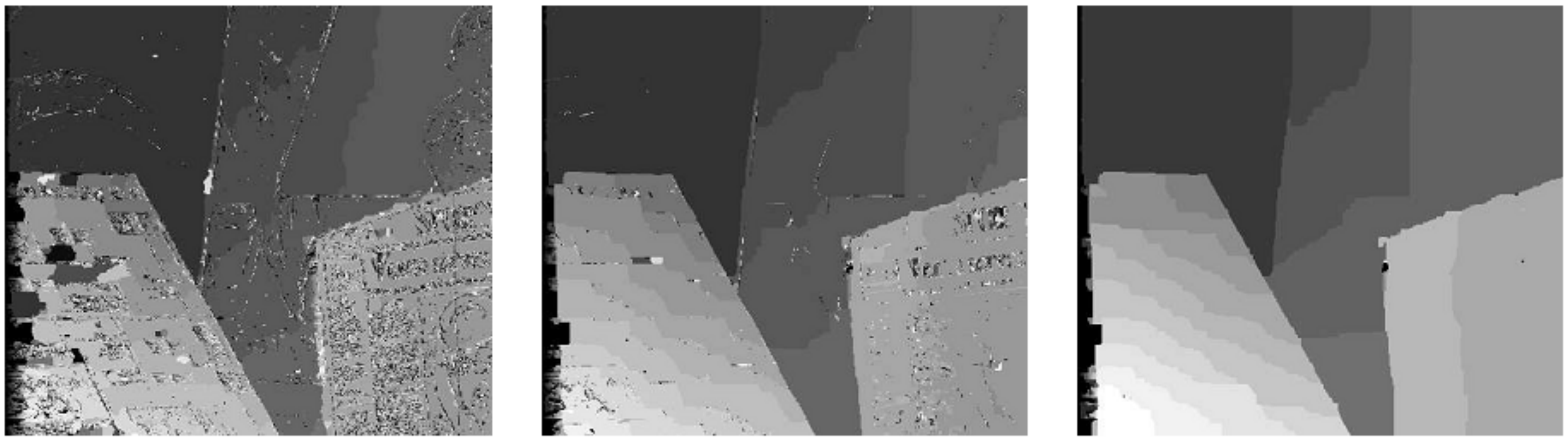
$$h_{pq}^k\left(x_p, x_q\right) = f\left(\left|\nabla I^{left}(p)\right|\right)\left[x_p \neq x_q\right]$$

- Goal: learn function $f(\cdot)$ for gradient-modulated Potts model



Stereo matching

- Potentials: $$u_p^k\left(x_p\right) = \left|I^{left}\left(p\right) - I^{right}\left(p - x_p\right)\right|$$
$$h_{pq}^k\left(x_p, x_q\right) = f\left(\left|\nabla I^{left}(p)\right|\right)\left[x_p \neq x_q\right]$$
- Goal: learn function $f(\cdot)$ for gradient-modulated Potts model

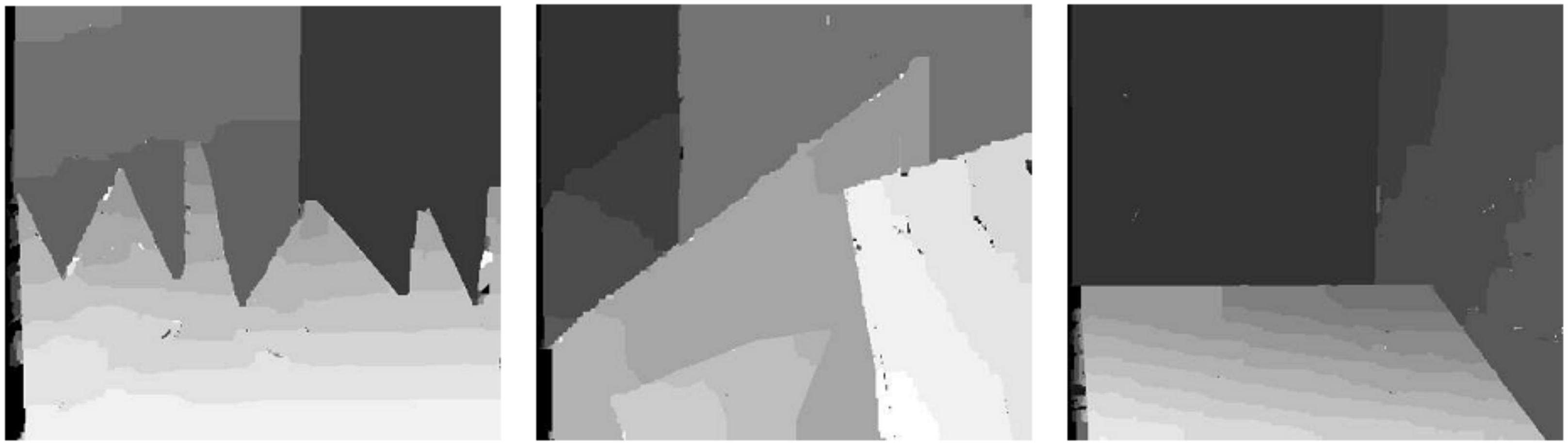


[Middlebury dataset]

"Venus" disparity using $f(\cdot)$ as estimated at different iterations of learning algorithm

Stereo matching

- Potentials: $$u_p^k\left(x_p\right) = \left|I^{left}\left(p\right) - I^{right}\left(p - x_p\right)\right|$$
$$h_{pq}^k\left(x_p, x_q\right) = f\left(\left|\nabla I^{left}(p)\right|\right)\left[x_p \neq x_q\right]$$
- Goal: learn function $f(\cdot)$ for gradient-modulated Potts model



[Middlebury dataset]

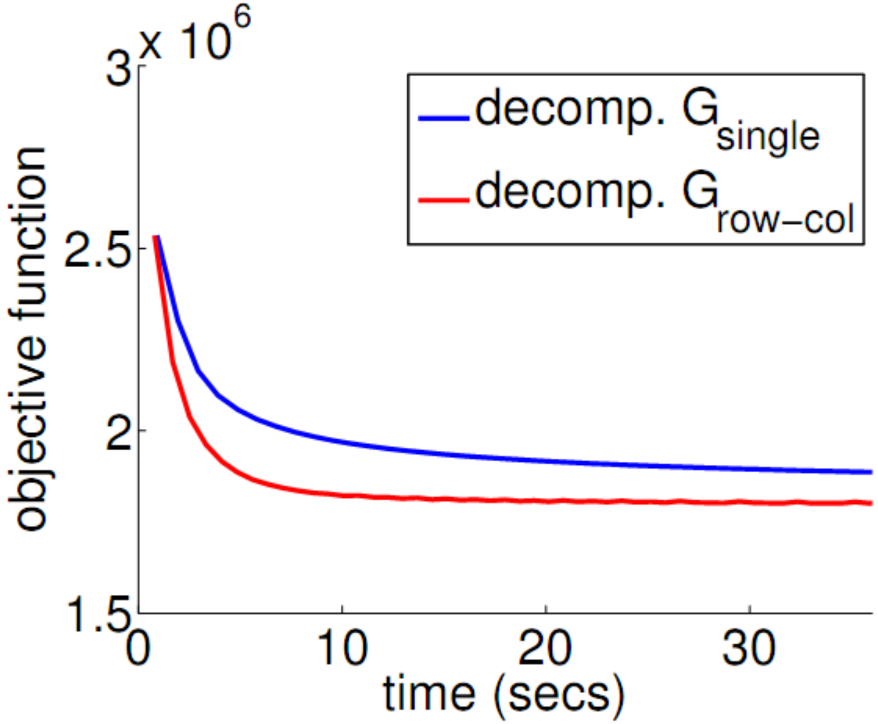
Sawtooth 4.9% | Poster 3.7% | Bull 2.8%

Stereo matching

- Potentials: $u_p^k\left(x_p\right) = \left|I^{left}\left(p\right) - I^{right}\left(p - x_p\right)\right|$

$$h_{pq}^k\left(x_p, x_q\right) = f\left(\left|\nabla I^{left}(p)\right|\right)\left[x_p \neq x_q\right]$$

- Goal: learn function $f(\cdot)$ for gradient-modulated Potts model



High-order P^n Potts model

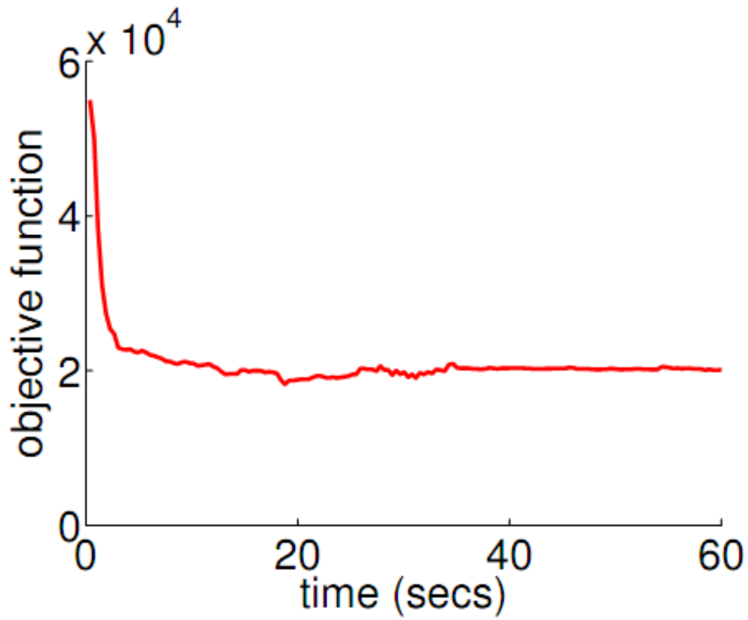
Goal: learn high order CRF with potentials given by

$$h_c(\mathbf{x}) = \begin{cases} \beta_l^c & \text{if } x_p = l,\ \forall p \in c \\ \beta_{\max}^c & \text{otherwise} \end{cases},$$

[Kohli et al. CVPR07]

$$\beta_l^c = \mathbf{w}_l \cdot z_l^c$$

Cost for optimizing slave CRF: O(|L|) ⇨ Fast training



- 100 training samples
- 50x50 grid
- clique size 3x3
- 5 labels (|L|=5)

Learning to cluster

Clustering

- A fundamental task in vision and beyond
- Typically formulated as an optimization problem based on a given distance function between datapoints
- Choice of distance crucial for the success of clustering
- **Goal 1:** learn this distance automatically based on training data
- **Goal 2:** learning should also handle the fact that the number of clusters is typically unknown at test time

Exemplar based clustering formulation

distance between
datapoints p and q

penalty for choosing q as
exemplar (cluster center)

$$\min_{Q \subseteq S} E(Q) = \sum_{p \notin Q} \min_{q \in Q} d_{p,q} + \sum_{q \in Q} d_{q,q}$$

set of exemplars
(cluster centers)

set of datapoints

The above formulation allows to:

- automatically estimate the number of clusters (i.e. size of Q)
- use arbitrary distances
  (e.g., non-metric, asymmetric, non-differentiable)

Exemplar based clustering formulation

distance between datapoints p and q

penalty for choosing q as exemplar (cluster center)

$$\min_{Q \subseteq S} E(Q) = \sum_{p \notin Q} \min_{q \in Q} d_{p,q} + \sum_{q \in Q} d_{q,q}$$

set of exemplars (cluster centers)

set of datapoints

Inference can be performed efficiently using:
Clustering via LP-based Stabilities [Komodakis et al., NIPS 2008]