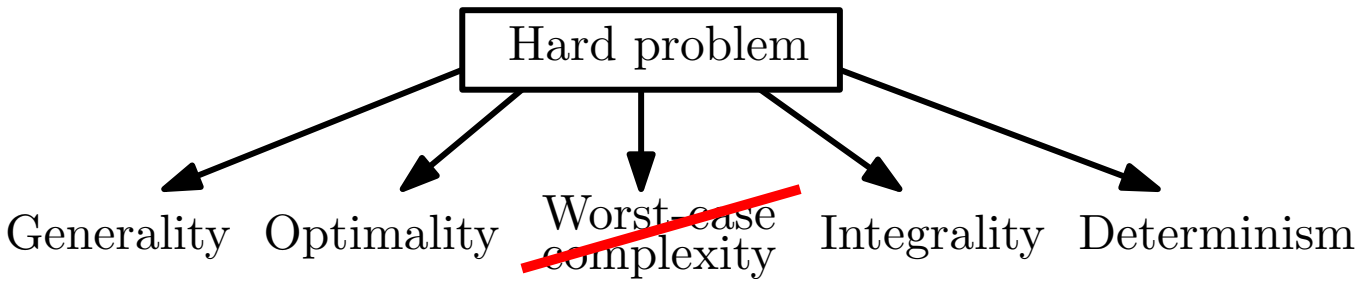
Part 7: Structured Prediction and Energy Minimization (2/2)

Sebastian Nowozin and Christoph H. Lampert

Colorado Springs, 25th June 2011


Hard problem
Generality
Optimality
Worst-case complexity
Integrality
Determinism

Giving up Worst-case Complexity

- Worst-case complexity is an asymptotic behaviour
- Worst-case complexity quantifies worst case
- Practical case might be very different
  - Issue: what is the distribution over inputs?

Popular methods with bad or unknown worst-case complexity

- Simplex Method for Linear Programming
- Hash tables
- Branch-and-bound search

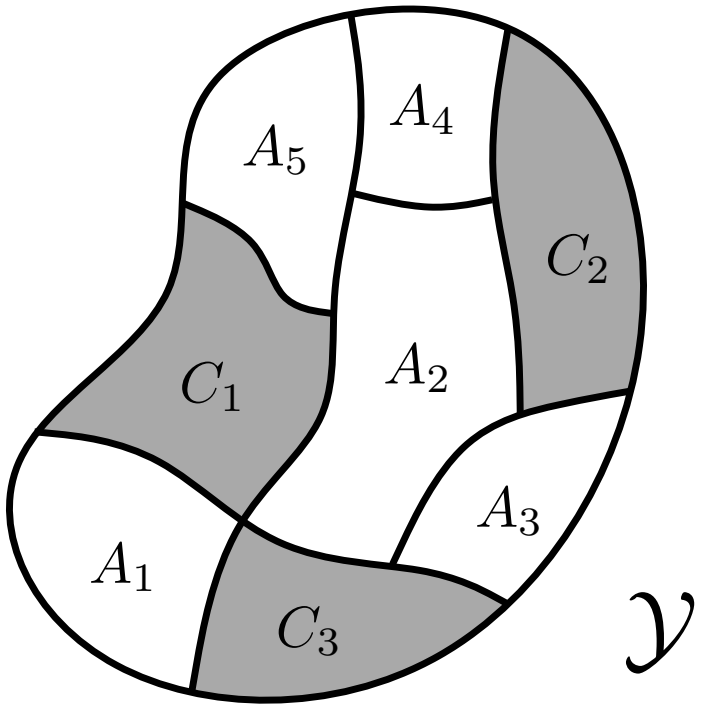
Branch-and-bound Search

- *Implicit enumeration*: globally optimal
- Choose
  1. *Partitioning* of solution space
  2. *Branching* scheme
  3. Upper and lower *bounds* over partitions

Branch and bound

- is very flexible, many tuning possibilities in partitioning, branching schemes and bounding functions,
- can be very efficient in practise,
- typically has worst-case complexity equal to *exhaustive* enumeration

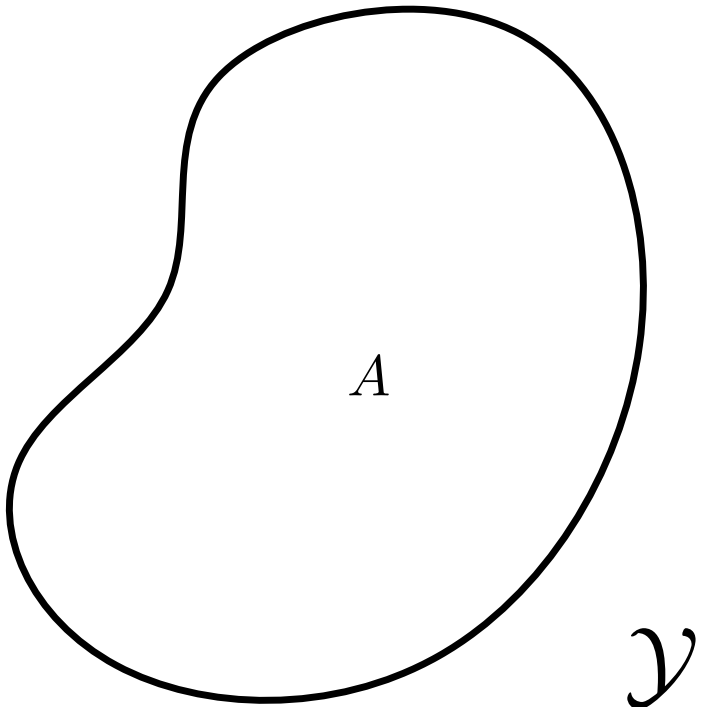
Branch-and-Bound (cont)

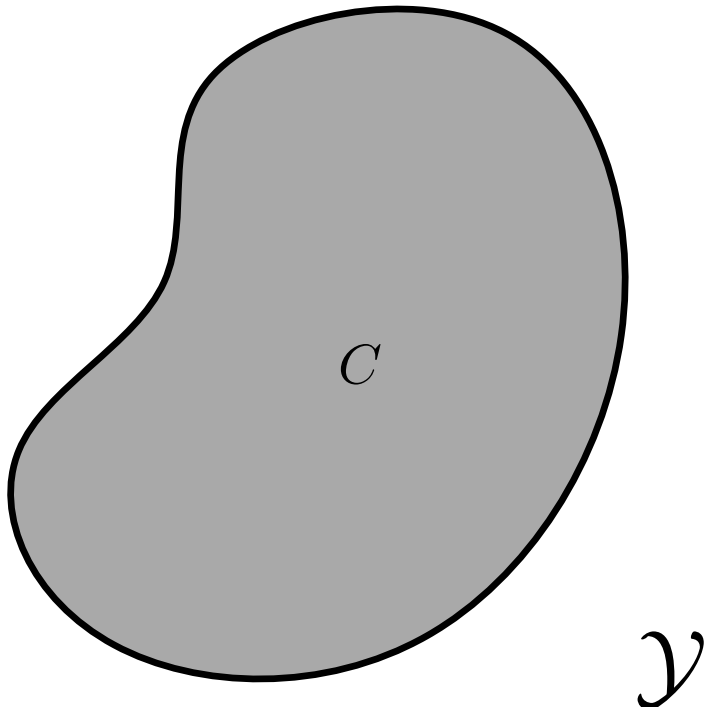


Work with partitioning of solution space $\mathcal{Y}$

- *Active* nodes (white)
- *Closed* nodes (gray)

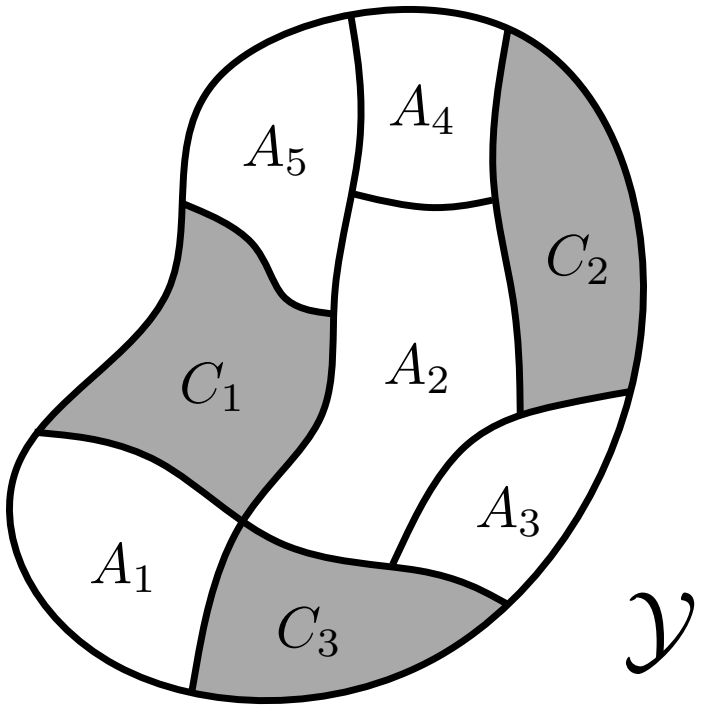
Branch-and-Bound (cont)



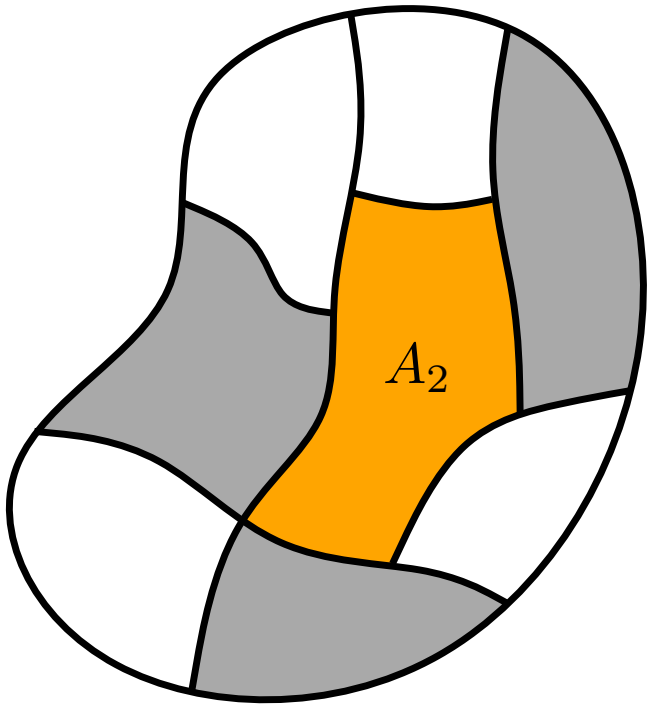


- Initially: everything active
- Goal: everything closed

Branch-and-Bound (cont)

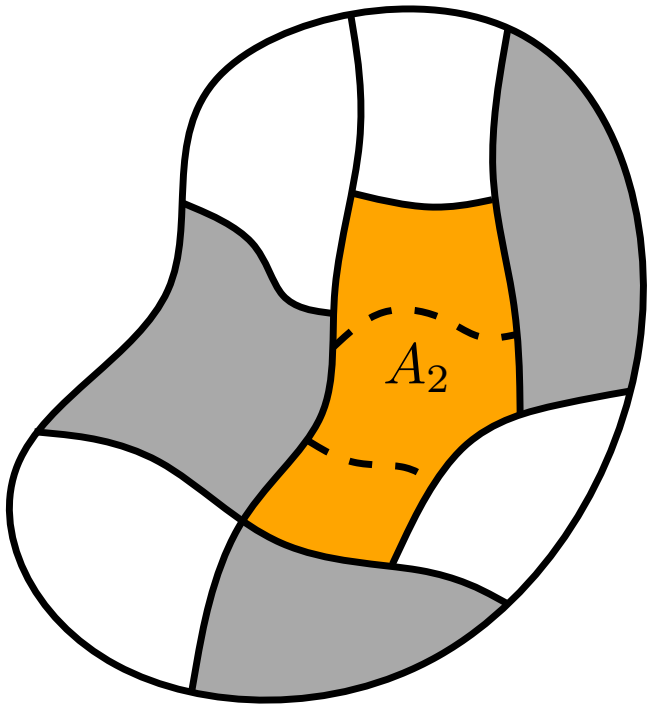


Branch-and-Bound (cont)



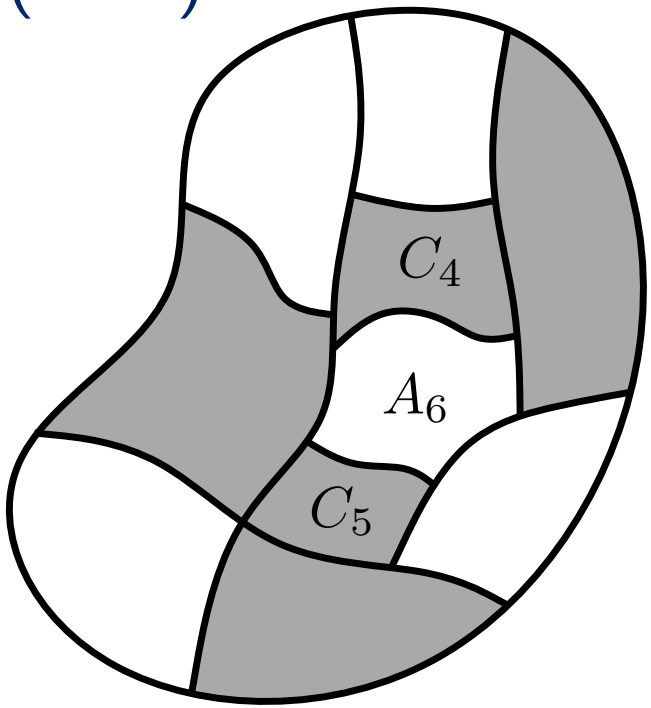
- Take an active element (A_2)

Branch-and-Bound (cont)



- Partition into two or more subsets of $\mathcal{Y}$

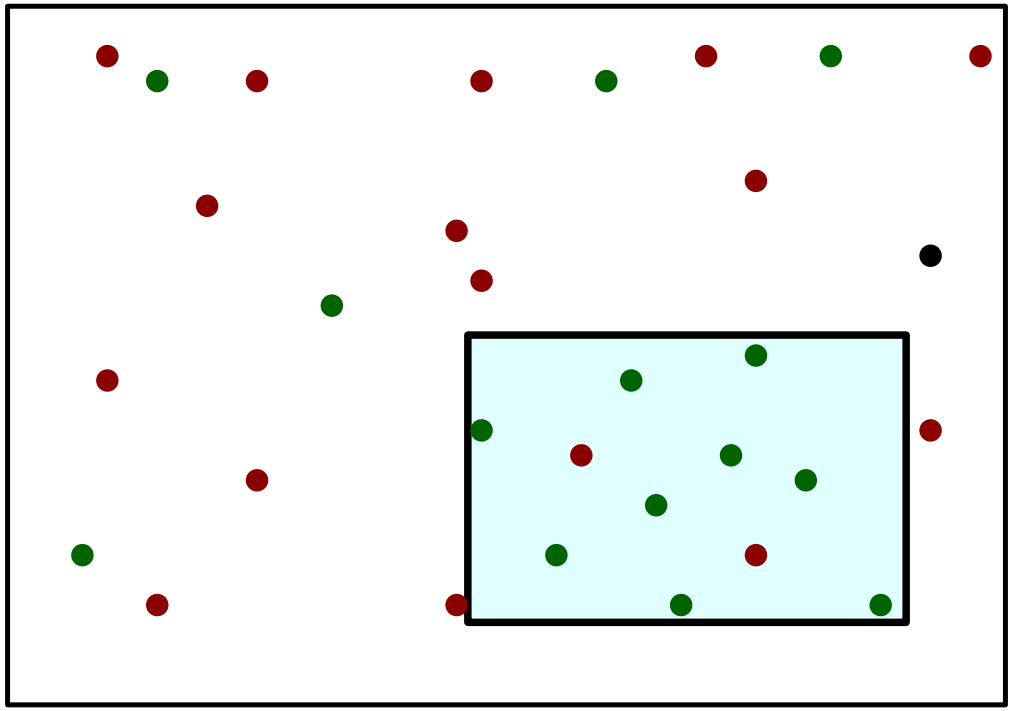
Branch-and-Bound (cont)



- Evaluate bounds and set node active or closed
- Closing possible if we can prove that no solution in a partition can be better than a known solution of value L
- $\overline{g}(A) \leq L \rightarrow \text{close } A$

Example 1: Efficient Subwindow Search

Efficient Subwindow Search (Lampert and Blaschko, 2008)

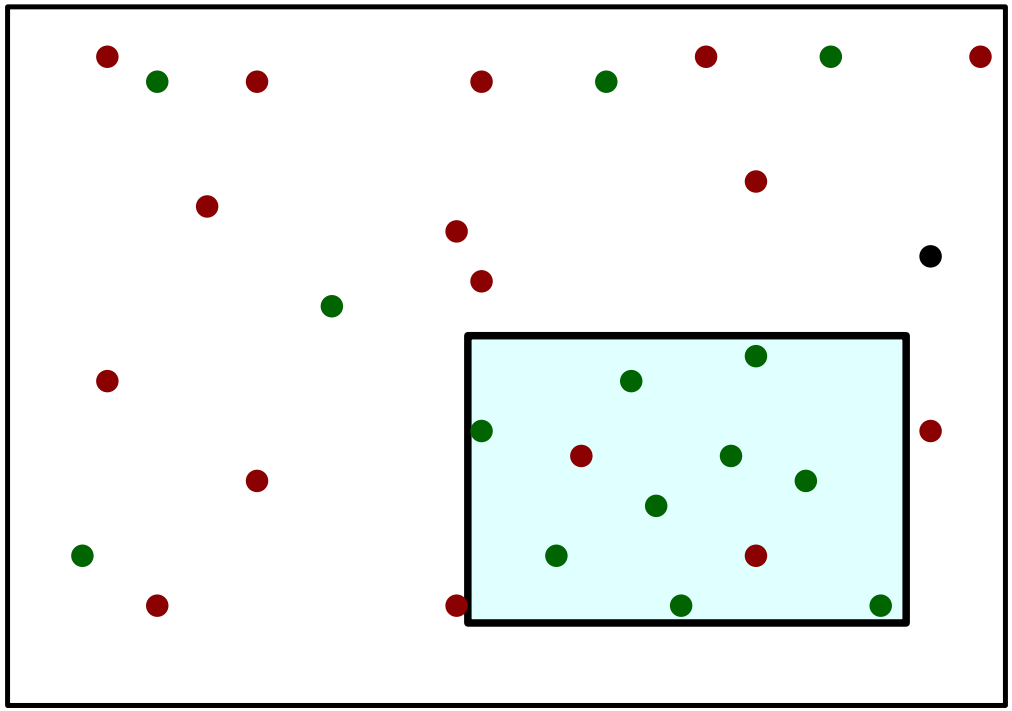


Find the bounding box that maximizes a linear scoring function

$$y^* = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \langle w, \phi(y, x) \rangle$$

Example 1: Efficient Subwindow Search

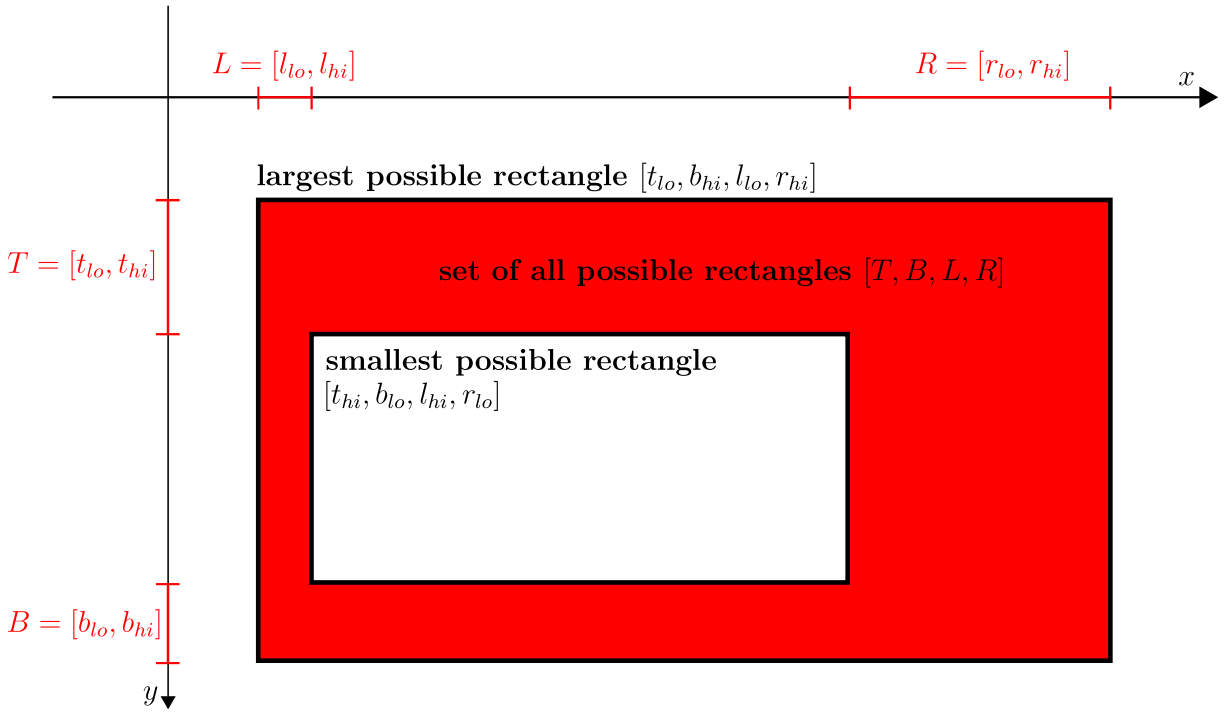
Efficient Subwindow Search (Lampert and Blaschko, 2008)



$$g(x, y) = \beta + \sum_{x_i \text{ within } y} w(x_i)$$

Example 1: Efficient Subwindow Search

Efficient Subwindow Search (Lampert and Blaschko, 2008)



Subsets $\mathcal{B}$ of bounding boxes specified by interval coordinates, $\mathcal{B} \subset 2^{\mathcal{Y}}$

Example 1: Efficient Subwindow Search

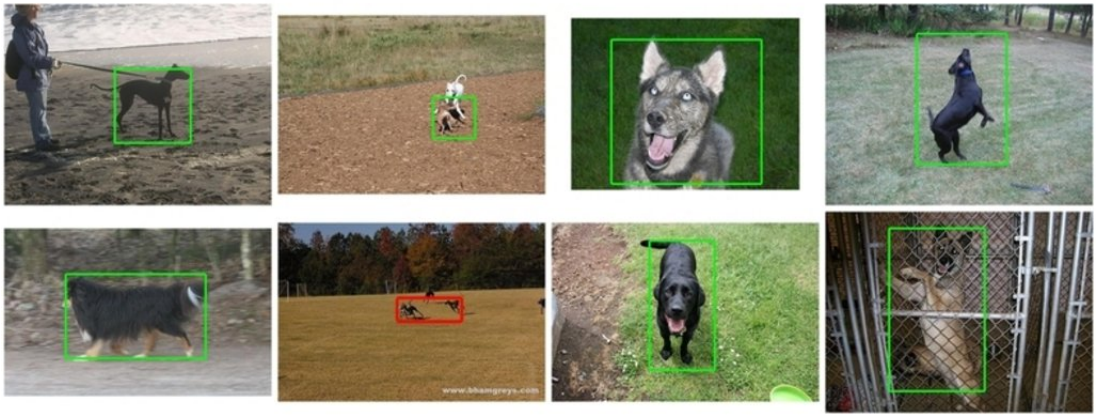
Efficient Subwindow Search (Lampert and Blaschko, 2008)

Upper bound: $\overline{g}(x, B) \geq g(x, y)$ for all $y \in B$

$$\begin{aligned} \overline{g}(x, B) &= \beta + \sum_{\substack{x_i \\ \text{within} \\ B_{\max}}} \max\{w(x_i), 0\} + \sum_{\substack{x_i \\ \text{within} \\ B_{\min}}} \min\{0, w(x_i)\} \\ &\geq \max_{y \in B} g(x, y) \end{aligned}$$

Example 1: Efficient Subwindow Search

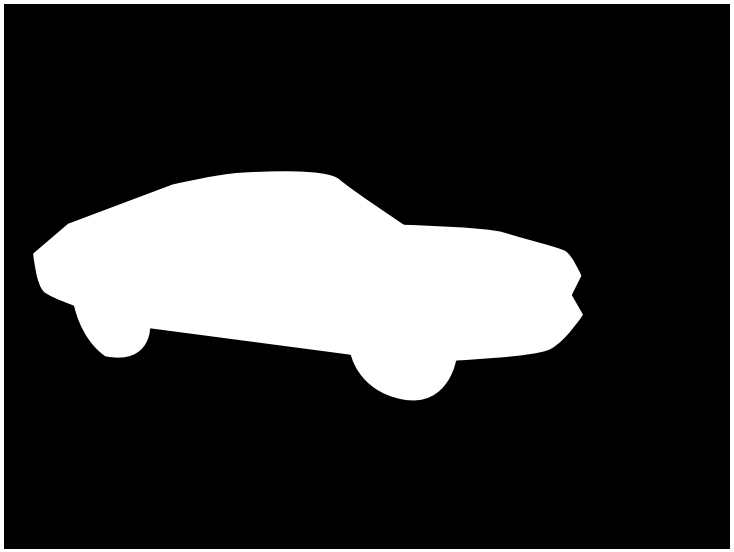
Efficient Subwindow Search (Lampert and Blaschko, 2008)

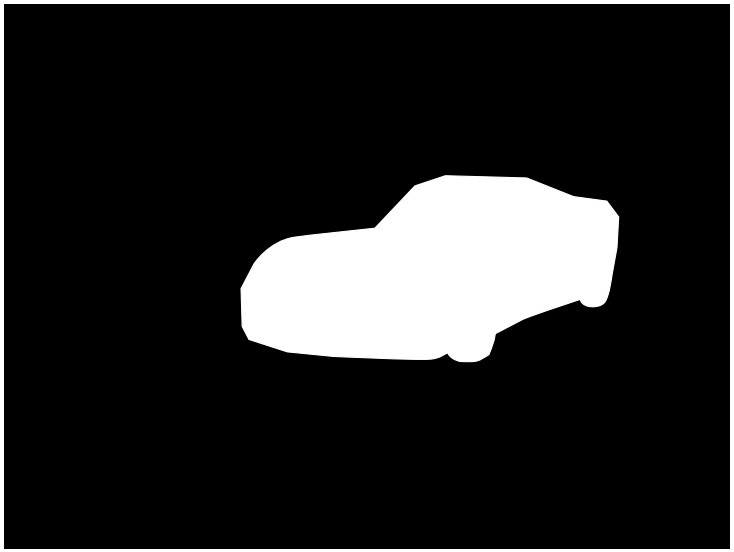


PASCAL VOC 2007 detection challenge bounding boxes found using ESS

Example 2: Branch-and-Mincut

Branch-and-Mincut (Lempitsky et al., 2008)
Binary image segmentation with non-local interaction





$$y_1, y_2 \in \mathcal{Y}$$

Example 2: Branch-and-Mincut

Branch-and-Mincut (Lempitsky et al., 2008)
Binary image segmentation with non-local interaction

$$E(z,y) = C(y) + \sum_{p \in V} F^p(y) z_p + \sum_{p \in V} B^p(y)(1-z_p) + \sum_{\{i,j\} \in \mathcal{E}} P^{pq}(y)|z_p - z_q|,$$

$$g(x,y) = \max_{z \in 2^V} -E(z,y)$$

- Here: $z \in \{0,1\}^V$ is a binary pixel mask
- $F^p(y)$, $B^p(y)$ are foreground/background unary energies
- $P^{pq}(y)$ is a standard pairwise energy
- Global dependencies on y

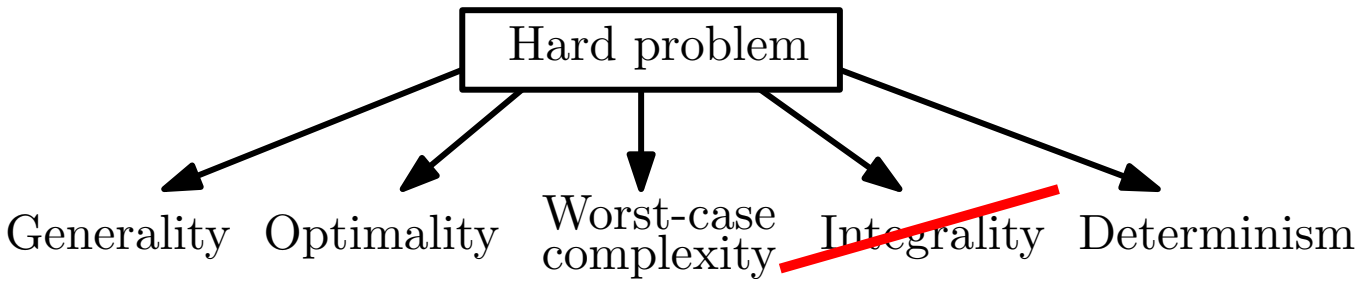
Example 2: Branch-and-Mincut

Branch-and-Mincut (Lempitsky et al., 2008)
Upper bound for any subset $A \subseteq \mathcal{Y}$:

$$\max_{y \in A} g(x,y) = \max_{y \in A} \max_{z \in 2^V} -E(z,y)$$

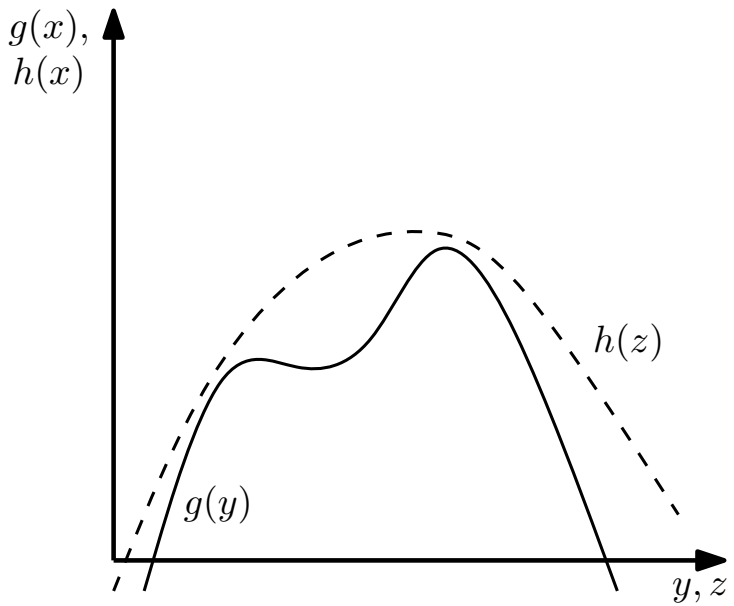
$$= \max_{y \in A} \max_{z \in 2^V} -\left[C(y) + \sum_{p \in V} F^p(y) z_p + \sum_{p \in V} B^p(y)(1-z_p) + \sum_{\{i,j\} \in \mathcal{E}} P^{pq}(y) |z_p - z_q| \right]$$

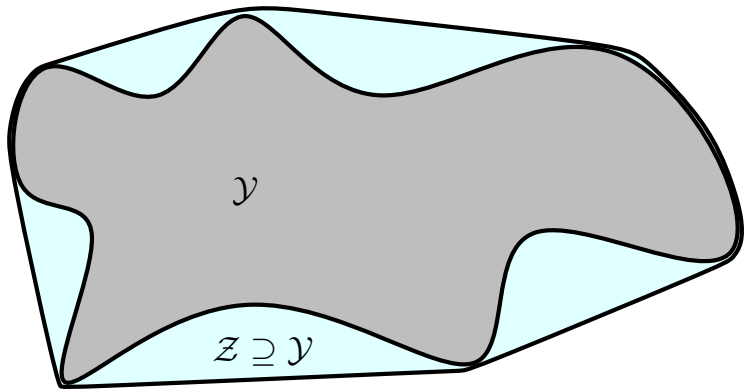
$$\leq \max_{z \in 2^V} \left[\left(\max_{y \in A} -C(y) \right) + \sum_{p \in V} \left(\max_{y \in A} -F^p(y) \right) z_p + \sum_{p \in V} \left(\max_{y \in A} -B^p(y) \right) (1-z_p) + \sum_{\{i,j\} \in \mathcal{E}} \left(\max_{y \in A} -P^{pq}(y) \right) |z_p - z_q| \right].$$


Hard problem
Generality
Optimality
Worst-case complexity
Integrality
Determinism

Problem Relaxations

- Optimization problems (minimizing $g : \mathcal{G} \rightarrow \mathbb{R}$ over $\mathcal{Y} \subseteq \mathcal{G}$) can become easier if
  - feasible set is enlarged, and/or
  - objective function is replaced with a bound.





Problem Relaxations

- Optimization problems (minimizing $g : \mathcal{G} \to \mathbb{R}$ over $\mathcal{Y} \subseteq \mathcal{G}$) can become easier if
  - feasible set is enlarged, and/or
  - objective function is replaced with a bound.

Definition (Relaxation (Geoffrion, 1974))

Given two optimization problems $(g, \mathcal{Y}, \mathcal{G})$ and $(h, \mathcal{Z}, \mathcal{G})$, the problem $(h, \mathcal{Z}, \mathcal{G})$ is said to be a *relaxation* of $(g, \mathcal{Y}, \mathcal{G})$ if,

1. $\mathcal{Z} \supseteq \mathcal{Y}$, i.e. the feasible set of the relaxation contains the feasible set of the original problem, and
2. $\forall y \in \mathcal{Y} : h(y) \geq g(y)$, i.e. over the original feasible set the objective function h achieves no smaller values than the objective function g.

Relaxations

- Relaxed solution z^* provides a bound:

$$h(z^*) \geq g(y^*), \text{ (maximization: upper bound)}$$
$$h(z^*) \leq g(y^*), \text{ (minimization: lower bound)}$$

- Relaxation is typically tractable
- Evidence that relaxations are "better" for learning (Kulesza and Pereira, 2007), (Finley and Joachims, 2008), (Martins et al., 2009)
- Drawback: $z^* \in \mathcal{Z} \setminus \mathcal{Y}$ possible

Are there principled methods to *construct* relaxations?

- Linear Programming Relaxations
- Lagrangian relaxation
- Lagrangian/Dual decomposition

Relaxations

- Relaxed solution z^* provides a bound:

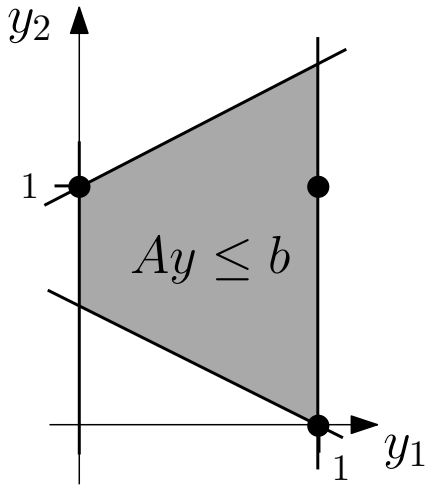
$$h(z^*) \geq g(y^*), \text{ (maximization: upper bound)}$$
$$h(z^*) \leq g(y^*), \text{ (minimization: lower bound)}$$

- Relaxation is typically tractable
- Evidence that relaxations are "better" for learning (Kulesza and Pereira, 2007), (Finley and Joachims, 2008), (Martins et al., 2009)
- Drawback: $z^* \in \mathcal{Z} \setminus \mathcal{Y}$ possible

Are there principled methods to *construct* relaxations?

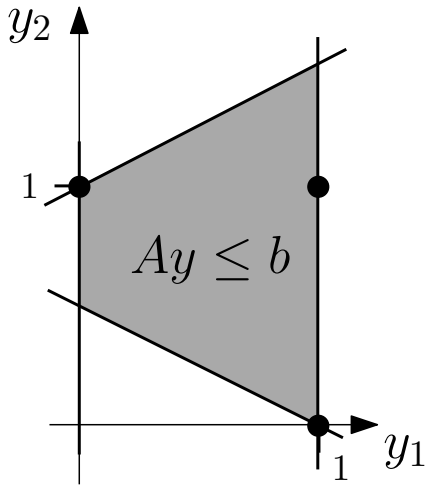
- Linear Programming Relaxations
- Lagrangian relaxation
- Lagrangian/Dual decomposition

Linear Programming Relaxation



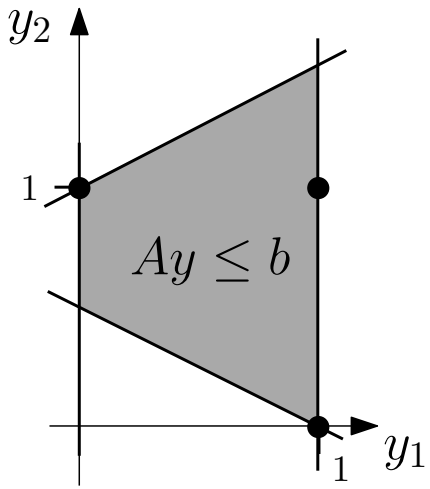
$$\begin{aligned} \max_{y} \quad & c^\top y \\ \text{sb.t.} \quad & Ay \leq b, \\ & y \text{ is integer.} \end{aligned}$$

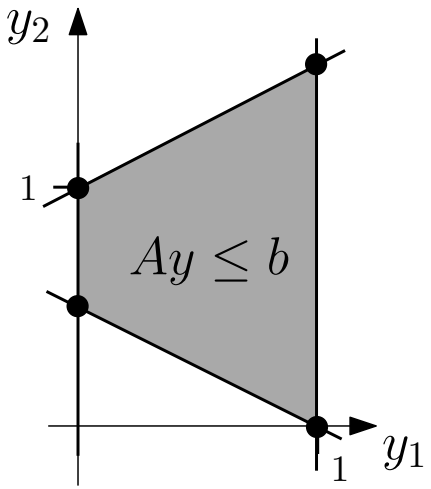
Linear Programming Relaxation



$$\begin{aligned} \max_{y} \quad & c^\top y \\ \text{sb.t.} \quad & Ay \leq b, \\ & y \text{ is integer.} \end{aligned}$$

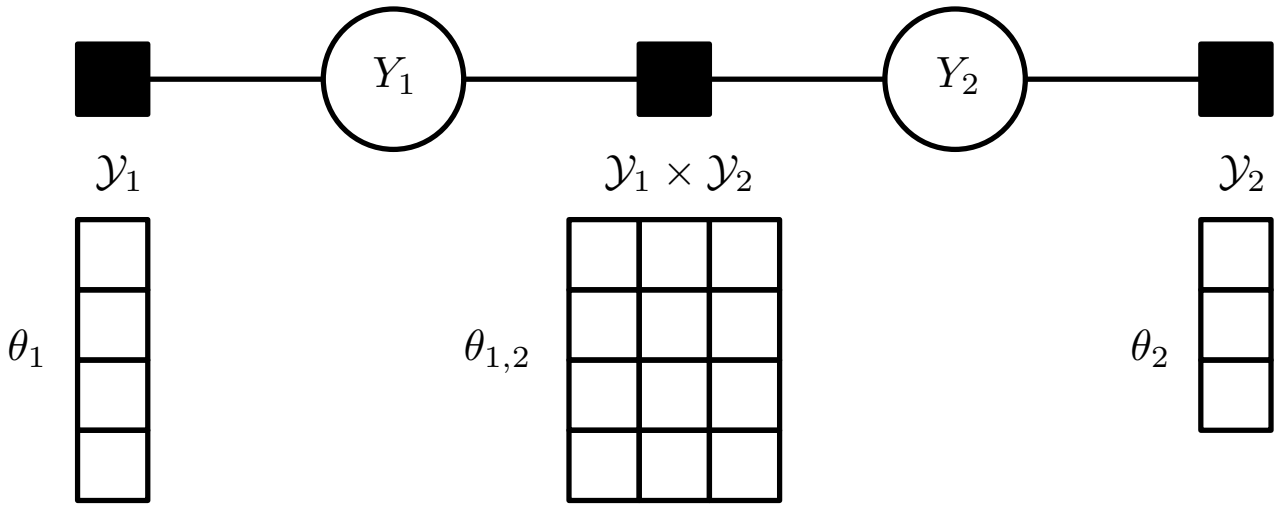
Linear Programming Relaxation



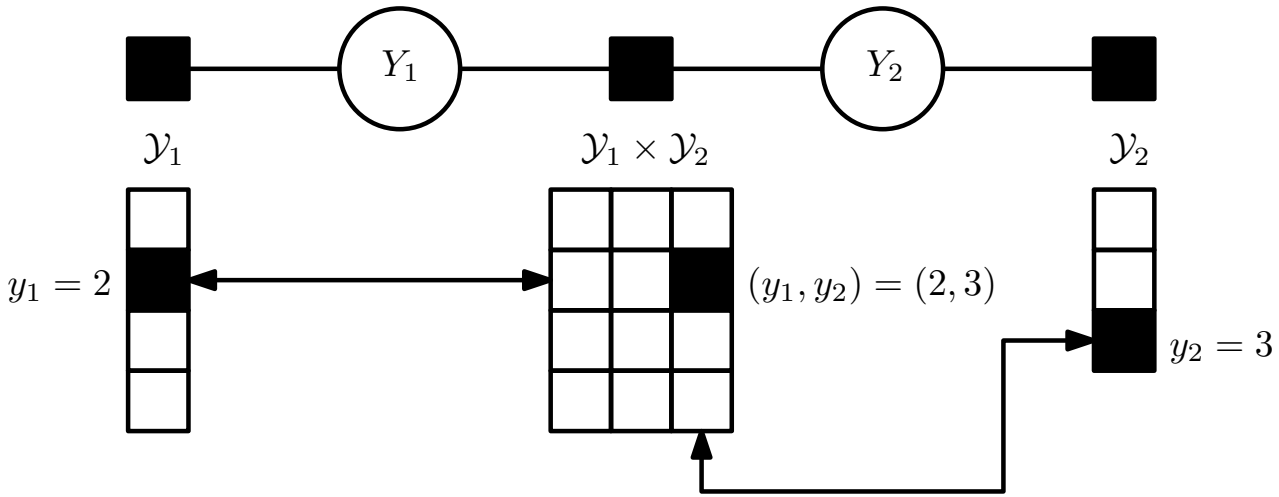


$$\begin{aligned} \max_{y} \quad & c^\top y \\ \text{sb.t.} \quad & Ay \leq b, \end{aligned}$$

MAP-MRF Linear Programming Relaxation

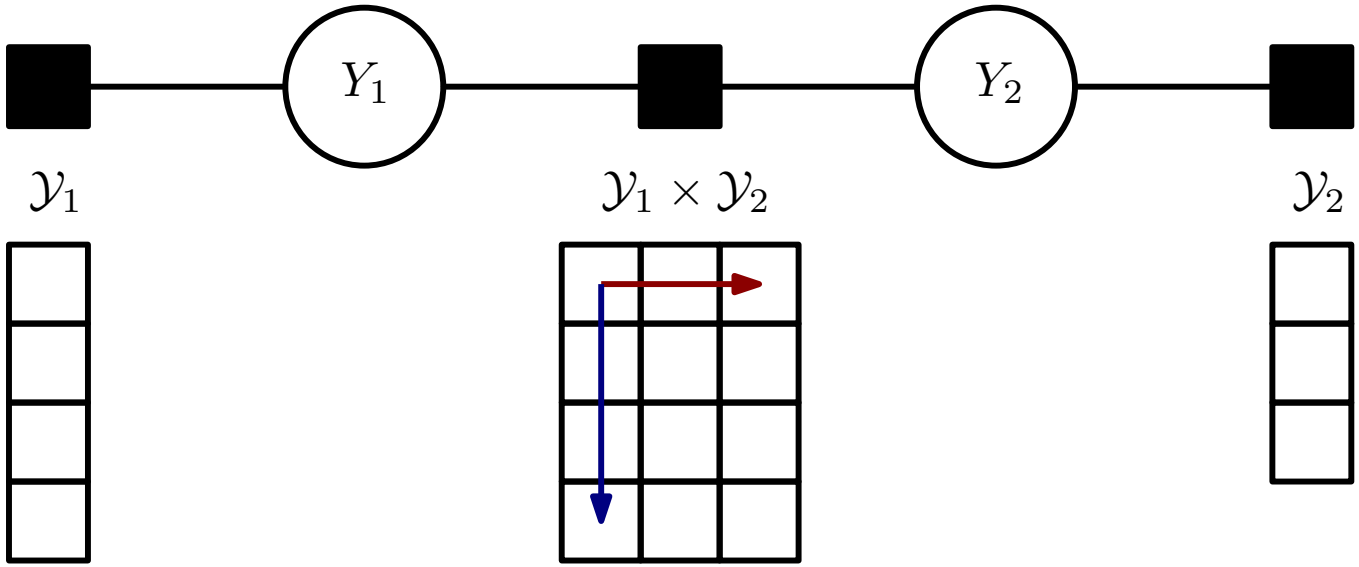


MAP-MRF Linear Programming Relaxation



- Energy $E(y; x) = \theta_1(y_1; x) + \theta_{1,2}(y_1, y_2; x) + \theta_2(y_2; x)$
- Probability $p(y|x) = \frac{1}{Z(x)} \exp\{-E(y; x)\}$
- MAP prediction: $\underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, p(y|x) = \underset{y \in \mathcal{Y}}{\operatorname{argmin}}\, E(y; x)$

MAP-MRF Linear Programming Relaxation



$\mu_1 \in \{0,1\}^{\mathcal{Y}_1}$ $\quad \mu_{1,2} \in \{0,1\}^{\mathcal{Y}_1 \times \mathcal{Y}_2}$ $\quad \mu_2 \in \{0,1\}^{\mathcal{Y}_2}$

$$\mu_1(y_1) = \sum_{y_2 \in \mathcal{Y}_2} \mu_{1,2}(y_1, y_2) \qquad \sum_{y_1 \in \mathcal{Y}_1} \mu_{1,2}(y_1, y_2) = \mu_2(y_2)$$

$$\sum_{y_1 \in \mathcal{Y}_1} \mu_1(y_1) = 1 \qquad \sum_{(y_1,y_2) \in \mathcal{Y}_1 \times \mathcal{Y}_2} \mu_{1,2}(y_1, y_2) = 1 \qquad \sum_{y_2 \in \mathcal{Y}_2} \mu_2(y_2) = 1$$

- Energy is now linear: $E(y; x) = \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle =: -g(y, x)$

MAP-MRF Linear Programming Relaxation (cont)

$$
\begin{aligned}
\max_{\boldsymbol{\mu}} \quad & \sum_{i \in V} \sum_{y_i \in \mathcal{Y}_i} \theta_i(y_i)\mu_i(y_i) + \sum_{\{i,j\} \in E} \sum_{(y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} \theta_{i,j}(y_i,y_j)\mu_{i,j}(y_i,y_j) \\
\text{sb.t.} \quad & \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) = 1, \quad \forall i \in V, \\
& \sum_{y_j \in \mathcal{Y}_j} \mu_{i,j}(y_i,y_j) = \mu_i(y_i), \quad \forall \{i,j\} \in E, \forall y_i \in \mathcal{Y}_i \\
& \mu_i(y_i) \in \{0,1\}, \quad \forall i \in V, \forall y_i \in \mathcal{Y}_i, \\
& \mu_{i,j}(y_i,y_j) \in \{0,1\}, \quad \forall \{i,j\} \in E, \forall (y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j.
\end{aligned}
$$

MAP-MRF Linear Programming Relaxation (cont)

$$\max_{\boldsymbol{\mu}} \quad \sum_{i \in V} \sum_{y_i \in \mathcal{Y}_i} \theta_i(y_i)\mu_i(y_i) + \sum_{\{i,j\} \in E} \sum_{(y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} \theta_{i,j}(y_i,y_j)\mu_{i,j}(y_i,y_j)$$

$$\text{sb.t.} \quad \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) = 1, \quad \forall i \in V,$$

$$\sum_{y_j \in \mathcal{Y}_j} \mu_{i,j}(y_i,y_j) = \mu_i(y_i), \quad \forall \{i,j\} \in E, \forall y_i \in \mathcal{Y}_i$$

$$\mu_i(y_i) \in \{0,1\}, \quad \forall i \in V, \forall y_i \in \mathcal{Y}_i,$$

$$\mu_{i,j}(y_i,y_j) \in \{0,1\}, \quad \forall \{i,j\} \in E, \forall (y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j.$$

- $\longrightarrow$ NP-hard integer linear program

MAP-MRF Linear Programming Relaxation (cont)

$$
\begin{aligned}
\max_{\boldsymbol{\mu}} \quad & \sum_{i \in V} \sum_{y_i \in \mathcal{Y}_i} \theta_i(y_i)\mu_i(y_i) + \sum_{\{i,j\} \in E} \sum_{(y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} \theta_{i,j}(y_i,y_j)\mu_{i,j}(y_i,y_j) \\
\text{s.t.} \quad & \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) = 1, \quad \forall i \in V, \\
& \sum_{y_j \in \mathcal{Y}_j} \mu_{i,j}(y_i,y_j) = \mu_i(y_i), \quad \forall \{i,j\} \in E, \forall y_i \in \mathcal{Y}_i \\
& \mu_i(y_i) \in [0,1], \quad \forall i \in V, \forall y_i \in \mathcal{Y}_i, \\
& \mu_{i,j}(y_i,y_j) \in [0,1], \quad \forall \{i,j\} \in E, \forall (y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j.
\end{aligned}
$$

- $\longrightarrow$ linear program

MAP-MRF LP Relaxation, Works

MAP-MRF LP Analysis

- (Wainwright et al., Trans. Inf. Tech., 2005), (Weiss et al., UAI 2007), (Werner, PAMI 2005)
- (Kolmogorov, PAMI 2006)

Improving tightness

- (Komodakis and Paragios, ECCV 2008), (Kumar and Torr, ICML 2008), (Sontag and Jaakkola, NIPS 2007), (Sontag et al., UAI 2008), (Werner, CVPR 2008), (Kumar et al., NIPS 2007)

Algorithms based on the LP

- (Globerson and Jaakkola, NIPS 2007), (Kumar and Torr, ICML 2008), (Sontag et al., UAI 2008)

MAP-MRF LP Relaxation, Known results

1. LOCAL is tight iff the factor graph is a forest (acyclic)
2. All labelings are vertices of LOCAL (Wainwright and Jordan, 2008)
3. For cyclic graphs there are additional fractional vertices.
4. If all factors have regular energies, the fractional solutions are never optimal (Wainwright and Jordan, 2008)
5. For models with only binary states: half-integrality, integral variables are certain

Lagrangian Relaxation

$$\begin{aligned} \max_{y} \quad & g(y) \\ \text{sb.t.} \quad & y \in \mathcal{D}, \\ & y \in \mathcal{C}. \end{aligned}$$

Assumption

- Optimizing $g(y)$ over $y \in \mathcal{D}$ is "easy".
- Optimizing over $y \in \mathcal{D} \cap \mathcal{C}$ is hard.

High-level idea

- Incorporate $y \in \mathcal{C}$ constraint into objective function

For an excellent introduction, see (Guignard, "Lagrangean Relaxation", TOP 2003) and (Lemaréchal, "Lagrangian Relaxation", 2001).

Lagrangian Relaxation (cont)

Recipe

1. Express $\mathcal{C}$ in terms of equalities and inequalities

$$\mathcal{C} = \{y \in \mathcal{G} : u_i(y) = 0, \forall i = 1, \ldots, I,\ v_j(y) \leq 0, \forall j = 1, \ldots, J\},$$

   - $u_i : \mathcal{G} \to \mathbb{R}$ differentiable, typically affine,
   - $v_j : \mathcal{G} \to \mathbb{R}$ differentiable, typically convex.

2. Introduce Lagrange multipliers, yielding

$$\begin{aligned} \max_{y} \quad & g(y) \\ \text{sb.t.} \quad & y \in \mathcal{D}, \\ & u_i(y) = 0 : \lambda, \quad i = 1, \ldots, I, \\ & v_j(y) \leq 0 : \mu, \quad j = 1, \ldots, J. \end{aligned}$$

Lagrangian Relaxation (cont)

2. Introduce Lagrange multipliers, yielding

$$\begin{aligned} \max_{y} \quad & g(y) \\ \text{sb.t.} \quad & y \in \mathcal{D}, \\ & u_i(y) = 0 : \lambda, \quad i = 1, \ldots, I, \\ & v_j(y) \leq 0 : \mu, \quad j = 1, \ldots, J. \end{aligned}$$

3. Build partial Lagrangian

$$\begin{aligned} \max_{y} \quad & g(y) + \lambda^\top u(y) + \mu^\top v(y) \\ \text{sb.t.} \quad & y \in \mathcal{D}. \end{aligned} \tag{1}$$

Lagrangian Relaxation (cont)

3. Build partial Lagrangian

$$\begin{aligned} \max_{y} \quad & g(y) + \lambda^\top u(y) + \mu^\top v(y) \\ \text{sb.t.} \quad & y \in \mathcal{D}. \end{aligned} \tag{1}$$

Theorem (Weak Duality of Lagrangean Relaxation)

For differentiable functions $u_i : \mathcal{G} \to \mathbb{R}$ and $v_j : \mathcal{G} \to \mathbb{R}$, and for any $\lambda \in \mathbb{R}^I$ and any non-negative $\mu \in \mathbb{R}^J$, $\mu \geq 0$, problem (1) is a relaxation of the original problem: its value is larger than or equal to the optimal value of the original problem.

Lagrangian Relaxation (cont)

3. Build partial Lagrangian

$$q(\lambda, \mu) := \max_{y} \quad g(y) + \lambda^\top u(y) + \mu^\top v(y) \qquad (1)$$
$$\text{sb.t.} \quad y \in \mathcal{D}.$$

4. Minimize upper bound wrt λ, μ

$$\min_{\lambda,\mu} \quad q(\lambda, \mu)$$
$$\text{sb.t.} \quad \mu \geq 0$$

- $\longrightarrow$ efficiently solvable if $q(\lambda, \mu)$ can be evaluated

Optimizing Lagrangian Dual Functions

4. Minimize upper bound wrt λ, μ

$$\begin{aligned} \min_{\lambda,\mu} \quad & q(\lambda,\mu) \\ \text{sb.t.} \quad & \mu \geq 0 \end{aligned} \tag{2}$$

Theorem (Lagrangean Dual Function)

1. *q is convex in λ and μ, Problem (2) is a convex minimization problem.*
2. *If q is unbounded below, then the original problem is infeasible.*
3. *For any λ, $\mu \geq 0$, let*

$$y(\lambda,\mu) = argmax_{y\in\mathcal{D}}\, g(y) + \lambda^\top u(y) + \mu^\top v(u)$$

Then, a subgradient of q can be constructed by evaluating the constraint functions at $y(\lambda,\mu)$ as

$$u(y(\lambda,\mu)) \in \frac{\partial}{\partial\lambda} q(\lambda,\mu), \quad and \quad v(y(\lambda,\mu)) \in \frac{\partial}{\partial\mu} q(\lambda,\mu).$$

Example: MAP-MRF Message Passing

$$
\begin{aligned}
\max_{\boldsymbol{\mu}} \quad & \sum_{i \in V} \sum_{y_i \in \mathcal{Y}_i} \theta_i(y_i)\mu_i(y_i) + \sum_{\{i,j\} \in E} \sum_{(y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} \theta_{i,j}(y_i,y_j)\mu_{i,j}(y_i,y_j) \\
\text{sb.t.} \quad & \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) = 1, \quad \forall i \in V, \\
& \sum_{(y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} \mu_{i,j}(y_i,y_j) = 1, \quad \forall \{i,j\} \in E, \\
& \sum_{y_j \in \mathcal{Y}_j} \mu_{i,j}(y_i,y_j) = \mu_i(y_i), \quad \forall \{i,j\} \in E, \forall y_i \in \mathcal{Y}_i \\
& \mu_i(y_i) \in \{0,1\}, \quad \forall i \in V, \forall y_i \in \mathcal{Y}_i, \\
& \mu_{i,j}(y_i,y_j) \in \{0,1\}, \quad \forall \{i,j\} \in E, \forall (y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j.
\end{aligned}
$$

Example: MAP-MRF Message Passing

$$
\begin{aligned}
\max_{\boldsymbol{\mu}} \quad & \sum_{i \in V} \sum_{y_i \in \mathcal{Y}_i} \theta_i(y_i)\mu_i(y_i) + \sum_{\{i,j\} \in E} \sum_{(y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} \theta_{i,j}(y_i,y_j)\mu_{i,j}(y_i,y_j) \\
\text{s.t.} \quad & \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) = 1, \quad \forall i \in V, \\
& \sum_{(y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} \mu_{i,j}(y_i,y_j) = 1, \quad \forall \{i,j\} \in E, \\
& \sum_{y_j \in \mathcal{Y}_j} \mu_{i,j}(y_i,y_j) = \mu_i(y_i), \quad \forall \{i,j\} \in E, \forall y_i \in \mathcal{Y}_i \\
& \mu_i(y_i) \in \{0,1\}, \quad \forall i \in V, \forall y_i \in \mathcal{Y}_i, \\
& \mu_{i,j}(y_i,y_j) \in \{0,1\}, \quad \forall \{i,j\} \in E, \forall (y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j.
\end{aligned}
$$

- If the constraint would not be there, problem is trivial!
- (Wainwright and Jordan, 2008, section 4.1.3)

Example: MAP-MRF Message Passing

$$
\begin{aligned}
\max_{\boldsymbol{\mu}} \quad & \sum_{i \in V} \sum_{y_i \in \mathcal{Y}_i} \theta_i(y_i)\mu_i(y_i) + \sum_{\{i,j\} \in E} \sum_{(y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} \theta_{i,j}(y_i,y_j)\mu_{i,j}(y_i,y_j) \\
\text{s.t.} \quad & \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) = 1, \quad \forall i \in V, \\
& \sum_{(y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} \mu_{i,j}(y_i,y_j) = 1, \quad \forall \{i,j\} \in E, \\
& \sum_{y_j \in \mathcal{Y}_j} \mu_{i,j}(y_i,y_j) = \mu_i(y_i) : \phi_{i,j}(y_i), \quad \forall \{i,j\} \in E, \forall y_i \in \mathcal{Y}_i \\
& \mu_i(y_i) \in \{0,1\}, \quad \forall i \in V, \forall y_i \in \mathcal{Y}_i, \\
& \mu_{i,j}(y_i,y_j) \in \{0,1\}, \quad \forall \{i,j\} \in E, \forall (y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j.
\end{aligned}
$$

- If the constraint would not be there, problem is trivial!
- (Wainwright and Jordan, 2008, section 4.1.3)

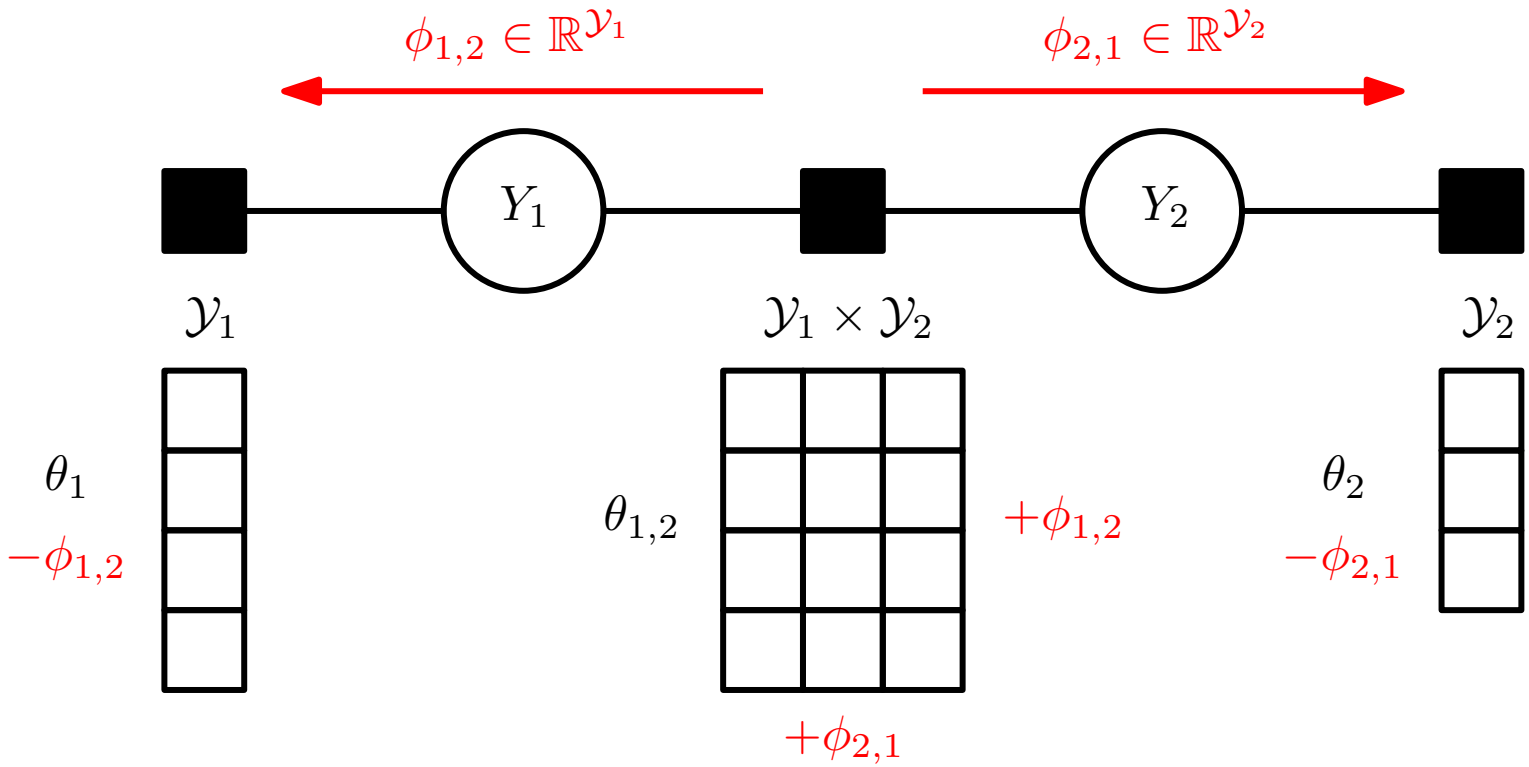
Example: MAP-MRF Message Passing

$$
\begin{aligned}
q(\phi) := \quad \max_{\mu} \quad & \sum_{i \in V} \sum_{y_i \in \mathcal{Y}_i} \theta_i(y_i)\mu_i(y_i) + \sum_{\{i,j\} \in E} \sum_{(y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} \theta_{i,j}(y_i,y_j)\mu_{i,j}(y_i,y_j) \\
& + \sum_{\{i,j\} \in E} \sum_{y_i \in \mathcal{Y}_i} \phi_{i,j}(y_i) \left(\sum_{y_j \in \mathcal{Y}_j} \mu_{i,j}(y_i,y_j) - \mu_i(y_i) \right) \\
\text{sb.t.} \quad & \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) = 1, \quad \forall i \in V, \\
& \sum_{(y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} \mu_{i,j}(y_i,y_j) = 1, \quad \forall \{i,j\} \in E, \\
& \mu_i(y_i) \in \{0,1\}, \quad \forall i \in V, \forall y_i \in \mathcal{Y}_i, \\
& \mu_{i,j}(y_i,y_j) \in \{0,1\}, \quad \forall \{i,j\} \in E, \forall (y_i,y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j.
\end{aligned}
$$

Example: MAP-MRF Message Passing

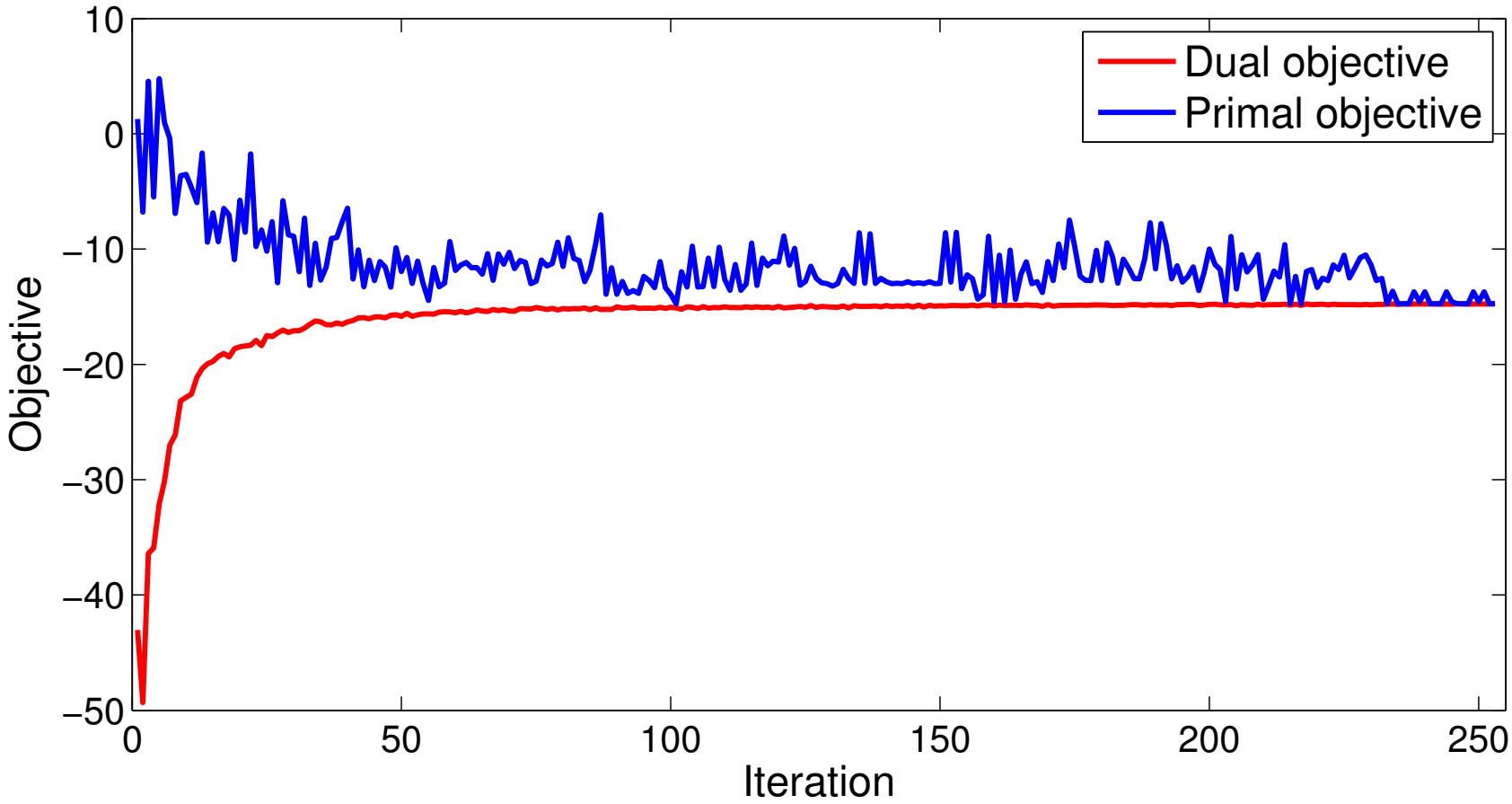
$$
\begin{aligned}
q(\phi) := \quad \max_{\mu} \quad & \sum_{i \in V} \sum_{y_i \in \mathcal{Y}_i} \left(\theta_i(y_i) - \sum_{j \in V : \{i,j\} \in E} \phi_{i,j}(y_i) \right) \mu_i(y_i) \\
& + \sum_{\{i,j\} \in E} \sum_{(y_i, y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} (\theta_{i,j}(y_i, y_j) + \phi_{i,j}(y_i))\, \mu_{i,j}(y_i, y_j) \\
\text{s.t.} \quad & \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) = 1, \quad \forall i \in V, \\
& \sum_{(y_i, y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j} \mu_{i,j}(y_i, y_j) = 1, \quad \forall \{i,j\} \in E, \\
& \mu_i(y_i) \in \{0,1\}, \quad \forall i \in V, \forall y_i \in \mathcal{Y}_i, \\
& \mu_{i,j}(y_i, y_j) \in \{0,1\}, \quad \forall \{i,j\} \in E, \forall (y_i, y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j.
\end{aligned}
$$

Example: MAP-MRF Message Passing



- Parent-to-child region-graph messages (Meltzer et al., UAI 2009)
- Max-sum diffusion reparametrization (Werner, PAMI 2007)

Example behaviour of objectives



Primal Solution Recovery

Assume we solved the dual problem for (λ^*, μ^*)

1. Can we obtain a primal solution y^*?
2. Can we say something about the relaxation quality?

Theorem (Sufficient Optimality Conditions)

If for a given λ, $\mu \geq 0$, we have $u(y(\lambda, \mu)) = 0$ and $v(y(\lambda, \mu)) \leq 0$ (primal feasibility) and further we have

$$\lambda^\top u(y(\lambda, \mu)) = 0, \quad \text{and} \quad \mu^\top v(y(\lambda, \mu)) = 0,$$

(complementary slackness), then

- *$y(\lambda, \mu)$ is an optimal primal solution to the original problem, and*
- *(λ, μ) is an optimal solution to the dual problem.*

Primal Solution Recovery

Assume we solved the dual problem for (λ^*, μ^*)

1. Can we obtain a primal solution y^*?
2. Can we say something about the relaxation quality?

Theorem (Sufficient Optimality Conditions)

If for a given λ, $\mu \geq 0$, we have $u(y(\lambda, \mu)) = 0$ and $v(y(\lambda, \mu)) \leq 0$ (primal feasibility) and further we have

$$\lambda^\top u(y(\lambda, \mu)) = 0, \quad \textit{and} \quad \mu^\top v(y(\lambda, \mu)) = 0,$$

(complementary slackness), then

- *$y(\lambda, \mu)$ is an optimal primal solution to the original problem, and*
- *(λ, μ) is an optimal solution to the dual problem.*

Primal Solution Recovery (cont)

But,

- we might never see a solution satisfying the optimality condition
- is the case only if there is no *duality gap*, i.e. $q(\lambda^*, \mu^*) = g(x, y^*)$

Special case: integer linear programs

- we can always reconstruct a *primal solution* to

$$\begin{aligned} \min_{y} \quad & g(y) \\ \text{sb.t.} \quad & y \in \text{conv}(\mathcal{D}), \\ & y \in \mathcal{C}. \end{aligned} \tag{3}$$

- For example for subgradient method updates it is known that (Anstreicher and Wolsey, MathProg 2009)

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} y(\lambda^t, \mu^t) \to y^* \text{ of } (3).$$

Dual/Lagrangian Decomposition

Additive structure

$$\begin{aligned} \max_y \quad & \sum_{k=1}^{K} g_k(y) \\ \text{sb.t.} \quad & y \in \mathcal{Y}, \end{aligned}$$

such that $\sum_{k=1}^{K} g_k(y)$ is hard, but for any k,

$$\begin{aligned} \max_y \quad & g_k(y) \\ \text{sb.t.} \quad & y \in \mathcal{Y} \end{aligned}$$

is tractable.

- (Guignard, "Lagrangean Decomposition", MathProg 1987), *the original paper for "dual decomposition"*
- (Conejo et al., "Decomposition Techniques in Mathematical Programming", 2006), continuous variable problems

Dual/Lagrangian Decomposition

Additive structure

$$\begin{aligned} \max_y \quad & \sum_{k=1}^{K} g_k(y) \\ \text{sb.t.} \quad & y \in \mathcal{Y}, \end{aligned}$$

such that $\sum_{k=1}^{K} g_k(y)$ is hard, but for any k,

$$\begin{aligned} \max_y \quad & g_k(y) \\ \text{sb.t.} \quad & y \in \mathcal{Y} \end{aligned}$$

is tractable.

- (Guignard, "Lagrangean Decomposition", MathProg 1987), *the original paper for "dual decomposition"*
- (Conejo et al., "Decomposition Techniques in Mathematical Programming", 2006), continuous variable problems

Dual/Lagrangian Decomposition

Idea

1. Selectively duplicate variables ("variable splitting")

$$\begin{aligned} \max_{y_1,\ldots,y_K,y} \quad & \sum_{k=1}^{K} g_k(y_k) \\ \text{sb.t.} \quad & y \in \mathcal{Y}, \\ & y_k \in \mathcal{Y}, \qquad k = 1,\ldots,K, \\ & y = y_k \;:\; \lambda_k, \qquad k = 1,\ldots,K. \end{aligned} \tag{4}$$

2. Add coupling equality constraints between duplicated variables
3. Apply Lagrangian relaxation to the coupling constraint

Also known as *dual decomposition* and *variable splitting*.

Dual/Lagrangian Decomposition (cont)

$$\begin{aligned} \max_{y_1,\ldots,y_K,y} \quad & \sum_{k=1}^{K} g_k(y_k) \\ \text{sb.t.} \quad & y \in \mathcal{Y}, \\ & y_k \in \mathcal{Y}, \qquad k = 1,\ldots,K, \\ & y = y_k \;:\; \lambda_k, \qquad k = 1,\ldots,K. \end{aligned} \tag{5}$$

Dual/Lagrangian Decomposition (cont)

$$\begin{aligned} \max_{y_1,\ldots,y_K,y} \quad & \sum_{k=1}^{K} g_k(y_k) + \sum_{k=1}^{K} \lambda_k{}^\top (y - y_k) \\ \text{sb.t.} \quad & y \in \mathcal{Y}, \\ & y_k \in \mathcal{Y}, \qquad k = 1, \ldots, K. \end{aligned}$$

Dual/Lagrangian Decomposition (cont)

$$\begin{aligned} \max_{y_1,\ldots,y_K,y} \quad & \sum_{k=1}^{K} \left(g_k(y_k) - \lambda_k{}^\top y_k \right) + \left(\sum_{k=1}^{K} \lambda_k \right)^\top y \\ \text{sb.t.} \quad & y \in \mathcal{Y}, \\ & y_k \in \mathcal{Y}, \qquad k = 1, \ldots, K. \end{aligned}$$

- Problem is *decomposed*

Dual/Lagrangian Decomposition (cont)

$$
\begin{aligned}
q(\lambda) := \quad & \max_{y_1,\ldots,y_K,y} \quad \sum_{k=1}^{K} \left(g_k(y_k) - \lambda_k{}^\top y_k\right) + \left(\sum_{k=1}^{K} \lambda_k\right)^\top y \\
& \text{sb.t.} \quad\quad y \in \mathcal{Y}, \\
& \qquad\quad\; y_k \in \mathcal{Y}, \qquad k = 1, \ldots, K.
\end{aligned}
$$

- Dual optimization problem is

$$
\begin{aligned}
& \min_{\lambda} \quad q(\lambda) \\
& \text{sb.t.} \quad \sum_{k=1}^{K} \lambda_k = 0,
\end{aligned}
$$

- where $\{\lambda | \sum_{k=1}^{K} \lambda_k \neq 0\}$ is the *domain* where $q(\lambda) > -\infty$
- *Projected subgradient method*, using

$$
\frac{\partial}{\partial \lambda_k} \ni (y - y_k) - \frac{1}{K} \sum_{\ell=1}^{K} (y - y_\ell) = \frac{1}{K} \sum_{\ell=1}^{K} y_\ell - y_k.
$$

Primal Interpretation

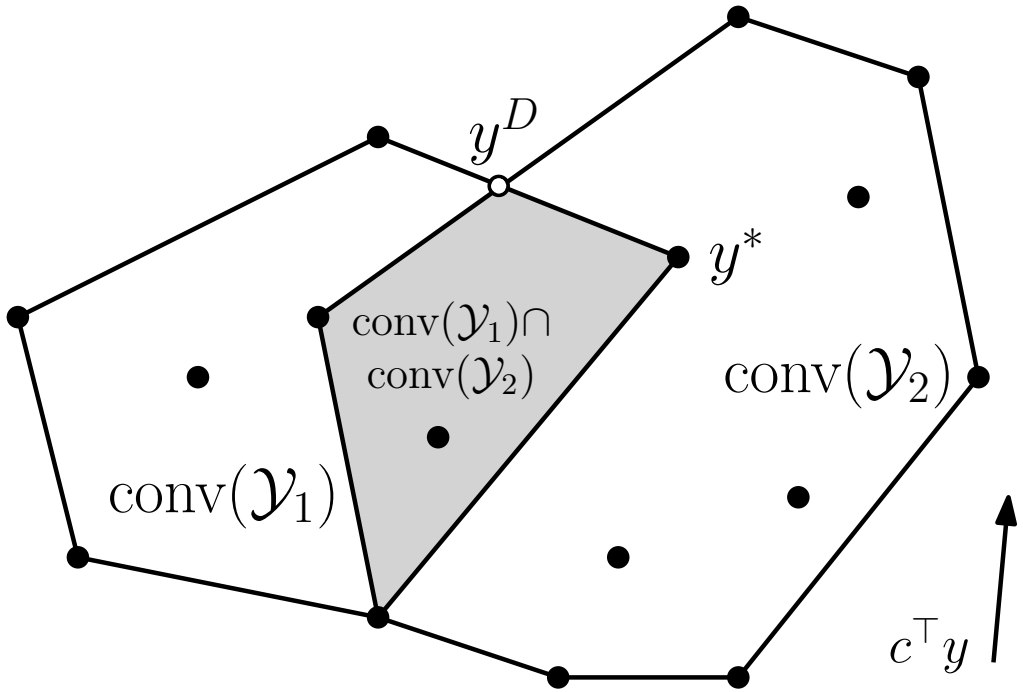
- Primal interpretation for linear case due to (Guignard, 1987)

Theorem (Lagrangian Decomposition Primal)

Let $g(x, y) = c(x)^\top y$ be linear, then the solution of the dual obtains the value of the following relaxed primal optimization problem.

$$\begin{aligned} \min_{y} \quad & \sum_{k=1}^{K} g_k(x, y) \\ sb.t. \quad & y \in conv(\mathcal{Y}_k), \quad \forall k = 1, \ldots, K. \end{aligned}$$

Primal Interpretation (cont)



$$\min_y \quad \sum_{k=1}^{K} g_k(x, y)$$
$$\text{sb.t.} \quad y \in \text{conv}(\mathcal{Y}_k), \quad \forall k = 1, \ldots, K.$$

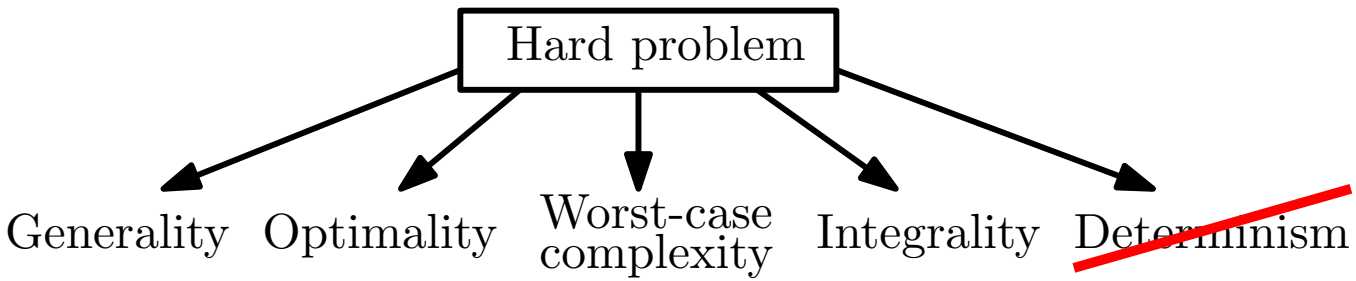
Applications in Computer Vision

Very broadly applicable

- (Komodakis et al., ICCV 2007)
- (Woodford et al., ICCV 2009)
- (Strandmark and Kahl, CVPR 2010)
- (Vicente et al., ICCV 2009)
- (Torresani et al., ECCV 2008)
- (Werner, TechReport 2009)

Further reading

- (Sontag et al., "Introduction to Dual Decomposition for Inference", OfML Vol, 2011)


Hard problem
Generality
Optimality
Worst-case complexity
Integrality
Determinism

Giving up Determinism

Algorithms that use randomness, and

- are non-deterministic, result does not exclusively depend on the input, and
- are allowed to return a wrong result (with low probability), or/and
- have a random runtime.

Is it a good idea?

- for some problems randomized algorithms are the only known efficient algorithms,
- using randomness for hard problems has a long tradition in sampling-based algorithms, physics, computational chemistry, etc.
- such algorithms can be simple and effective, but proving theorems about them can be much harder

Simulated Annealing

Basic idea (Kirkpatrick et al., 1983)

1. Define a distribution $p(y; T)$ that concentrates mass on states y with high values $g(x, y)$
2. Simulate $p(y; T)$
3. Increase concentration and repeat

Defining $p(y; T)$

- Boltzmann distribution

Simulating $p(y; T)$

- MCMC: usually done using a Metropolis chain or a Gibbs sampler

Simulated Annealing (cont)

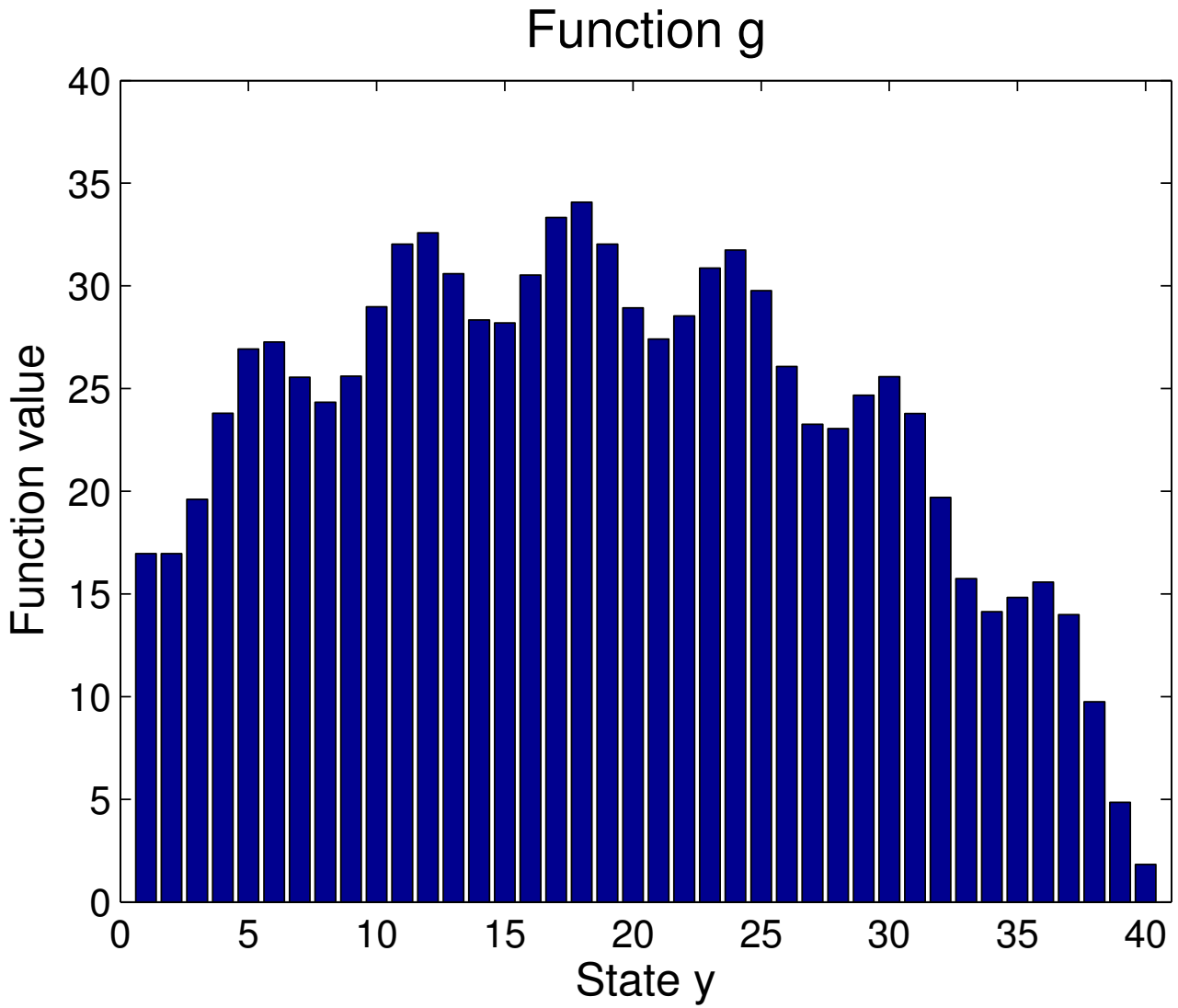
Definition (Boltzmann Distribution)

For a finite set $\mathcal{Y}$, a function $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and a *temperature parameter* $T > 0$, let

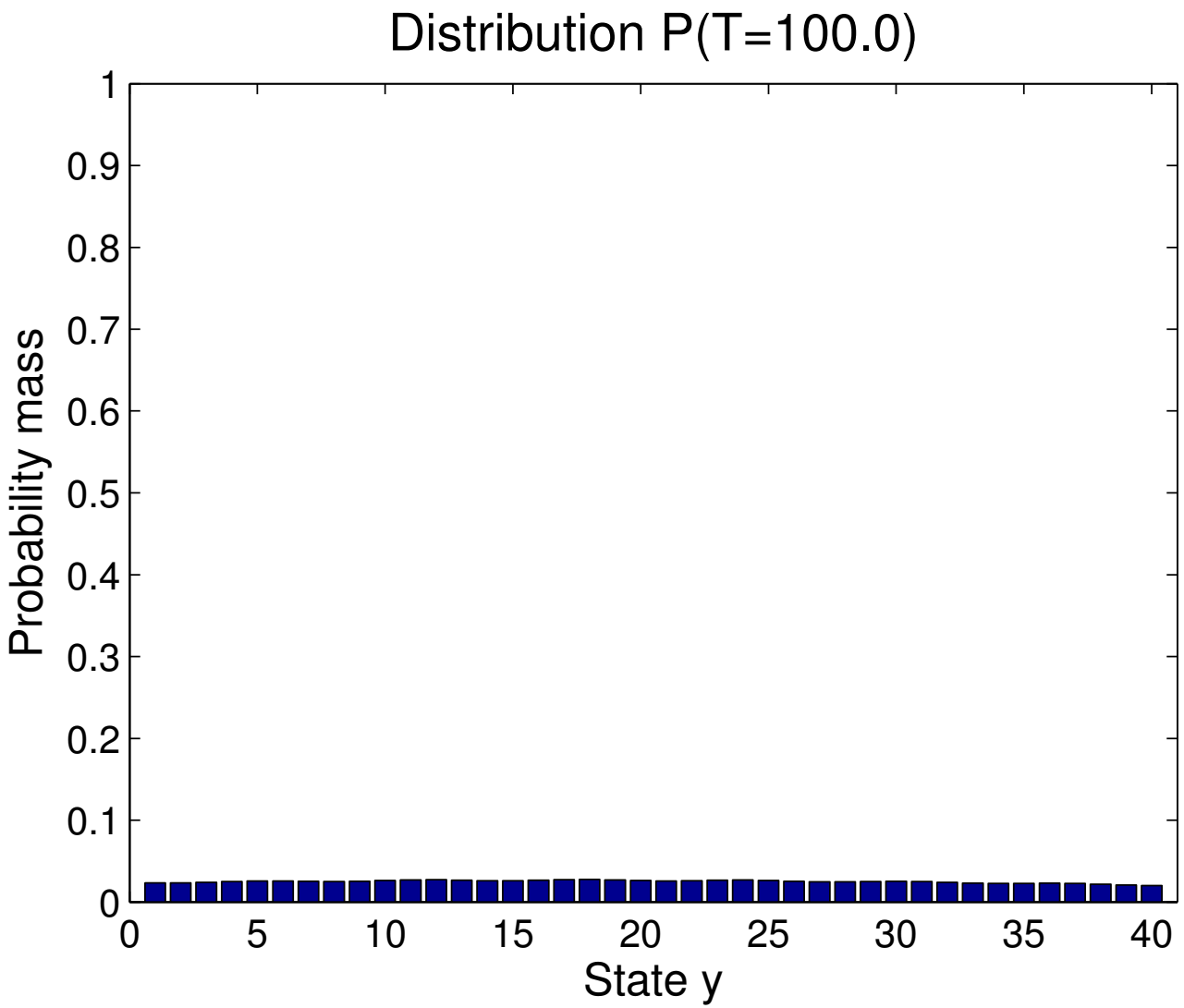
$$p(y; T) = \frac{1}{Z(T)} \exp\left(\frac{g(x, y)}{T}\right), \tag{5}$$

with $Z(T) = \sum_{y \in \mathcal{Y}} \exp(\frac{g(x,y)}{T})$ be the *Boltzmann distribution* for g over $\mathcal{Y}$ at temperature T.

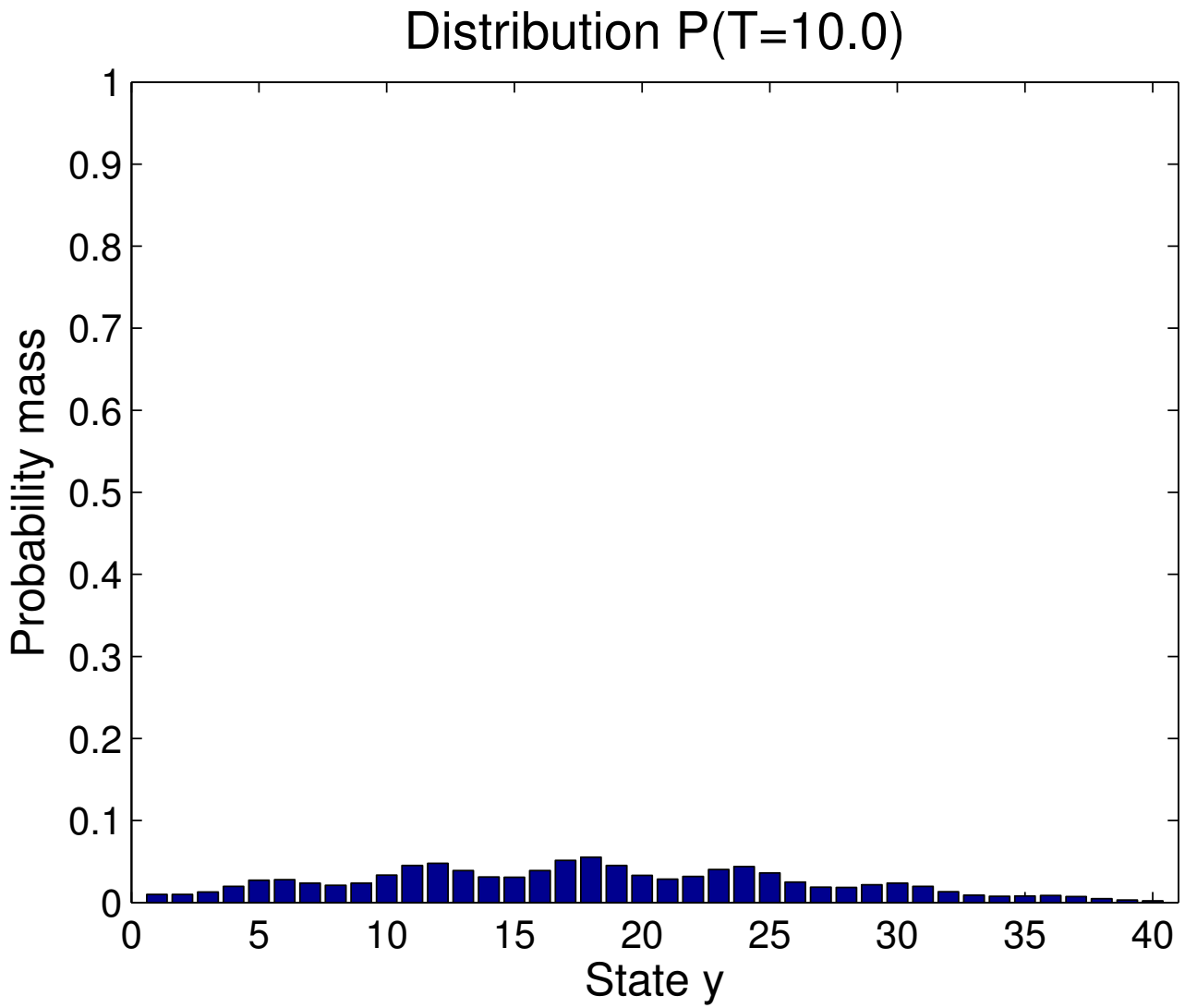
Simulated Annealing (cont)



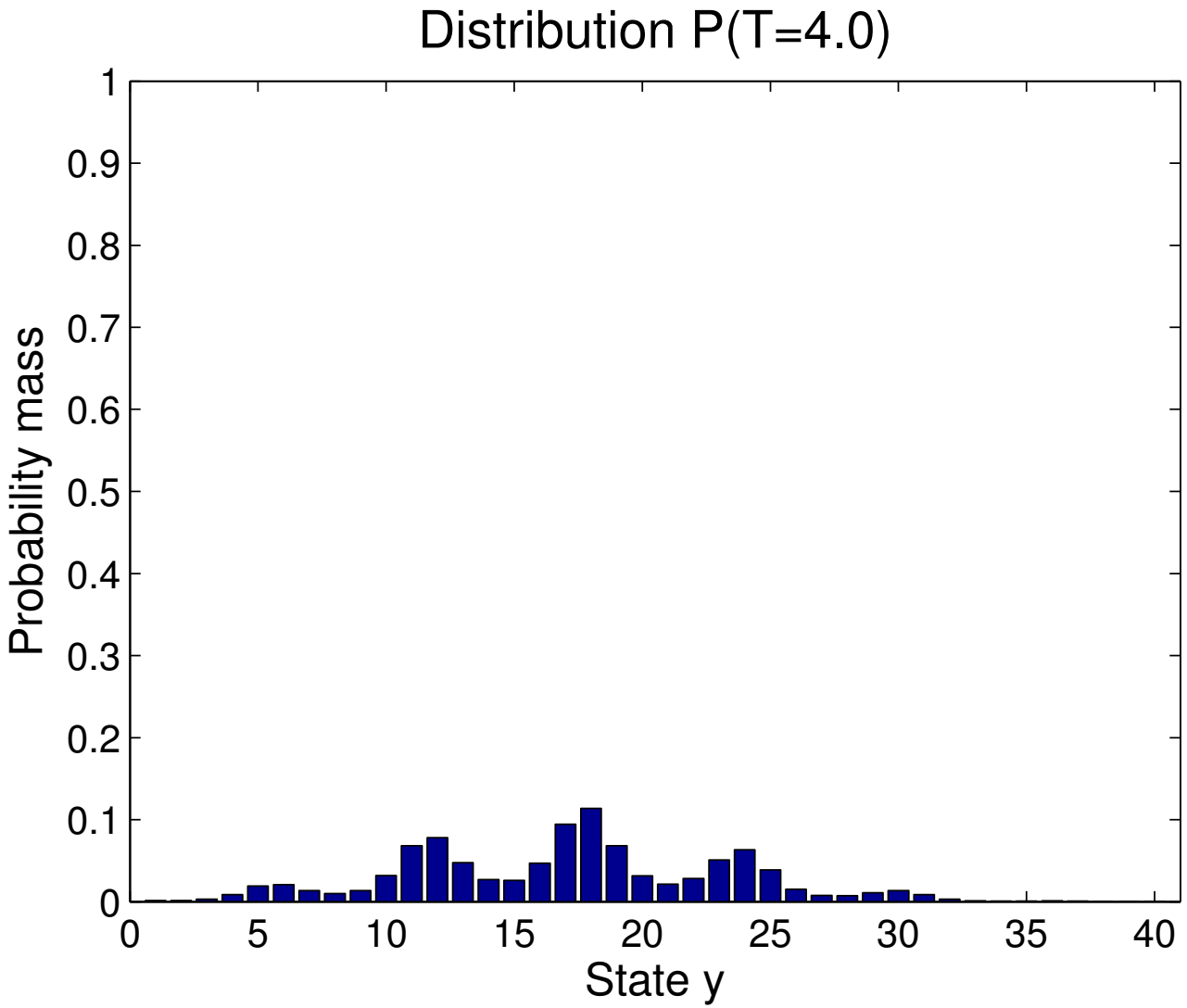
Simulated Annealing (cont)



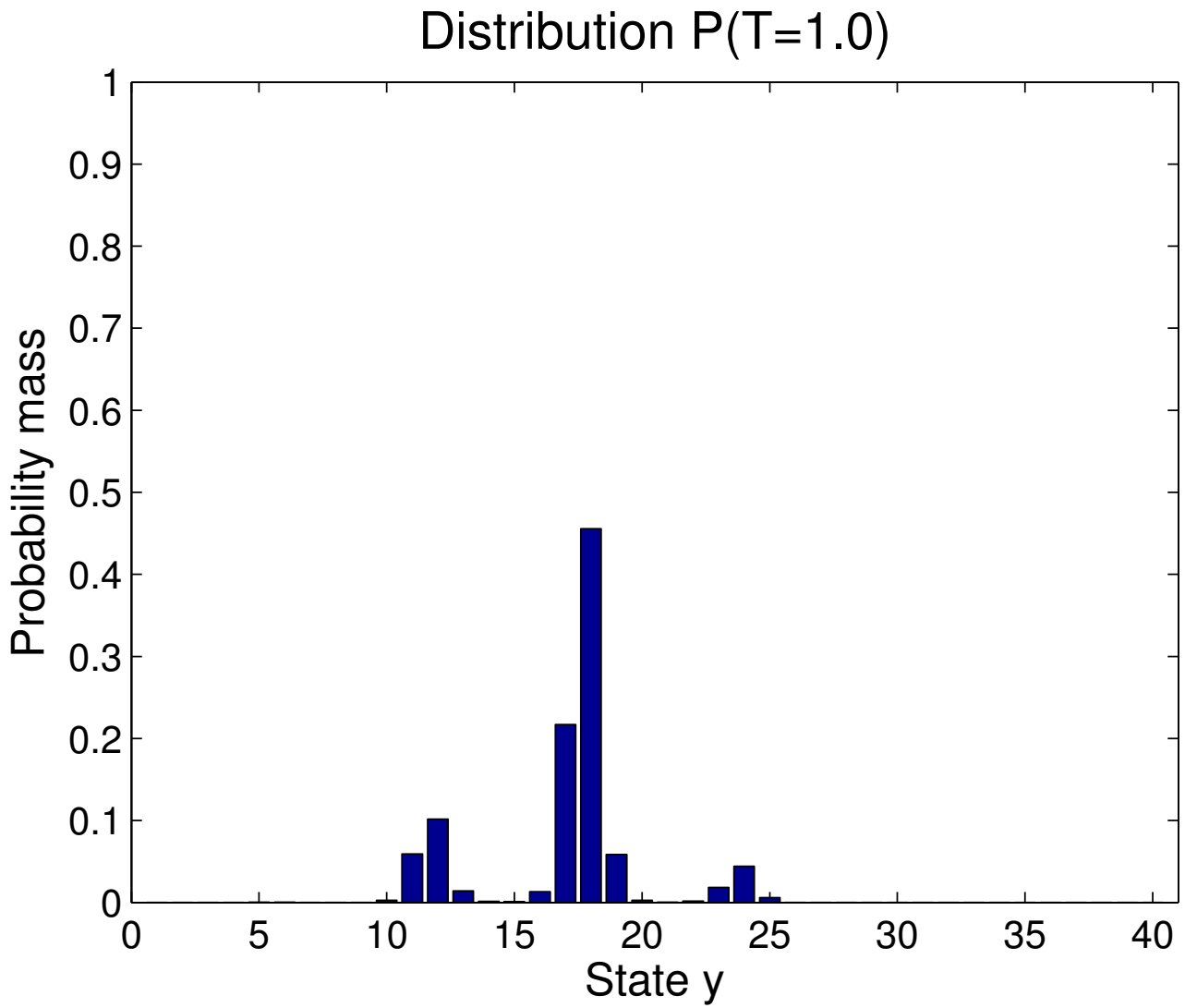
Simulated Annealing (cont)



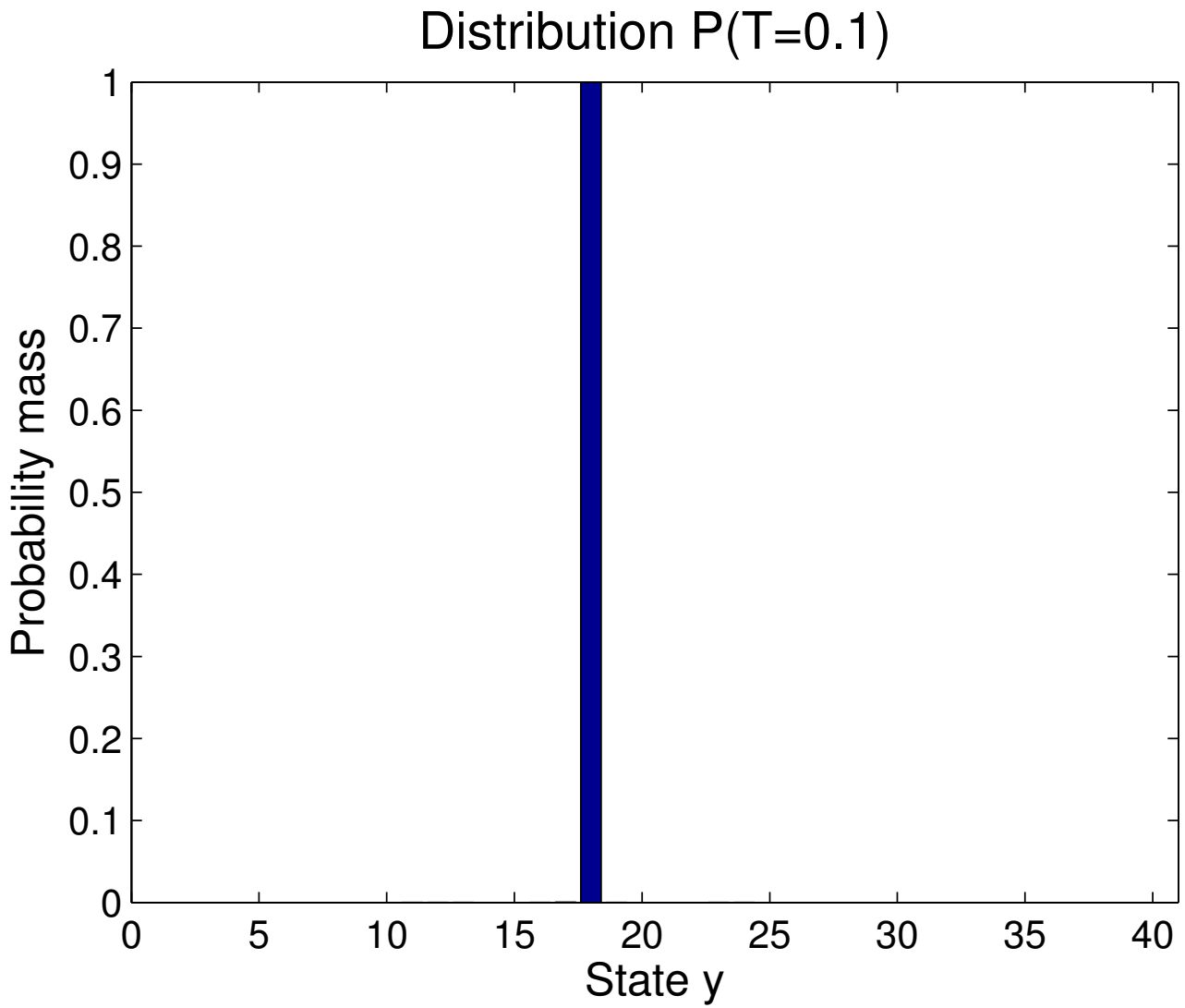
Simulated Annealing (cont)



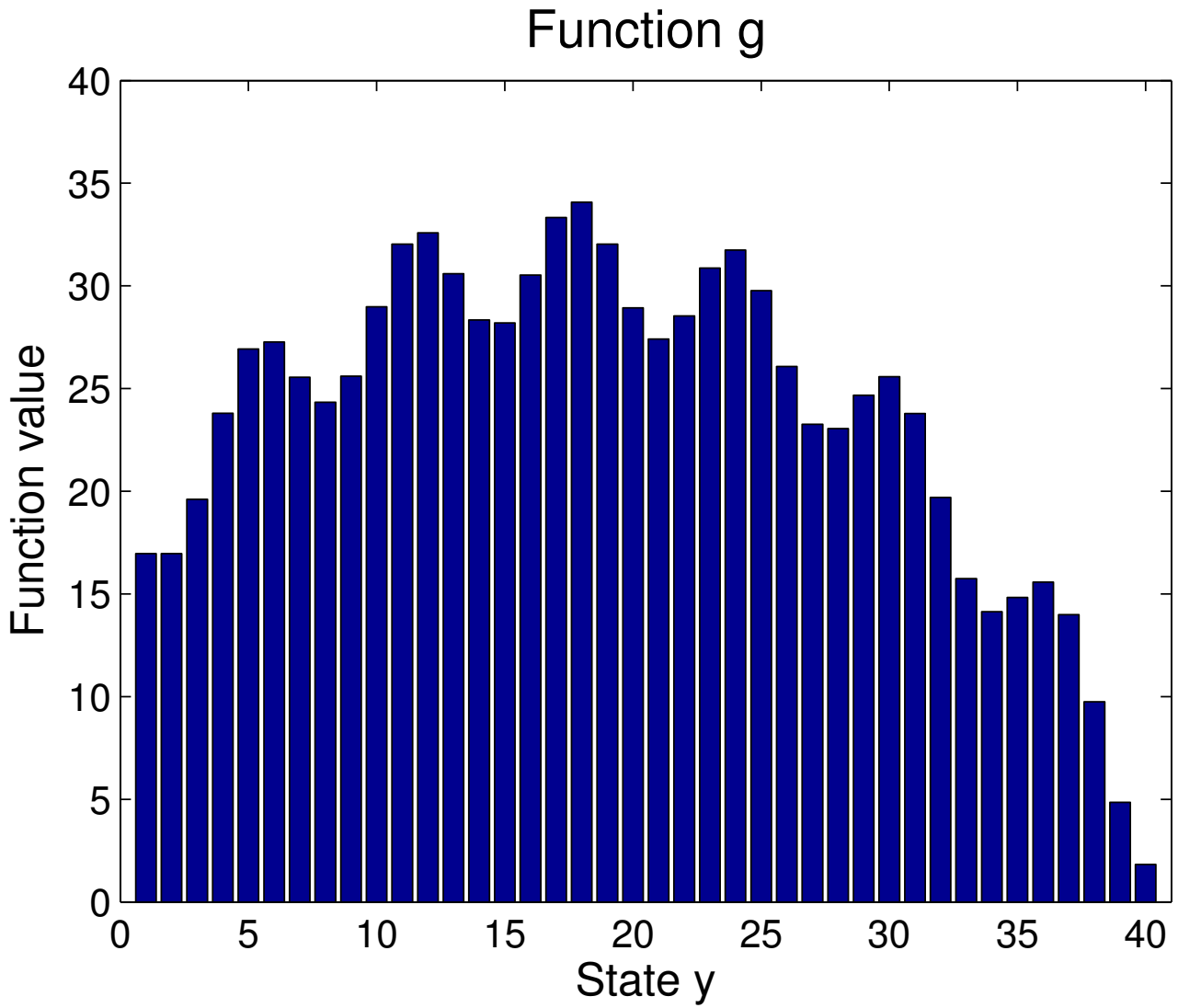
Simulated Annealing (cont)



Simulated Annealing (cont)



Simulated Annealing (cont)



Simulated Annealing (cont)

$y^* = \textsc{SimulatedAnnealing}(\mathcal{Y}, g, T, N)$
Input:
  $\mathcal{Y}$ finite feasible set
  $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ objective function
  $T \in \mathbb{R}^K$ sequence of K decreasing temperatures
  $N \in \mathbb{N}^K$ sequence of K step lengths
$(y, y^*) \leftarrow (y_0, y_0)$
for $k = 1, \ldots, K$ **do**
  $y \leftarrow$ simulate Markov chain $p(y; T(k))$ starting from y for $N(k)$ steps
end for
$y^* \leftarrow y$

Guarantees

Theorem (Guaranteed Optimality (Geman and Geman, 1984))

If there exist a $k_0 \geq 2$ such that for all $k \geq k_0$ the temperature $T(k)$ satisfies

$$T(k) \geq \frac{|\mathcal{Y}| \cdot (\max_{y \in \mathcal{Y}} g(x, y) - \min_{y \in \mathcal{Y}} g(x, y))}{\log k},$$

then the probability of seeing the maximizer y^* of g tends to one as $k \to \infty$.

- too slow in practise
- faster schedules are used in practise, such as

$$T(k) = T_0 \cdot \alpha^k.$$

Guarantees

Theorem (Guaranteed Optimality (Geman and Geman, 1984))

If there exist a $k_0 \geq 2$ such that for all $k \geq k_0$ the temperature $T(k)$ satisfies

$$T(k) \geq \frac{|\mathcal{Y}| \cdot (\max_{y \in \mathcal{Y}} g(x, y) - \min_{y \in \mathcal{Y}} g(x, y))}{\log k},$$

then the probability of seeing the maximizer y^* of g tends to one as $k \to \infty$.

- too slow in practise
- faster schedules are used in practise, such as

$$T(k) = T_0 \cdot \alpha^k.$$

Example: Simulated Annealing

(Geman and Geman, 1984)

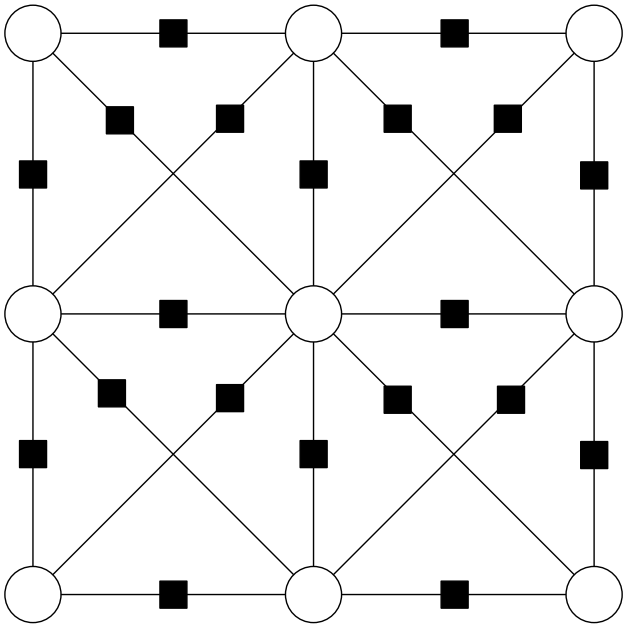


Figure: Factor graph model

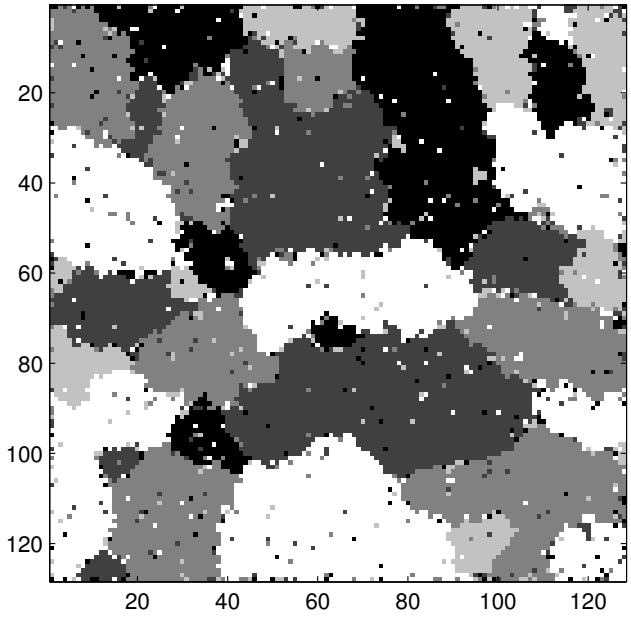


Figure: Approximate sample (200 sweeps)

- 2D 8-neighbor 128-by-128 grid, 5 possible labels
- Pairwise Potts potentials
- Task: restoration

Example: Simulated Annealing

(Geman and Geman, 1984)

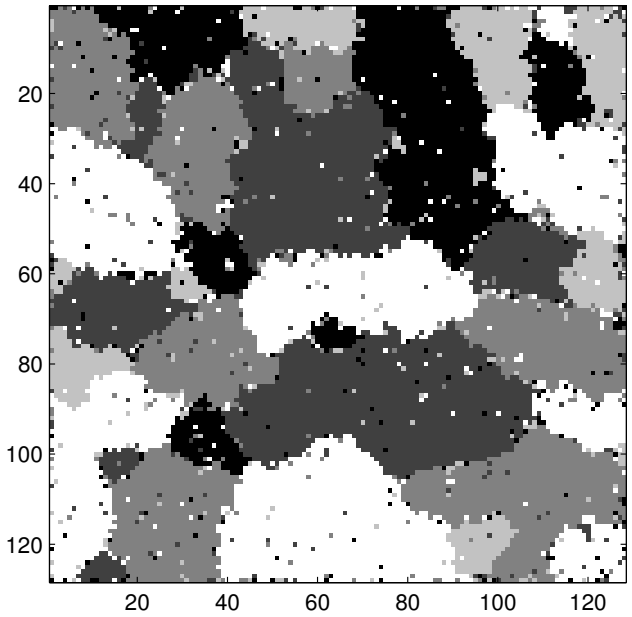


Figure: Approximate sample (200 sweeps)

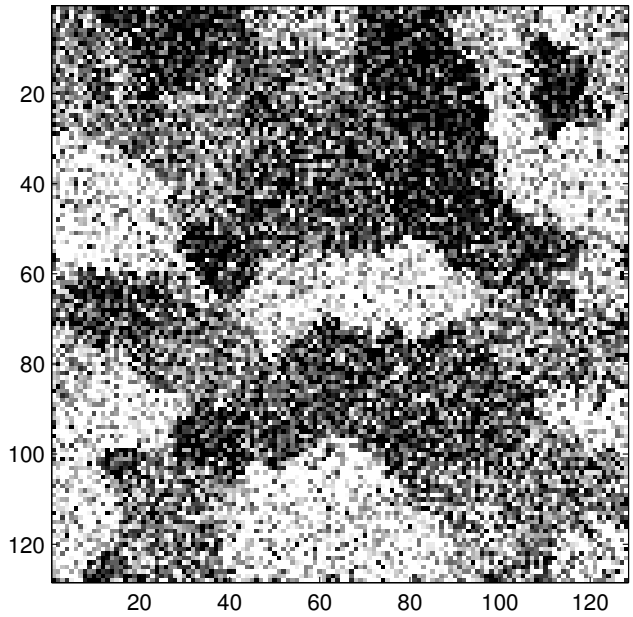


Figure: Corrupted with Gaussian noise

- 2D 8-neighbor 128-by-128 grid, 5 possible labels
- Pairwise Potts potentials
- Task: restoration

Example: Simulated Annealing

(Geman and Geman, 1984)

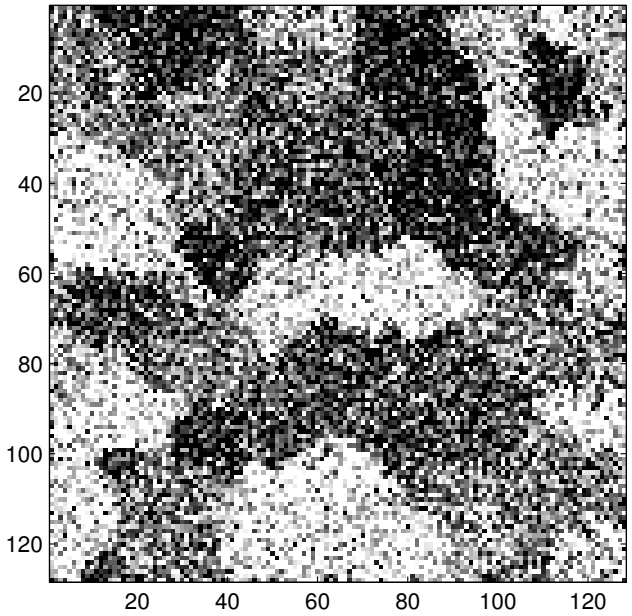


Figure: Corrupted input image

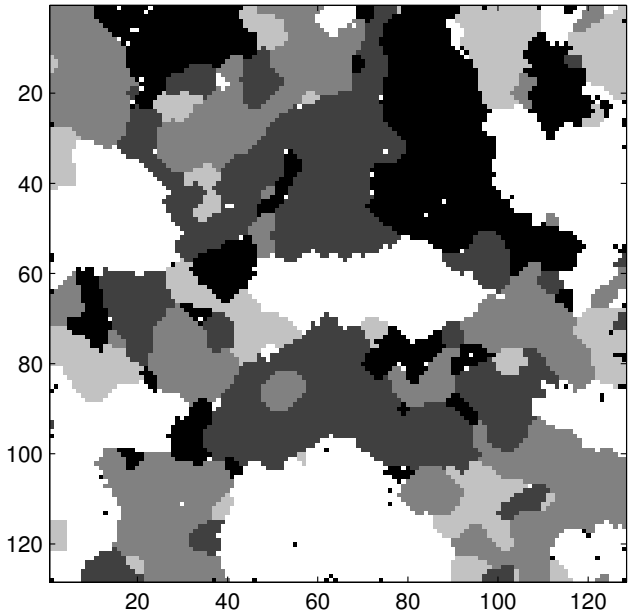


Figure: Restoration, 25 sweeps

- Derive unary energies from corrupted input image (optimal)
- Simulated annealing, 25 Gibbs sampling sweeps

Example: Simulated Annealing

(Geman and Geman, 1984)

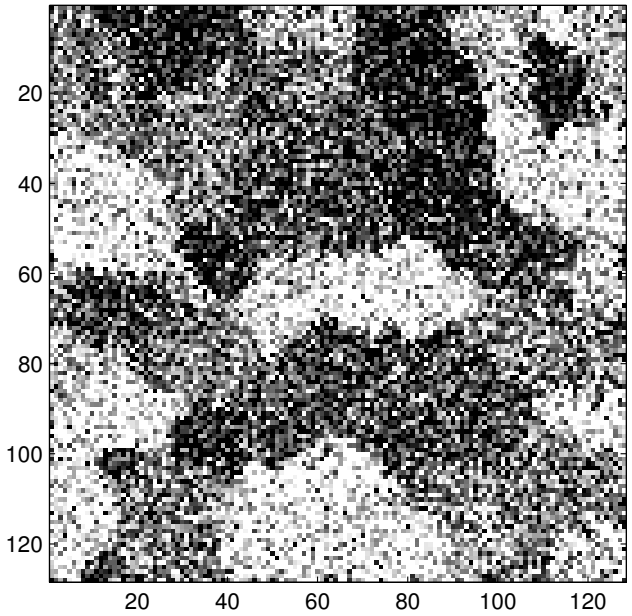


Figure: Corrupted input image

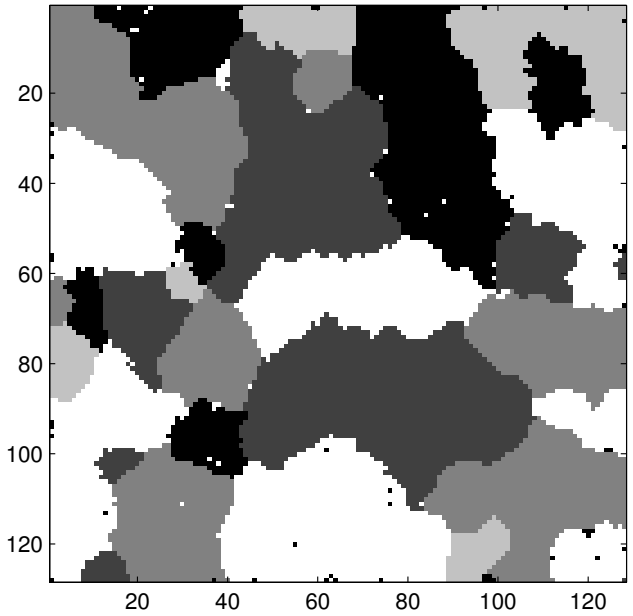


Figure: Restoration, 300 sweeps

- Simulated annealing, 300 Gibbs sampling sweeps
- Schedule $T(k) = 4.0/\log(1+k)$

Example: Simulated Annealing

(Geman and Geman, 1984)

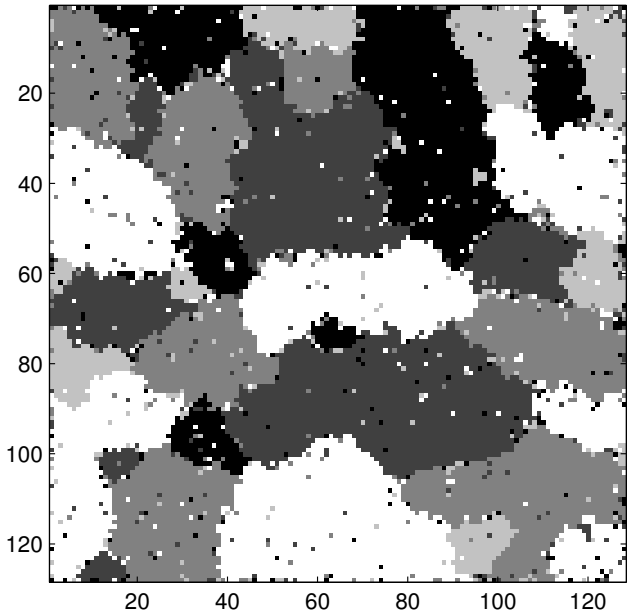


Figure: Noise-free sample

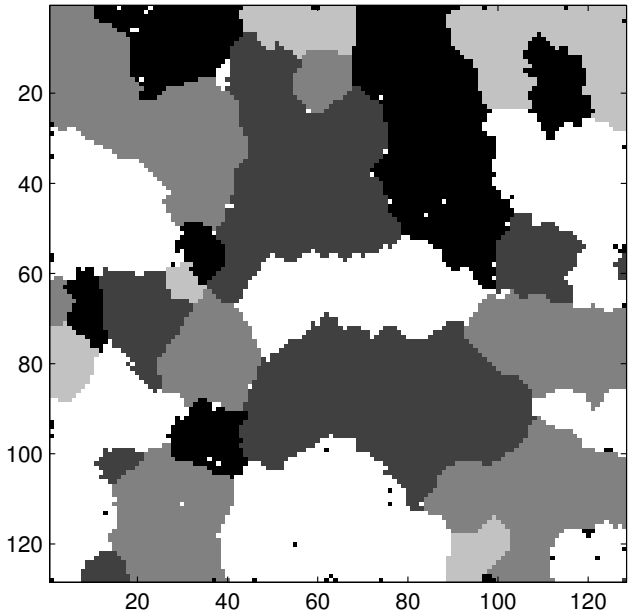
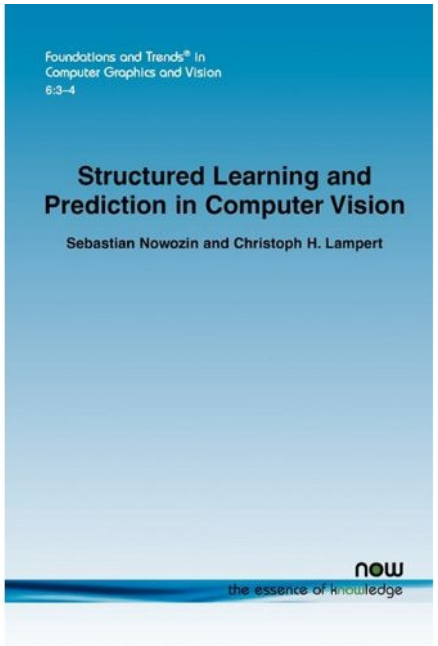


Figure: Restoration, 300 sweeps

- Simulated annealing, 300 Gibbs sampling sweeps
- Schedule $T(k) = 4.0/\log(1+k)$

The End...

- Tutorial in written form
- now publisher's FnT Computer Graphics and Vision series
- `http://www.nowpublishers.com/`
- PDF available on authors' homepages



Thank you!