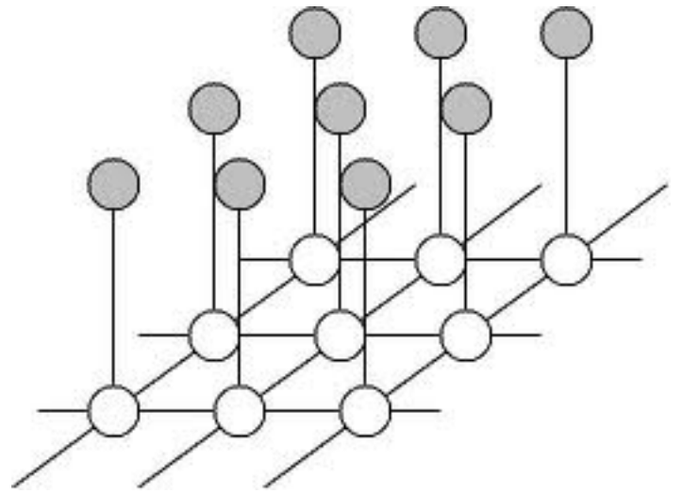
Sampling-based methods for diverse solutions

Objective function

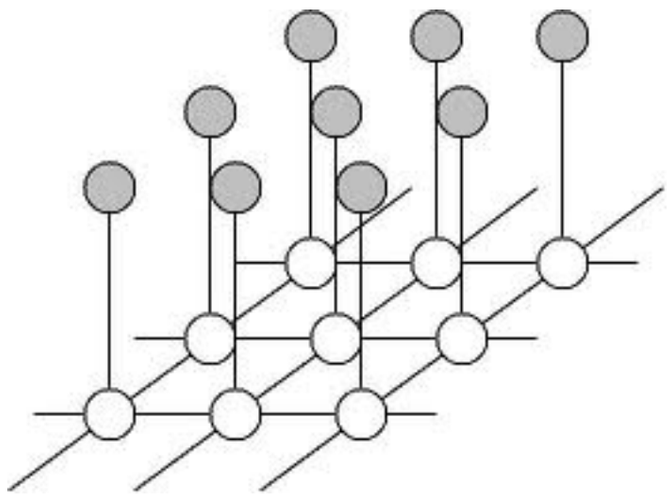
$$S(y, x) = \sum_{i \in V} \phi_i(y_i, x) + \sum_{ij \in E} \psi_{ij}(y_i, y_j, x)$$

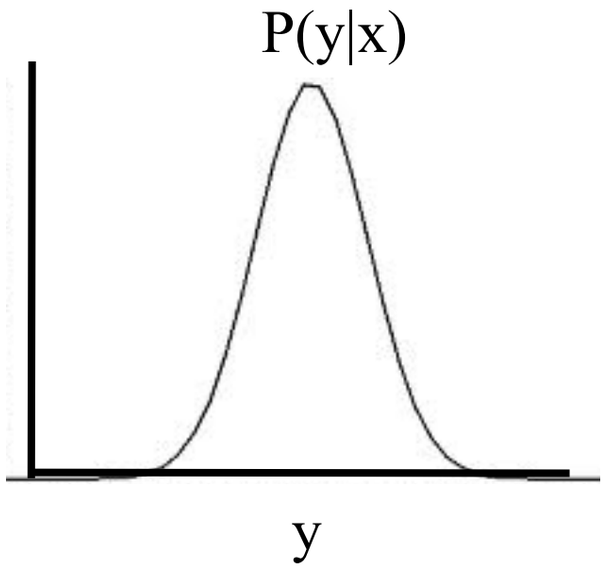


$$\max_y S(y, x)$$

Probabilistic interpretation

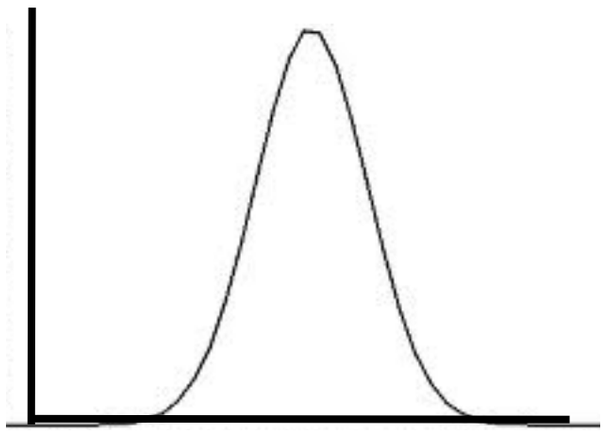
$$P(y|x) \propto e^{S(y,x)}$$





Bayesian view

$$P(y|x) \propto e^{S(y,x)}$$



y

MAP: Infer single y*

N (diverse) best: Infer a discrete set $\{y^{1*},y^{2*},...y^{3*},...\}$

Bayes: Work with posterior P(y|x)

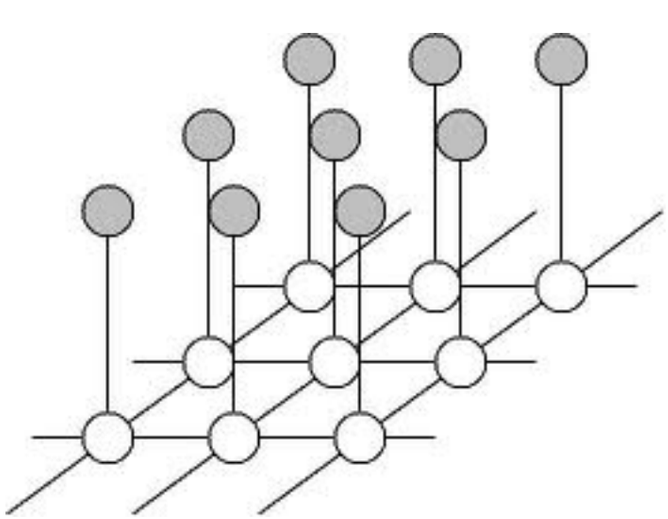
"Right way" to handle ambiguity, but difficult to represent

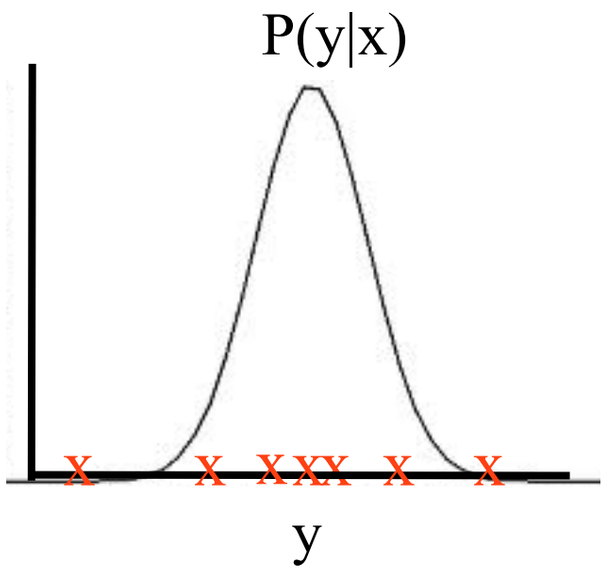
Turns out, in some cases we can!

Approximate strategy

$$P(y|x) \propto e^{S(y,x)}$$

Represent posterior with samples





Generate high-scoring-ish and diverse-ish solutions by sampling

Let's simplify notation

$$S(y, x) = \sum_{i \in V} \phi_i(y_i, x) + \sum_{ij \in E} \psi_{ij}(y_i, y_j, x)$$

$$\max_y S(y, x)$$

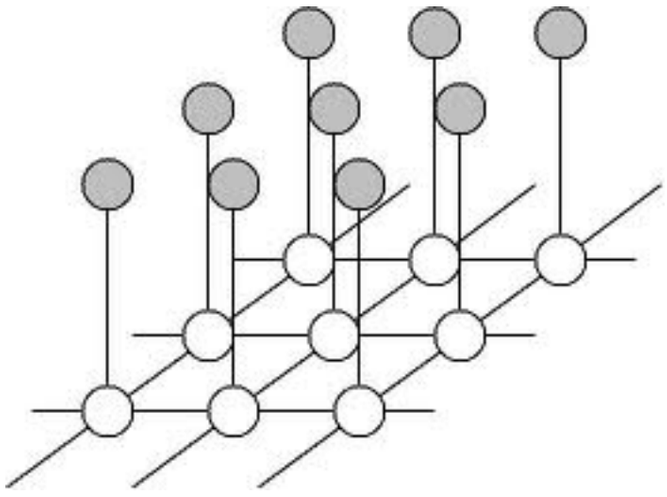
For notational simplicity, let's not write out evidence "x" and implicitly condition on it

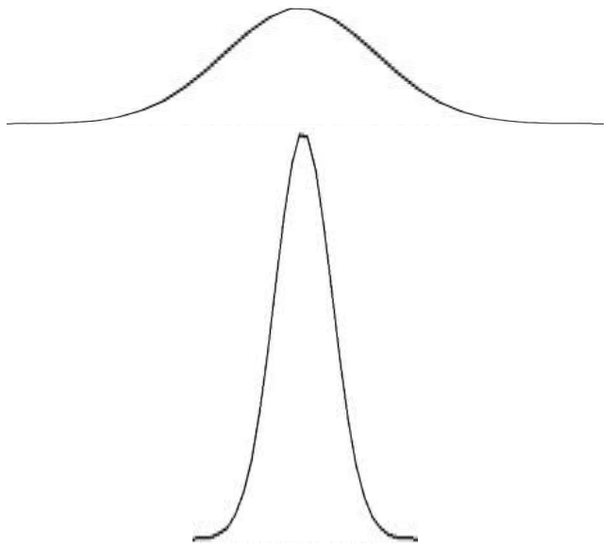
$$S(y) = \sum_{i \in V} \phi_i(y_i) + \sum_{ij \in E} \psi_{ij}(y_i, y_j)$$

Temperature parameter

$$S(y) = \sum_{i \in V} \phi_i(y_i) + \sum_{ij \in E} \psi_{ij}(y_i, y_j)$$

$$P(y) \propto e^{tS(y)}$$





t is optimal temperature parameter

t→0 Sample uniformly (more diverse)

t→∞ Sample MAP (more high-scoring)

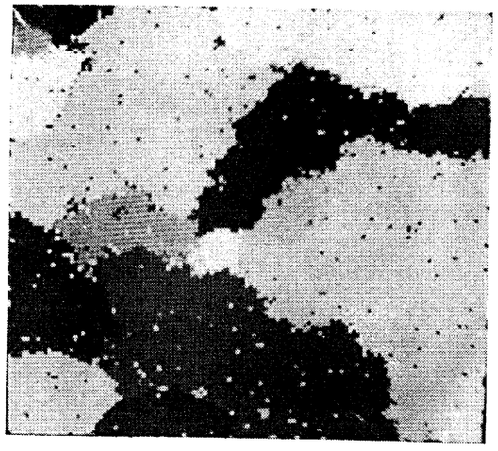
Gibbs sampling

Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images

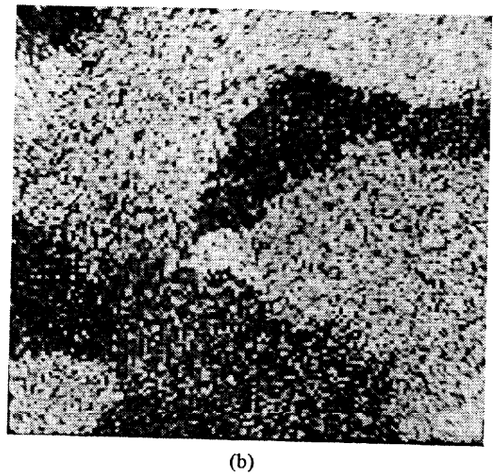
STUART GEMAN AND DONALD GEMAN

Abstract—We make an analogy between images and statistical mechanics systems. Pixel gray levels and the presence and orientation of edges are viewed as states of atoms or molecules in a lattice-like physical system. The assignment of an energy function in the physical system determines its Gibbs distribution. Because of the Gibbs distribution, Markov random field (MRF) equivalence, this assignment also determines an MRF image model. The energy function is a more convenient and natural mechanism for embodying picture attributes than are the local characteristics of the MRF. For a range of degradation mechanisms, including blurring, nonlinear deformations, and multiplicative or additive noise, the posterior distribution is an MRF with a structure

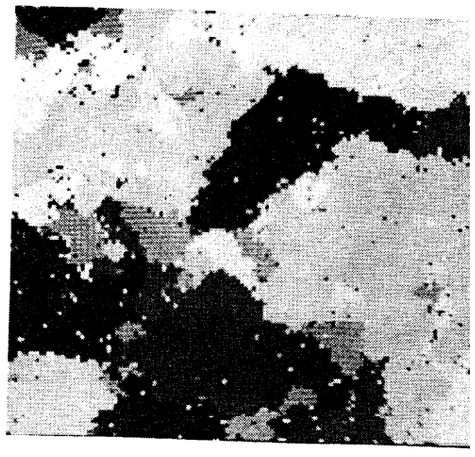
The essence of our approach to restoration is a stochastic relaxation algorithm which generates a sequence of images that converges in an appropriate sense to the MAP estimate. This sequence evolves by *local* (and potentially *parallel*) changes in pixel gray levels and in locations and orientations of boundary elements. Deterministic, iterative-improvement methods generate a sequence of images that monotonically increase the posterior distribution (our "objective function"). In contrast, stochastic relaxation permits changes that *decrease* the pos



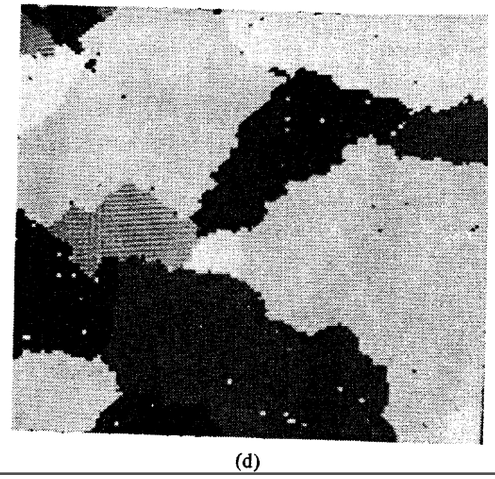
(a)



(b)



(c)



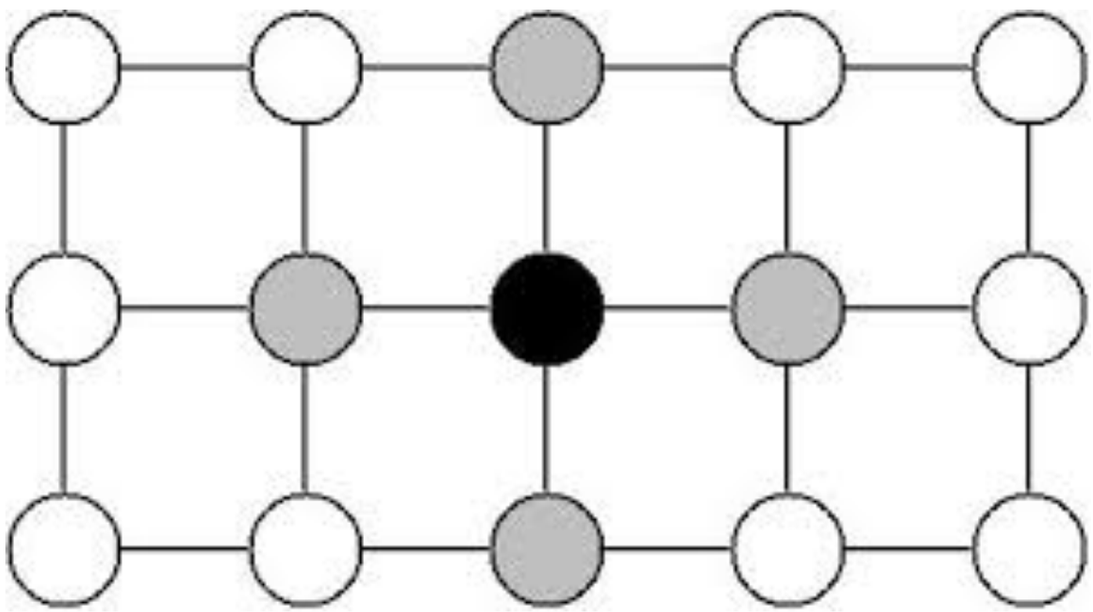
(d)

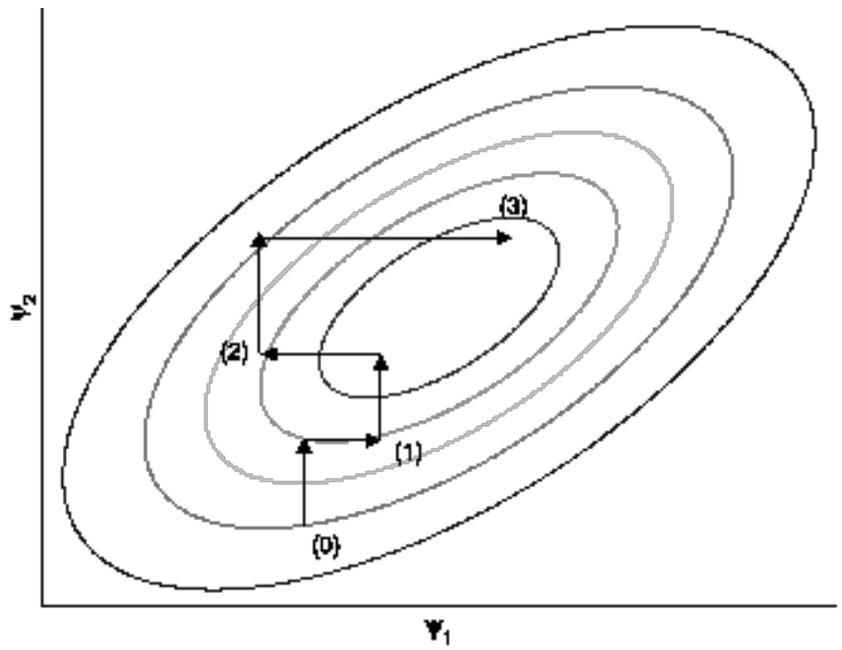
Algorithm (wikipedia)

Gibbs sampling, in its basic incarnation, is a special case of the Metropolis–Hastings algorithm. The point of Gibbs sampling is that given a multivariate distribution it is simpler to sample from a conditional distribution than to marginalize by integrating over a joint distribution. Suppose we want to obtain k samples of $\mathbf{X} = \{x_1, \ldots, x_n\}$ from a joint distribution $p(x_1, \ldots, x_n)$. Denote the ith sample by $\mathbf{X}^{(i)} = \{x_1^{(i)}, \ldots, x_n^{(i)}\}$. We proceed as follows:

1. We begin with some initial value $\mathbf{X}^{(0)}$ for each variable.
2. For each sample $i = \{1 \ldots k\}$, sample each variable $x_j^{(i)}$ from the conditional distribution $p(x_j | x_1^{(i)}, \ldots, x_{j-1}^{(i)}, x_{j+1}^{(i-1)}, \ldots, x_n^{(i-1)})$. That is, sample each variable from the distribution of that variable conditioned on all other variables, making use of the most recent values and updating the variable with its new value as soon as it has been sampled.

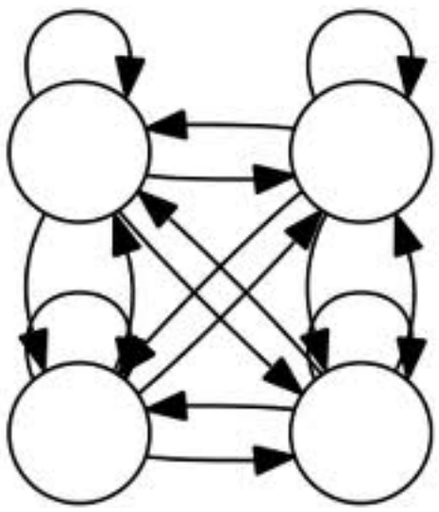
Iterate over variables "i" and sample: $P(y_i | y_{\neg i}) = P(y_i | y_{N(i)})$

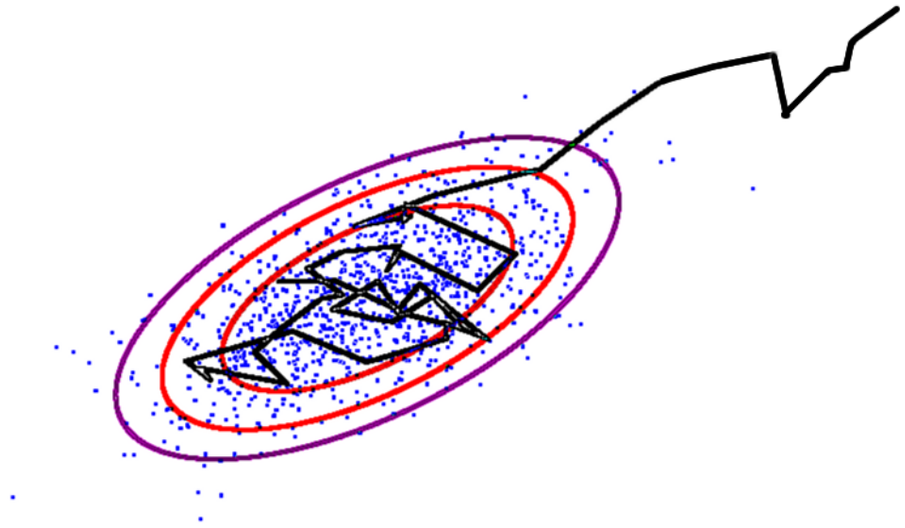




MCMC
(markov-chain monte carlo)

Approach: design a markov chain who's stationary distribution = model posterior





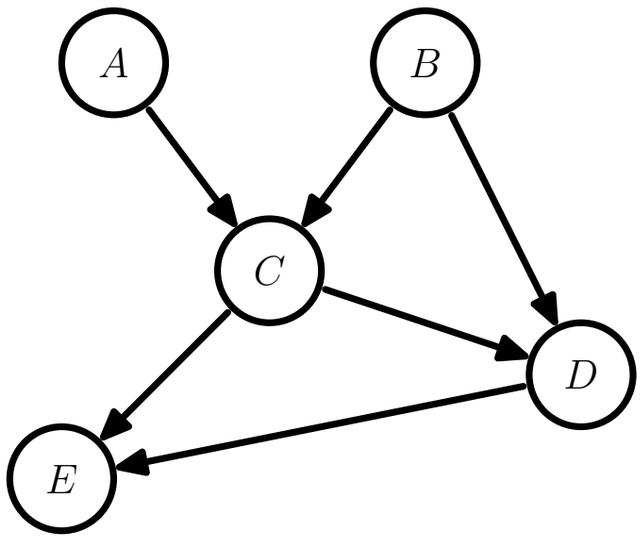
(Key issue) mixing time: Amount of time until samples reach stationary distribution (i.e.,"amount of time until sampler forgets its initialization")

Sampling from any directed graphical model

Ancestral pass for directed graphical models:

— sample each top level variable from its marginal

— sample each other node from its conditional once its parents have been sampled

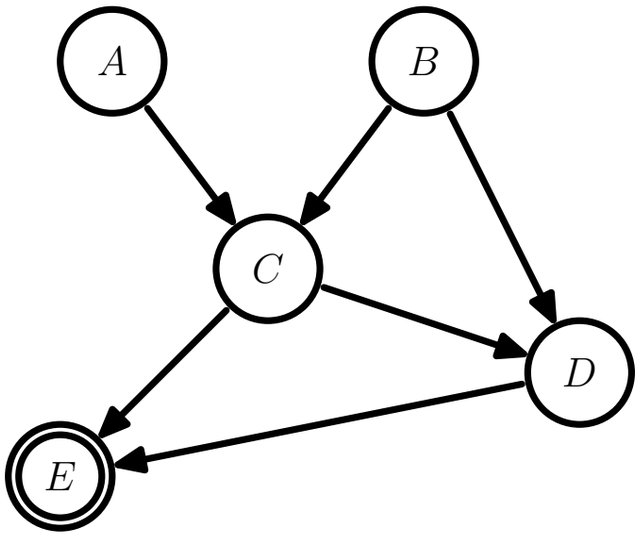


Sample:

$A \sim P(A)$
$B \sim P(B)$
$C \sim P(C \mid A, B)$
$D \sim P(D \mid B, C)$
$E \sim P(D \mid C, D)$

$$P(A, B, C, D, E) = P(A)\, P(B)\, P(C \mid A, B)\, P(D \mid B, C)\, P(E \mid C, D)$$

Problem

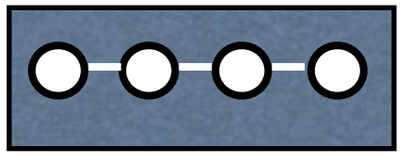


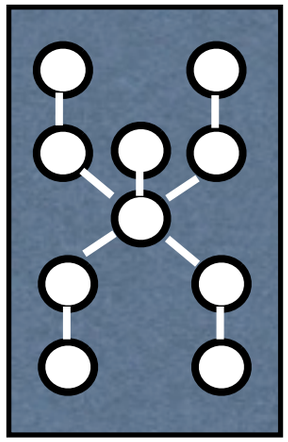
Posterior of a directed graphical model

$$P(A, B, C, D \mid E) = \frac{P(A, B, C, D, E)}{P(E)}$$

Computing P(E) is just as hard as computing normalizer of MRF (partition function)

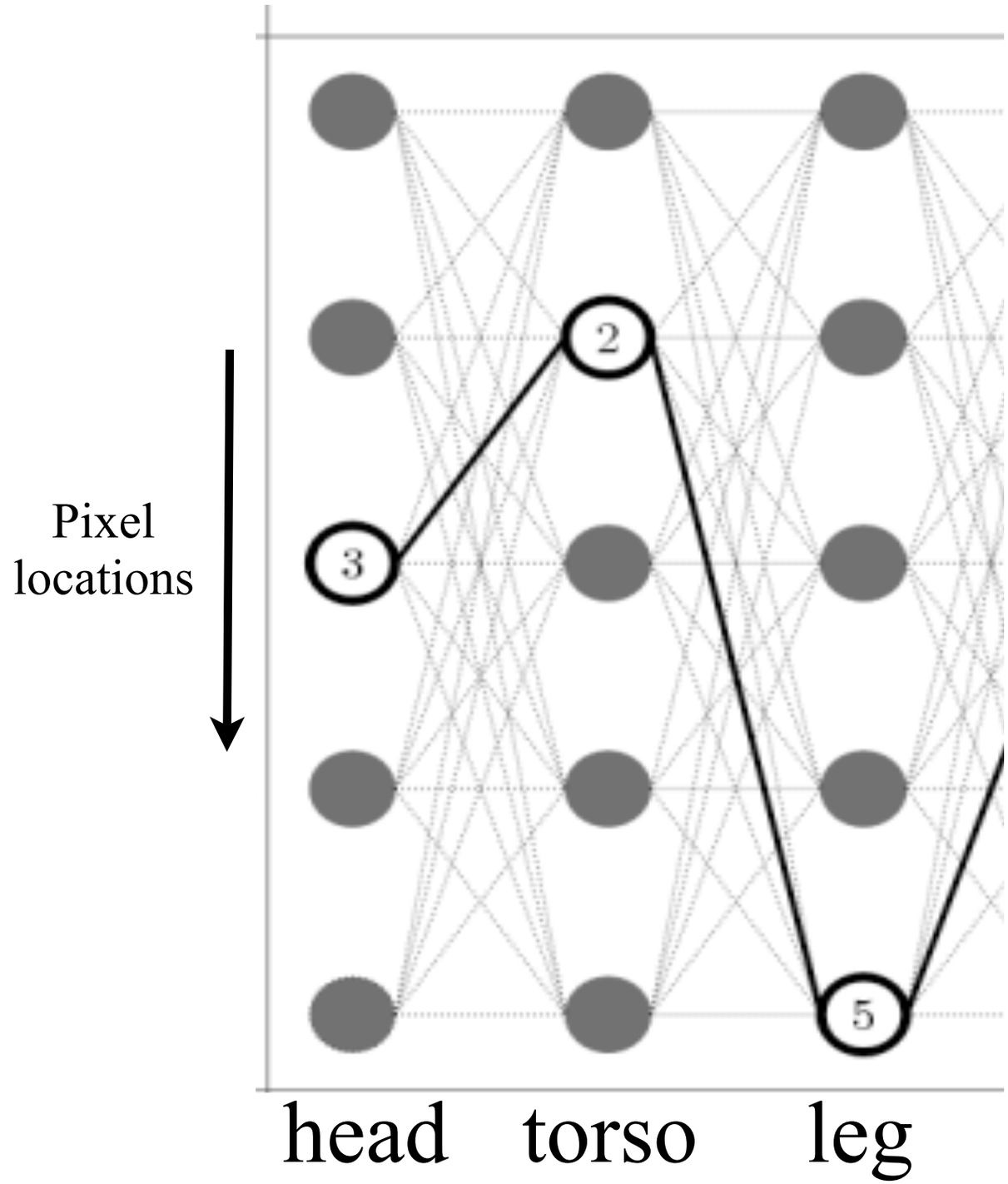
Special case

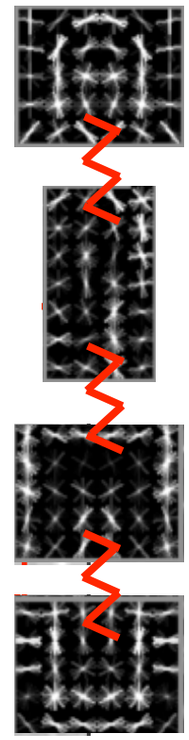




We can compute P(E) (or normalizer) for these graphs

Recall: dynamic programming on chains





1) Initialize nodes with match score

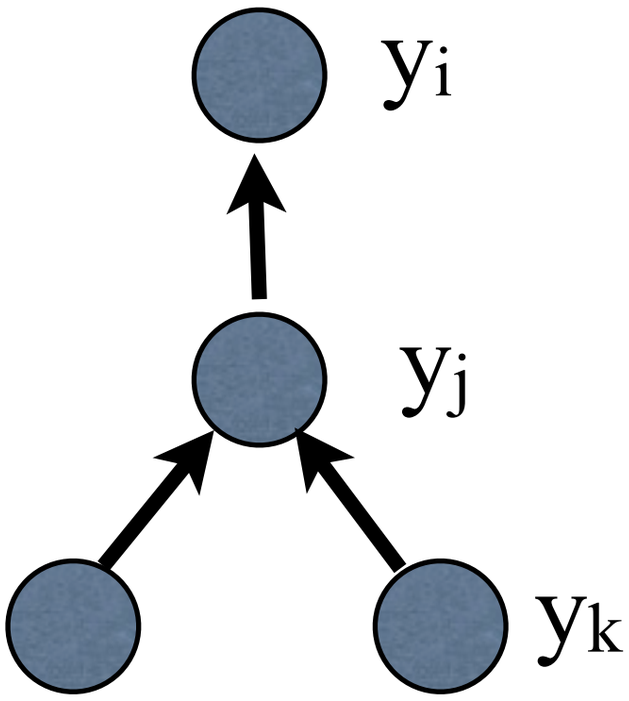
2) Initialize edges with spring score

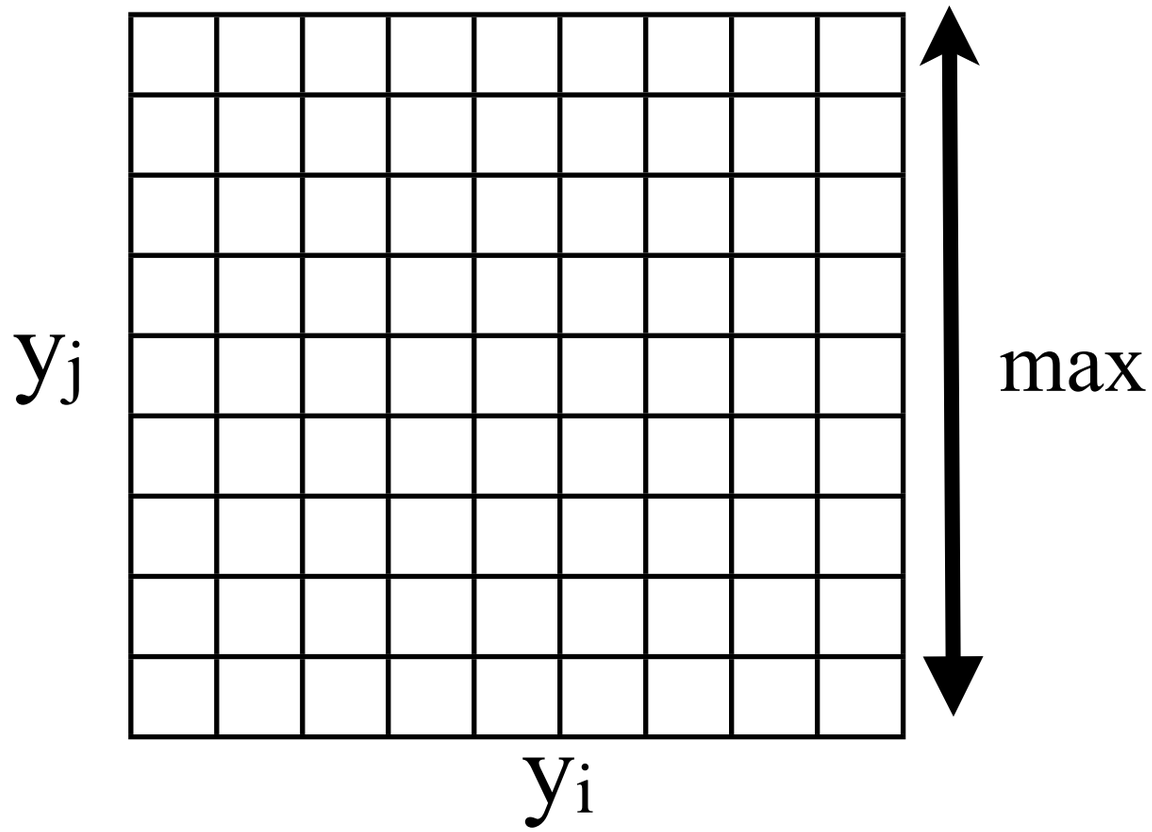
3) Find best path from left to right

Message passing

$$S(y) = \sum_{i \in V} \phi_i(y_i) + \sum_{ij \in E} \psi_{ij}(y_i, y_j)$$

$$m_j(y_i) = \max_{y_j} \left[\phi_j(y_j) + \psi_{ij}(y_i, y_j) + \sum_{k \in kids(j)} m_k(y_j) \right]$$





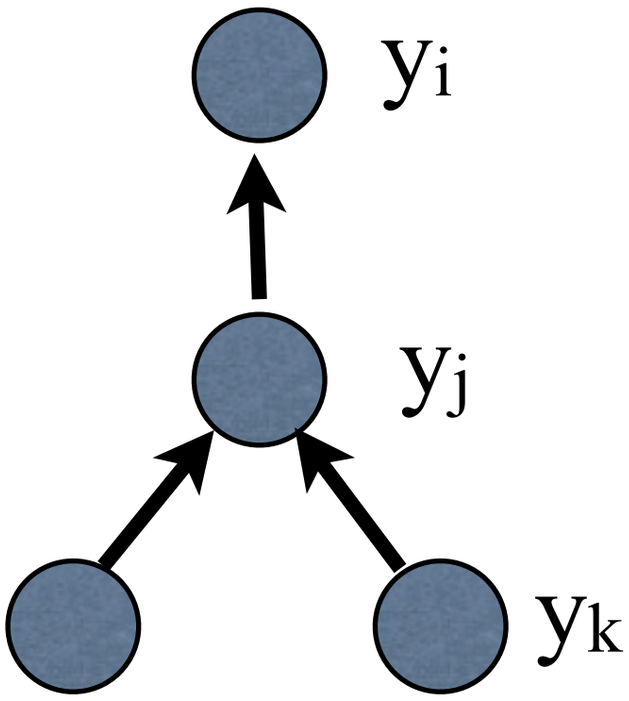
Sum-product messages

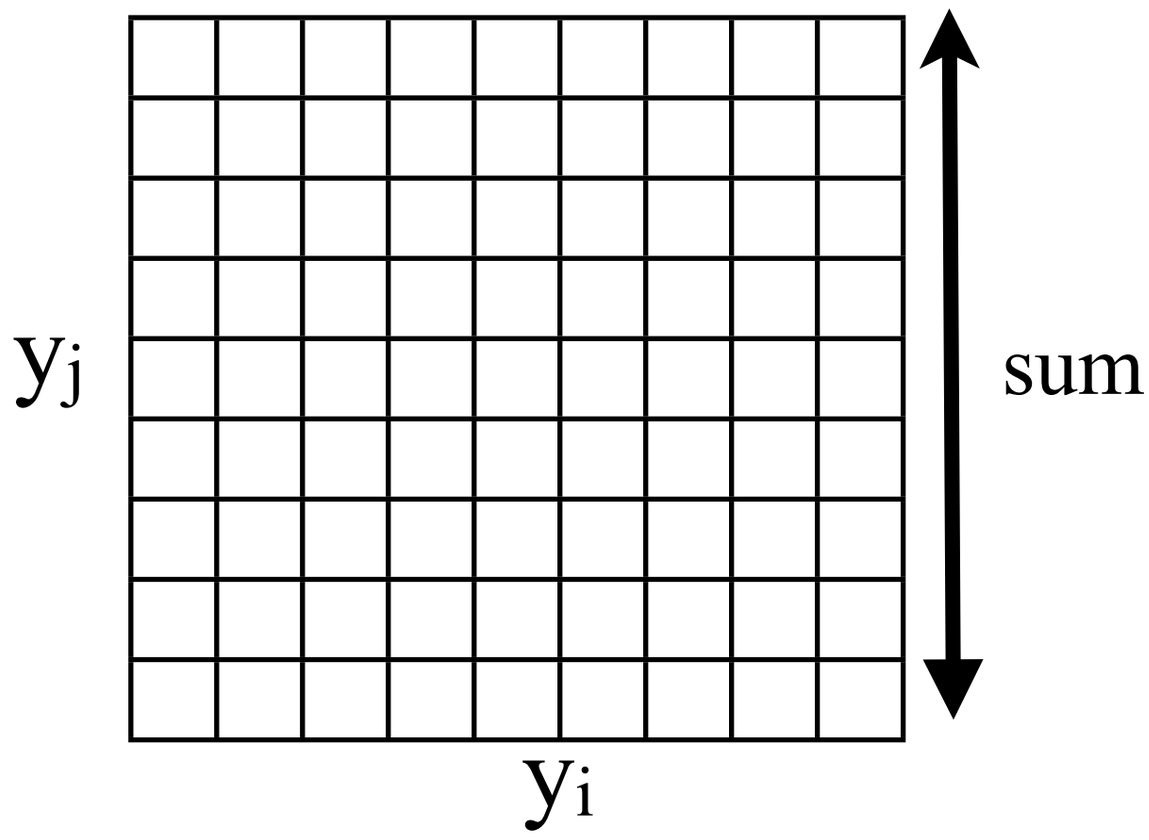
sum→prod

max→sum

$$P(y) = \frac{1}{Z} e^{S}(y)$$

$$m_j(y_i) \propto \sum_{y_j} \left[e^{\phi_i(y_i) + \psi_{ij}(y_i, y_j)} \prod_{k \in kids(j)} m_k(y_j) \right]$$



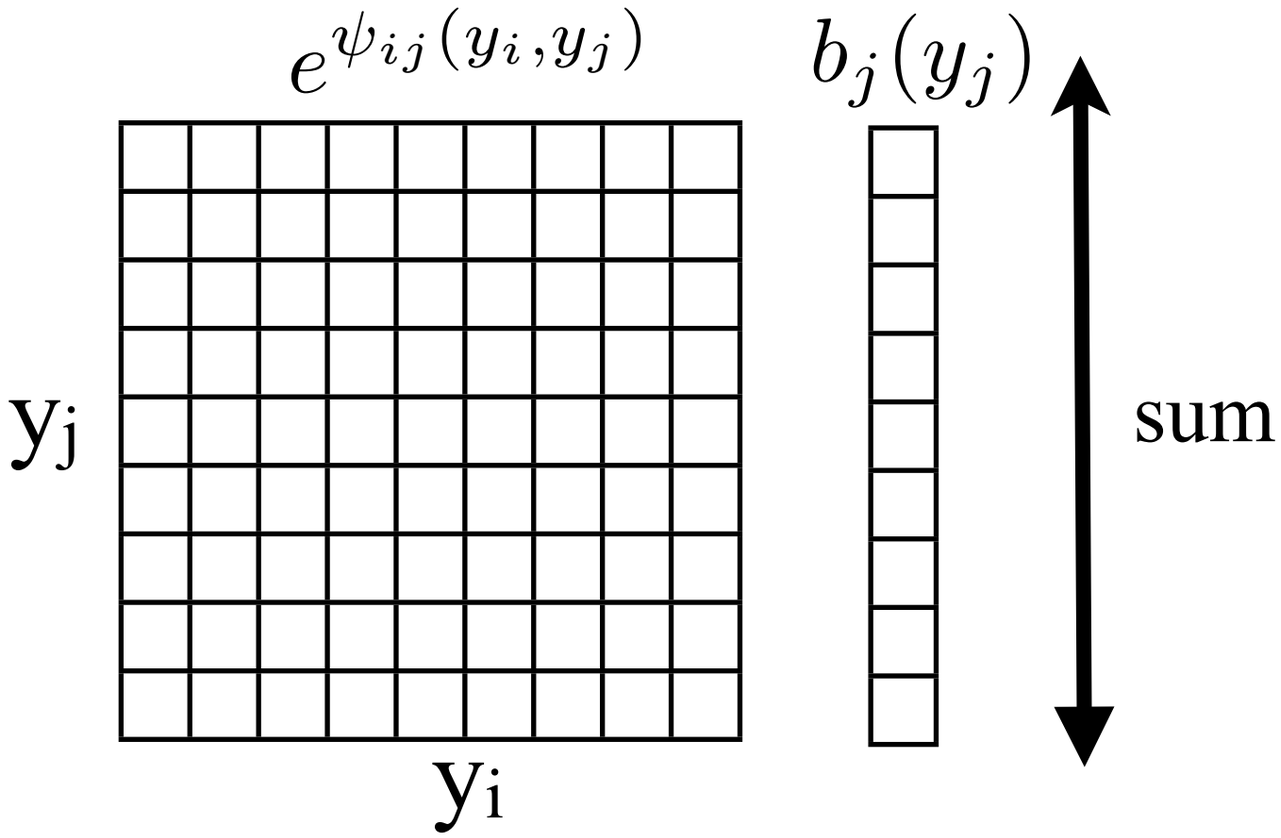


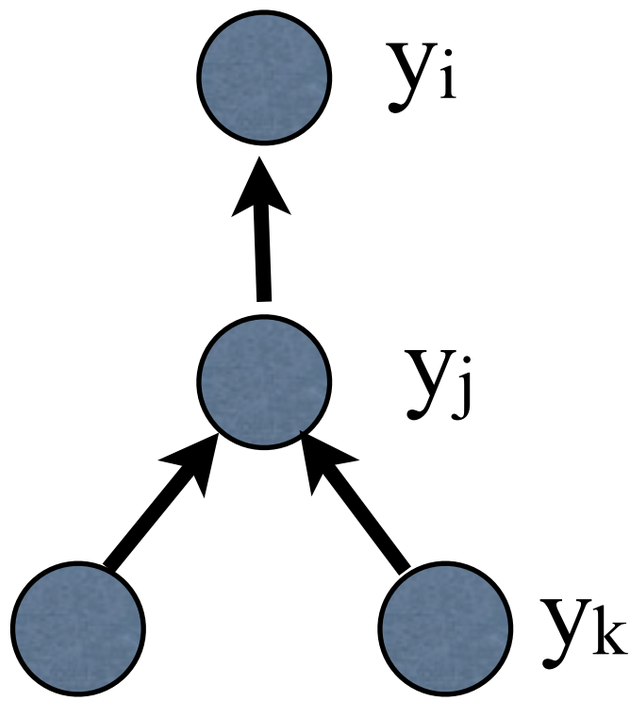
Sum-product messages

Cached (and normalized) pairwise matrices are conditional probabilities!

$$P(y_j|y_i) \propto e^{\psi_{ij}(y_i,y_j)} b_j(y_j)$$

$$b_j(y_j) \propto e^{\phi_j(y_j)} \prod_{k \in kids(j)} \sum_{y_k} P(y_k|y_j)$$

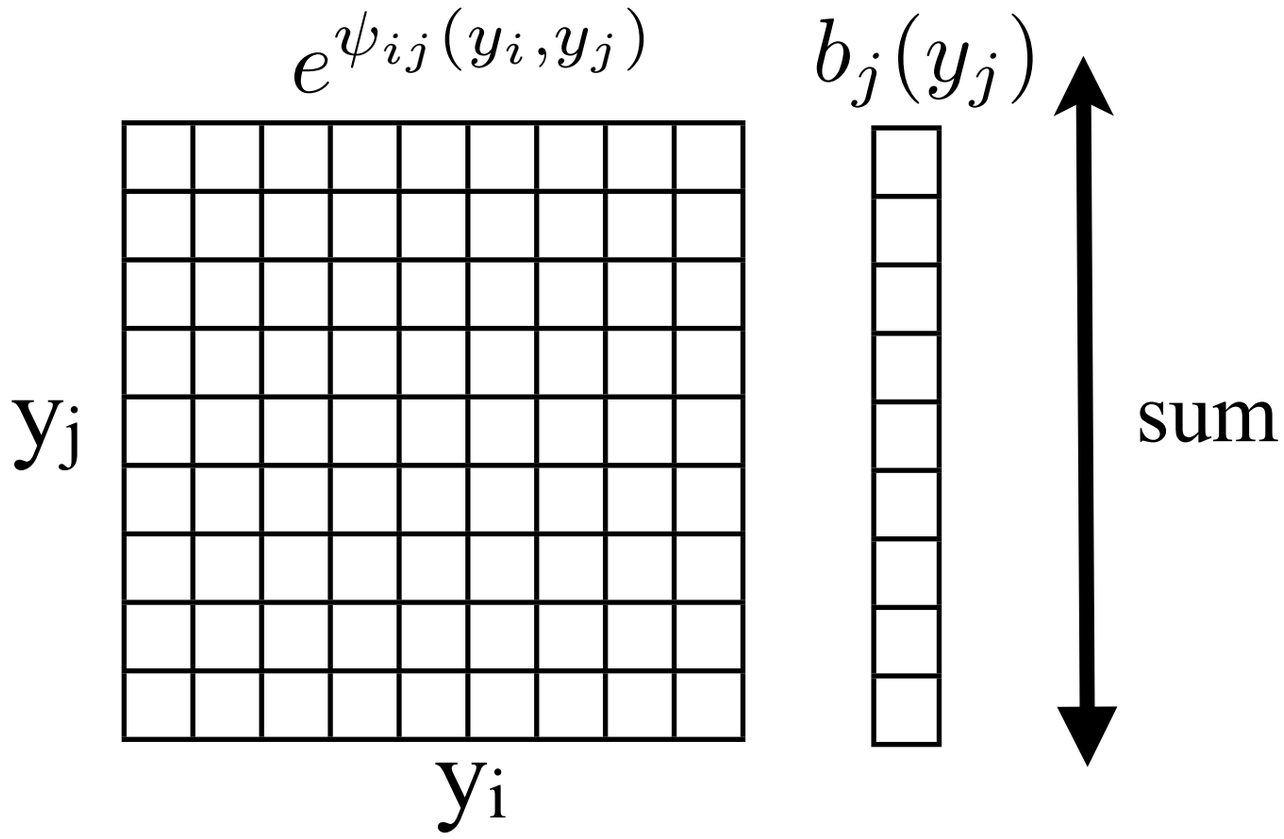


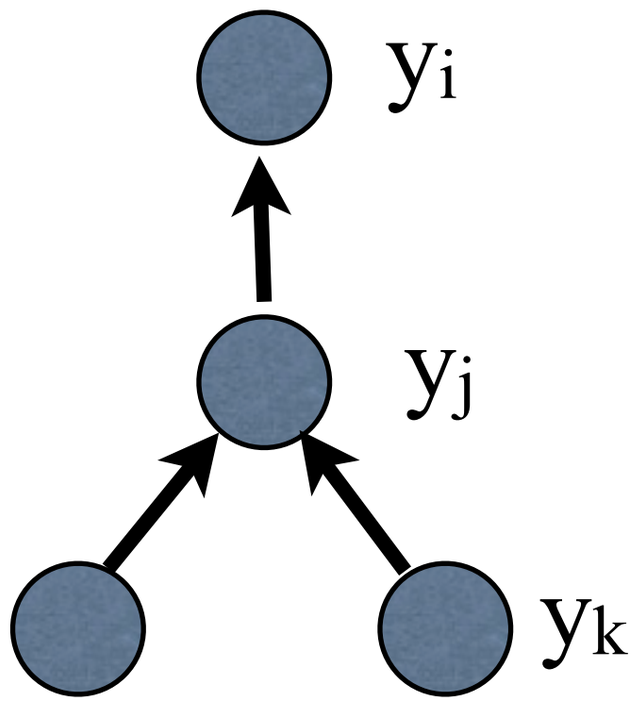


(Conditional) sum-product messages

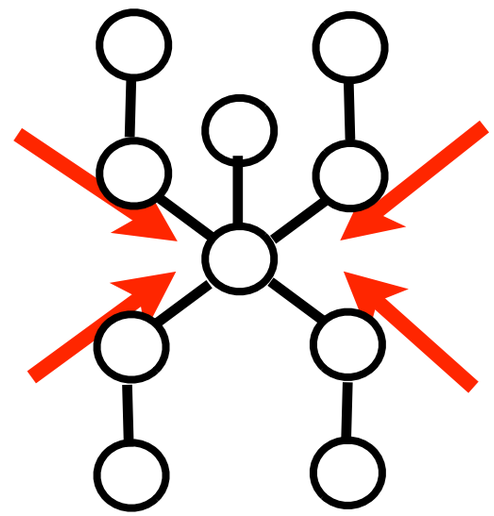
$$P(y_j|y_i, x) \propto e^{\psi_{ij}(y_i, y_j, x)} b_j(y_j)$$

$$b_j(y_j) \propto e^{\phi_j(y_j, x)} \prod_{k \in kids(j)} \sum_{y_k} P(y_k|y_j, x)$$



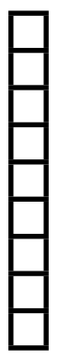


“Forward” pass



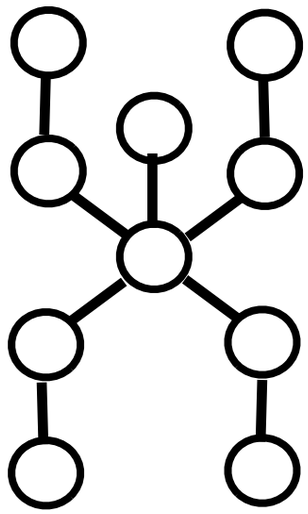
$$b_j(y_j) \propto e^{\phi_j(y_j,x)} \prod_{k \in kids(j)} \sum_{y_k} P(y_k|y_j,x)$$

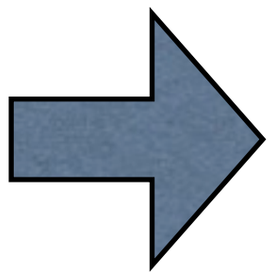
At root (j=0), $b_0(y_0) = P(y_0)$

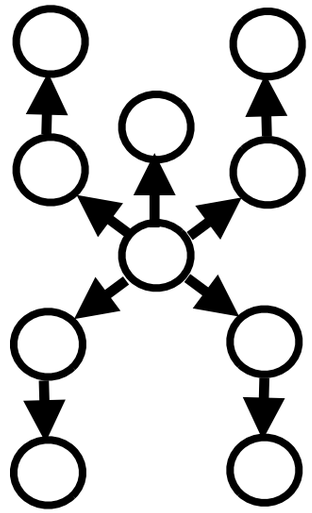


Root marginal

Message passing as reparameterization







$$P(y|x) \propto e^{S(x,y)}$$

Undirected tree

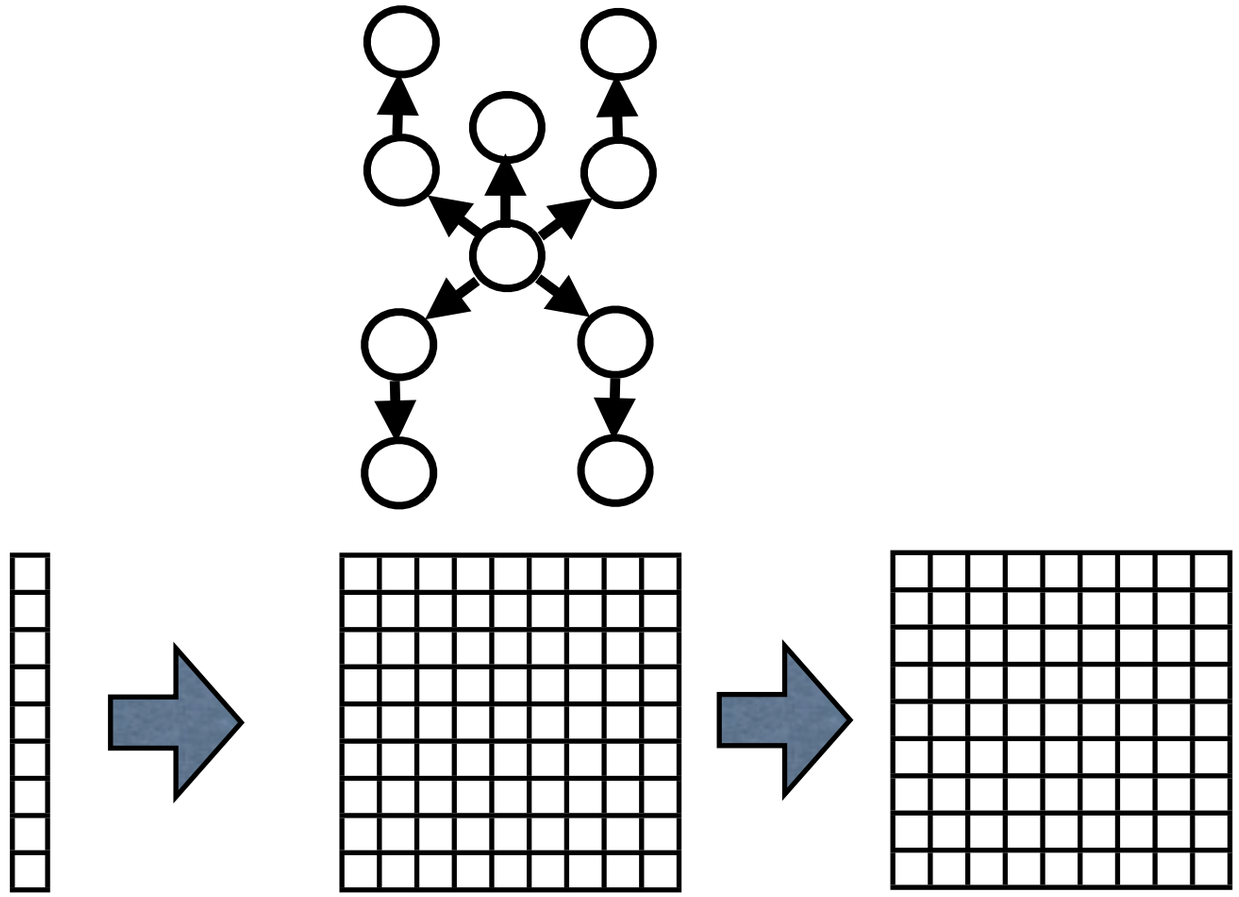
$$P(y|x) = P(y_1|x) \prod_{ij \in E} P(y_i|y_j, x)$$

Directed tree

Wainwright et al, IEEE Trans. Info Theory 03

“Backward” pass

Sample from reparameterized directed tree



Root marginal

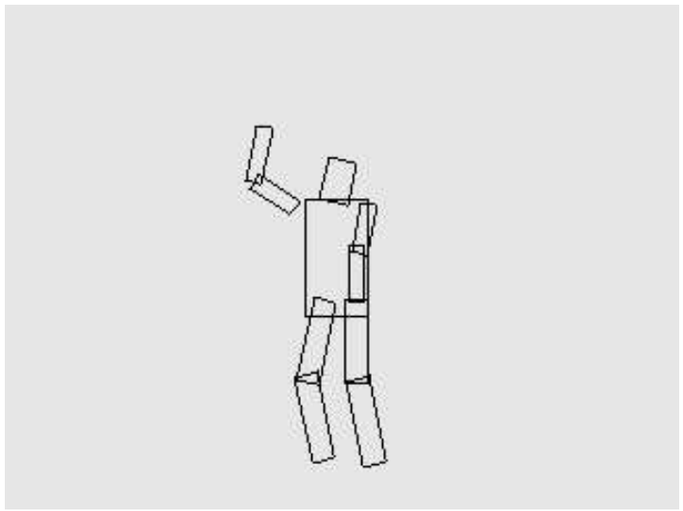
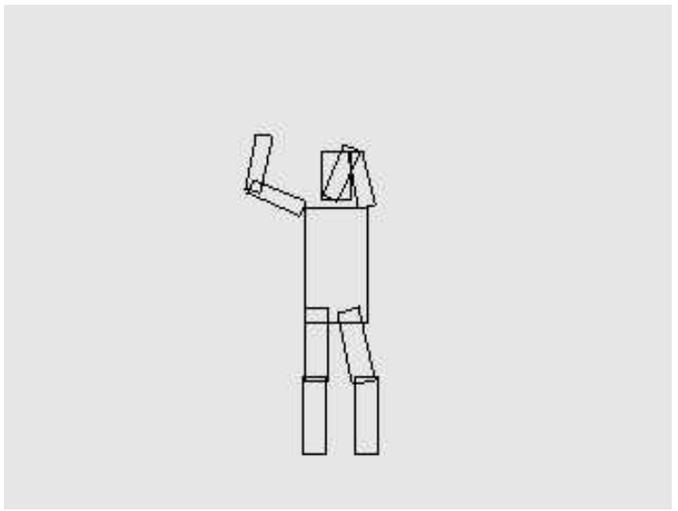
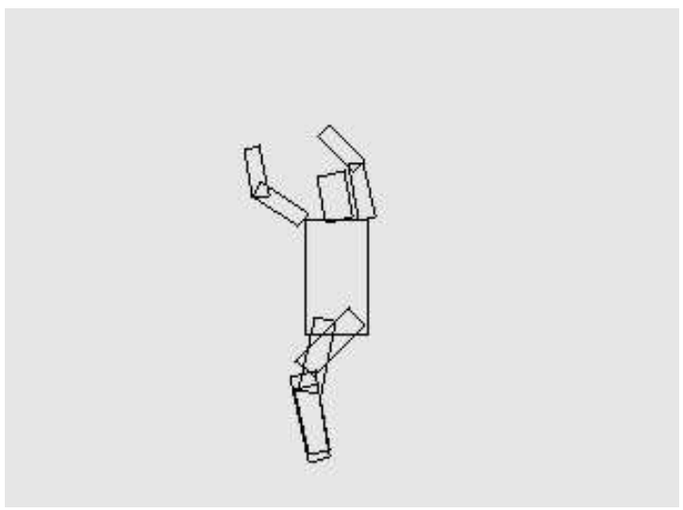
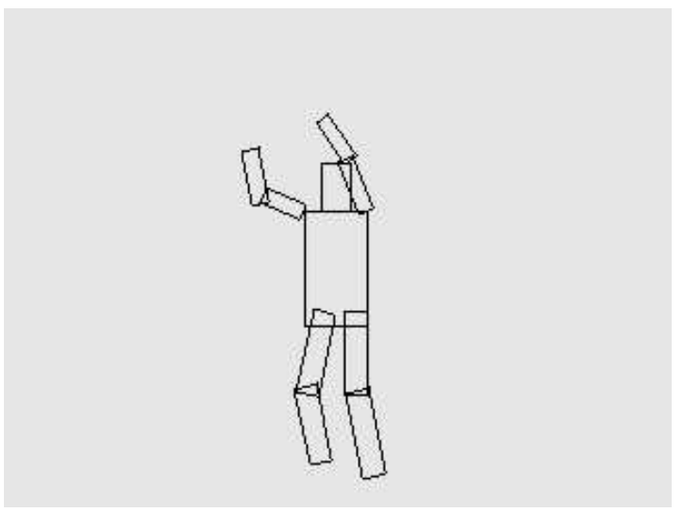
Pairwise CPTs

At root (j=0), $b_0(y_0) = P(y_0)$

Sample and walk back down tree: $P(y_0) \Rightarrow P(y_1|y_0) \Rightarrow \ldots$

Each sample requires a table look-up per part
Zero mix-in time!



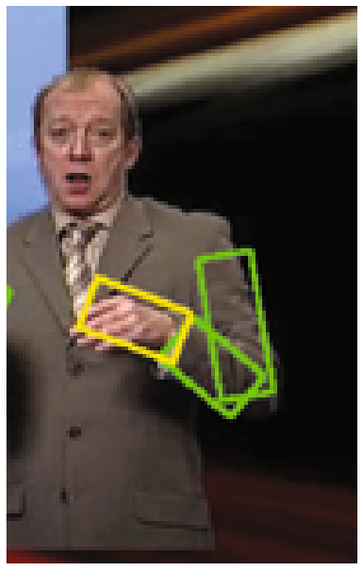





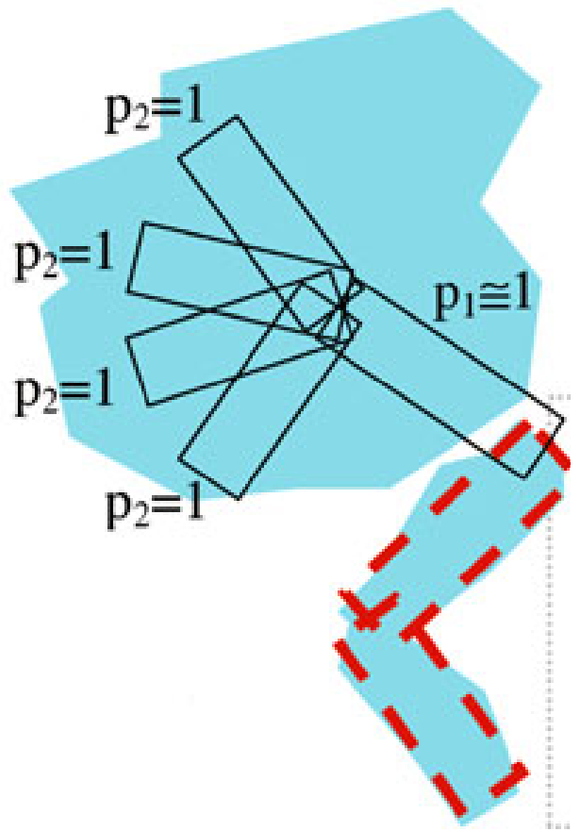
Felzenszwalb & Huttenlocher, IJCV05

An approximate strategy

Sample from max-product messages

Beuler, Everingham, Huttenlocher, & Zisserman IJCV 11





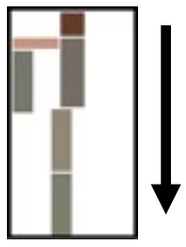
Argument: probabilistic sampling can't distinguish between 1 high scoring pose and lots of low-scoring poses

Superimpose samples

Ramanan et al 07

Superimpose samples

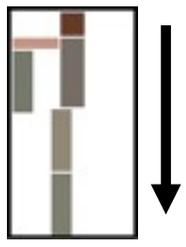
Ramanan et al 07



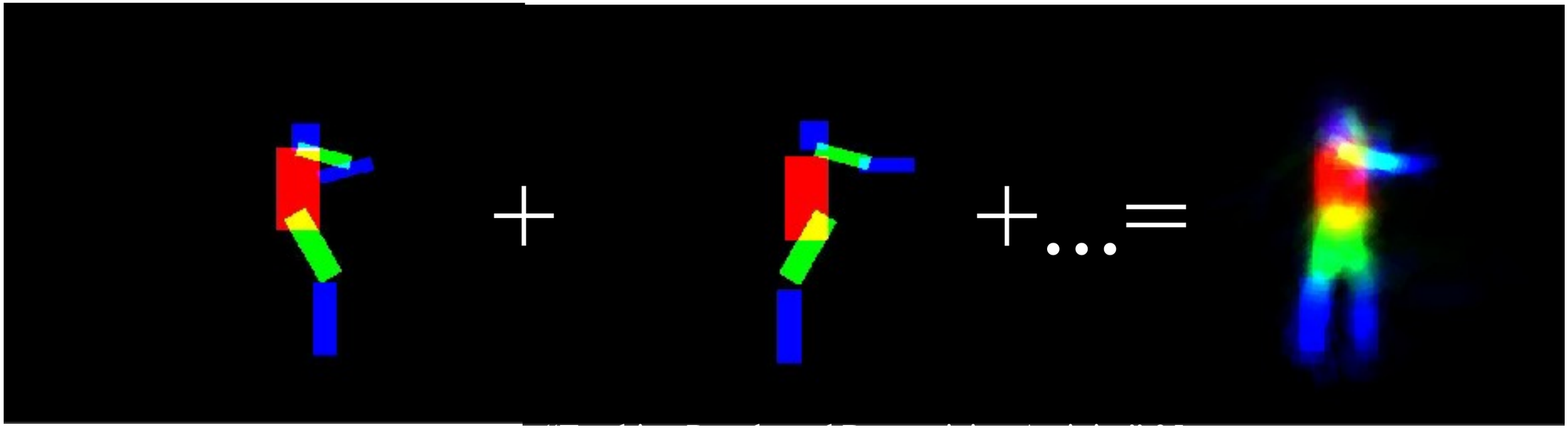
sample from
$\Pr(y_{tor}, y_{arm}, \ldots | x)$

Superimpose samples

Ramanan et al 07

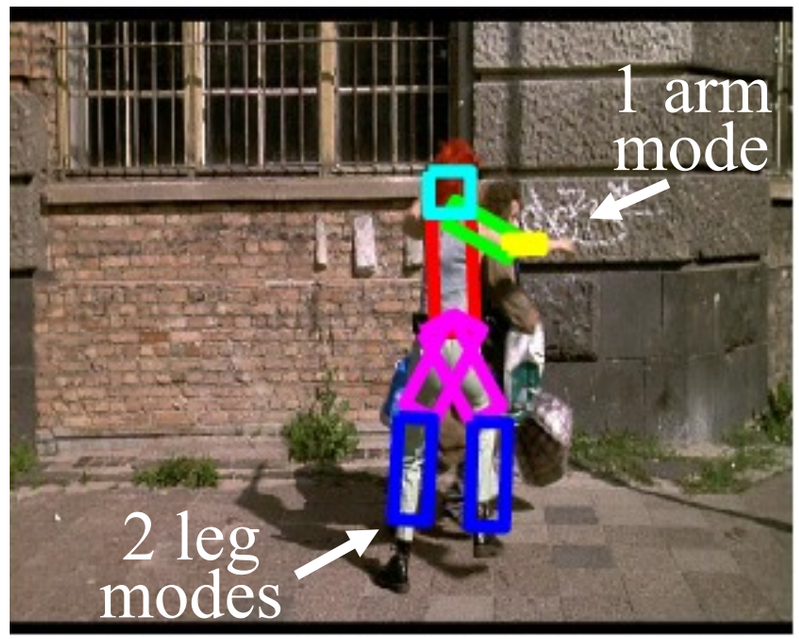


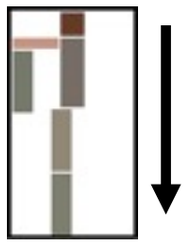
sample from
$\Pr(y_{tor}, y_{arm}, \ldots | x)$



Superimpose samples

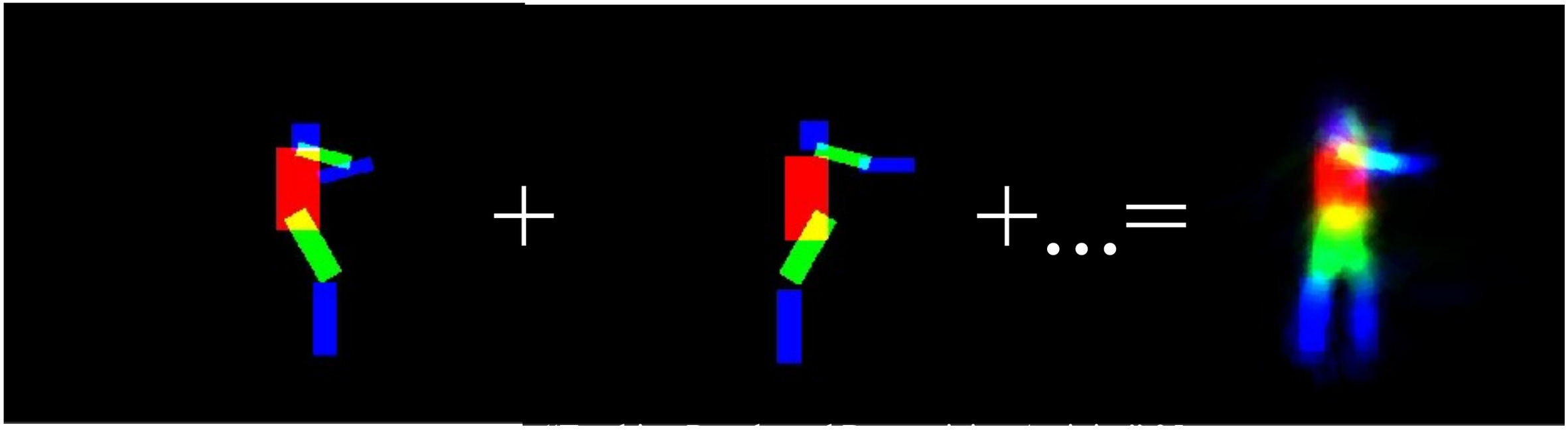
Ramanan et al 07





sample from
$\Pr(y_{tor}, y_{arm}, \ldots | x)$

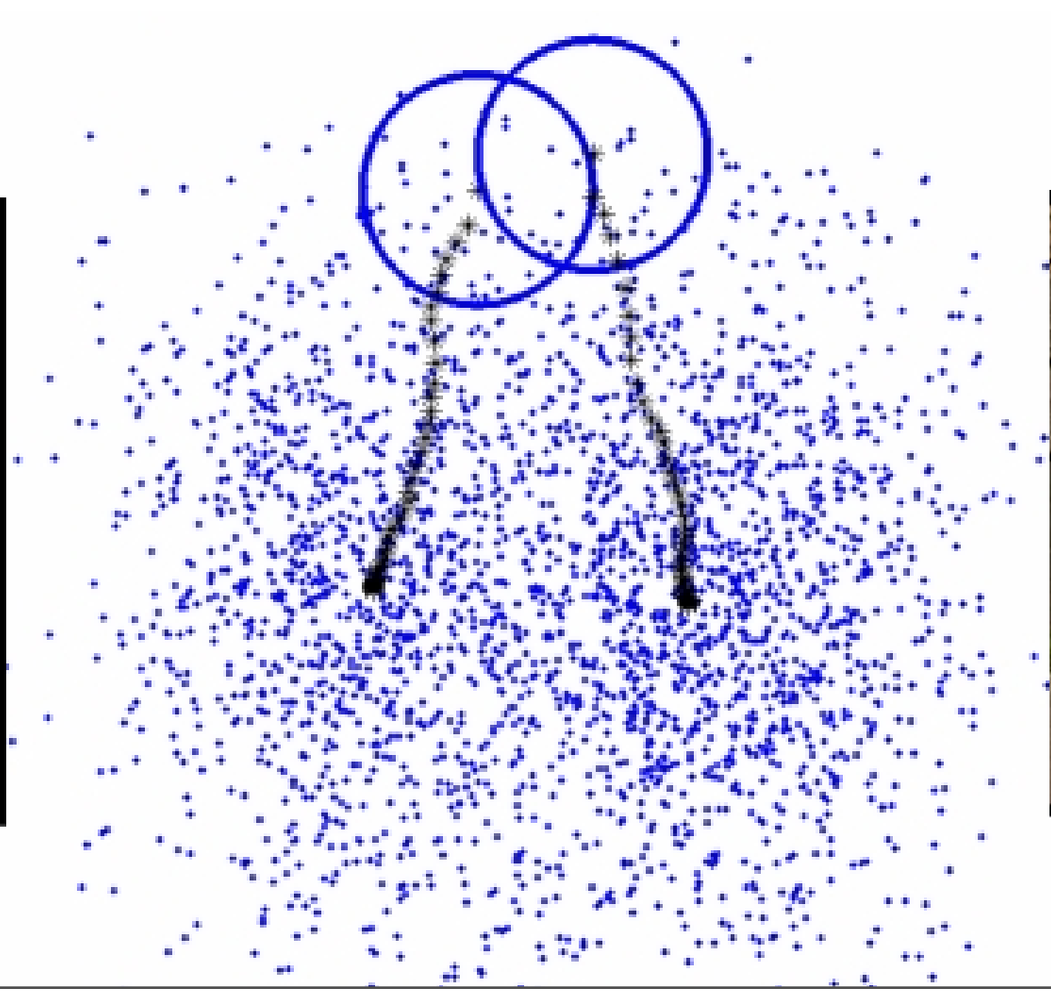
find modes

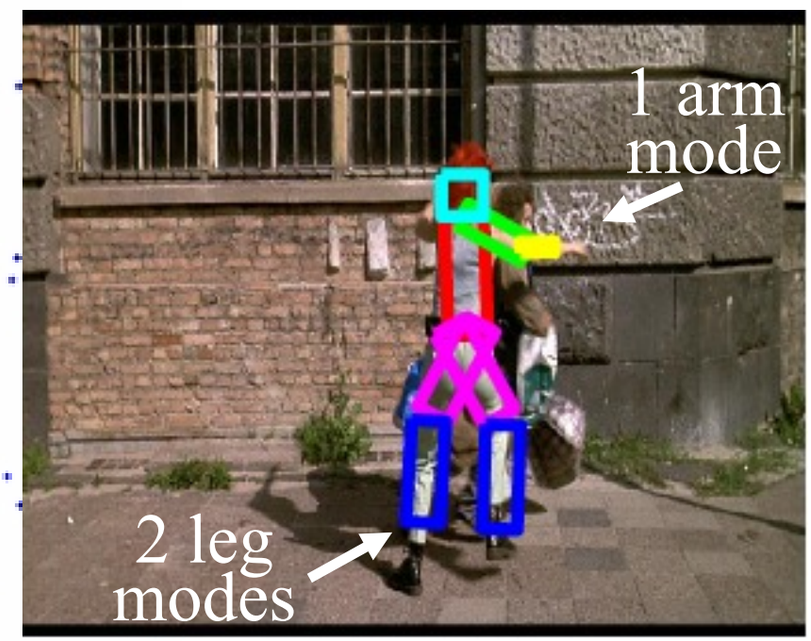


Find modes in samples

Meanshift modefinding

1. Start at random sample
2. Find all “closeby” samples within a distance of ‘b’
3. Move to center of closeby samples and repeat (2)



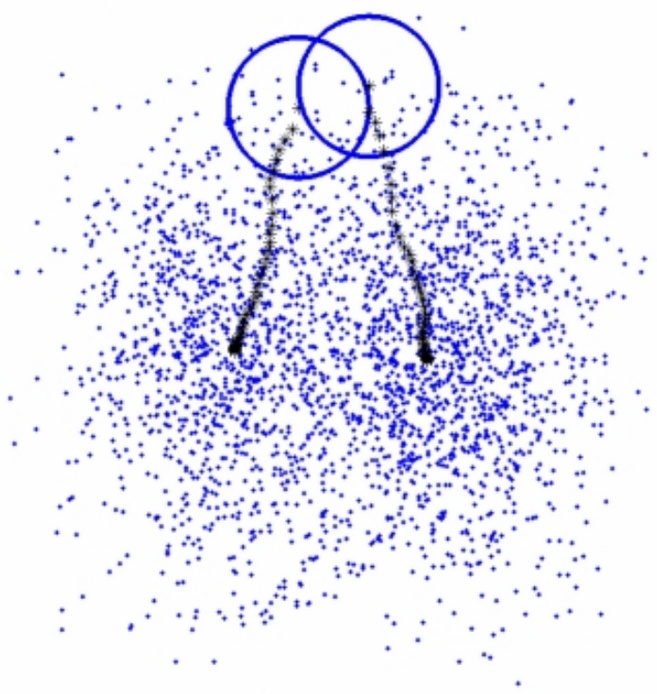


Closed-form mode-finding

$$P(y|x) = P(y_1|x) \prod_{ij \in E} P(y_i|y_j, x)$$

With limit of infinite samples, its straight forward to show mean computation is an “truncated” expectation (which can be computed exactly with marginal posterior for certain kernels)

$$y_{new} = \int_{y \in Near(y_{old})} P(y|x)dy$$



(Never tried, but seems quite slow)

Directly work with posterior

$$P(y|x) = P(y_1|x) \prod_{ij \in E} P(y_i|y_j, x)$$

Pass conditional tables to higher-level system (?)

Linear-parameterized MRFs

Assume scoring function is linearly parameterized

$$S_w(y) = \sum_{i \in V} w_i \cdot \phi_i(y_i) + \sum_{ij \in E} w_{ij} \cdot \psi_{ij}(y_i, y_j)$$

$$S_w(y) = w \cdot \Phi(y)$$

MAP: easy for some functions (e.g., graphcuts) $\max_y S_w(y)$

Sampling: can be hard $y \sim \frac{1}{Z} e^{S_w(y)}$

Perturb and MAP

Papandreou and Yuille

$$S_w(y) = w \cdot \Phi(y)$$

Approach:

1) Randomly perturb weights $\tilde{w} = w^* + \epsilon, \quad \epsilon \sim p(\epsilon)$

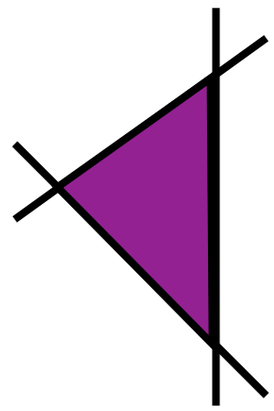
2) Solve perturbed model $\max_y S_{\tilde{w}}(y)$

But we are not sampling from P(y)

well, what are we sampling from?

Let's define the set of weights who's MAP solution is a particular state "y"

$$C_y = \{\tilde{w} : \tilde{w} \cdot (\Phi(y) - \Phi(y')) \geq 0, \forall y'\}$$



Convex set in space of $\tilde{w}$

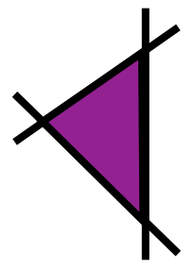
A state "y" will be generated if sampled weights $\tilde{w}$ fall inside C_y

But we are not sampling from P(y)

well, what are we sampling from?

Let's define the set of weights who's MAP solution is a particular state "y"

$$C_y = \{\tilde{w} : \tilde{w} \cdot (\Phi(y) - \Phi(y')) \geq 0, \forall y'\}$$



Convex set in space of $\tilde{w}$

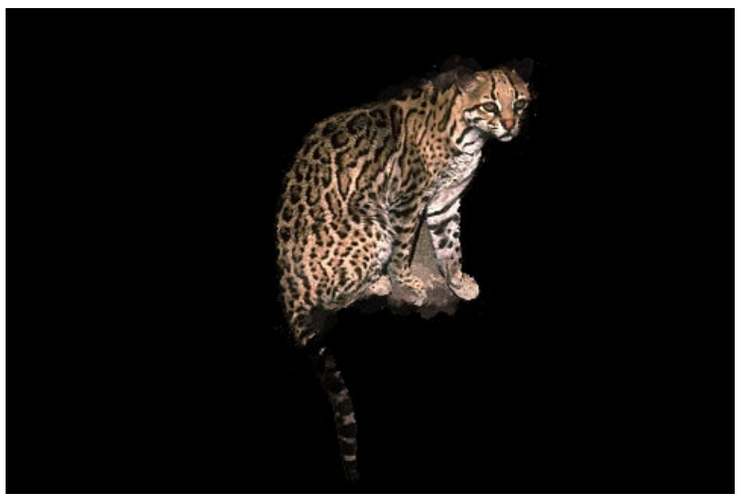
A state "y" will be generated if sampled weights $\tilde{w}$ fall inside C_y

We are sampling from $P(y) \propto \int_{\tilde{w} \in C_y} P(\tilde{w}) d\tilde{w}$

$P(\tilde{w})$

Interactive segmentation

Papandreou and Yuille



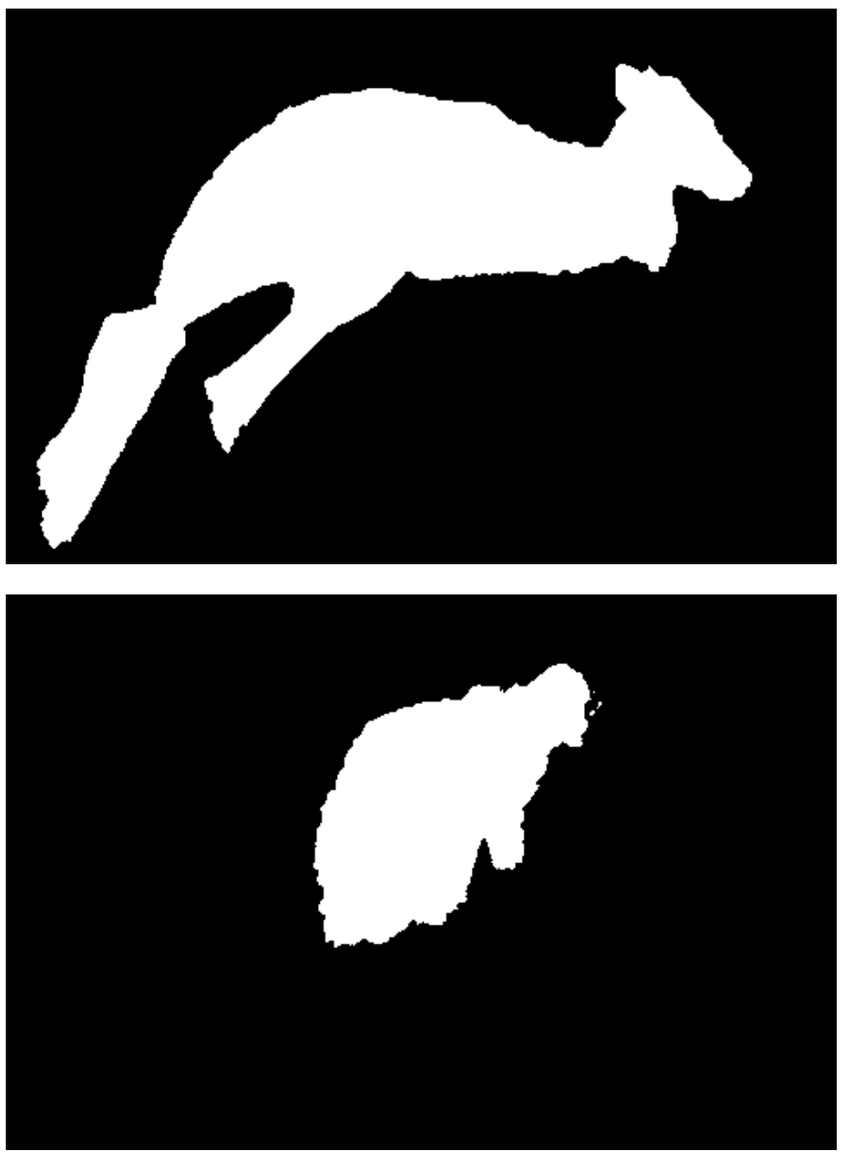
Image

GrabCut

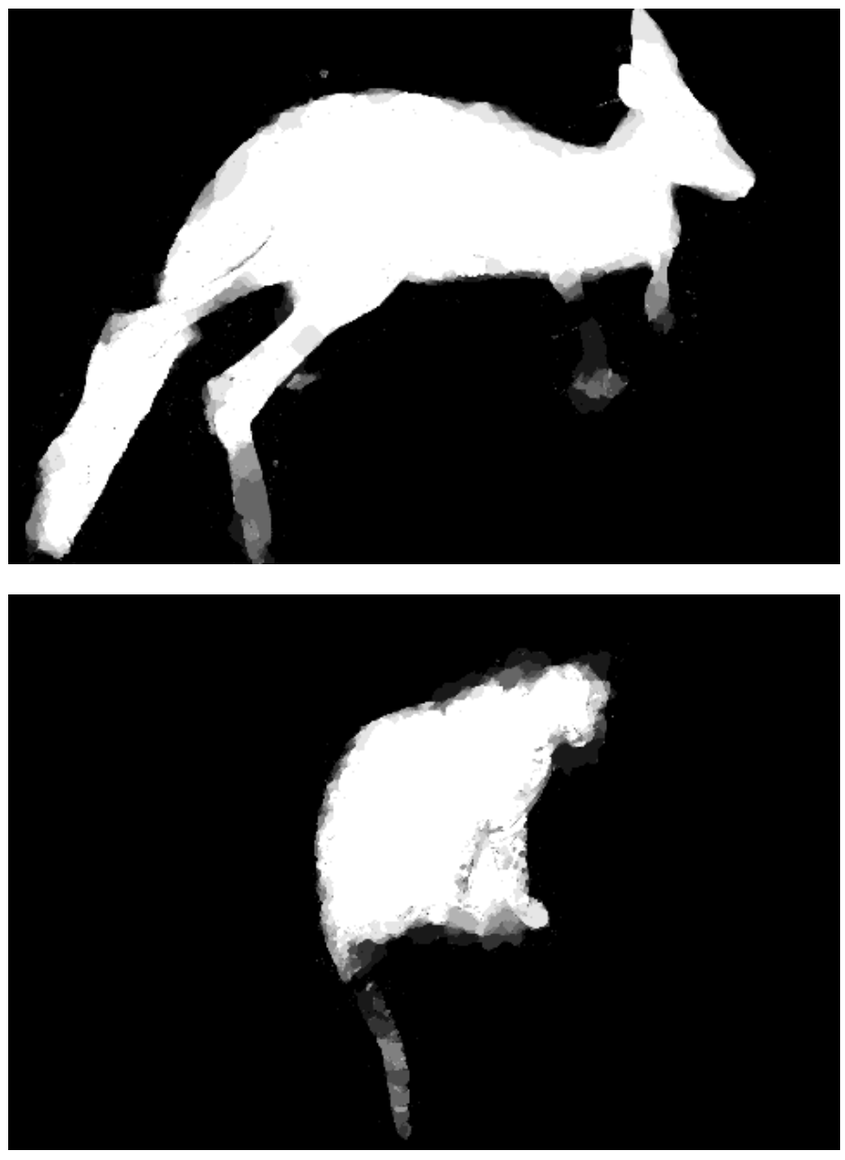
Perturb-and-MAP

(A particular sample)

Interactive segmentation



GrabCut mask



Perturb-and-MAP mask

A look back

0. Bayesian perspective: represent entire posterior rather than point estimates
   -Can compute in some situations (trees)

1. Use samples as a ad-hoc method of generating multiple solutions
   -Control samples by pre-processing (temperature) or post-processing (mode-finding)

2. Exploit rich literature on sampling
   -MCMC
   -“No-mixing” required on trees
   - Sample with MAP