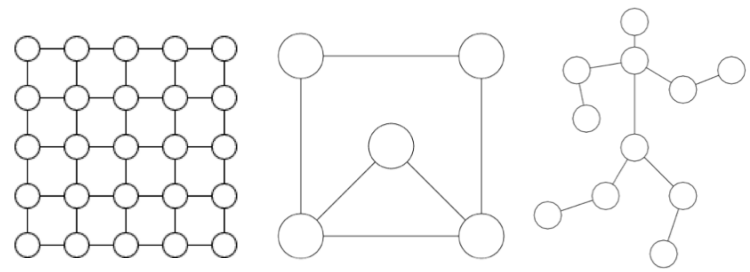
Learning and Inference in Probabilistic Graphical Models

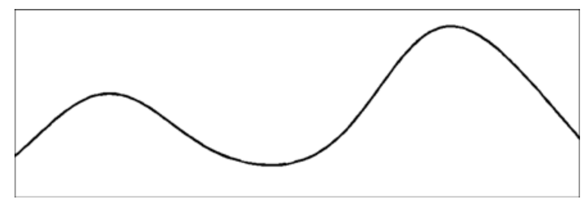
Nonparametric Belief Propagation
April 26, 2010

Introduction

Nonparametric Belief Propagation (NBP)

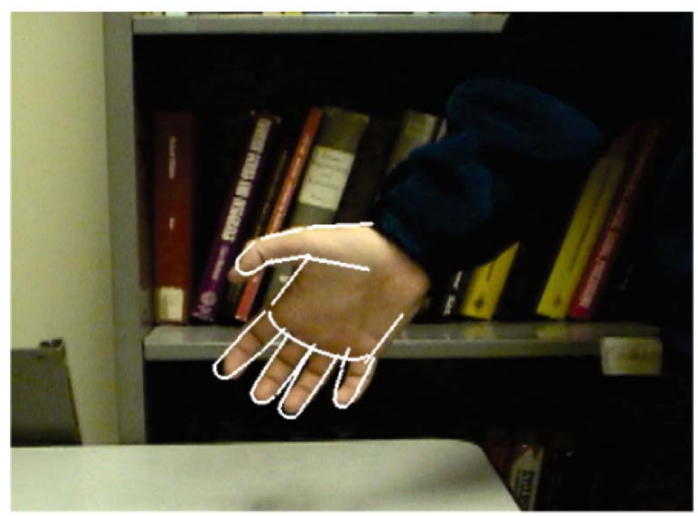


- **GOAL:** Inference for graphical models with variables which are
  - Continuous
  - High-dimensional
  - Non-Gaussian



- Efficiently extends particle filtering methods to general graphs

NBP for Visual Tracking



- Graphical formulation of hand structure, kinematics, & dynamics
- NBP tracker which accounts for finger self-occlusions

Outline

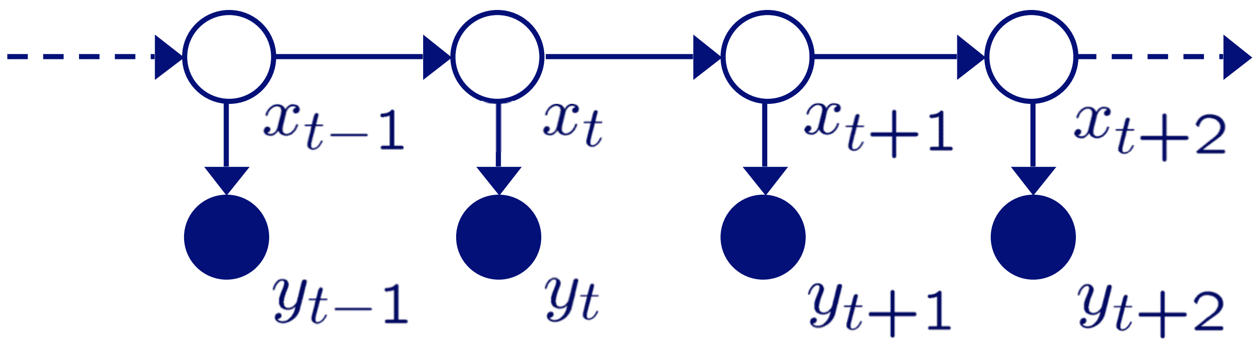
Nonparametric Belief Propagation

- Graphical models and belief propagation
- Nonparametric message propagation
- Efficient multiscale sampling from mixture products

Visual Hand Tracking

- Prior constraints & image likelihoods
- NBP for occlusion-compensated hand tracking
- Temporal constraints & tracking results

Hidden Markov Models



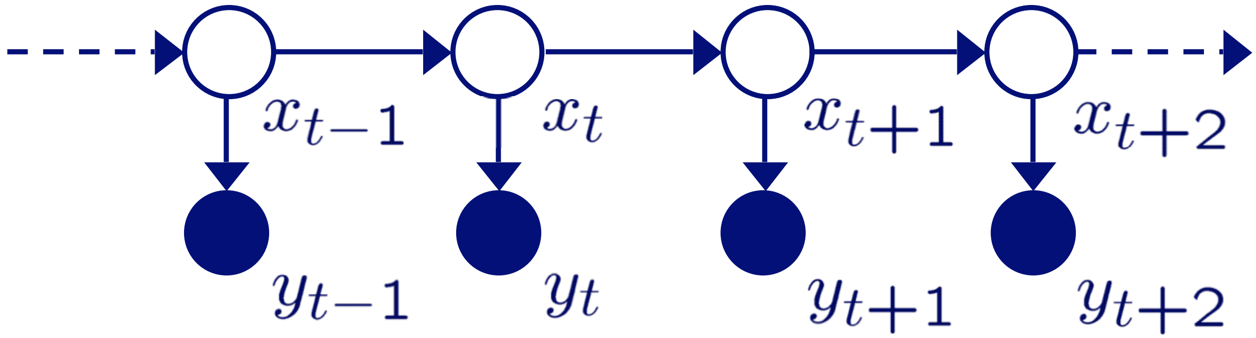
x_t → state variable at time t (unobserved or hidden)

y_t → local observation at time t

$$p(x,y) = p(x_0) \prod_{t=1}^{T} p(x_t \mid x_{t-1}) p(y_t \mid x_t)$$

"Conditioned on the present, the past and future are statistically independent"

Probabilistic Inference



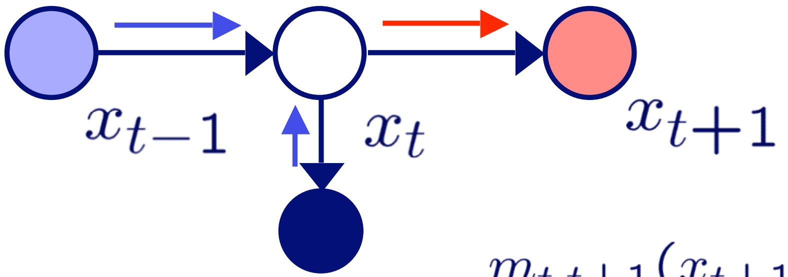
GOAL: Determine the conditional *marginal* distributions

$$p(x_t \mid y) = \alpha \int_{x_{\mathcal{V}\setminus t}} p(x, y)\, dx_{\mathcal{V}\setminus t}$$

- Provides many different estimates:
  - Bayes' least squares
  - Maximizer of Posterior Marginals (MPM)
- Degree of confidence in those estimates

Belief Propagation for HMMs

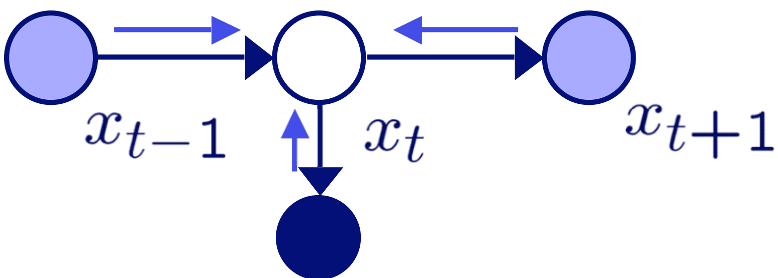
MESSAGES: Sufficient statistics of observations



$$m_{t,t+1}(x_{t+1}) = \int_{x_t} \underbrace{p(x_{t+1} \mid x_t)}_{\text{Message Propagation}} \underbrace{p(y_t \mid x_t) m_{t-1,t}(x_t)}_{\text{Message Product}} \, dx_t$$

Message Propagation Message Product

BELIEFS: Posterior distributions over state variables



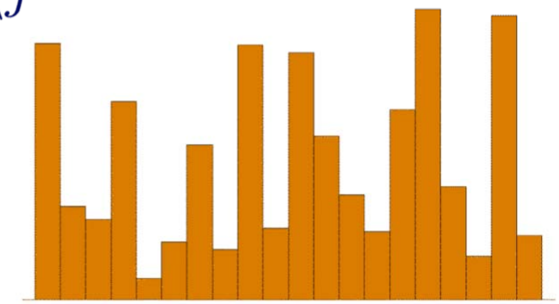
$$p(x_t|y) \propto p(y_t|x_t) m_{t-1,t}(x_t) m_{t+1,t}(x_t)$$

Present Past Future

Message Representations

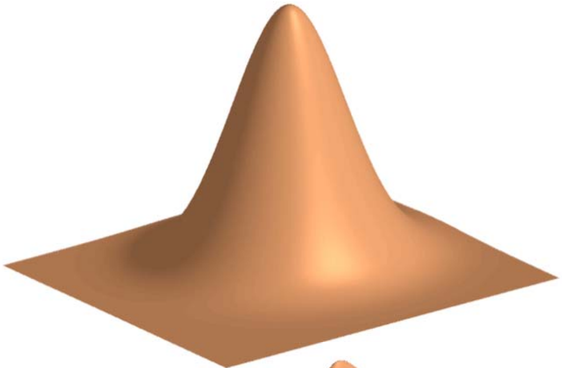
$$m_{ij}(x_j) \propto \int_{x_i} \psi_{j,i}(x_j, x_i)\psi_i(x_i, y) \prod_{k \in \Gamma(i)\setminus j} m_{ki}(x_i)\, dx_i$$

Discrete State Variables



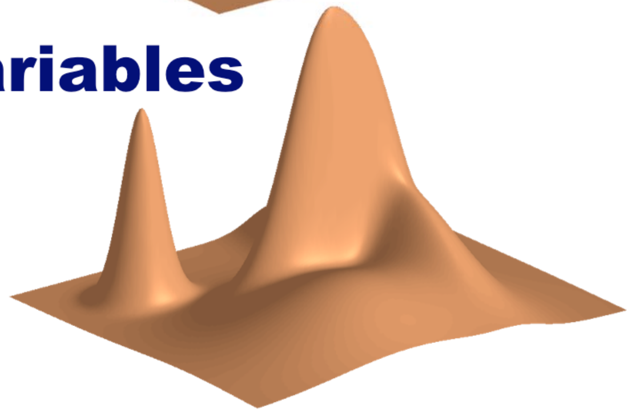
- Messages are *finite vectors*
- Updated via matrix-vector products

Gaussian State Variables



- Messages are *mean & covariance*
- Updated via information Kalman filter

Continuous Non-Gaussian State Variables

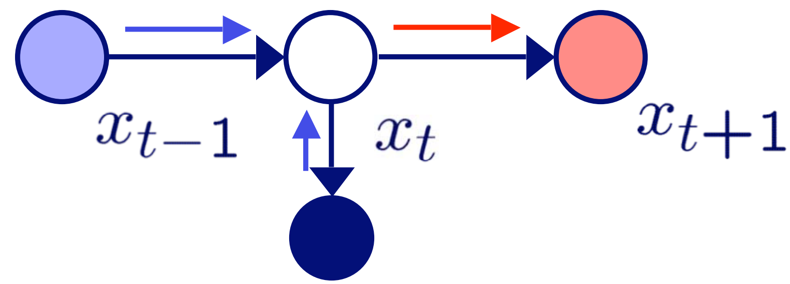


- Closed parametric forms unavailable
- Discretization can be *intractable* even with 2 or 3 dimensional states

Particle Filters

Condensation, Sequential Monte Carlo, Survival of the Fittest,…

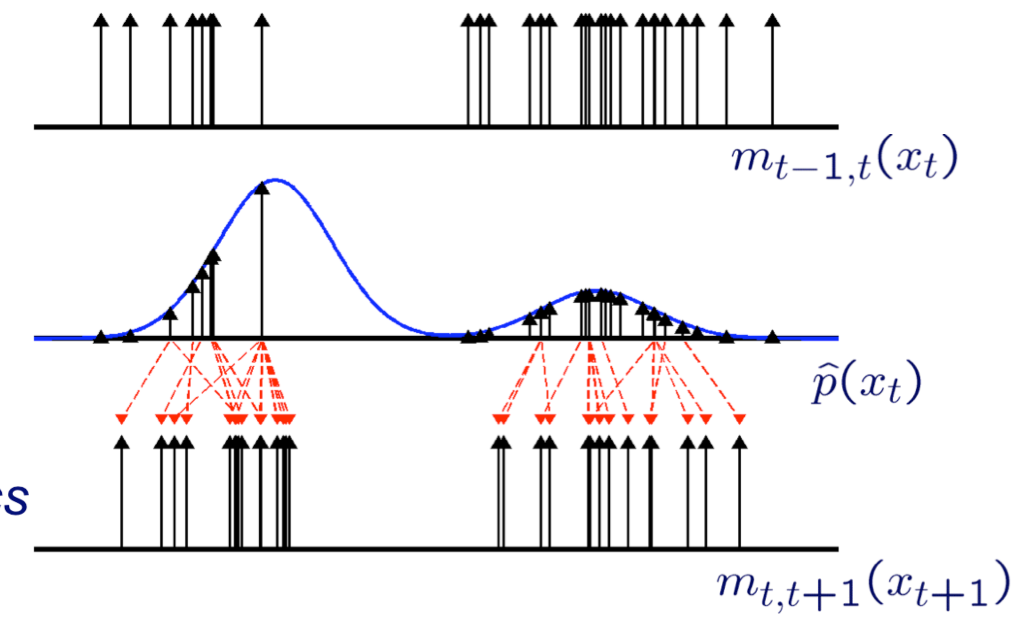
- Nonparametric approximation to optimal BP estimates
- Represent messages and posteriors using a set of samples, found by simulation



Sample-based density estimate

Weight by observation likelihood

Resample & propagate by dynamics



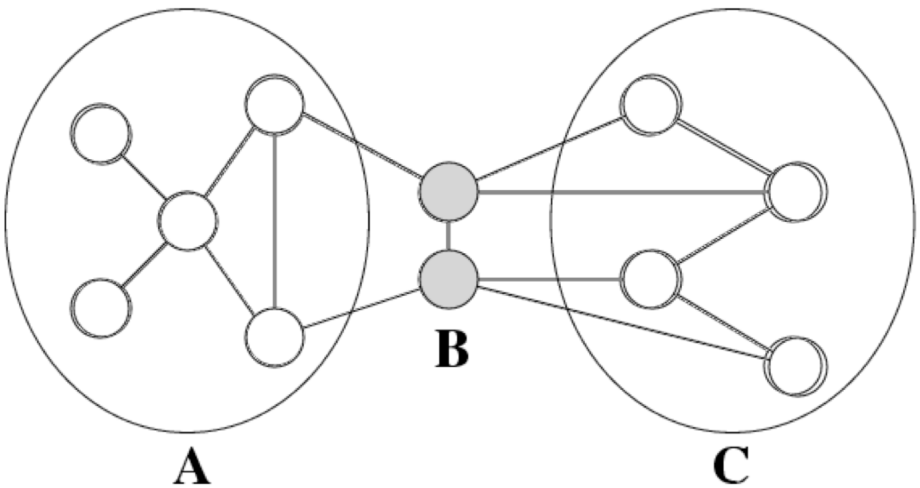
Graphical Models

An undirected graph $\mathcal{G}$ is defined by

$\mathcal{V}$ $\longrightarrow$ set of N nodes $\{1, 2, \ldots, N\}$

$\mathcal{E}$ $\longrightarrow$ set of edges (i, j) connecting nodes $i, j \in \mathcal{V}$

Nodes $i \in \mathcal{V}$ are associated with random variables x_i



Graph Separation

$\Updownarrow$

Conditional Independence

$$p(x_A, x_C | x_B) = p(x_A | x_B) p(x_C | x_B)$$

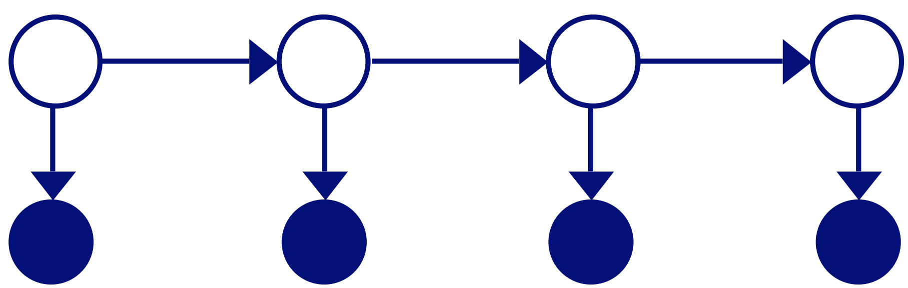
Pairwise Markov Random Fields

$$p(x,y) = \frac{1}{Z} \prod_{(i,j)\in\mathcal{E}} \psi_{i,j}(x_i, x_j) \prod_{i\in\mathcal{V}} \psi_i(x_i, y)$$

- Product of arbitrary positive *clique potential* functions
- Guaranteed Markov with respect to corresponding graph

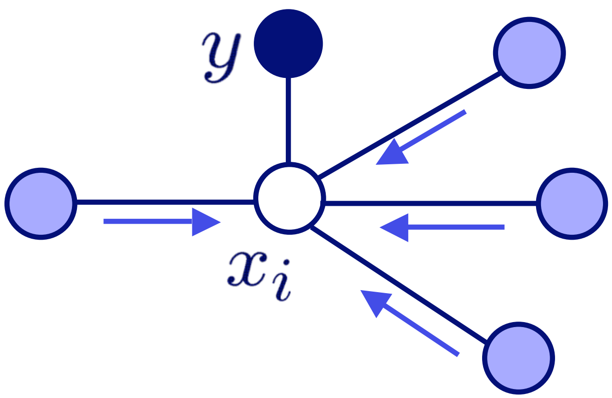
Special Case: Temporal HMM

$$p(x,y) = p(x_0) \prod_{t=1}^{T} p(x_t \mid x_{t-1}) p(y_t \mid x_t)$$



Belief Propagation

BELIEFS: Approximate posterior marginal distributions



$$\hat{p}(x_i \mid y) = \alpha \psi_i(x_i, y) \prod_{k \in \Gamma(i)} m_{ki}(x_i)$$

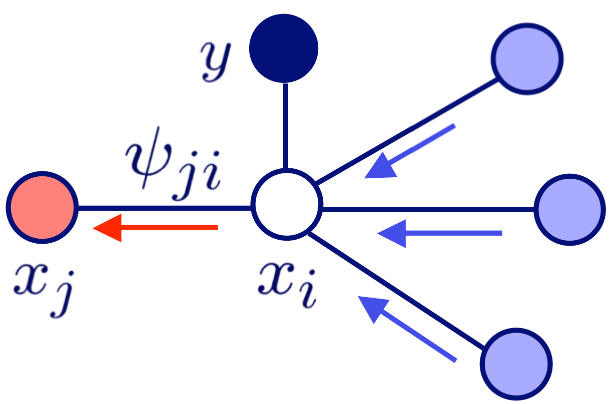
$\Gamma(i) \longrightarrow$ *neighborhood* of node *i* (adjacent nodes)

MESSAGES: Approximate sufficient statistics

$$m_{ij}(x_j) = \alpha \int_{x_i} \psi_{j,i}(x_j, x_i) \psi_i(x_i, y) \prod_{k \in \Gamma(i) \setminus j} m_{ki}(x_i)\, dx_i$$

$$= \alpha \int_{x_i} \psi_{j,i}(x_j, x_i) \frac{\hat{p}(x_i|y)}{m_{ji}(x_i)}\, dx_i$$

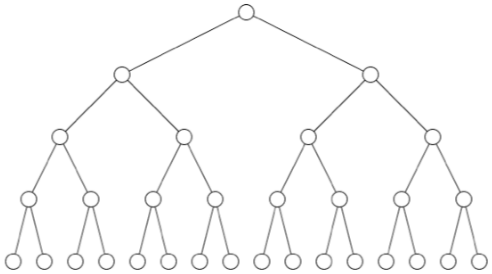
(Koller, *UAI 1999*)



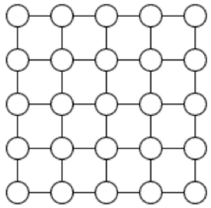
I. Belief Update (Message Product)

II. Message Propagation (Integral)

BP Justification



- Produces *exact* conditional marginals for discrete or Gaussian tree-structured graphs (variant of dynamic programming)



- For general graphs, exhibits excellent empirical performance in many applications (especially coding)

Statistical Physics & Free Energies *(Yedidia, Freeman, and Weiss)*

Variational interpretation, improved region-based approximations

BP as Reparameterization *(Wainwright, Jaakkola, and Willsky)*

Characterization of fixed points, error bounds

Many others…

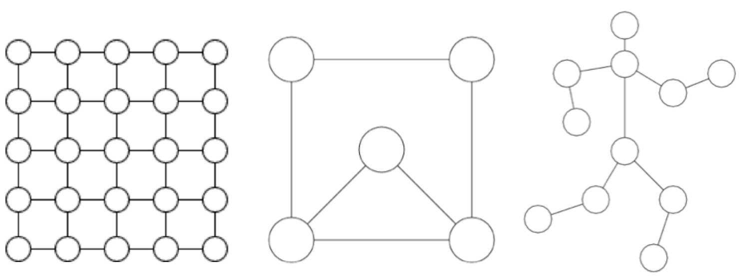
Nonparametric Inference for General Graphs

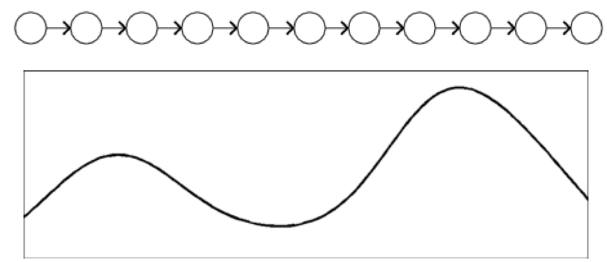
Belief Propagation

- General graphs
- Discrete or Gaussian

Particle Filters

- Markov chains
- General potentials





Nonparametric BP

- General graphs
- General potentials

Outline

Nonparametric Belief Propagation

- Graphical models and belief propagation
- Nonparametric message propagation
- Efficient multiscale sampling from mixture products

Visual Hand Tracking

- Prior constraints & image likelihoods
- NBP for occlusion-compensated hand tracking
- Temporal constraints & tracking results

Nonparametric Density Estimates

Kernel (Parzen Window) Density Estimator

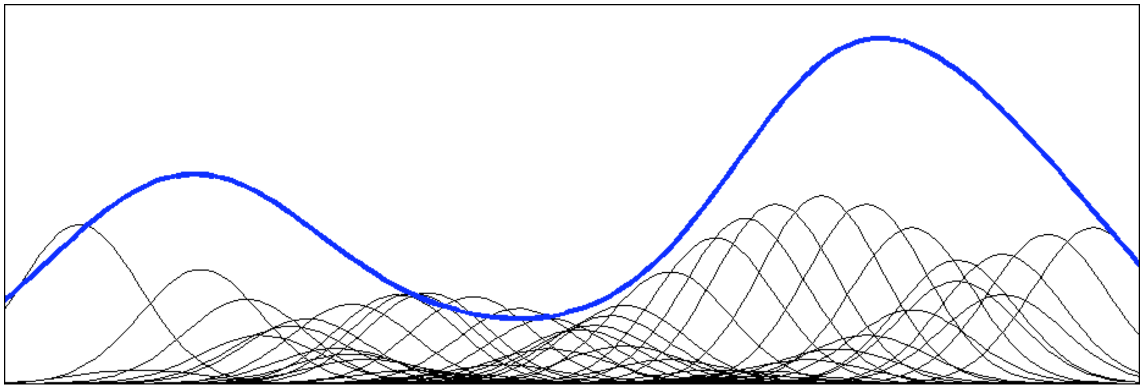
Approximate PDF by a set of smoothed data samples

$$\widehat{p}(x) = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{\sigma} K\left(\frac{x - X_i}{\sigma}\right)$$

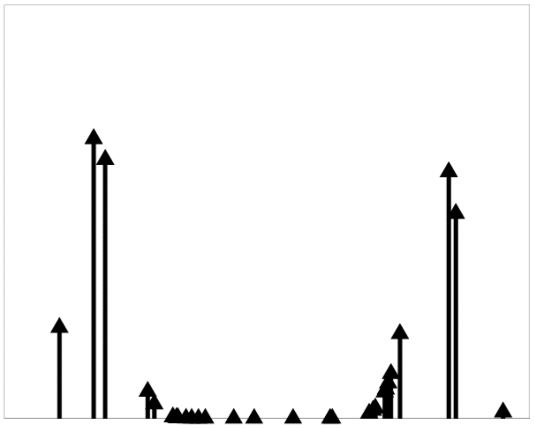
$\{X_i\}$ → *M* independent samples from *p(x)*

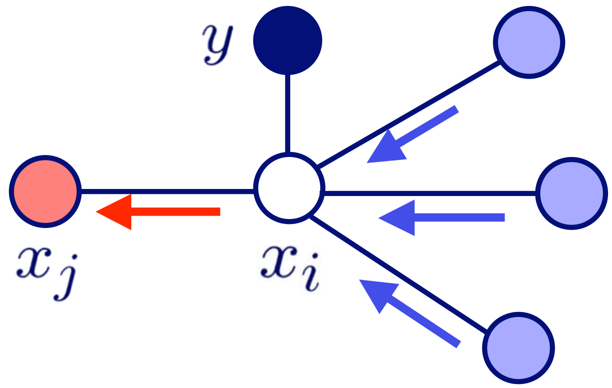
$K(\cdot)$ → *Gaussian* kernel function (self-reproducing)

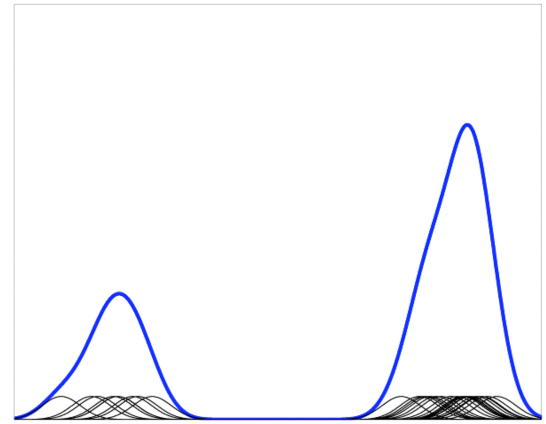
σ → Bandwidth (chosen automatically)



Nonparametric BP

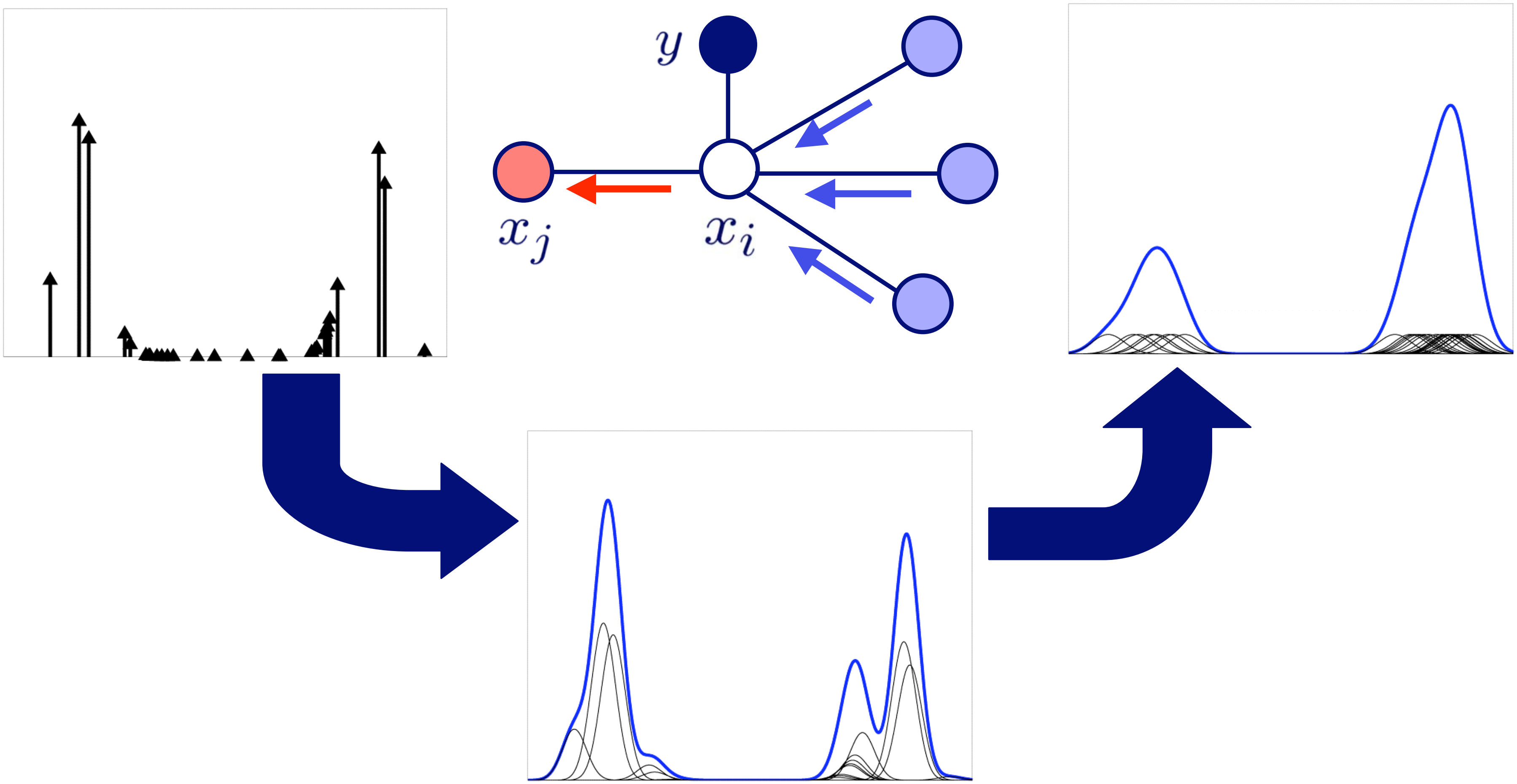






- Input messages are *kernel density estimates* (Gaussian)
- Message product: $$x_i^{(\ell)} \sim \psi_i(x_i, y) \prod_{k \in \Gamma(i) \setminus j} m_{ki}(x_i)$$
  Draw L samples
- Message propagation: $$x_j^{(\ell)} \sim \psi_{ji}(x_j, x_i^{(\ell)})$$
  Monte Carlo integration

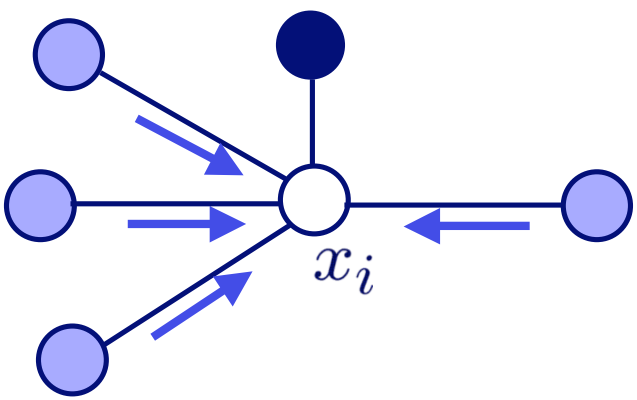
Nonparametric BP



- Output message estimated from weighted samples via a *bandwidth selection* rule

Extensive literature: asymptotic analysis, cross-validation, etc.

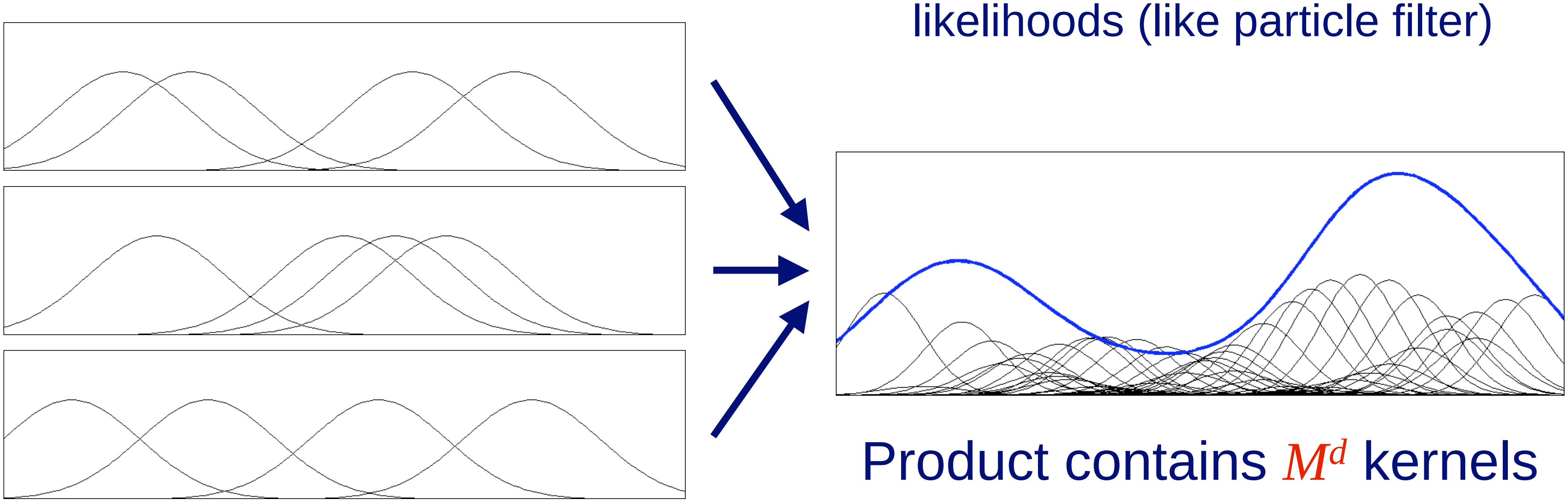
NBP Marginal Update



$$x_i^{(\ell)} \sim \psi_i(x_i, y) \prod_{k \in \Gamma(i)} m_{ki}(x_i)$$

Importance Sampling:

- Sample from product of all Gaussian mixture messages
- Reweight samples by likelihoods (like particle filter)



Product contains M^d kernels

d messages, M kernels each

Sampling from Mixture Products

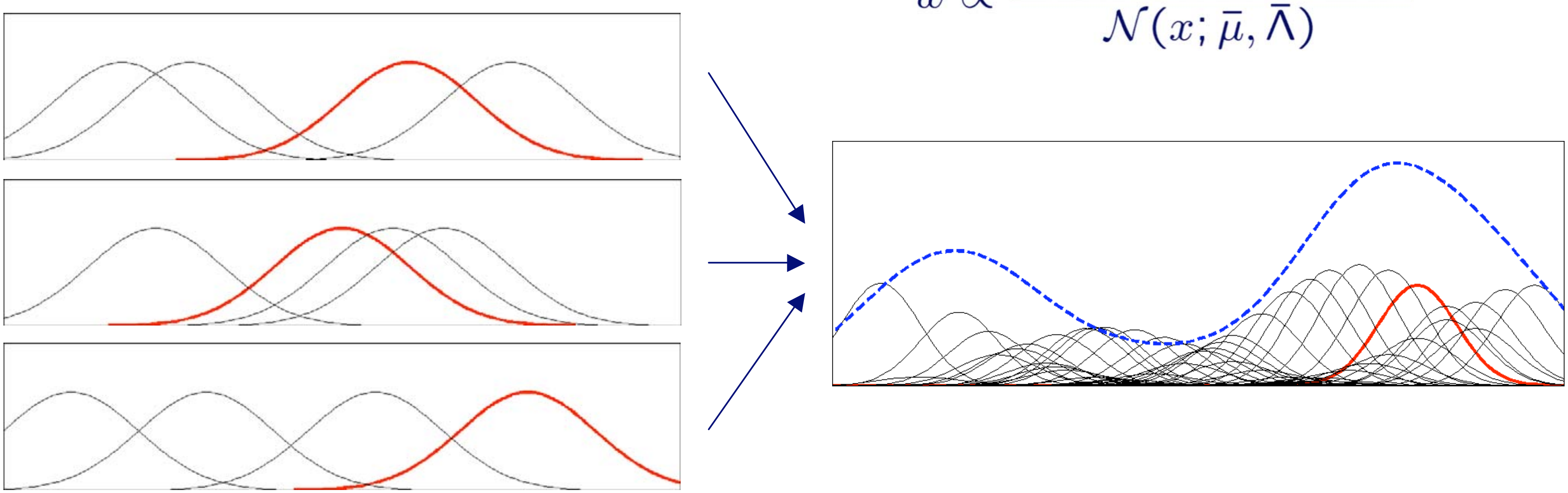
- Product density kernels generated by *combinations* of input density kernels
- Structure exploited by *Gibbs sampling* algorithms

- Products of Gaussians are also *weighted* Gaussians:

$$\prod_{i=1}^{d} \mathcal{N}(x; \mu_i, \Lambda_i) \propto \mathcal{N}(x; \bar{\mu}, \bar{\Lambda})$$

$$\bar{\Lambda}^{-1} = \sum_{i=1}^{d} \Lambda_i^{-1} \qquad \bar{\Lambda}^{-1}\bar{\mu} = \sum_{i=1}^{d} \Lambda_i^{-1}\mu_i$$

$$\bar{w} \propto \frac{\prod_{i=1}^{d} w_i \mathcal{N}(x; \mu_i, \Lambda_i)}{\mathcal{N}(x; \bar{\mu}, \bar{\Lambda})}$$



Product Density Sampling

d mixtures of M Gaussians → *mixture of M^d Gaussians*

$$p_i(x) = \sum_{l_i} w_{l_i} \mathcal{N}\left(x; \mu_{l_i}, \Lambda_i\right) \longrightarrow p(x) \propto \prod_{i=1}^{d} p_i(x)$$

- Exact sampling
- Importance sampling: *mixture vs. Gaussian*
- Gibbs sampling: *parallel vs. sequential*
- Multiscale sampling: *Gibbs vs. ε-exact*

Exact Sampling

l_i → mixture component label for *i^th^* *input* density

$L = [l_1, \ldots, l_d]$ → label of component in *product* density

$$w_L = \frac{\prod_{i=1}^{d} w_{l_i} \mathcal{N}(x; \mu_{l_i}, \Lambda_i)}{\mathcal{N}(x; \mu_L, \Lambda_L)} \qquad \Lambda_L^{-1} = \sum_{i=1}^{d} \Lambda_i^{-1} \qquad \Lambda_L^{-1} \mu_L = \sum_{i=1}^{d} \Lambda_i^{-1} \mu_{l_i}$$

- Calculate the weight partition function in *O(M^d^)* operations: $Z = \sum_L w_L$
- Draw and sort M uniform [0,1] variables
- Compute the cumulative distribution of

$$p(L) = \frac{w_L}{Z}$$

Importance Sampling

$p(x)$ ⟶ true distribution (difficult to sample from)
assume may be evaluated *up to normalization Z*

$q(x)$ ⟶ proposal distribution (easy to sample from)

- Draw N >> M samples from proposal distribution:

$$x_i \sim q(x) \qquad w_i \propto p(x_i)/q(x_i)$$

- Sample M times (with replacement) from

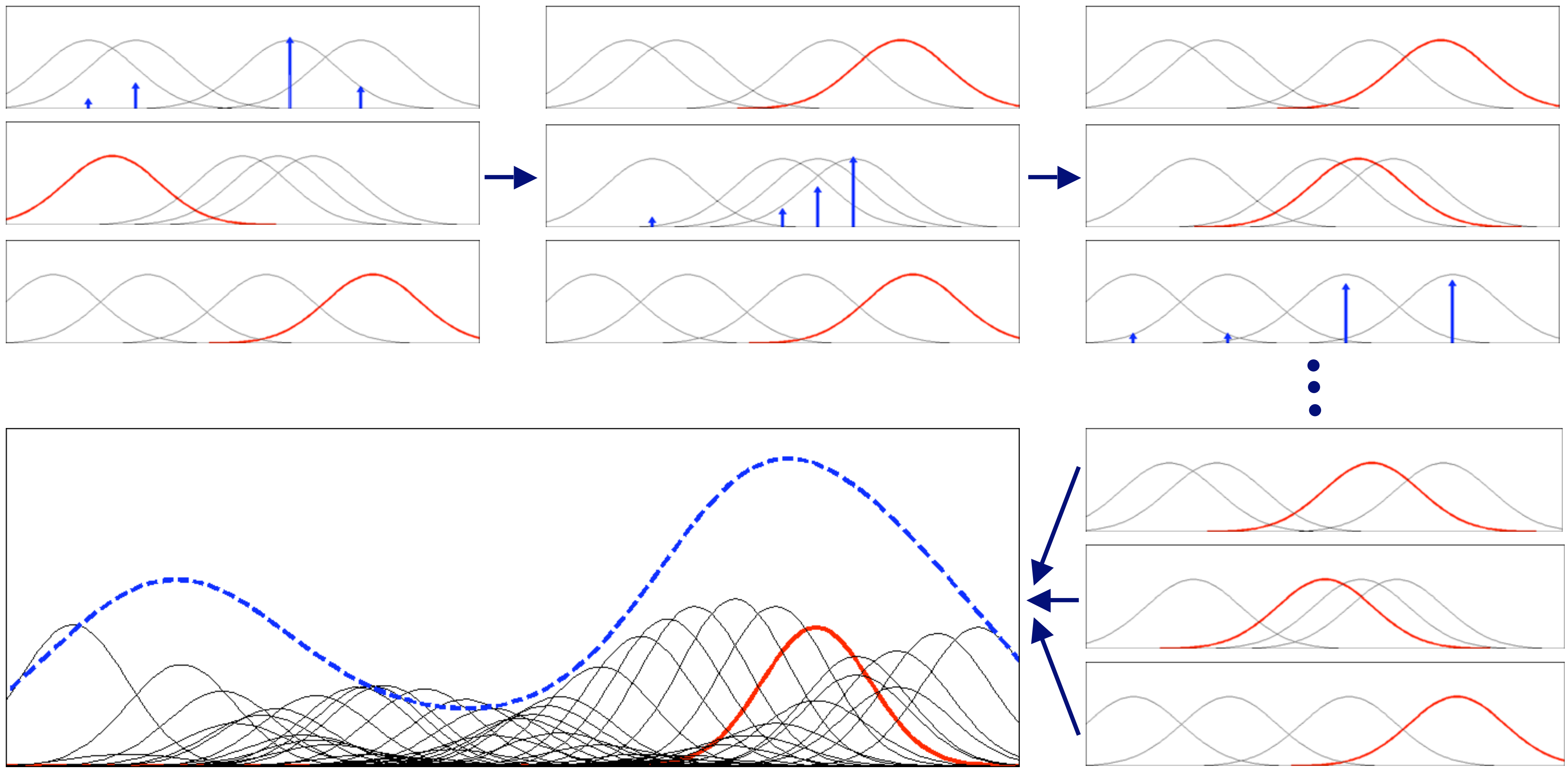
$$\bar{p}(x_i) = w_i/Z$$

Mixture IS: Randomly select a different mixture $p_i(x)$ for each sample (other mixtures provide weight)

Gaussian IS: Approximate each mixture by single Gaussian

Sequential Gibbs Sampler

Product of 3 messages, each containing 4 Gaussian kernels

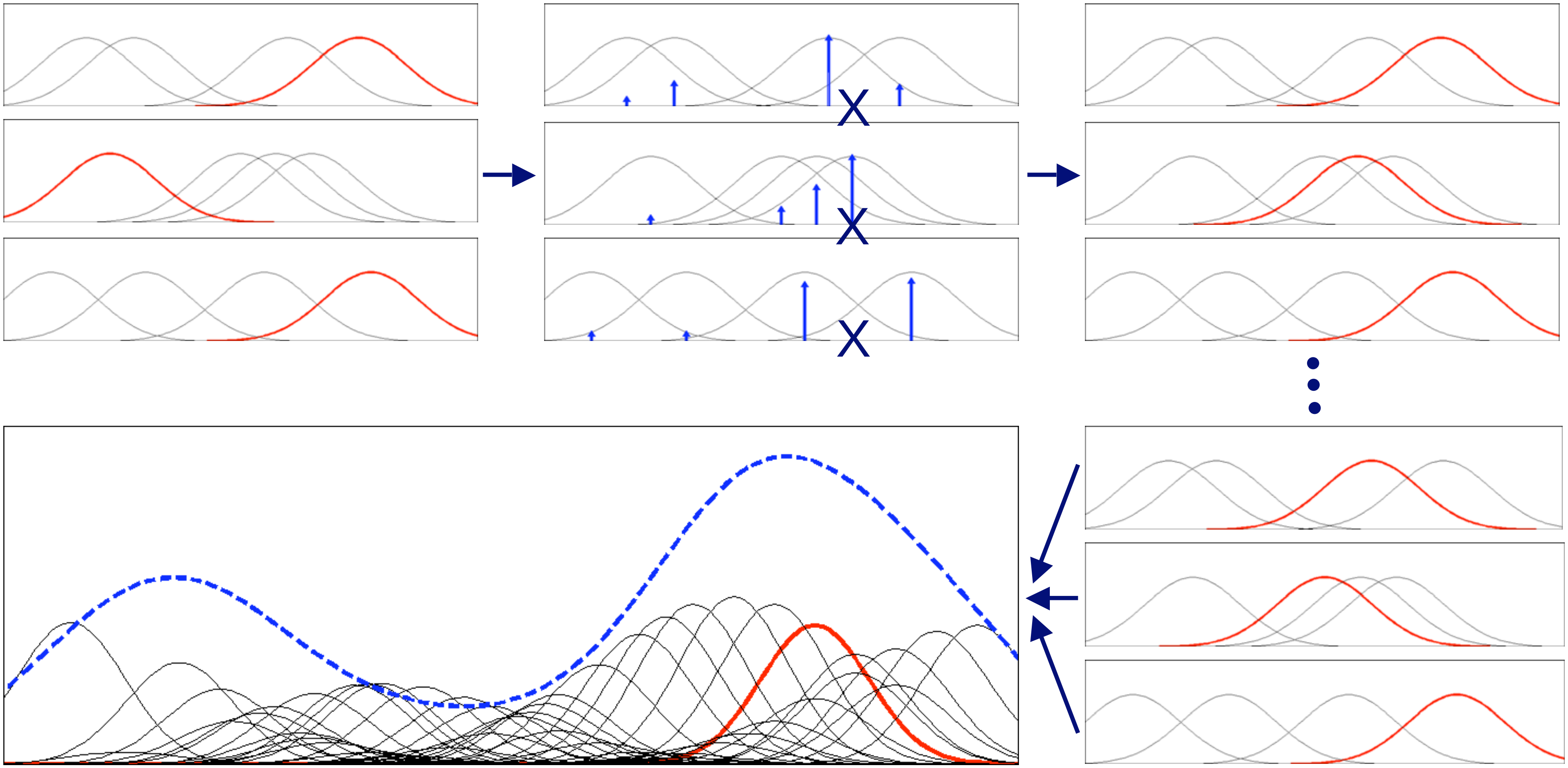


Labeled Kernels
Highlighted Red

Sampling Weights
Blue Arrows

Parallel Gibbs Sampler

Product of 3 messages, each containing 4 Gaussian kernels

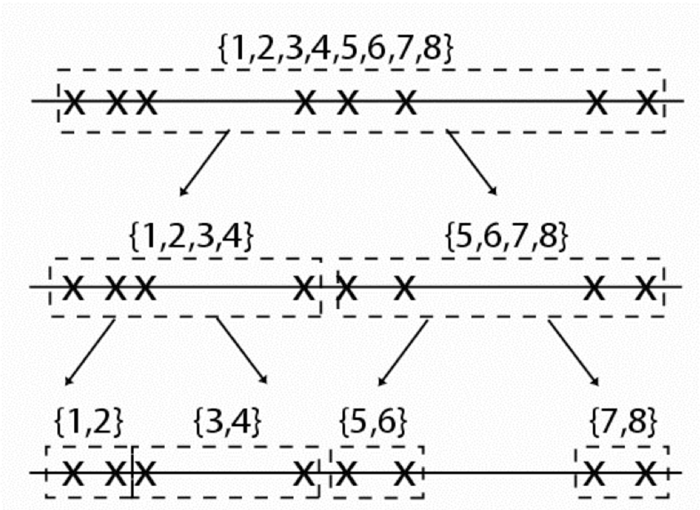


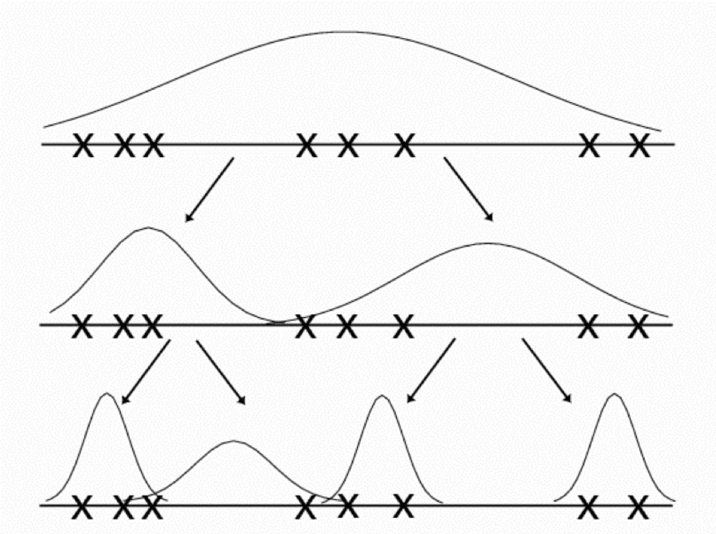
Labeled Kernels
Highlighted Red

Sampling Weights
Blue Arrows

Multiscale: KD-Trees

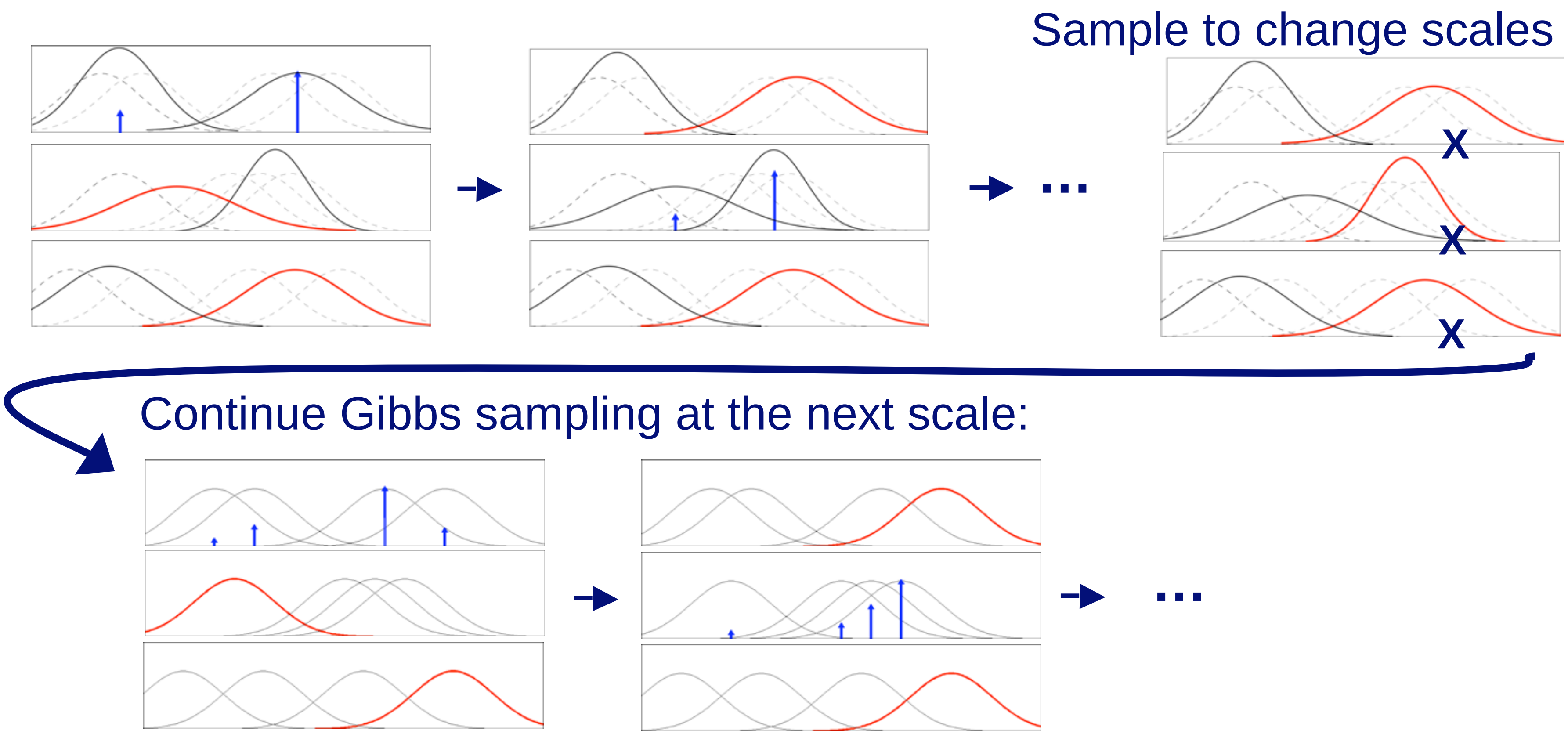
- "K-dimensional Trees"
- Multiscale representation of data set
- Cache *statistics of points* at each level:
  - Bounding boxes
  - Mean & covariance
- Original use: efficient search algorithms





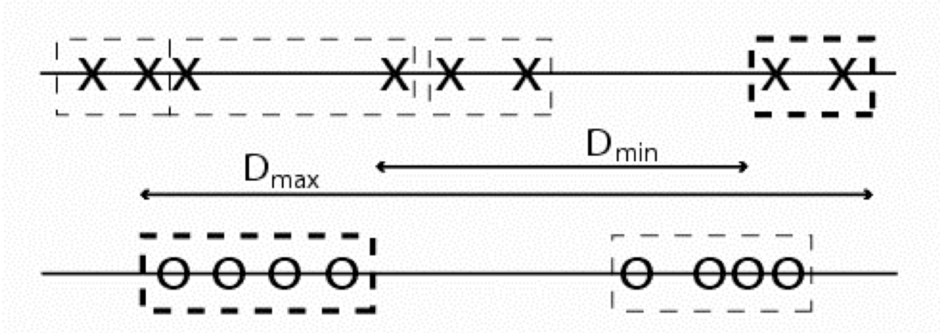
Multiscale Gibbs Sampling

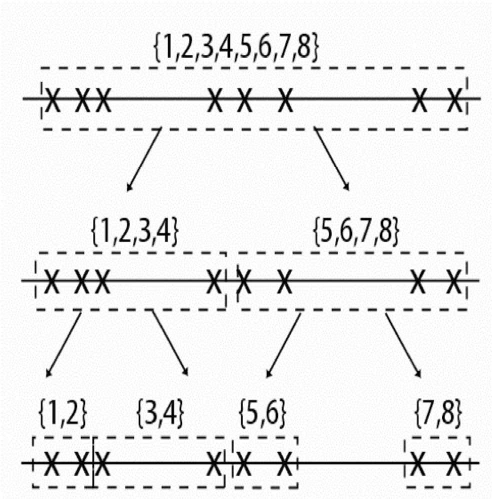
- Build KD-tree for each input density
- Perform Gibbs over progressively finer scales:



→ *Annealed Gibbs sampling (analogies in MRFs)*

ε-Exact Sampling

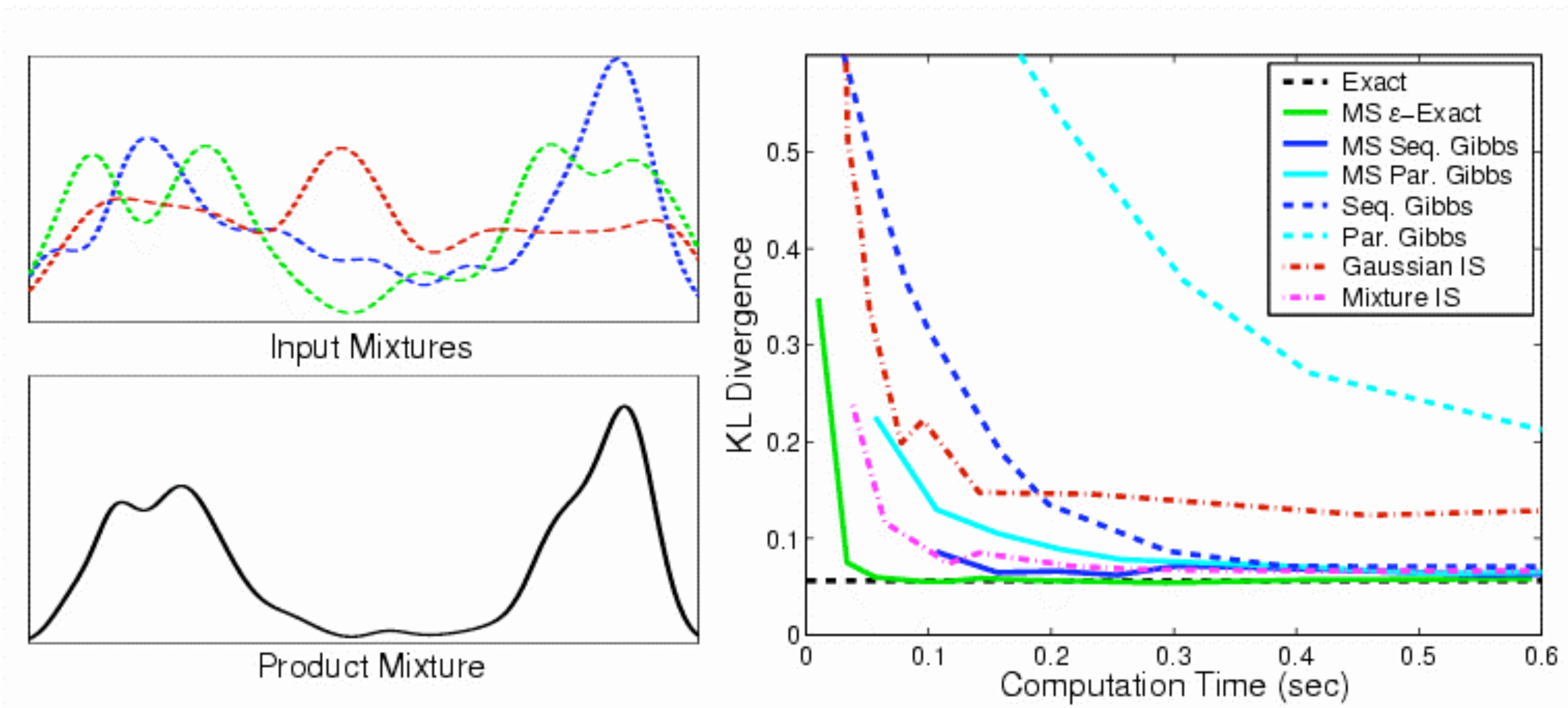




- Bounding boxes used for fast, approximate kernel density evaluation *(Gray & Moore, 2003)*:
  - Find sets with nearly constant weight
  - Provides evaluations within fractional error ε
- Similar method approximates partition function:
  - Express product mixture weights via density pairs
  - KD-tree recursions approximate sum, and then sample
- Tunable accuracy level: $|\widehat{p}_L - p_L| = \left| \frac{\widehat{w}_L}{\widehat{Z}} - \frac{w_L}{Z} \right| \leq \epsilon$

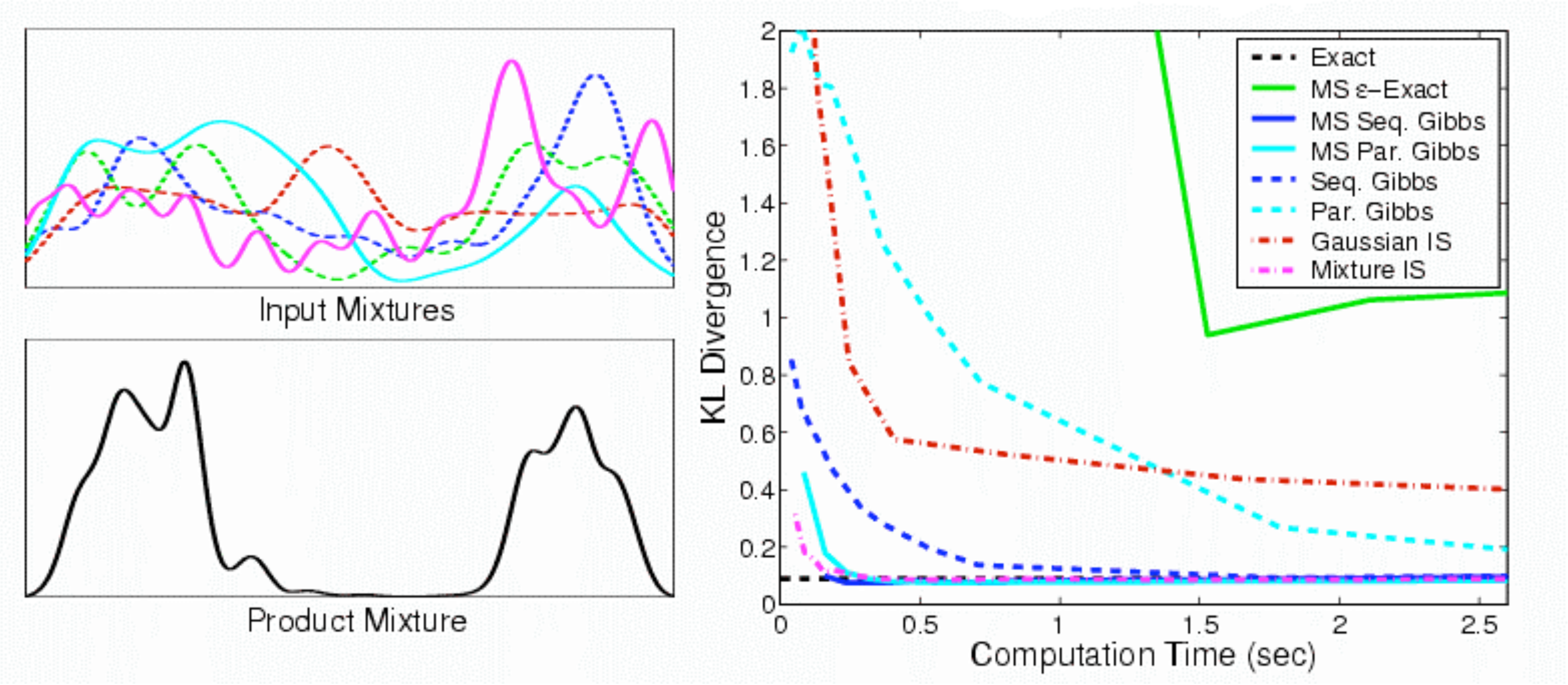
Taking Products: 3 Mixtures

- ***TASK:*** Draw 100 samples & construct density estimate
- All multiscale samplers perform well



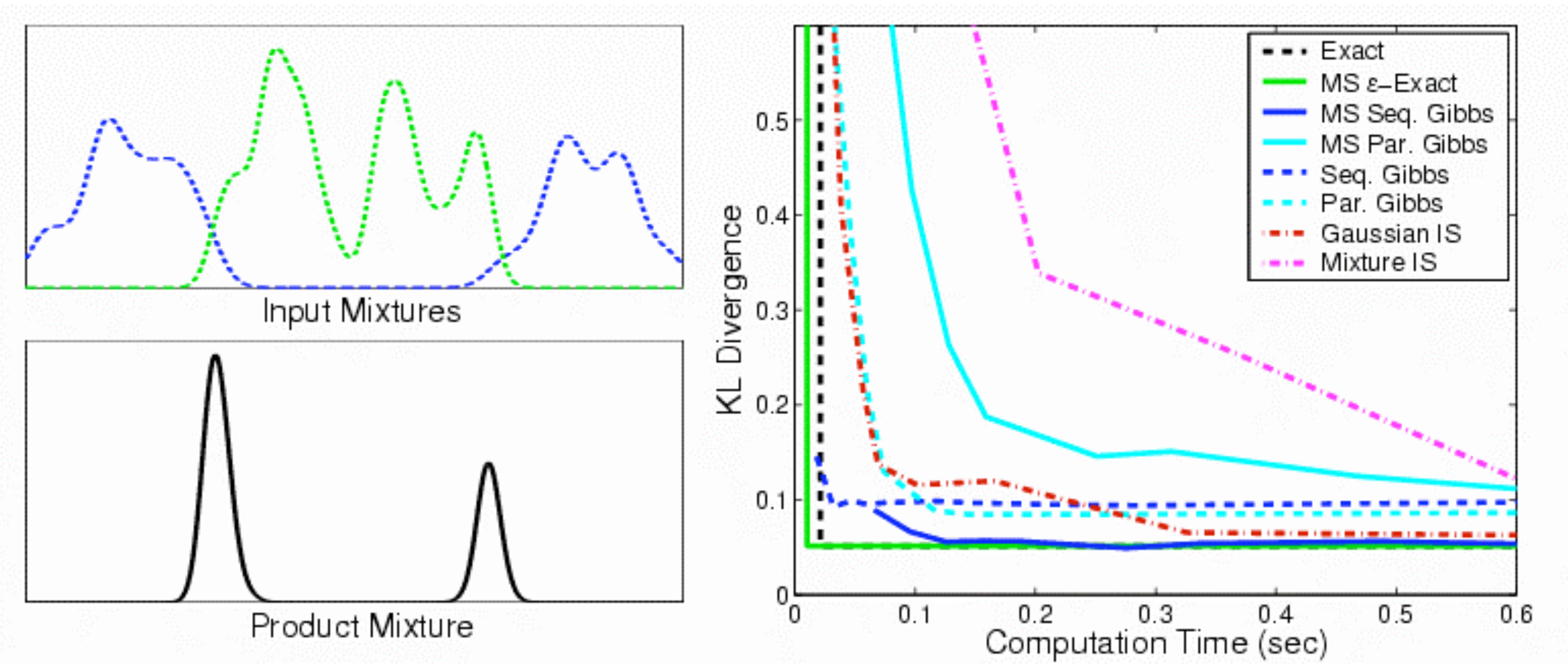
Taking Products: 5 Mixtures

- Exact sampling takes 7.6 hours
- Multiscale Gibbs performs comparably in 0.2 seconds



Taking Products: 2 Mixtures

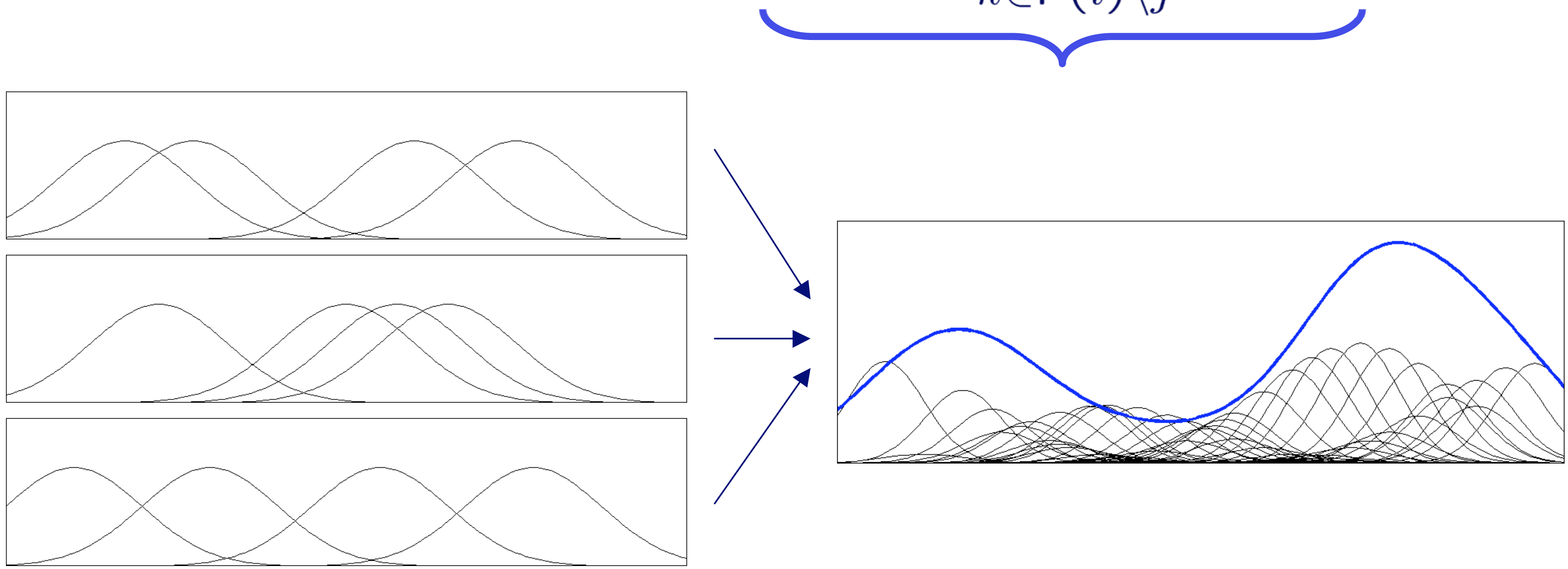
- Importance sampling sensitive to message alignment
- Multiscale methods show greater robustness



NBP Message Updates I:

Message Products

$$m_{ij}(x_j) = \alpha \int_{x_i} \psi_{j,i}(x_j, x_i)\psi_i(x_i, y) \prod_{k\in\Gamma(i)\setminus j} m_{ki}(x_i)\, dx_i$$



d messages

M kernels each

Product contains

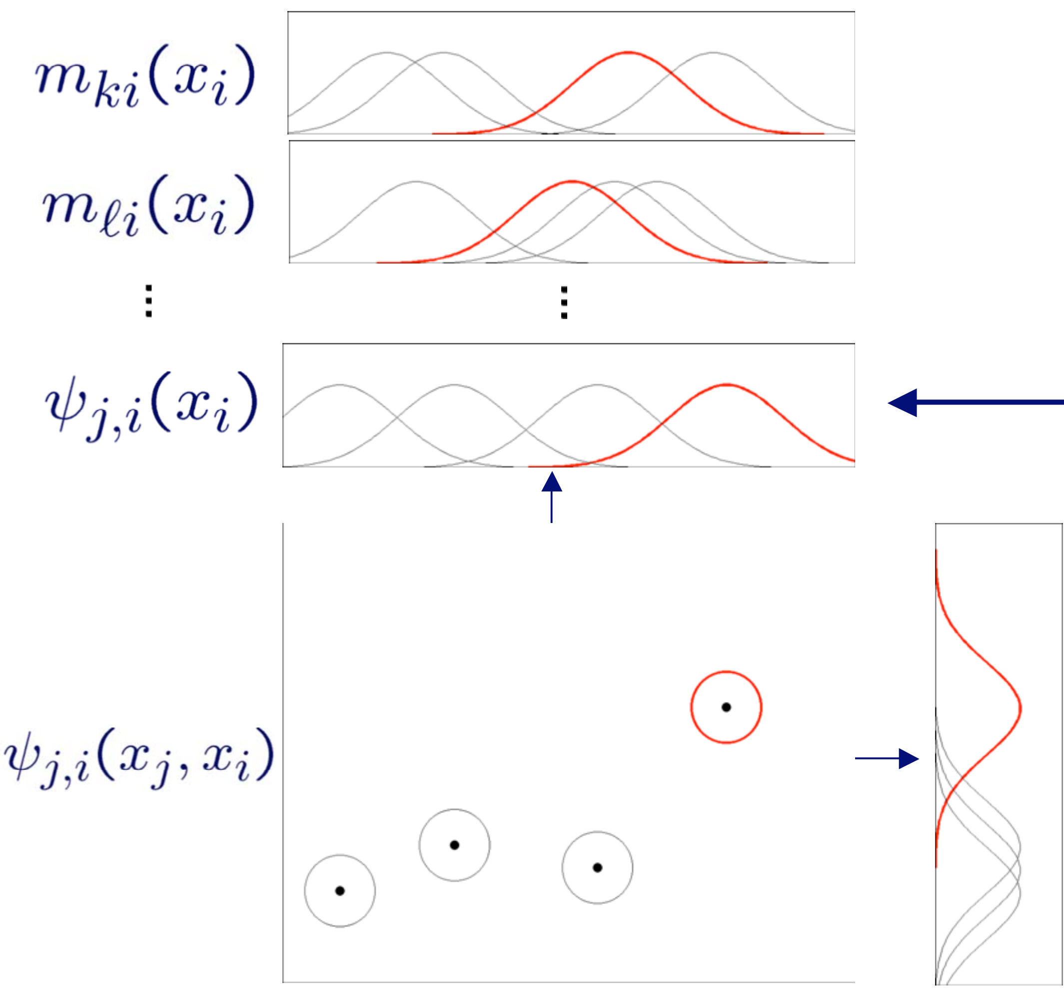
M^d kernels

We can now sample from this message product very efficiently

NBP Message Updates II:

Message Propagation (Gaussian Mixture)

$$m_{ij}(x_j) = \alpha \int_{x_i} \underbrace{\psi_{j,i}(x_j, x_i)\psi_i(x_i, y)}\prod_{k\in\Gamma(i)\setminus j} m_{ki}(x_i)\,dx_i$$



$\psi_{j,i}(x_j)$

View $\psi_{j,i}(x_j, x_i)$
as a joint distribution

Add marginal $\psi_{j,i}(x_i)$
to the product mix

Label selected by
sampler locates
kernel center in $\psi_{j,i}(x_j)$

⟶ Draw sample

Outline

Nonparametric Belief Propagation

- Graphical models and belief propagation
- Nonparametric message propagation
- Efficient multiscale sampling from mixture products

Visual Hand Tracking

- Prior constraints & image likelihoods
- NBP for occlusion-compensated hand tracking
- Temporal constraints & tracking results

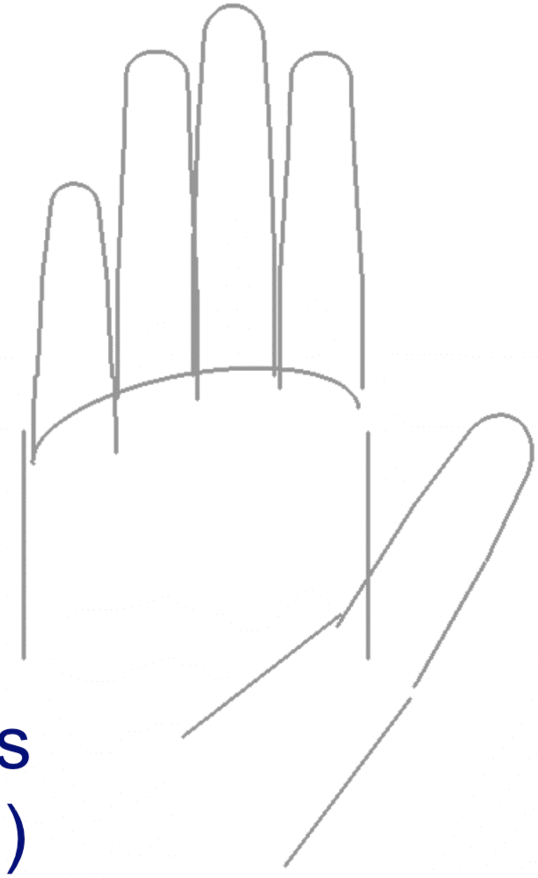
Motivation

- Accurately locating a few fingers highly constrains the set of possible global poses
- **GOAL:** Robustly propagate local image evidence to track arbitrary hand motions

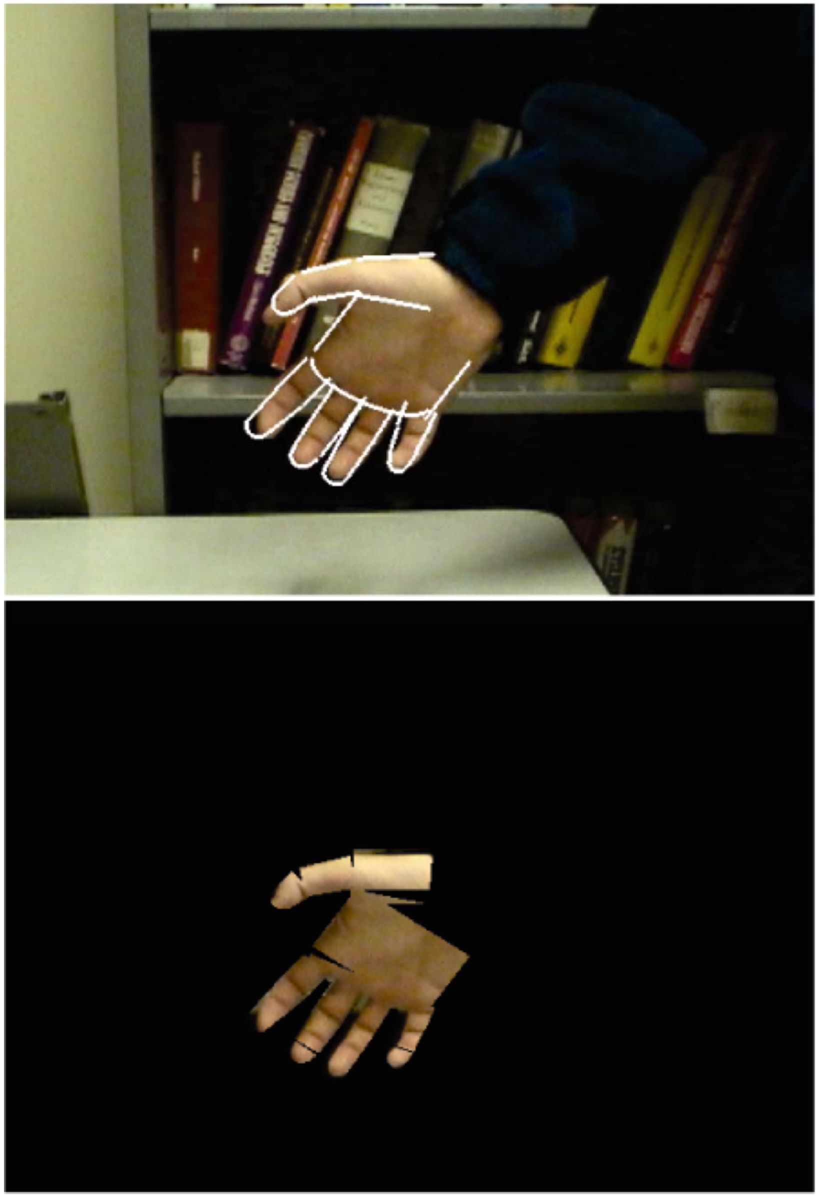
Structural Model

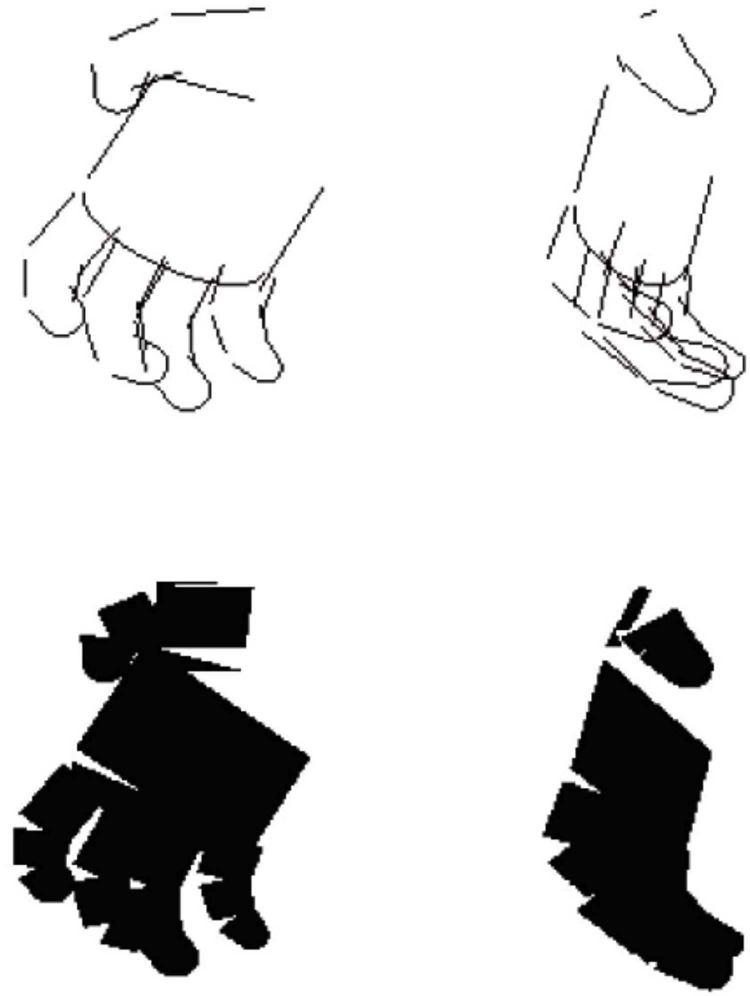
- Hand described by 16 *rigid bodies*
- 3D geometry of each rigid body modeled by truncated *quadric surfaces* (*Stenger et. al., CVPR 2001*)
  - Ellipsoids, cones, & cylinders
- Perspective projection maps quadrics to conics (ellipses, pairs of lines, etc.)

  → *efficient edge & silhouette calculation*
- Fixed geometry measured offline



Hand Model Projections

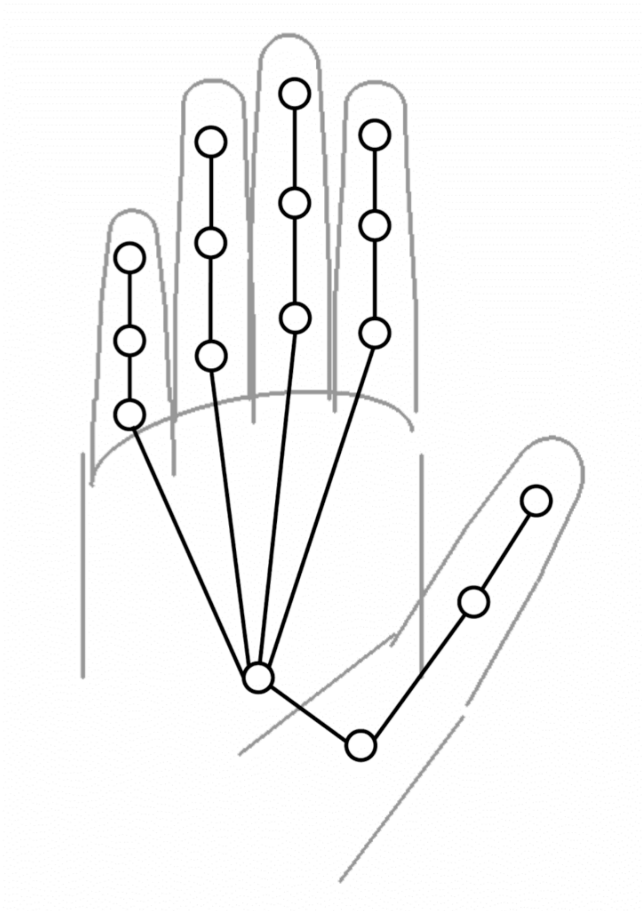




35° 70°

Kinematic Model

- Rigid bodies kinematically related by *revolute joints*
- Model has total of 26 DOF
  - 20 joint angles (4 per finger)
  - Palm's global position & orientation
- Likelihood calculation requires *global* coordinates of all bodies
  - No *direct* evidence for joint angle
- Forward kinematics maps joint angles to 3D poses



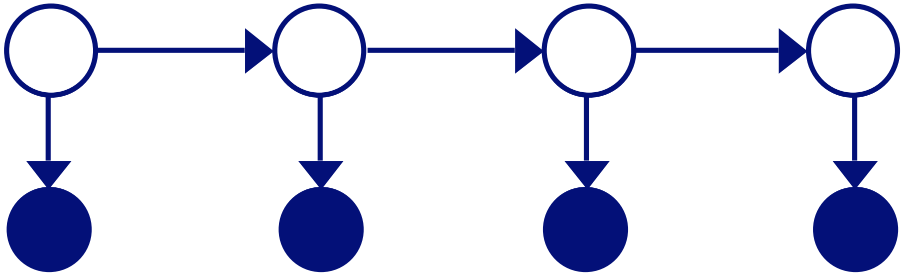
nodes ⟷ *rigid bodies*

edges ⟷ *joints*

Existing Hand Trackers

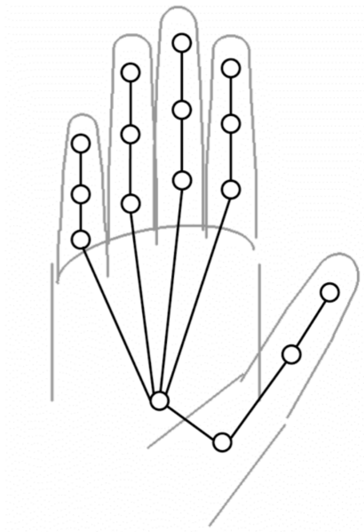
26-Dimensional State:

- 20 joint angles
- Global pose of palm



Unstructured Geometric Model Tracking

- Extended Kalman filter (Rehg 1994)
- Unscented Kalman filter (Stenger 2001)
- Particle filter (MacCormick 2000, Wu 2001)
- Tree-Based multiscale filter (Stenger 2003)



→ *all require simplified models (fewer DOF); many also employ complex prior models*

Local State Representation

- Describe each hand component by 3D pose:

$q_i \longrightarrow$ position of rigid body i

$r_i \longrightarrow$ orientation of rigid body i (unit quaternion)

$$x_i = \begin{bmatrix} q_i \\ r_i \end{bmatrix} \qquad x = \{x_1, \ldots, x_{16}\}$$

- Tradeoffs in this representation:
  - ***Redundant:*** Additional DOF (16 £ 6 = 96), *but*
  - Image appearance *directly* relates to local state

Related approach to 3D person tracking: Sigal, Black, Isard, et. al.

Kinematic Constraints

- Define an indicator function for each joint edge $(i, j) \in \mathcal{E}_K$

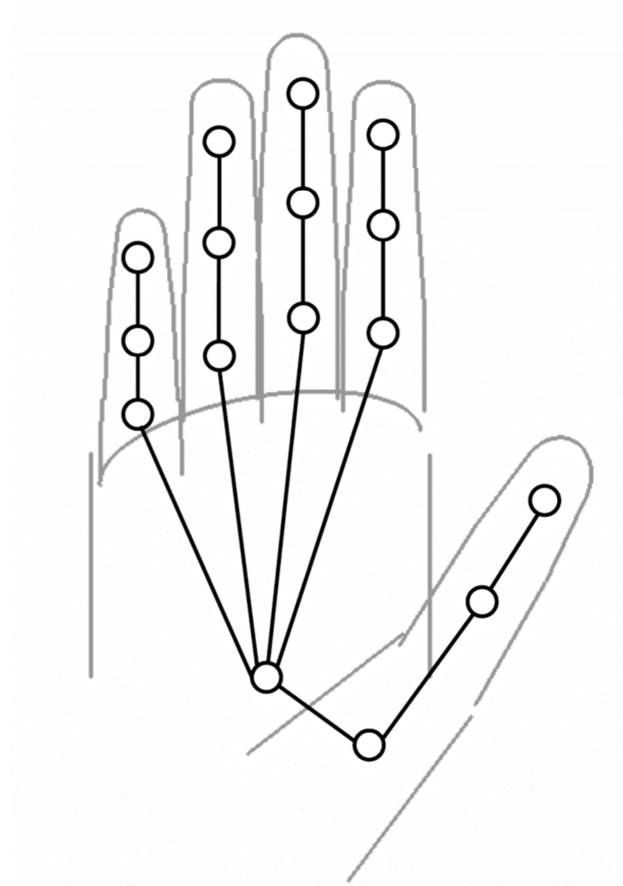
$$\psi_{i,j}^K(x_i, x_j) = \begin{cases} \text{1 if } (x_i, x_j) \text{ valid} \\ \text{for } \textit{some} \text{ choice of} \\ \text{joint angles, else 0} \end{cases}$$

- Kinematic prior model:

$$p_K(x) \propto \prod_{(i,j)\in\mathcal{E}_K} \psi_{i,j}^K(x_i, x_j)$$

- Graphical model *exactly* enforcing original joint angle constraints:

"Conditioned on the palm, the fingers are statistically independent"



$\mathcal{E}_K \longrightarrow$ *edges from joint constraints*

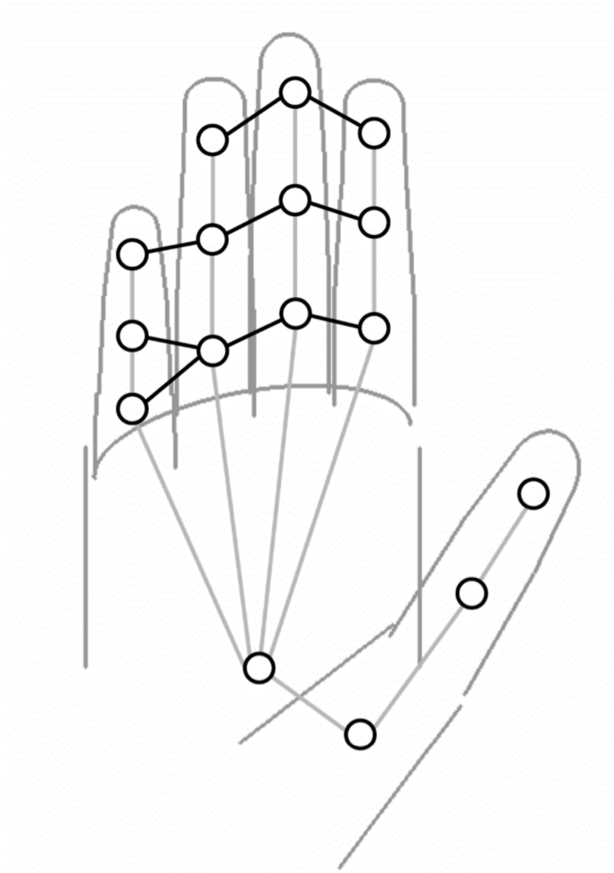
Structural Constraints

- Kinematics do not prevent finger intersection (joints *not* independent)
- "Ideal" structural constraint prevents 3D quadric surface intersection
- Approximate structural constraint:

$$\psi^S_{i,j}\left(x_i, x_j\right) = \begin{cases} 1 & \|q_i - q_j\| > \delta_{i,j} \\ 0 & \text{otherwise} \end{cases}$$

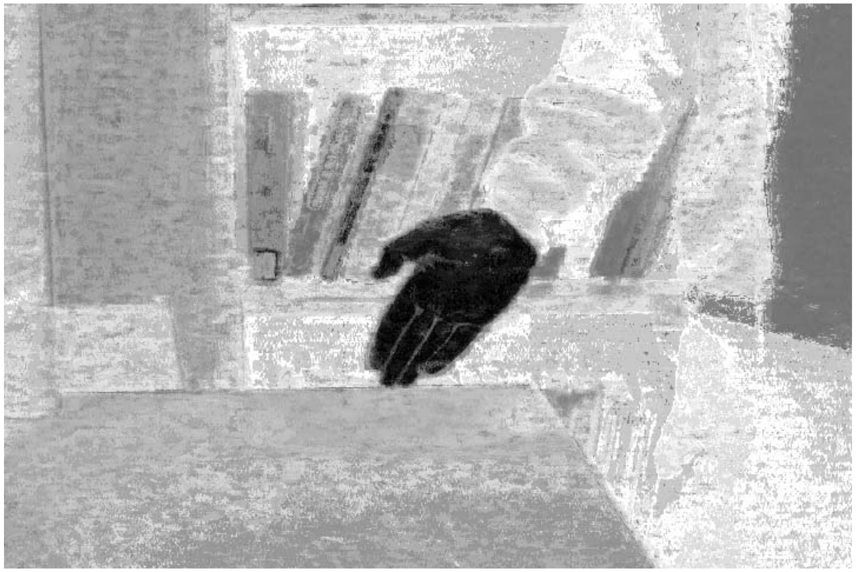
- Structural prior model:

$$p_S(x) \propto \prod_{(i,j)\in\mathcal{E}_S} \psi^S_{i,j}\left(x_i, x_j\right)$$



$\mathcal{E}_S$ → *edges from physical constraints*

Observation Model



Skin Color

Edge Intensity

Silhouette Matching: Skin Color

- Assume RGB values at each pixel independent:

$p_{\text{skin}} =$ histogram estimated from labeled skin pixels

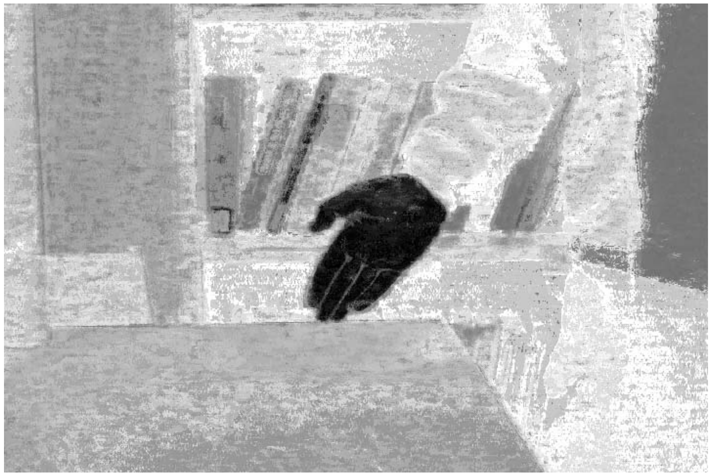
$p_{\text{bkgd}} =$ histogram estimated from hand-free background images

$\Omega(x) \longrightarrow$ pixels in silhouette of projection of model *x*

$\Upsilon \longrightarrow$ set of all image pixels

$$p_C(y \mid x) = \prod_{u \in \Omega(x)} p_{\text{skin}}(u) \prod_{v \in \Upsilon \setminus \Omega(x)} p_{\text{bkgd}}(v)$$

$$\propto \prod_{u \in \Omega(x)} \frac{p_{\text{skin}}(u)}{p_{\text{bkgd}}(u)}$$



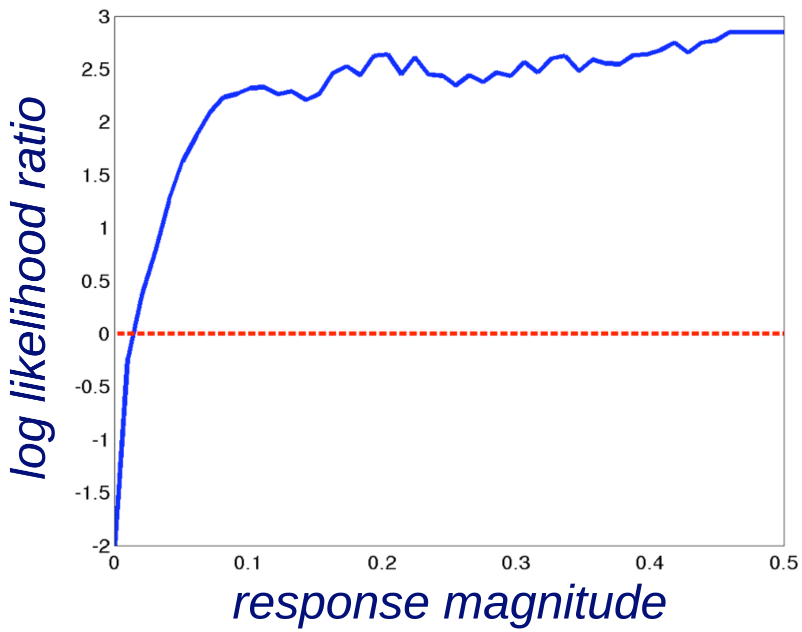
Must only evaluate likelihood ratio over projected silhouette

Edge Matching: Steered Gradient

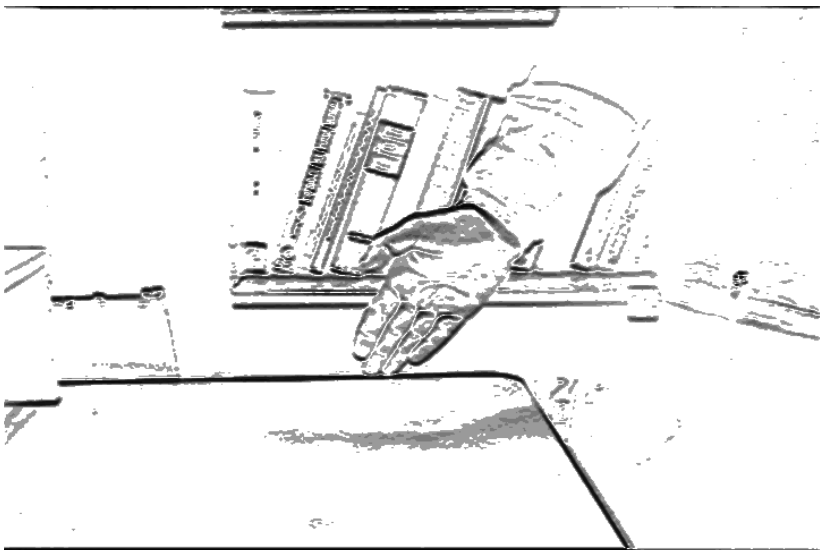
- Steer derivative of Gaussian response to orientation of projected hand boundary

$p_{\mathrm{edge}} =$ histogram estimated from labeled edge pixels

$p_{\mathrm{bkgd}} =$ histogram estimated from background images

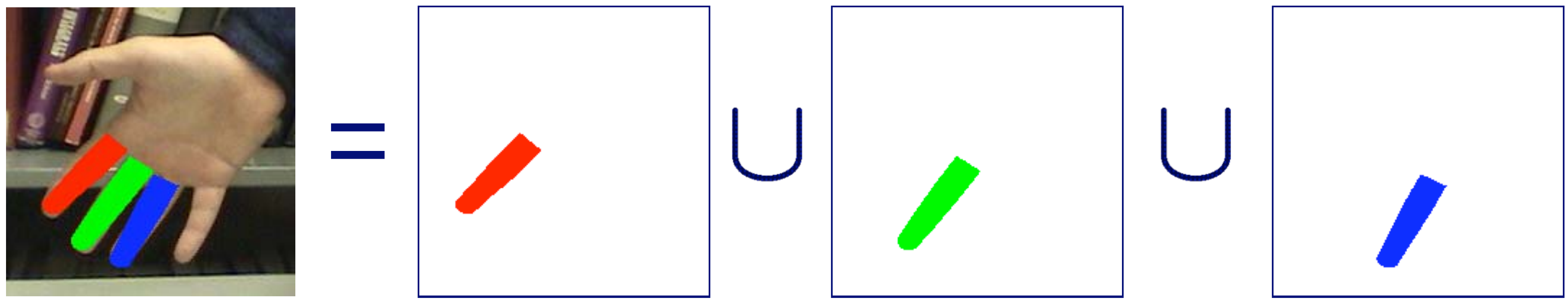


Dx



Dy

Local Likelihood Decomposition



If two hand components do not occlude each other, they will project to disjoint subsets of the image.

$$p_C(y \mid x) \propto \prod_{u \in \Omega(x)} \frac{p_{\text{skin}}(u)}{p_{\text{bkgd}}(u)}$$

$$\propto \prod_{i=1}^{16} \prod_{u \in \Omega(x_i)} \frac{p_{\text{skin}}(u)}{p_{\text{bkgd}}(u)} = \prod_{i=1}^{16} p_C(y \mid x_i)$$

- Edge likelihood ratio decomposes similarly
- Reasoning about self-occlusions discussed later…

Outline

Nonparametric Belief Propagation

- Graphical models and belief propagation
- Nonparametric message propagation
- Efficient multiscale sampling from mixture products

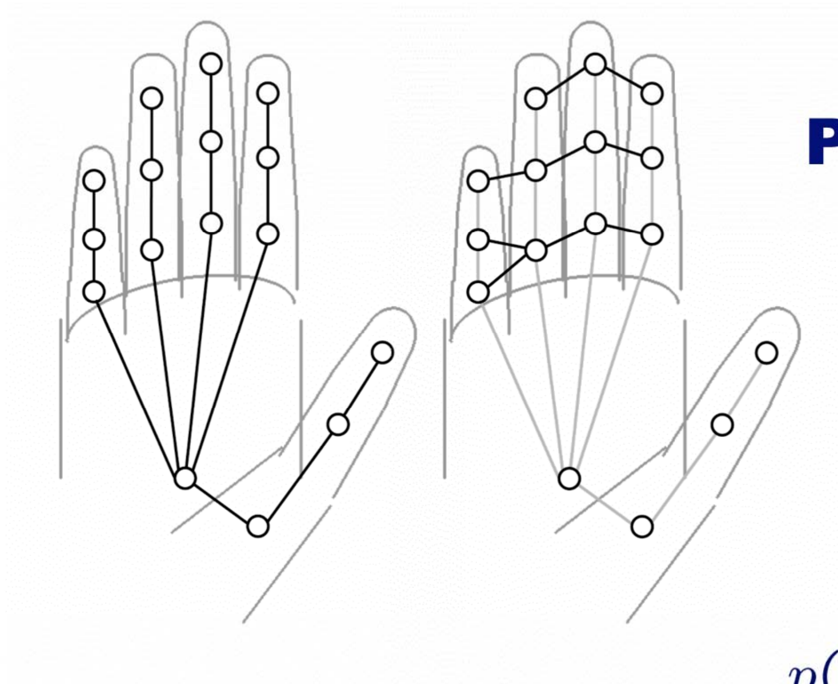
Visual Hand Tracking

- Prior constraints & image likelihoods
- NBP for occlusion-compensated hand tracking
- Temporal constraints & tracking results

Inferring Hand Position

$$p(x \mid y) \propto p_K(x) p_S(x) \left[\prod_{i=1}^{16} p_C(y \mid x_i) p_E(y \mid x_i) \right]$$

Color & Edge Likelihoods



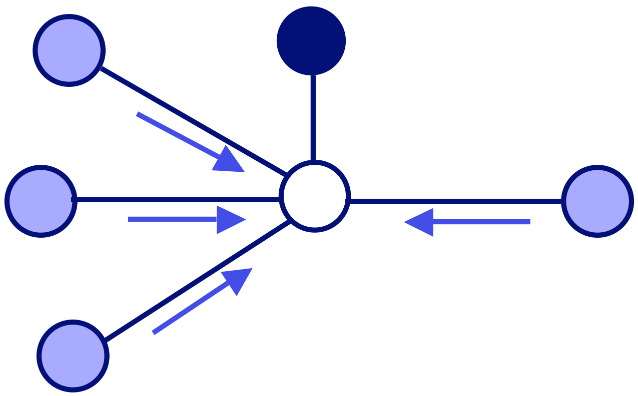
Kinematic Prior *Structural Prior*

Pairwise Markov Random Field

- $\mathcal{V}$ → *nodes* (random variables)
- $\mathcal{E}$ → *edges* (dependencies)
- x_i → hidden variable at node i
- y → observations

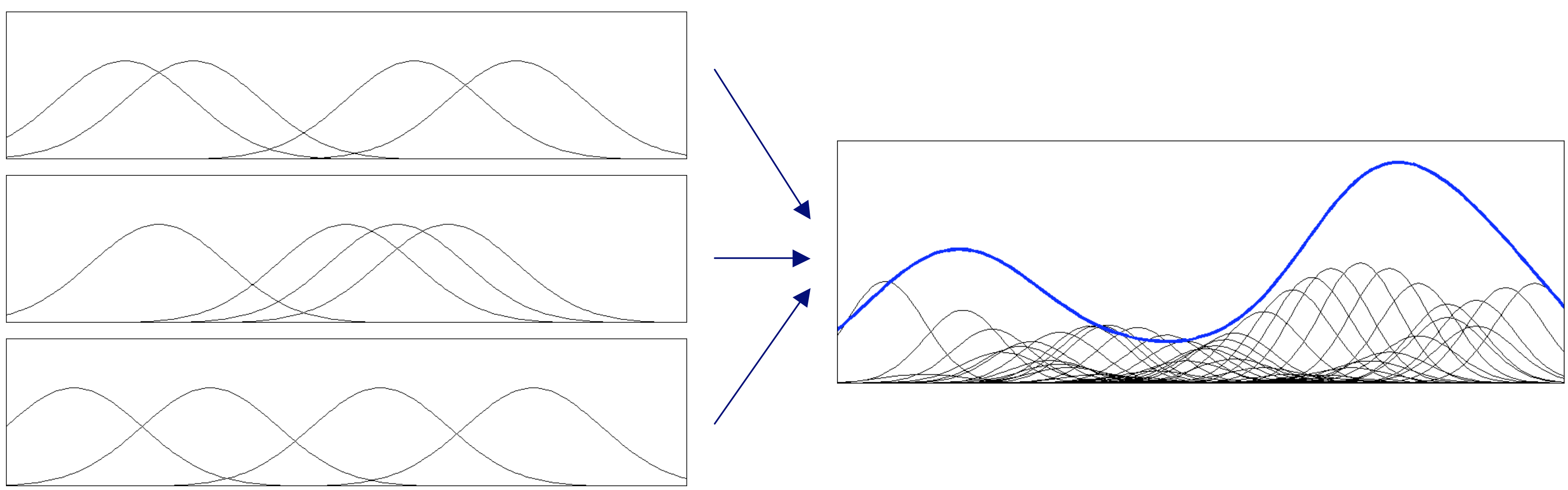
$$p(x \mid y) = \frac{1}{Z} \prod_{(i,j) \in \mathcal{E}} \psi_{i,j}(x_i, x_j) \prod_{i \in \mathcal{V}} \psi_i(x_i, y)$$

NBP Hand Tracker Marginal Update

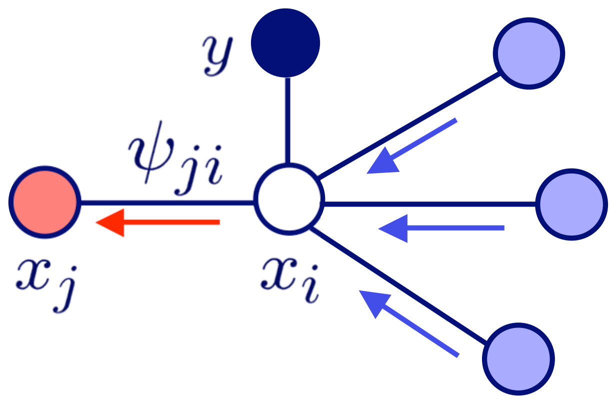


Importance Sampling:

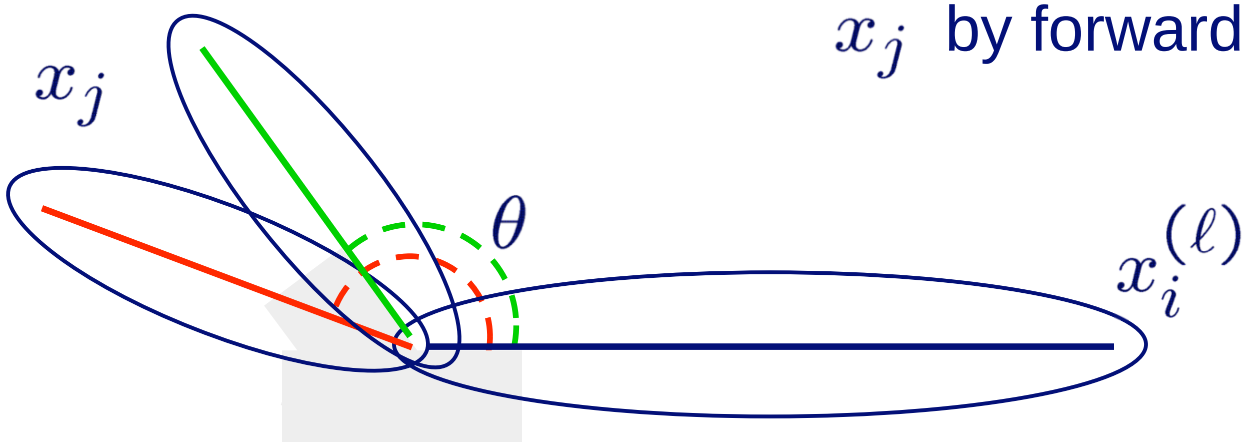
- Sample from product of all Gaussian mixtures
- Reweight samples by analytic functions (like particle filter)



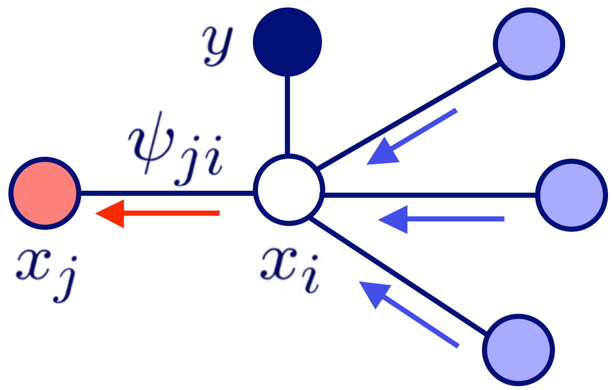
Kinematic Message Propagation



- Start with weighted samples $x_i^{(\ell)}$ from last marginal update
- Kinematic potential gives all valid poses equal weight:
  - Sample uniformly among allowable joint angles θ
  - Compute corresponding pose of x_j by forward kinematics



Structural Message Propagation

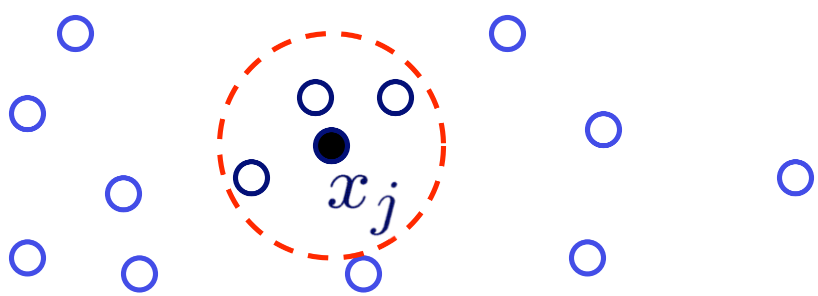


$$m_{ij}(x_j) = \alpha \int_{x_i} \psi^S_{j,i}(x_j, x_i) \frac{\widehat{p}(x_i|y)}{m_{ji}(x_i)} \, dx_i$$

$$\psi^S_{i,j}\left(x_i, x_j\right) = \begin{cases} 1 & ||q_i - q_j|| > \delta_{i,j} \\ 0 & \text{otherwise} \end{cases}$$

- Exact: Integrate belief over all poses outside some ball centered at the candidate pose x_j
- Approximate: Sum weights of all Gaussians with centers outside that ball

Reduces weight of particles which overlap with likely positions of neighboring nodes



Single Frame Inference

0

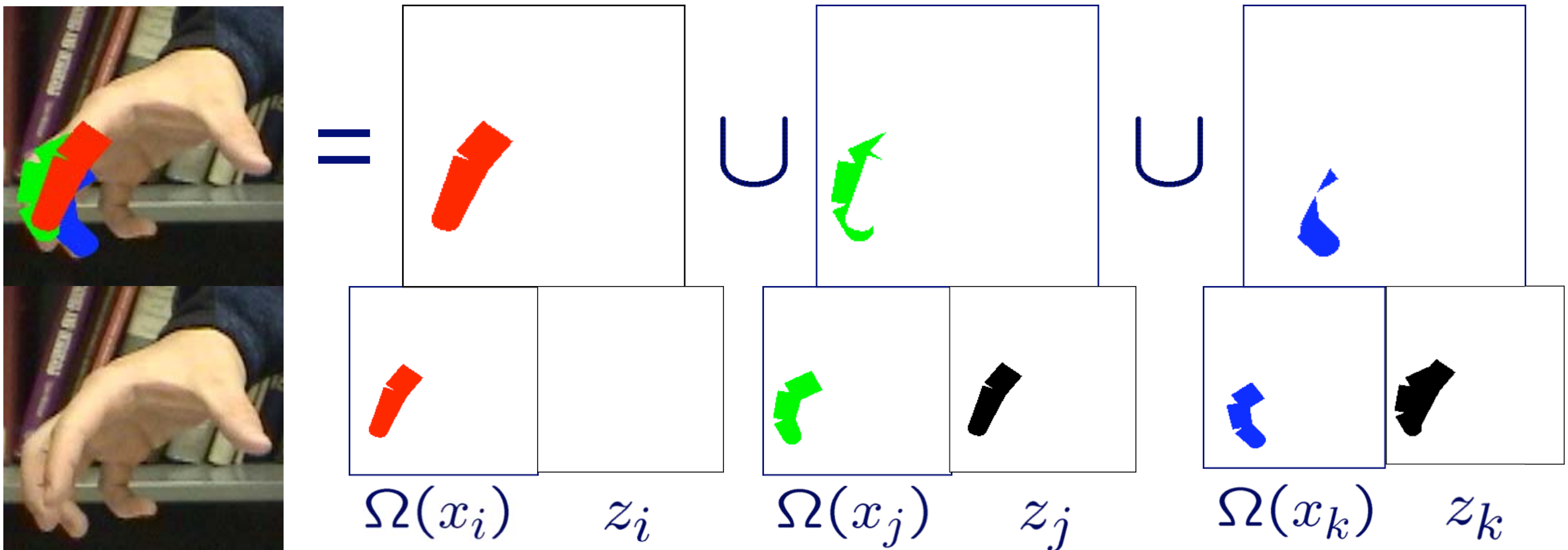
1

2

4

Self-Occlusion Masks

$$z_{i(u)} = \begin{cases} 0 & \textit{pixel u in the projection of body i is occluded} \\ 1 & \textit{otherwise} \end{cases}$$



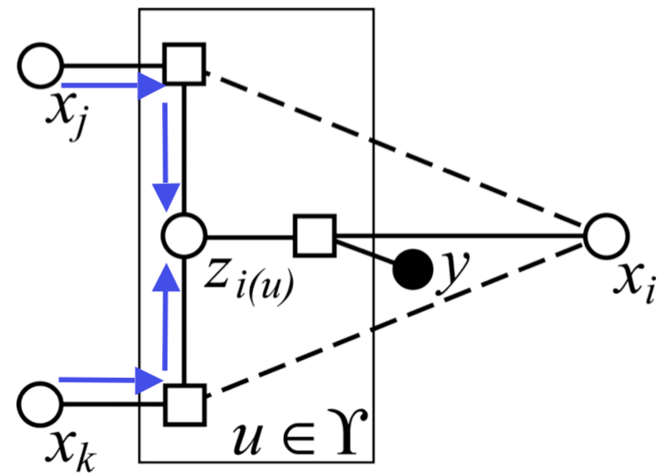
$$p_C(y \mid x, z) \propto \prod_{i=1}^{16} \prod_{u \in \Omega(x_i)} \left(\frac{p_{\mathsf{skin}}(u)}{p_{\mathsf{bkgd}}(u)} \right)^{z_{i(u)}}$$

Conditioning on occlusion masks z allows exact likelihood decomposition.

Distributed Occlusion Reasoning

- Factor graph imposes constraints ensuring occlusion consistency
- Use BP to analytically estimate probability of pixel's occlusion:

$$\nu_{i(u)} \triangleq \Pr[z_{i(u)} = 0]$$



- Neglecting correlations among the occlusion variables, the likelihood function (integrating over occlusions) becomes

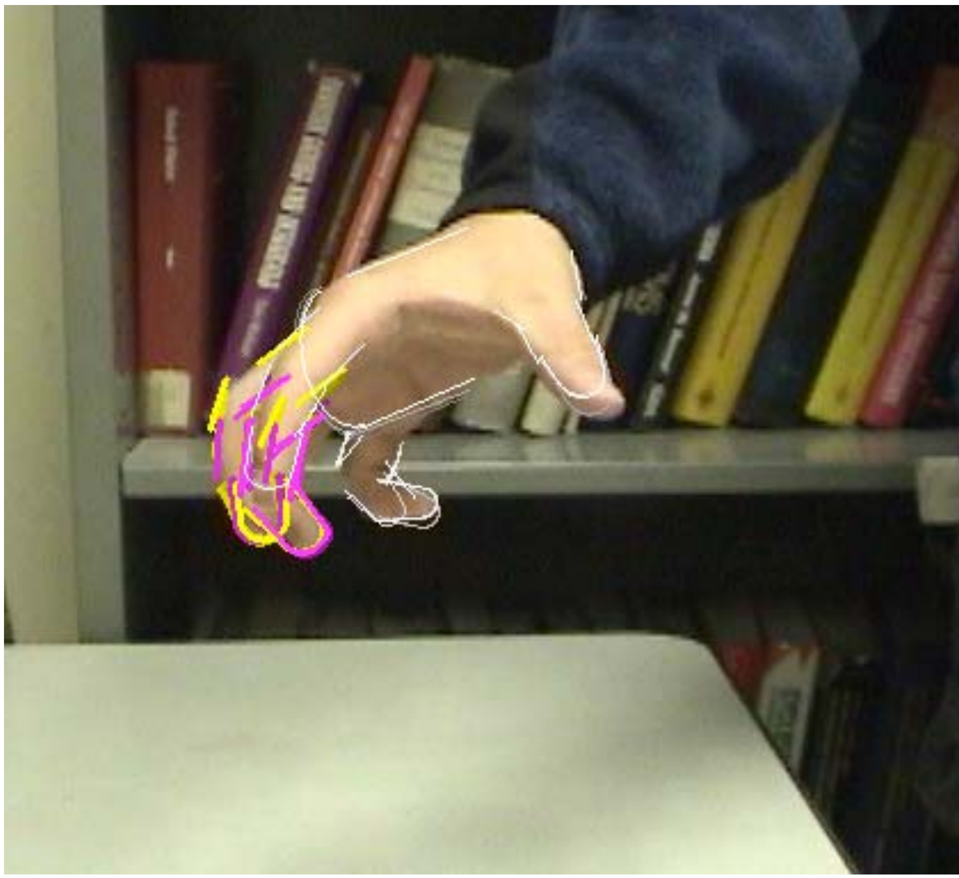
$$p_C(y \mid x_i) \propto \prod_{u \in \Omega(x_i)} \left[\nu_{i(u)}(1) + (1 - \nu_{i(u)}) \left(\frac{p_{\mathsf{skin}}(u)}{p_{\mathsf{bkgd}}(u)} \right) \right]$$

Uninformative Likelihood Ratio

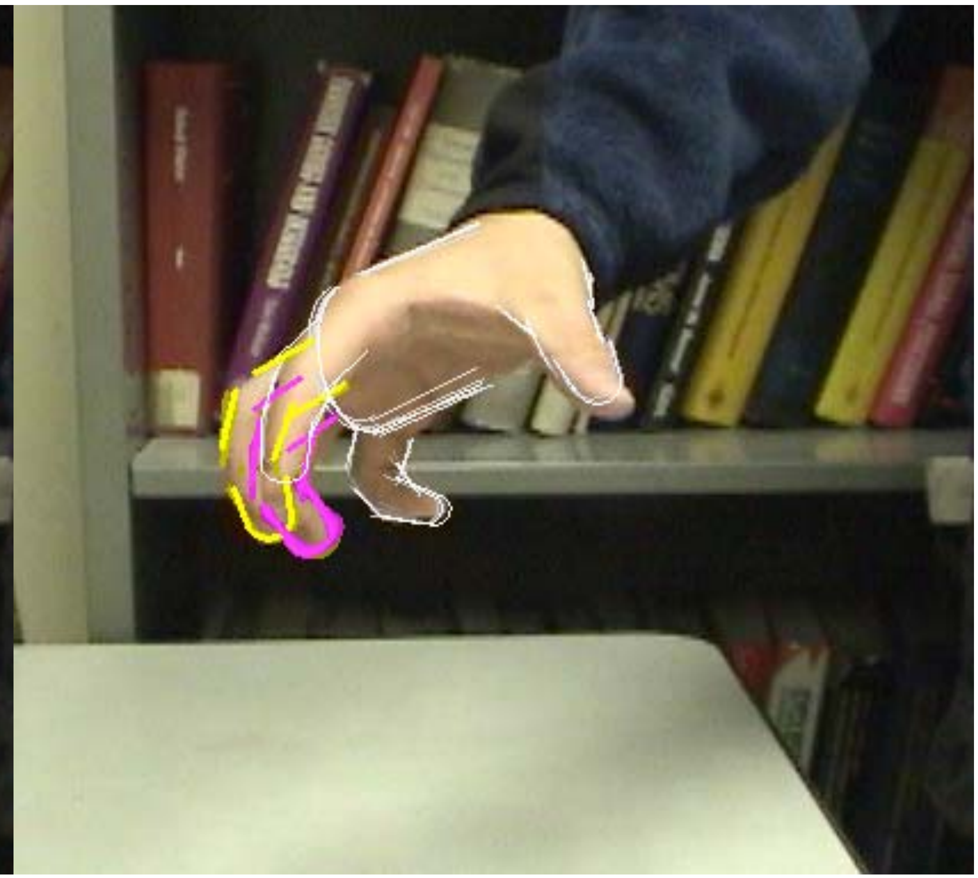
Skin Color Likelihood Ratio

Occlusion Reasoning Example

Middle (Third) Finger **Ring (Fourth) Finger**



No Occlusion Reasoning



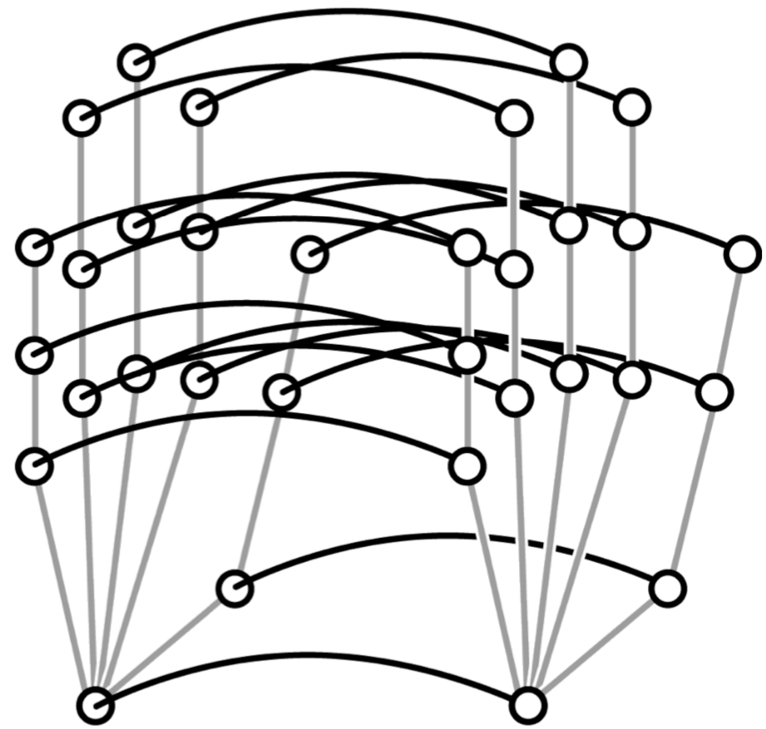
Occlusion Reasoning

Temporal Constraints & Tracking

- Add Gaussian potentials between adjacent time steps:

$$\psi\left(x_{t-1,i}, x_{t,i}\right) = \mathcal{N}(x_{t-1,i} - x_{t,i}; 0, \Lambda_{t,i})$$

- Interpretations:
  - Maximum entropy model given marginal variances in 3D pose
  - Random walks implicitly coupled by kinematic & structural constraints



$\mathcal{E}_T$ → *edges from temporal constraints*

Tracking Hand Rotation

Tracking Finger Motion

Conclusions

Publications & Code: *http://ssg.mit.edu/nbp/*

Nonparametric Belief Propagation

- Inference in continous, non-Gaussian graphical models
- Very flexible, easy to adapt to diverse applications
- Multiscale samplers lead to computational efficiency

Framework for Tracking Problems

- Modular state representation
- Graphical model of kinematics, structure, & dynamics
- NBP may accommodate complexities such as occlusions
- Many other potential applications…