# MCMC Tutorial at ICCV

<table>
<thead>
<tr>
<th>Time</th>
<th>Speaker</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.30-9.15</td>
<td>SCZ</td>
<td>Intro</td>
</tr>
<tr>
<td>10.15-11.15</td>
<td>FD</td>
<td><strong>MCMC Basics</strong></td>
</tr>
<tr>
<td>10.15-10.45</td>
<td></td>
<td>Break</td>
</tr>
<tr>
<td>10.45-11.30</td>
<td>SCZ</td>
<td>A variety of tricks for MCMC design</td>
</tr>
<tr>
<td>11.30-12.15</td>
<td>ZT</td>
<td>Exact sampling techniques</td>
</tr>
<tr>
<td>12.15-2.00</td>
<td></td>
<td>Lunch</td>
</tr>
<tr>
<td>2.00-2.45</td>
<td>FD</td>
<td>Trans-dimensional MCMC</td>
</tr>
<tr>
<td>2.45-3.30</td>
<td>SCZ</td>
<td>Cluster sampling</td>
</tr>
<tr>
<td>3.30-4.00</td>
<td></td>
<td>Break</td>
</tr>
<tr>
<td>4.00-4.45</td>
<td>ZT</td>
<td>Data-driven MCMC</td>
</tr>
</tbody>
</table>

Let5: **Data-driven MCMC:**

Integrating MCMC Search with Discriminative Computing

1. Generative and discriminative models
2. DDMCMC case study
3. DDMCMC approaches for vision applications
4. Conclusions
Challenges

1. Modeling: How to model various patterns in images?

\[ p(I|W), \text{ likelihood} \]

Appearances of scenes are highly complex.

\[ p(W), \text{ prior about } W \]

Complexity of scene configurations is enormous.

2. Computing: How to make inference of these patterns?

\[ W^* = \arg \max p(W|I) = \arg \max p(I|W)p(W) \]

Motivation for MCMC

1. Analytical solutions, if available, are always preferred.

Rao-Blackwellization theorem.

2. Otherwise, we should seek to sample \( p(W|I) \) directly, if we can.

3. Use proposals to guide the search of Markov chain.

Importance sampling, Metropolis-Hastings, Slice sampling, Reject sampling, …
Why do we use Metropolis-Hastings?

As an optimization technique: \( W^* = \arg\max p(W), \ W \in \Omega \)

1. \( W \) has complicated form and lies in a complex space, \( \Omega \), which is composed of sub-spaces of different dimensions.
2. There are no closed form solutions.
3. \( p(W|I) \) is not convex and PDE approaches find local minimum.
4. Well-computed proposals can quickly guide the MC jumping among promising modes.

What is the goal of vision? (Freeman and Blake ICCV 2001 short course)

If you are asking, “Are there any faces in this image?”, then you would probably want to use discriminative methods.

If you are asking, “Find a 3-d model that describes the runner”, then you would use generative methods.
Generative vs. Discriminative

Bayesian: (Top-down)
\[ p(W) \]
\[ W = \left( w_1, w_2, \ldots, w_k \right) \]
\[ p(I|W) \]
\[ p(W|I) \]
\[ W^* = \arg \max p(W|I) = \arg \max p(I|W)p(W) \]

Data-driven: (Bottom-up)
\[ W \sim \left( w_1, w_2, \ldots, w_k \right) \]
\[ F_1(I), F_2(I), \ldots, F_k(I) \]
\[ q(w_j|F_j(I)) \rightarrow p(w_j|I), j = 1 \ldots k \]

Generative (descriptive) Models: MiniMax

Entropy Principle

\[ W = \left( y, \alpha, \Theta \right) \]

I are samples of generative model: \[ I \sim p(x|y, \alpha : \Theta) \]


\[ p_\lambda(I|y) = \frac{1}{\sum_I \exp \left( -\sum_{j=1}^I \lambda_j h_j(I) \right)} \exp \left( -\sum_{j=1}^I \lambda_j h_j(I) \right) \]

Observation
Discriminative Models: Boosting

**AdaBoost and Its Variants:** (Freund and R. Schapire 1996, Friedman et al. 1998, Lebanon and Lafferty 2003)

\[
p_\lambda (y|I) = \frac{\prod_{j=1}^{T} \exp\{- \sum_{j=1}^{1} \lambda_j f_j(I, y)\}}{\sum_{y} \exp\{- \sum_{j=1}^{T} \lambda_j f_j(I, y)\}}
\]

\(f_{j=1..M} \) are weak classifiers.

---

**MiniMax Entropy Principle and Boosting**

*MiniMax Entropy:*

\[
p_\lambda (I|y) = \frac{\prod_{j=1}^{T} \exp\{- \sum_{j=1}^{1} \lambda_j h_j(I)\}}{\sum_{y} \exp\{- \sum_{j=1}^{T} \lambda_j h_j(I)\}}
\]

*Boosting:*

\[
p_\lambda (y|I) = \frac{\prod_{j=1}^{T} \exp\{\sum_{j=1}^{T} \lambda_j f_j(I, y)\}}{\sum_{y} \exp\{\sum_{j=1}^{T} \lambda_j f_j(I, y)\}}
\]

- Both have the feature selection procedure (greedy).
- Both follow the maximum-likelihood principle.
- Generative models focus on single class of interest.
- But Boosting is much easier to use since its normalization term is on \(y\).
- Although generative model is always preferred, we are forced to use discriminative models in many cases.
Discriminative models are often not good enough.

---

Discriminative Models (bottom-up) and Generative Models (top-down)

Methods of integrating between top-down (generative models) and bottom-up information have been widely applied in vision.

Integrating Discriminative and Generative Models

1. To integrate generative and discriminative models in a principled way.

2. Use discriminative models for fast computation.

3. Use generative models as verification.

4. DDMCMC provides such a framework.

Metropolis-Hastings

To design transition kernel $K$

$p \bullet K = p$

Detailed balance:

$$p(W_A)K(W_A \rightarrow W_R) = p(W_R)K(W_R \rightarrow W_A).$$

Metropolis-Hastings:

$$K(W_A \rightarrow W_B) = Q(W_B|W_A; F(I)) \alpha(W_A \rightarrow W_B)$$

$$\alpha(W_A \rightarrow W_B) = \min\{1, \frac{Q(W_A|W_B; F(I))}{Q(W_B|W_A; F(I))} \frac{p(W_B|I)}{p(W_A|I)} \}$$

$$Q(W_B|W_A) \Rightarrow Q(W_B|W_A; F(I))$$
DDMCMC Approaches for Vision Applications

Target and object recognition:

Segmentation and Perceptual grouping:

Tracking:

Stereo and 3D:
- Dellaert et al. 2001, Han and Zhu 2003, Barbu and Zhu 2005

Color constancy: Forsyth 1999

A Case Study: Switching Linear Dynamic Systems

* SLDSs have been widely researched – computer vision etc.
* Complex phenomenon
  = Switching among a set of linear dynamic models over time.
Honeybee Dance

* Honeybee dance comprises of three patterns.
  * left-turn (blue), waggle (green), right-turn (red)

Inference in SLDS

Inference in an SLDS model is intractable. (Lerner & Parr, UAI-01)

Approximate Inference Algorithms

* Approximate Viterbi, Variational method (Pavlovic & Rehg, CVPR-00)
* GPB2 (Bar-Shalom & Li, 1993)
* Kalman Filtering (Bregler, CVPR-97)
* Expectation propagation (Zoester & Heskes, PAMI-03)
  and many others...

The Ultimate: MCMC inference method.
- Golden standard. Theoretically, it converges to the true posterior.
- Characterizes the accuracy of deterministic Approx. algorithms.
A DDMCMC Approach

* Target Distribution: \( P(L | Z) \)

1. Start with a valid initial label sequence \( L^{(1)} \).

2. Propose a new label sequence \( L' \) from \( L \) using a proposal density \( Q \).

3. Calculate the acceptance ratio:
   \[
   a = \frac{P(L | Z) Q(L'; L)}{P(L | Z) Q(L' | L)}
   \]

4. Accept or reject the new sample in the MH framework.

Experimental Results

* Comparison with the approximate Viterbi method.

- Approx. Viterbi VS. MCMC MAP
  - A large number of over-segmentations disappear.
  - Additionally able to analyze the classification results.
A Case Study: Image Segmentation by DDMCMC

\[ W = (n, \{(R_i, l_i, \theta_i), i = 1, \ldots, n\}) \]

- \( R_i \): partition (by boundaries)
- \( l_i \): intensity type
- \( \theta_i \): parameters

Sampling the Posterior Distribution

\[ W^* = \arg \max_{W \in \Omega} p(W|I) = \arg \max_{W \in \Omega} p(I|W)p(W) \]

Markov Chain: \( MC = (\pi, K, p_0) \)

To design transition kernel: \( \pi \cdot K^\pi \rightarrow \pi \)
## Likelihood Models

1. iid Gaussian for pixel intensities
   \[ p(I_R; l_1, \theta) = \prod_{v \in R} G(I_v - \mu; \sigma^2) \]

2. non-parametric histograms
   \[ p(I_R; l_2, \theta) = \prod_{v \in R} h(I_v) \]

3. Markov random fields for texture
   \[ p(I_R; l_3, \theta) = \prod_{v \in R} p(I_v; I_{b_v}; \theta) \]
   \[ = \prod_{v \in R} \frac{1}{Z_v} \exp\left( - \theta \cdot h(I_v; I_{b_v}) \right) \]

4. spline model for lighting variations
   \[ p(I_R; l_4, \theta) = \prod_{v \in R} G(I_v - B_v; \sigma^2) \]

5. iid Gaussian for color (LUV)

6. mixture of Gaussians for color

7. spline model for smooth color variations (e.g. sky, shading, …)

---

## Top-down Approaches

- Gibbs Sampler (Geman and Geman 1984)
- Variational Method (Mumford and Shah 1989)
- Jump-Diffusion (Grenander and Miller 1994)
- Region Competition, PDE (Zhu and Yuille 1996, Osher and Sethian, 1988), …

\[ W = \{ n, \{ (R_i, l_i, \theta_i), i = 1, \ldots, n \} \} \]

\[ \text{Generator} \]

\[ p(I|W) \]

\[ \text{Inference} \]

\[ p(W|I) \]

**Pros:**
- Easy to incorporate various levels of knowledge.

**Cons:**
- Either local minimal or very slow.
**Bottom-up Approaches**

- Edge detection (Canny 1986)
- Clustering (Jain and Dubes 1988, Comaniciu and Meer 1998),
- Graph-Cuts (Shi and Malik 1997),

\[ W \quad (w_1, w_2, \ldots) \]

\[ F_{\text{edge}}(I) \]

\[ F_c(I) \]

**Pros:**
- Usually very fast.

**Cons:**
- Cues obtained are often local and inconsistent.
- Data-driven approaches are not directly exploring the solution space.

---

**Proposals by Edge Detection at Different Scales**

Partition maps: \( q(\text{partition} | F_{\text{edge}}(I)) \)

Scale 1

Scale 2

Scale 3
Proposals for Models by Clustering

\[ q(\theta | F_c(I)) = \sum_{i=1}^{K} \omega_i G(\theta - \theta_i); \]

Saliency maps (the brightness represents how likely a pixel belongs to a cluster.)

MCMC Kernels

Transition kernel \( K \):

\[ K(W_A \rightarrow W_B) = \sum_{a} q_a K_a(W_A \rightarrow W_B) \]

\[ p(W_A)K_a(W_A \rightarrow W_R) = p(W_R)K_a(W_R \rightarrow W_A) \]
Design Issues for the Markov Chain Search

1. We want to have efficient moves—big scope of $x$. $\Omega(x) \sim \Omega$

2. The proposals should be as close to the true distribution as possible. (Generalized Metropolis-Gibbs sampler)

$$Q_m(x, y) = \sum_{y' \in \Omega_m(x)} \frac{p(y)/p(x)}{p(y'/p(x)} \sim q(y' | x)$$

MCMC Moves (Jumps and Diffusions)

$K_1$: Splitting of a region into two.

$K_2$: Merging two regions into one.

$K_3$: Switching the model type for a region.

$K_4$: Diffusion of region boundary — region competition (Zhu and Yuille 1996).

It integrates a variety of existing segmentation methods in computer vision such as:

Edge Detection, Clustering, Split-merge, Region Competition (Snake, MDL, PDE) ...
**Split and Merge**

Consider a reversible jump: $W_A = (1, (R_1, l_1, \theta_1)) \leftrightarrow W_B = (2, (R_2, l_2, \theta_2), (R_3, l_3, \theta_3))$.

\[
\begin{align*}
K(W_A \rightarrow W_B) &= \frac{Q(W_B|W_A; F(I))}{Q(W_A|W_B; F(I))} \alpha(W_A \rightarrow W_B) \\
Q(W_B|W_A, F(I)) &= Q(R_2, R_3|W_A; F_{\text{edge}}(I)) \cdot Q(l_2, \theta_2|R_2, W_A; F_{\text{edge}}(I)) \cdot Q(l_3, \theta_3|R_3, W_A; F_{\text{edge}}(I))
\end{align*}
\]

**Stochastic Diffusion and PDE**

The continuous Langevin equation simulates a Markov Chain with stationary density

\[
p(W|I) \propto \exp\{-E(W)/T\}
\]

For example, the movement of changing point is driven by

\[
\frac{dx(t)}{dt} = \left\{ \left[ \log p(x|l_i, \theta_i) - \log p(x|l_j, \theta_j) \right] - \kappa x + \sqrt{2T(t)}N(0, 1) \right\} \tilde{\epsilon}
\]
**Speed Comparison**

![Graph showing Speed Comparison]

**A Demo**

- **Segmentation**
- **Synthesis**

- **Snapshot of solution sampled by DDMCMC**
Revisit of Top-down And Bottom-up

Two approaches: top-down and bottom-up

Top-down (Generative)

\[ p(W|I) \propto p(I|W)p(W) \]

- Bayes Theory (Bayes 1763, Pearl 1988, Gelman, Carlin, Stern, and Rubin 1995,...)
- General Pattern Theory (Grenada 1993, Mumford 2001,...)

Bottom-up (Discriminative)

\[ g(w_j|F_j(I)) \rightarrow p(w_j|I) \]

- Neural Networks (Rosenblatt 1962, LeCun 1986, Rumelhart and McClelland 1986,...)
- Support Vector Machines (Vapnik 1995,...)
- Boosting, AdaBoost, Bagging (Freund and Schapire 1996, Friedman et al. 1998,...)
- Decision Tree (Wang and Suen 1984, Amit and Geman 1997,...)

Some Failure Examples: Need to Engage Middle-level and High-level Knowledge
Image Parsing

\[ W = (n, \{(\zeta_i, R_i, l_i, \theta_i)_i, i = 1, \ldots n\}) \]

Results

<table>
<thead>
<tr>
<th>Input</th>
<th>Regions</th>
<th>Objects</th>
<th>Synthesis</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Face images of FERET dataset

Text images of San Francisco street scenes.
Applications—Segmentation with Gestalt Laws


To integrate the low level segmentations with mid-level Gestalt cues.

\[
f(S) = \sum_{S \in S} \left( \sum_{j} c_j F_j(S) - \theta \right)
\]

Applications—SWC with Shape Prior


\[
p(W|I, S) \propto p(I|S, W)p(S|W)p(W) \\
\propto \prod_i p(I_i|S_i, W) \prod_i p(S_i|W_i)p(W) \\
\propto \prod_i p(I_i|\theta_i, S_i) \prod_i p(S_i|C_{\theta_i}) p(W)
\]
Applications—Tracking Multiple Objects


\[
\theta^{(t)*} = \operatorname{argmax} p(\theta^{(t)}|\theta^{(t-1)}, Bg^{(t-1)}) \\
= \operatorname{argmax} p(I^{(t)}|\theta^{(t)}, Bg^{(t-1)})p(\theta^{(t)}|\theta^{(t-1)})
\]

Applications—Body Configuration

Theoretical Side

Analyze the convergence rate of the system.

\[
\frac{1}{\min\{p(x), q(x)\}} \leq E[f(x)] \leq \frac{1}{\min\{p(x), q(x)\}} \cdot \frac{1}{1 - \|p - q\|}
\]

Maciuca and Zhu 2003

To study the optimal control strategy.

- Ordering of the kernels given the current cues (tests).

\[
W_t \sim \mu_t(W) = \nu(W) \circ K_{\alpha(1)} \circ K_{\alpha(2)} \circ \cdots \circ K_{\alpha(t)}
\]

\[
\delta_{\alpha(t)} = KL(p(W|\mu_t(W))) - KL(p(W|q_t(W))) = E[KL(K_{\alpha(t)}(W_t|W_{t+1}))]\|
\]

- Ordering the tests by their power.

\[
S(w|F_{+}) = E[KL(p(w|I)|q(w|Tst(I)))] - KL(p(w|I)|q(w|Tst(I), F_{+})]
\]

\[
= MI(w||Tst(I), F_{+}) - MI(w||Tst(I)) = KL(q(w|Tst(I), F_{+})||q(w|Tst(I)))
\]

- Ordering kernels and tests by their computation cost.

Take-home Messages for DDMCMC Approaches

1. DDMCMC is an open framework which can integrate various methods (top-down, bottom-up, and PDEs).

2. DDMCMC can deal with solutions of complex form.

3. The performance of DDMCMC is largely decided by proposals.

4. To decrease the complexity of the approach and increase its scalability.

5. We should tightly couple discriminative and generative models and bring more learning aspects to DDMCM.