Bayesian Phase Unwrapping with Factor Graphs

Eric Jonas

MIT Department of Brain and Cognitive Sciences

May 12, 2009

Markov Random Fields and Factor Graphs

Markov random field, undirected graphical model, etc.
Factor Graph: a particular language / notation for markov random fields
Show CPT talk about markov blanket
[?]

Factor Graphs for Low-Level Vision

Properties of image MRFs large number of verticies O(1) (constant local) connectivity

Bayesian Factor Graphs

Decompose into prior, likelihood, etc.

MRFs for Phase: Frey's approach

more classical image MRF with delta functions, continuous state

discrete latent state, uniform factors

discrete latent state, unique factors

My formulation

Inference in MRFs

Our MRF has given us $p*(x|D)$, which is not convex, and not even a valid probability distribution.

We would like to somehow "solve" this system to get a rough sense of the distribution $p(x|D)$.

Two generic approaches:

- draw samples from $p(x|D)$ to empirically estimate
- optimize to find MAP solution

Inference in MRFs

Our MRF has given us $p*(x|D)$, which is not convex, and not even a valid probability distribution.

We would like to somehow "solve" this system to get a rough sense of the distribution $p(x|D)$.

Two generic approaches:

- draw samples from $p(x|D)$ to empirically estimate
- optimize to find MAP solution

We focus on sampling.

Markov-Chain Monte Carlo

Markov Property: next state only depends on current state

$$p(x_{t+1}|x_{1:t} = p(x_{t+1}|x_t) \tag{1}$$

Ergodic markov chains have stationary distributions
Set up a state space so that the expectation is the target distribution Used in situations where you know $\pi^*(x)$ but not $\pi(x)$.

Metropolis Hastings

One way to construct this Markov Chain

$$a = \min(1, \frac{p(x^*)}{p(x)} \cdot \frac{q(x \rightarrow x^*)}{q(x^* \rightarrow x)}) \quad (2)$$

[?]

Gibbs Sampling

like MH but along an axis, useful when we can condition on other variables. Look, we can gibbs sample in image MRFs with discrete state spaces [**?**]

Tempering

Like Simulated Annealing

Swendsen-Wang

Work Through

MRFs and Parallelism

The conditional independence assumptions allow fine-grained parallelism

Our Implementation

use SW, etc. python, numpy, scipy, c++, boost, etc.
multithreaded

How to measure performance? I'm going to go for log-likelihood,

2-D Synthetic Data

3-D Synthetic Data

Div and Audrey

PRELUDE

Where to now?

Exact sampling using Systematic Stochsatic Search Better neighborhood connectivity / likelihood? GPU implementation Better visualization of posterior?

More information

Source is on github