Lecture 6 — Sampling / MCMC / Graph Cuts / Linear Programming

Note Title 1/24/2010

Brief introduction to other techniques for optimization.

First, mention dynamic programming & Max-weights TRW.

Linear Programming →

What is the limit of a Bethe / Convex Free energy as $T \to 0$?

$$\text{minimize} \quad \sum_{ij} b_{ij}(x_i, x_j)\psi_{ij}(x_i, x_j) + \sum_i b_i(x_i)\phi_i(x_i).$$

w.r.t. $\langle b_{ij}, b_i \rangle$

subject to linear constraints

$$\sum_{x_j} b_{ij}(x_i, x_j) = b_i(x_i)$$

$$\sum_{x_i} b_i(x_i) = 1.$$

This is an example of linear programming (LP).

See wikipedia.

$$\text{maximize} \quad \underline{c}^T \underline{x} \quad \text{subject to} \quad A\underline{x} \le \underline{b}$$

There are courses taught on this at UCLA

eg Prof. Vandenberghe

There are software packages available online

Caution – general purpose packages probably won't exploit the special structure of computer vision problems – so they may be faster algorithms

Note: the minimum of ## usually does not correspond to the minimum

$$(*) \quad \langle \hat{x}_i \rangle = \arg\min_{\langle x_i \rangle} \left\{ \sum_{ij} \psi_{ij}(x_i, x_j) + \sum_i \phi_i(x_i) \right\},$$

Current research investigates the relationship of BP/TRW for small T and linear programming and the solution of the optimization problem (*).

→ e.g. dual methods like BP/TRW may be more effective than standard linear programming methods for some problems.

(2) # Stochastic Sampling & MCMC methods

Also taught at UCLA. Introduction Stat 202C.
More advanced classes. → Stat 231 & (Bio)

Basic Idea –

— if we can sample from a distribution $P(\underline{x})$
to obtain random independent samples $\underline{x}_i \sim P(\underline{x})$
then we can estimate properties of interest by

$$\sum_{\underline{x}} f(\underline{x})P(\underline{x}) \approx \frac{1}{N}\sum_{i=1}^{N} f(\underline{x}_i)$$
$$\int f(\underline{x})P(\underline{x})d\underline{x}$$

How many samples N required?

Key Result → the estimate $\frac{1}{N}\sum_{i=1}^{N} f(\underline{x}_i)$ is a random variable, its expectation is $\sum_{\underline{x}} f(\underline{x})P(\underline{x})$ $\int d\underline{x}$

its variance is independent of the dimension of the space of $\underline{x}$! (Decreases like $1/N$.)

Claim → big advantage if $\underline{x}$ lies in a high-dim space.
compared to alternative ways to approximate $\sum_{\underline{x}} f(\underline{x})P(\underline{x})$ $\int d\underline{x}$

In particular, we can estimate quantities such as

$$\sum_{\underline{x}} \underline{x}P(\underline{x}) \quad \int d\underline{x}$$

which can be used for inference

How to get independent samples from a distribution $P(\underline{x})$?

For simple distributions → e.g. $P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ 1-D Gaussian

this can be done directly. (Random no. generators)

Many techniques –

- importance sampling
  - ~ sample from $q(x)$ to get $x_1 \ldots x_m$ ($q(x)$ easy to sample from.)
  - ~ weight samples by $w_i = \frac{P(x_i)}{q(x_i)}$
    and set $\tilde{w}_i = w_i / \sum_j w_j$.
  - ~ resample from $\{x_1 \ldots x_m\}$ by probs $\tilde{w}_i$

(3) In general, we cannot sample directly from probability distributions → certainly not from MRFs

Instead, need a procedure/algorithm that results in a sample from $P(\underline{x})$.

Markov Chain Monte Carlo (MCMC)

Define a transition kernel $K(\underline{x}|\underline{x}')$

$$\sum_{\underline{x}} K(\underline{x}|\underline{x}') = 1, \quad \forall \underline{x}'$$
$$K(\underline{x}|\underline{x}') \geq 0$$

condition $\sum_{\underline{x}'} K(\underline{x}|\underline{x}')P(\underline{x}') = P(\underline{x})$ (fixed point)

sufficient – detail balance $K(\underline{x}|\underline{x}')P(\underline{x}') = K(\underline{x}'|\underline{x})P(\underline{x})$

require for any states $\underline{x}, \underline{y}$, there exist a trajectory $(\underline{x}_1, \ldots \underline{x}_N)$ s.t. $\underline{x}_1 = \underline{x}$, $\underline{x}_N = \underline{y}$ & $K(\underline{x}_i|\underline{x}_{i-1}) > 0$.
(you can get to $\underline{y}$ from $\underline{x}$)

MCMC: initialize $\underline{x}_0$ at random;

sample $\underline{x}_1$ from $K(\underline{x}|\underline{x}_0)$
$\underline{x}_2$ " $K(\underline{x}|\underline{x}_1)$
$\vdots$
$\underline{x}_N$ " $K(\underline{x}|\underline{x}_{N-1})$

then $\underline{x}_N$ will be a random sample from $P(\underline{x})$ for suff. large N?

How quick does this converge? How large N?
Treat $K(\underline{x}'|\underline{x})$ as a matrix
Calculate its eigenvalues & eigenvectors
MCMC converges like $e^{-N\lambda_2}$
where λ_2 is the second largest (mod) eigenvalue of $K(\underline{x}'|\underline{x})$
(largest eigenvalue is 1)

Convergence is exponentially fast – but usually impossible to calculate λ_2
How do you know you have a sample? Apply set of tests.

(4) How to find a kernel $K(\underline{x}|\underline{x}')$?

Two important kernels that obey detailed balance.

Gibbs sampler:

$$K(\underline{x}|\underline{x}') = \sum_r p(r)\, K_r(\underline{x}|\underline{x}')$$

$p(r)$ is a dist. on graph nodes (or groups of nodes)

$$K(\underline{x}|\underline{x}') = p(x_r | \underline{x}'_{N(r)})\, \delta_{\underline{x}'_{N(r)}, \underline{x}_{N(r)}}$$

$N(r)$ all the nodes except for r.

conditional distribution

Note: conditional distribution is independent of the normalization constant Z of the distribution:

$$p(\underline{x}) = \frac{1}{Z} e^{-\sum_{ij}\psi_{ij}(x_i,x_j) - \sum_i \phi_i(x_i)}$$

- The Z is hard to compute in general and is one reason why "direct sampling" is hard & MCMC required.
- $p(x_i | \underline{x}_{\setminus i}) = \dfrac{e^{-\sum_j \psi_{ij}(x_i,x_j) - \phi_i(x_i)}}{\sum_{x_i} e^{-\sum_j \psi_{ij}(x_i,x_j) - \phi_i(x_i)}}$ usually easy to do.

Example → simple MRF, with $x_i \in \{0,1\}$

$\psi_{ij}(x_i,x_j) = \sum_{ij} \psi_{ij} x_i x_j$

$\phi_i(x_i) = \sum_i x_i \phi_i$

Gibbs: choose node i at random, choose x_i by sampling from

$$p(x_i | \underline{x}_{\setminus i}) = \frac{1}{1 + e^{(x_i (\sum_j \psi_{ij} x_j + \phi_i))}}$$

Note: Gibbs sampling can be approximated by MFT. (DIA)

compute $b_i^{t+1} = \dfrac{1}{1 + \exp\{2\sum_j \psi_{ij} b_j^t + \phi_i\}}$

also Gibbs sampling can be approximated by Belief Propagation (Rosen-Zvi, Jordan, Yuille 2005)

Gibbs is easy to implement.

Probably not the fastest MCMC (judged by empirical comparisons)

Metropolis Hastings:

$$K(\underline{x}|\underline{x}') = q(\underline{x}|\underline{x}') \min\left\{1, \frac{p(x)\, q(x'|x)}{p(x')\, q(x|x')}\right\}$$

$q(\underline{x}|\underline{x}')$ proposal probability for $\underline{x} \neq \underline{x}'$

(5) M-H has two steps:

(1) sample from proposal $q(\underline{x}|\underline{x}')$

(2) accept or reject proposal with prob $\min\left\{1, \frac{p(x)q(x'|x)}{p(x')q(x|x')}\right\}$ $x \neq x'$

Note: (1) $q(\underline{x}|\underline{x}')$ can be anything (provided it enables you to get to all parts of the space)

(2) M-H only depends on the ratio $p(x)/p(x')$ which eliminates the normalization term!

Note: Gibbs sampler is an M-H for which the proposals are always accepted

Connection to steepest descent.

→ suppose $q(\underline{x}'|\underline{x}) = q(\underline{x}|\underline{x}') = 1$

$$p(\underline{x}) = \frac{1}{Z} e^{-E(\underline{x})}$$

then proposal $\underline{x}' \to \underline{x}$ is accepted with certainty, if $E(\underline{x}) \leq E(\underline{x}')$ and with prob. $e^{E(\underline{x}')-E(\underline{x})}$, if $E(\underline{x}) > E(\underline{x}')$

→ so prob. that M-H can go 'uphill' and escape local minima.

Note: MCMC does not converge to a fixed state. Only to a probability distribution.

Note: add temperature (annealing)

$E(\underline{x}) \to E(\underline{x})/T$

can go uphill more easily if T is large

" " with difficulty if T is small.

Note: the convergence rate of an Metropolis-Hastings depends on the proposal probability $q(\underline{x}|\underline{x}')$.

→ how to choose $q(\underline{x}|\underline{x}')$? This requires knowledge about the problem domain.

Example in vision → data-driven MCMC.

(Tu & Zhu, PAMI 2002)

Extensions: • suppose $\{K_\mu(\underline{x}|\underline{x}') : \mu \in \Lambda\}$ are transition kernels that obey detailed balance,

then $\sum_{\mu \in \Lambda} \alpha_\mu K_\mu(\underline{x}|\underline{x}')$, $\alpha_\mu \geq 0$, $\sum_\mu \alpha_\mu > 1$ obeys detailed balance.

so, all these kernels can be used.

- kernels can be used to create and destroy nodes
- kernels can be used to group regions
- kernels can do anything

(6) Max-Flow / Min Cut. CS Algorithms

Check: Cormen, Leiserson, Rivest.

Max-Flow: Source-to-Sink.

(wiki)

Graph $G=(V,E)$ (vertices, edges) source & sink.

edges $(u,v)\in E$ have capacity $c(u,v)\geq 0$

Flow $f: V\times V\to \mathbb{R}$

(1) capacity: $\forall u,v\in V,\quad f(u,v)\leq c(u,v)$

(2) skew-symmetry: $\forall u,v\in V,\quad f(u,v)=-f(v,u)$

(3) conservation: $\forall u\in V\setminus\{s,t\},\quad \sum_{v\in V} f(u,v)=0.$

(water flowing down pipes (some pipes allow backward flow))

Total Flow $|f|=\sum_{v\in V} f(s,v)$

Algorithms: e.g. Ford-Fulkerson enable us to compute the maximum flow.

Theorem: the maximal flow corresponds to the min-cut.

cut (S,T) partition node V into $V=S\cup T$ with $S\cap T=\emptyset$, $s\in S$, $t\in T$.

net flow across cut is $f(S,T)$ $\sum_{u\in S,\, v\in T} f(u,v)$

Theorem: Certain classes of binary energy minimization problems can be converted into min-cut \ max-flow — hence solved in polynomial time.

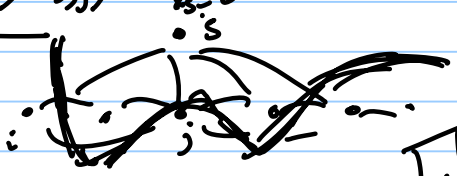
$$E(\underline{x})=\sum_{i,j}\psi_{ij}x_i x_j+\sum_i \phi_i x_i \qquad x_i\in\{0,1\}$$

convert to energy function that depends only on terms for which x_i & x_j take different values

e.g. $x_i x_j \to -x_i(1-x_j)+x_i$.

source node $x_s=0$, sink node $x_t=1$.

(required condition $\psi_{ij}\leq 0, \forall i,j$)



Many variations and extensions.

An assignment of variables $x_i\to 0,1$ puts the graph nodes into two sets $S=\{i: x_i=0\}$, $T=\{i: x_i=1\}$

The energy contributions only occur if at edges $i\circ\!-\!\circ j$ s.t. $x_i\neq x_j$, or $s\circ\!-\!\circ i$ s.t. $x_i\neq 0$ or $i\circ\!-\!\circ t$ s.t. $x_i\neq 1$.

The min cut minimizes the energy.