

Monte-Carlo methods: an introduction

A/Prof Richard Yi Da Xu

Yida.Xu@uts.edu.au

Wechat: aubedata

<https://github.com/roboticcam/machine-learning-notes>

University of Technology Sydney (UTS)

February 18, 2018

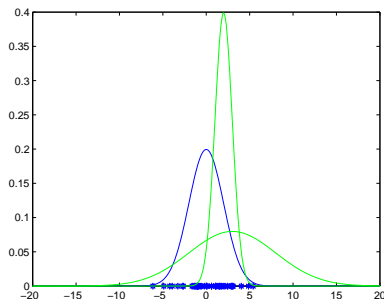
Point estimations

- ▶ Most machine learning students are used to “**point estimators**”, i.e., $\arg \max_{\theta} (f_{\theta}(X))$
- ▶ In estimating parameters of a distribution: Maximum Likelihood Estimation (MLE), Maximum A Posterior (MAP),
- ▶ In situations where we cant solve $\arg \max_{\theta} (f_{\theta}(X))$ analytically, we resort to some iterative methods, for examples, Expectation-Maximisation (EM)

MLE Example

Normal distributed data

- ▶ You believe data is Normal distributed:



Maximum Likelihood Estimation

- ▶ which “normal” distribution parameter $\theta = (\mu, \sigma)$ is more likely?

$$\begin{aligned}\theta^{\text{MLE}} &= \arg \max_{\theta} (\log[p(X|\theta)]) \\ &= \arg \max_{\theta} \left(\sum_{i=1}^N \log[\mathcal{N}(x_i; \mu, \sigma)] \right)\end{aligned}$$

- ▶ How to solve “argmax”? Well easy, take the derivative and let it equal zero. Works in the Gaussian case.

MAP Example

- ▶ What if I have some prior knowledge of μ , for example, $\mu \sim \mathcal{N}(\mu_0, \sigma_0)$. This type of estimation is called Maximum a Posterior (MAP):

$$\theta^{\text{MAP}} = \arg \max_{\theta} (\log[p(X|\theta)p(\theta)])$$

Say what you need is to find the mean, i.e.,

$$\mu^{\text{MAP}} = \arg \max_{\mu} \left(\sum_{i=1}^N \log[\mathcal{N}(x_i|\mu, \sigma)\mathcal{N}(\mu; \mu_0, \sigma_0)] \right)$$

- ▶ How to solve “argmax”? Well easy, take the derivative and let it equal zero. Works in the Gaussian case.

MAP Example Conti.

- ▶ Same trick applies: take the derivative with respect of μ and let it equal zero
- ▶ If you write out the expression for Gaussian fully, you will get:

$$\mu^{\text{MAP}} = \frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2} \left(\frac{1}{n} \sum_{j=1}^n x_i \right) + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \mu_0$$

- ▶ see what happens if $\sigma_0 \rightarrow \infty$

Expectation Maximization

- ▶ If lucky, can find $\arg \max_{\theta} \log[p(X|\theta)p(\theta)]$, i.e., take the derivative and let it equal zero analytically
- ▶ In many cases, we have to use some numerical methods, such as Expectation-Maximization (EM)
<http://www-staff.it.uts.edu.au/~ydxu/stat/incremental.pdf>
- ▶ Given an initial parameter θ^1 , we obtain a set of parameter estimate $\{\theta^1, \dots, \theta^g, \theta^{g+1}, \dots\}$, such that:

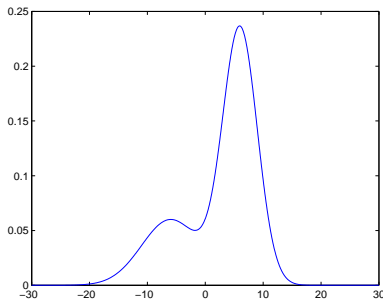
$$\log[p(X|\theta^{g+1})p(\theta^{g+1})] \geq \log[p(X|\theta^g)p(\theta^g)]$$

Two examples

- ▶ Gaussian Mixture model: $p(x|\theta) = \sum_{l=1}^M \mathcal{N} w_l(x; \mu_l, \sigma_l)$
- ▶ An example of my research: (Xu & Kemp, 2010 & 2013)
- ▶ both are solved using **expectation maximization**

The moral of the story

- ▶ Doesn't matter how sophisticated they are, these algorithms are point estimators, as they simply give you a “best” **single** θ .
- ▶ In many machine learning problems, you are actually interested in the posterior distribution $p(\theta|\text{Data}) \propto p(\text{data}|\theta)p(\theta)$
- ▶ Ok, let's look at an example:



A simple “almost real” posterior inference problem

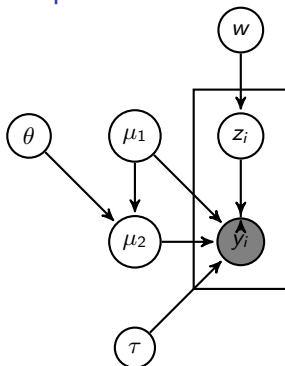
- ▶ A simple problem where data points $Y = \{y_1, \dots, y_N\}$ are distributed from a bi-modal Gaussian Mixture Model
- ▶ The two gaussians have their means at μ_1 and μ_2 , separately by a distance θ
- ▶ Both Gaussians have the identical precision τ
- ▶ First Gaussian has a weight w_1 , and the second has w_2
- ▶ We also assume the latent variable $z_i \in \{1, 2\}$, indicating which Gaussian has generated y_i
- ▶ The Generative model and its Graphical model is shown in the next page

A simple “almost real” posterior inference problem

Generative model

$$\begin{aligned}w &\sim \text{Dir}(\alpha, \alpha) \\ \tau &\sim \text{Gamma}(a, b) \\ \theta &\sim \mathcal{N}(0, \sigma_\theta^2) \\ \mu_1 &\sim \mathcal{N}(0, \sigma_\lambda^2) \\ z_i | w &\sim \text{Mult}(w) \\ \mu_2 &\sim \mathbf{1}(\mu_1 + \theta) \\ y_i | \mu_{z_i}, \tau &\sim \mathcal{N}(\mu_{z_i}, 1/\tau)\end{aligned}$$

Graphical model



- ▶ What you are hoping to get is of course $p(z_i, w, \mu_1, \mu_2, \tau, \theta | \{y_1, \dots, y_N\})$
- ▶ **Exercise** Write down the posterior densities for each of the variables, using appropriate **conditional independence** depicted in the Graphic model.

Any easy way out? To use black-box sampler

- ▶ What if I don't want to write my own sampling code?
- ▶ The good news is that you don't have to. You can simply use WINBUGS.
- ▶ <http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml>
- ▶ Or, modern-day STAN: <http://mc-stan.org>
- ▶ To use **WINBUGS**, the previous example can be coded into:

```
model eyes;
const
  N=48;
var
  y[N],
  z[N],
  mu[2],
  theta,
  tau,
  sigma,
  w[2],
  alpha[2];
data y in "eyes.dat";
```

```
inits in "eyes.in";
{for (i in 1:N){
  y[i] ~ dnorm(mu[z[i]],tau);
  T[i] ~ dcat(P[])}}
sigma <- 1/sqrt(tau);
tau ~ dgamma(0.01,0.01);
mu[1] ~ dnorm(0,1.0E-6);
mu[2] <- mu[1]+theta;
theta ~ dnorm(0,1.0E-6) I (0,);
w[] ~ ddirch(alpha[]);
alpha[1] <- 1;
alpha[2] <- 1;}
```

Can I leave now?

No :)

- ▶ Can't just use WINBUGS or STAN for all models; They are black-box sampler.
- ▶ In many scenarios, you need to develop your own efficient sampling
- ▶ ...

Ok, let's start sampling!

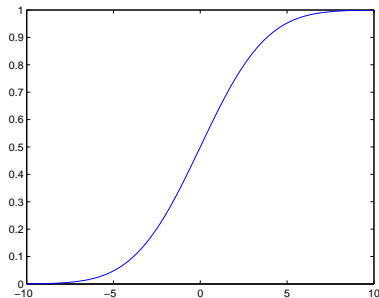
Any other methods for for posterior inference

Other methods also exist for posterior inference:

- ▶ Variational Bayes - good starting point: chapter 10 of Bishop's textbook, and/or **my notes**
- ▶ Laplace approximation
- ▶ ...

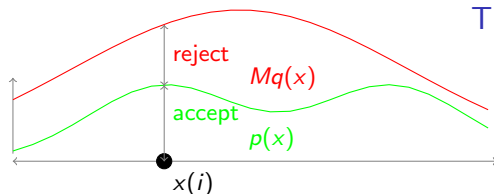
Simplist method: inverse of CDF

Simplist sampling method: sample the inverse of CDF!



$$u \sim U(0, 1) \quad x = CDF^{-1}(u)$$

Rejection Sampling



- ▶ Sampling is all about efficiency
- ▶ Rejection sampling can give you quite low acceptance ratio, should you choose a non-compatible $q(\cdot)$

The algorithm

```
i = 0
while i  $\neq$  N
  x(i)  $\sim$  q(x) and u  $\sim$  U(0, 1)
  if u <  $\frac{p(x(i))}{Mq(x(i))}$  then
    accept x(i)
    i = i + 1
  else
    reject x(i)
  end
end
```

Adaptive Rejection Sampling

Sometimes, rejection sampling can be made more efficient. One example is when $p(x)$ is log-concave. For example, in Dirichlet Process, the concentration factor α has probability:

$$p(\alpha|k, n) \propto \frac{\alpha^{k-3/2} \exp(-1/(2\alpha)) \Gamma(\alpha)}{\Gamma(n + \alpha)}$$

where $p(\cdot)$ is log-concave in terms of $\log(\alpha)$

Homework prove the above is in fact log-concave in terms of $\log(\alpha)$

- ▶ Let $\{x_i, \dots, x_k\}$ be the k starting points.
- ▶ Calculate $u_k(x)$, the piece-wise linear upper bound formed from the tangents to $h(x)$ at each point x_i
- ▶ $s_k(x) = \frac{\exp(u_k(x))}{\int \exp(u_k(x')) dx'}$
- ▶ $z_j = \frac{h(x_{j+1}) - h(x_j) - x_{j+1}h'(x_{j+1}) + x_jh'(x_j)}{h'(x_j) - h'(x_{j+1})}$
- ▶ Piece-wise upper bound:
 $u_k(x) = h(x_j) + (x - x_j)h'(x_j)$
for $x \in [z_{j1}, z_j]$ and $j = 1, \dots, k$

The Sampling steps

- ▶ Sample $x^* \sim s_k(x)$ and $u^* \sim U(0, 1)$.
- ▶ **If** $u^* \leq \exp\{h(x^*)u_k(x^*)\}$ **then** accept x^* , otherwise reject x^* .
- ▶ Include x^* in the list, so it has $K + 1$ elements, and rearrange in ascending order and reconstruct functions $u_{k+1}(x), s_{k+1}(x)$

- ▶ Watch demo of sampling a Gaussian distribution (no need to use ARS, but ok for demo)
- ▶ **Homework** Find some other log-concave distributions
- ▶ What happens to distribution in which it is NOT log-concave?
- ▶ It may be break up into piece-wise, concave/convex:
Further readings: Grr, Dilan, and Yee Whye Teh. "Concave-convex adaptive rejection sampling." Journal of Computational and Graphical Statistics 20.3 (2011): 670-691

Further improve ARS efficiency

- ▶ The algorithm you saw in the MATLAB demo require the computation of a new envelope each time.
- ▶ Is it really necessary after the envelop becomes “pretty good”?
- ▶ **Exercise** What is the most computational step in the envelope computation?
- ▶ **Exercise** Can you accept samples without recompute the envelope?

Importance Sampling

Say, for example, the aim for this task is to compute the integral:

$$\begin{aligned} \mathbb{E}_{p(z)}[f(z)] &= \int f(z)p(z)dz \\ &= \int \underbrace{f(z)\frac{p(z)}{q(z)}}_{\text{new } \tilde{f}(z)} q(z)dz \\ &\approx \frac{1}{N} \sum_{n=1}^N f(z^n) \frac{p(z^n)}{q(z^n)} \end{aligned}$$

\tilde{p} is the un-normalized pdf, i.e., $p(z) = \frac{\tilde{p}}{Z}$