Seminar on Advanced Topics in Statistical Learning

Mou Minghao

SSE, CUHK(SZ)

July 26, 2022

Outline

- EM Algorithm
- K-Means Clustering
- Gaussian Mixed Model
- Hidden Markov Model
- Kalman Filter
- Particle Filter
- Bayesian Inference for Latent Hawkes Process
- Applications
- References

Expectation Maximization

Expectation Maximization Algorithm is an iterative algorithm which is used to calculate the maximum likelihood estimation or maximum a posterior in parametric probabilistic models with hidden variables.
It can be decomposed into two steps

1. Expectation: estimate the parameters based on the data and the model then calculate the expectation
2. Maximization: find the parameter which maximizes the likelihood function

EM Algorithm

Given the data $\mathcal{X}$, assume independence of samples, we want to fit the model $p(x;\theta)$, the log likelihood function is given as

$$\begin{aligned}
L(\theta) := & \sum_{i=1}^{n} \log p(x_i;\theta) \\
= & \sum_{i=1}^{n} \log \sum_{z} p(x_i, z;\theta) \\
= & \sum_{i=1}^{n} \log \sum_{z} Q_i(z) \frac{p(x_i, z;\theta)}{Q_i(z)} \\
\geq & \sum_{i=1}^{n} \sum_{z} Q_i(z) \log \frac{p(x_i, z;\theta)}{Q_i(z)} \\
:= & J(z, Q;\theta)
\end{aligned} \tag{1}$$

EM Algorithm

In the last inequality, we used Jensen's inequality.
Since the objective is to maximize the log likelihood and we notice that the last quantity in (1) gives a lower bound to the log likelihood. Hence, we try to maximize the lower bound.
In Jensen's inequality, the equality holds iff $X = \mathbb{E}(X)$, consider

$$\frac{p(x_i, z; \theta)}{Q_i(z)} = c \tag{2}$$

Sum both sides of 2 over z and use $\sum_z Q_i(z) = 1$, we have

$$\sum_z p(x_i, z; \theta) = c \tag{3}$$

Thus,

$$Q_i(z) = \frac{p(x_i, z; \theta)}{\sum_z p(x_i, z, \theta)} = p(z|x_i; \theta) \tag{4}$$

which is actually the *posterior distribution* of z.

EM Algorithm

Algorithm (Expectation Maximization)

```
Input: $\theta \leftarrow \theta_0$
For $j = 1, 2, ..., N$, do
  1. Expectation Step:
     1.1 $Q_i(z) \leftarrow p(z|x_i;\theta), \forall i$
     1.2 compute $J(z, Q;\theta) = \sum_{i=1}^{n} \sum_{z} Q_i(z) \frac{p(x_i, z;\theta)}{Q_i(z)}$
  2. Maximization Step:
     2.1 $\theta \leftarrow arg \max_{\theta} J(z, Q;\theta)$
```

Remark

Idea: *first fix θ, tune $Q(z)$ so that the lower bound $J(z, Q;\theta)$ equals $L(\theta)$. Fix $Q(z)$, find the θ which maximizes $J(z, Q;\theta) = L(\theta)$. Repeat this procedure until θ converges (i.e. $\|\theta_{i+1} - \theta_i\| < \epsilon$)*

Latent Hawkes Process[1]

While Poisson processes are foundational models for spatiotemporal data, many real-world systems violate the assumption of independent intervals. Hawkes processes remedy this shortcoming by allowing events to influence the future rate.

Definition (Conditional Intensity Function)

$$\lambda_v(t, y|\mathcal{H}_t, \theta) = b_v(t, y; \theta) + \sum_{n=1}^{N} f_{v_n \to v}(t, y; t_n, y_n; \theta)\mathbb{I}[t > t_n] \quad (5)$$

Given this conditional intensity function, the log likelihood of a set of events decomposes into two terms: the negative integrated rate and the sum of instantaneous log rates,

[1] The content is from [Linderman et al., 2017]

Latent Hawkes Process

$$\log p(\{v_n, t_n, y_n\} \, |\theta) \\ = -\sum_{v=1}^{V} \int_0^T \int_{\mathcal{Y}} \lambda_v(t, y|\mathcal{H}_t, \theta) dtdy + \sum_{n=1}^{N} \log \lambda_{v_n}(t_n, y_n|\mathcal{H}_{t_n}, \theta) \tag{6}$$

in the partially-observed case, we must perform joint inference of both the model parameters and the latent data.

Learning and inference in latent Hawkes processes is fundamentally a latent variable problem. As such, we start with an *expectation-maximization* algorithm that alternates between

1. **Inference**: computing expected log likelihoods

$$\mathcal{L}(\theta) := \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\theta_{curr})}[\log p(\boldsymbol{z}, \boldsymbol{x}|\theta)]$$

2. **Learning**: taking gradients with respect to the model parameters $\nabla_\theta \mathcal{L}$

Data-driven Sequential Monte Carlo

The unique challenge of latent Hawkes processes is that the latent variables take the form of a marked point process. Specifically, since the number of latent events is undetermined, we propose a variety of methods for performing inference over sets of unknown cardinality.

It leverages the autoregressive nature of Hawkes processes (the instantaneous rate is only a function of preceding events) to sequentially propose and resample particles.

Data-driven Sequential Monte Carlo

1. define a scaffold $\{s_i\}_{i=1}^{I}$ that partitions the time range $[0, T]$ into I disjoint intervals
2. the value of the p-th particle in the i-th interval $z_i^{(p)} := \left\{(v_n^{(p)}, t_n^{(p)}, y_n^{(p)}) : s_{i-1} < t_n^{(p)} \leq s_i\right\}$ is a set of latent events. We only propose latent events for vertices whose data is missing in that interval
3. We generate a candidate set of latent events for interval i by sampling a proposal distribution $z_i^{(p)} \sim r(z_i | x_{1:i}, z_{1:i-1}^{(p)}, \theta)$, and weighting the newly updated particles with the function,

$$w(z_{1:i}^{(p)}) = \frac{p(x_{1:i}, z_{1:i}^{(p)} | \theta)}{p(x_{1:i-1}, z_{1:i-1}^{(p)} | \theta) r(z_i^{(p)} | x_{1:i}, z_{1:i-1}^{(p)}, \theta)} \tag{7}$$

the high dimensionality of the marks calls for delicate choices of the proposal distribution to control the variance of the SMC estimates. To this end, we utilize data-driven proposals, leveraging our intuition that latent marks are often similar to observed marks.

Rao-Blackwellized Sequential Monte Carlo

note that in the un-marked case, inference of the latent times is relatively simple, and standard SMC works well. This motivates a Rao-Blackwellized approach, in which we marginalize the latent y_n and infer only the timestamps t_n and vertices v_n. The weights are then given by $w(\tilde{z}_{1:i}^{(p)})$, where $\tilde{z}_{1:i}^{(p)}$ denotes the particles without marks

the weights now need the marginal likelihood $p(x_{1:i}, \tilde{z}_{1:i}^{(p)}, \theta)$, we have

$$p(x_{1:i}, \tilde{z}_{1:i}^{(p)} | \theta) = \frac{p(x_{1:i}, z_{1:i}^{(p)}, \theta)}{p(y_{1:i}^{(p)} | x_{1:i}, \tilde{z}_{1:i}, \theta)} \tag{8}$$

However, we do not know $p(y_{1:i}^{(p)} | x_{1:i}, \tilde{z}_{1:i}, \theta)$. Thus, we perform VI on it. Specifically, we optimize a parametric variational distribution $q(y_{1:i}^{(p)}; \eta) \approx p(y_{1:i}^{(p)} | x_{1:i}, \tilde{z}_{1:i}, \theta)$, its ELBO is

$$ELBO(p, q) = \mathbb{E}_q[\log p(x_{1:i}, z_{1:i}^{(p)}, \theta)] - \mathbb{H}[q(y_{1:i}^{(p)}; \eta)] \tag{9}$$

Rao-Blackwellized Sequential Monte Carlo

We just maximize (9), it can be achieved by applying Coordinate Ascent or Gradient Ascent algorithms.

Remark

This approximation biases our SMC estimates, but the Rao-Blackwellization should reduce its variance. In other words, we trade bias in variational approximation for lower variance due to Rao-Blackwellization.

Data-driven Sequential Monte Carlo: Algorithm Summary

Algorithm (Variational Inference)

```
VariationalInference(p,q)
Input: target $p(y_{1:i}|x_{1:i}, \tilde{z}_{1:i}, \theta)$, variational family $q(y_{1:i}; \eta)$
Output: $q(y_{1:i}; \eta^*)$
$\eta^* = arg \max_\eta ELBO(p, q)$ as given in (9)
End
```

Data-driven Sequential Monte Carlo: Algorithm Summary

Algorithm (Data-driven Rao-Blackwellized SMC)

```
Input: model $p(x_{1:i}, z_{1:i}, \theta)$, variational family $q(y_{1:i}; \eta)$, proposal distribution $r(z_i|x_{1:i}, z_{1:i-1}^{(p)}, \theta)$, number of partitions $I$, number of particles $M$
Output: variation parameter $\eta^*$, model parameter $\theta^*$
$q(y_{1:i}, \eta^*) \leftarrow \textsc{VarationalInference}(p, q)$
For $p = 1, 2, ..., M$ do
  1. sample $z_1^{(p)} \sim r(z_1|x_1, \theta)$, $w(\tilde{z}_1^{(p)}) \leftarrow \frac{p(x_1, z_1^{(p)}, \theta)}{q(y_1^{(p)}; \eta^*) r(\tilde{z}_1^{(p)}|x_1, \theta)}$
For $i = 2, 3, ..., I$ do
    For $p = 1, 2, ..., M$ do
  1. sample $z_i^{(p)} \sim r(z_i|x_{1:i}, z_{1:i-1}^{(p)}, \theta)$
  2. $w(\tilde{z}_{1:i}^{(p)}) \leftarrow \frac{p(x_{1:i}, z_{1:i}^{(p)}|\theta) q(y_{1:i-1}^{(p)}; \eta^*)}{p(x_{1:i-1}, z_{1:i-1}^{(p)}|\theta) q(y_{1:i}^{(p)}; \eta^*) r(z_i^{(p)}|x_{1:i}, \tilde{z}_{1:i-1}^{(p)}, \theta)}$
Base on the samples $\tilde{z}_{1:i}^{(p)}$, we can approximate $p(\tilde{z}_{1:i}|y_{1:i}, x_{1:i}, \theta)$
```

Data-driven Sequential Monte Carlo: Algorithm Summary

Algorithm (Data-driven Rao-Blackwellized SMC, **ctd.**)

```
$\theta^* \leftarrow \textsc{ExpectationMaximization}(p)$
End
```

Algorithm (EM Algorithm)

```
$\textsc{ExpectationMaximization(p)}$
Input: latent variable posterior $p(\tilde{z}_{1:i}|y_{1:i}, x_{1:i}, \theta)$
Output: $\theta^*$
$\theta_0 \leftarrow 0$
For $i = 1, 2, ..., N$ do
    $\mathcal{L}(\theta) \leftarrow \mathbb{E}_{p(\tilde{z}_{1:i}|y_{1:i}, x_{1:i}, \theta_{i-1})}[\log p(x_{1:i}, z_{1:i}, \theta)]$
    $\theta_i \leftarrow arg\ \max_\theta \mathcal{L}(\theta)$
    If $\theta$ converges do
        Output $\theta_i$
End
```

References I

Linderman, S. W., Wang, Y., and Blei, D. M. (2017).
Bayesian inference for latent hawkes processes.
Advances in Neural Information Processing Systems.