APMLA: Lecture 6

Caterina De Bacco and Isabel Valera

1 Introduction

In previous lectures, we studied probabilistic approaches for clustering data, where the data were assumed to be i.i.d. samples of a mixture distribution, such as a Gaussian or a categorical mixture. In addition, in the last lecture, we introduced a probabilistic framework, known as temporal point processes, to model the time when an action/event occurs as a random variable. There, we discussed the difficulties of designing a valid and intuitive pdf for the event time, and show that these challenges are overcame by using the intensity function instead.

Next, we will combine both ideas to derive (and infer the parameters of) a generative model, which we will name as "CRP+HP", where the data are discrete events localized in continuous time characterized by a mark that is, in turn, sampled from a mixture distribution. This model allows us to cluster documents (posts, news, etc.) that are generated over time in terms of topics, using both the content and the temporal information in the data. Among other applications, we will see how this model is thus useful to model the interest and dynamics of the different users in online platforms such a Twitter and Stack Overflow, among others. The CRP+HP model was introduced in [2] and referred as Hierachical Dirichlet Hawkes process (HDHP). This model was initially proposed to capture the dynamics of online learning in Stack Overflow.

2 The Chinese Restaurant Franchise Process (CRF)

The first step towards our goal is to group the documents generated by each user, so that we can capture the different interests and behaviors observed between users. To do so, we first introduce a hierarchical extension of the Dirichlet process, called the *hierarchical Dirichlet process* HDP. The HDP has been useful for clustering grouped data [3], since it allows for an unbounded number of clusters whose parameters are shared across groups. It has been broadly applied for topic modeling as the nonparametric counterpart of the Latent Dirichlet Allocation (LDA). In our example, a group of documents corresaponds to all the documents (e.g. Tweets) generated by each user.

More in detail, the HDP defines a hierarchy of DPs, in which a set of random probability measures $G_j \sim DP(\alpha_1, G_0)$ (one for each group of data) are distributed as DPs with concentration parameter α_1 and base distribution G_0. The latter is also distributed as a DP, i.e., $G_0 \sim DP(\alpha_0, H)$. In the HDP, all the distributions G_j share the same support as G_0, and are conditionally independent given G_0.

Analogously to the CRP construction for a DP, the count-based generative construction of an HDP is the Chinese Restaurant Franchise Process (CRF). The CRF assumes a franchise with as many restaurants as the groups of data (e.g., the number of users), where all of the restaurants share the same menu with an infinite number of dishes (or clusters). In particular, one can obtain samples from the HDP as follows:

1. Initialize the total number of dishes $L = 0$, and the total number of tables in each restaurant $K_r = 0$, for $r = 1, \ldots, R$, with R being the total number of restaurants in the franchise.

2. For each restaurant $r = 1, \ldots, R$, and for all customers in each restaurant $i = 1, \ldots, N_r$ (being N_r the total number of customers entering restaurant r):
   - Sample the table for the i-th customer in restaurant r from a multinomial distribution with probabilities:

$$\begin{array}{lcll} p(z_{ri} = k) & = & \frac{n_{rk}}{\beta_1+i-1}, & \text{for } \; k = 1, \ldots, K_r \\ p(z_{ri} = K_r + 1) & = & \frac{\beta_1}{\beta_1+i-1}, & \text{for opening anew table} \end{array} \tag{1}$$

     where $n_{rk} = \sum_{j=1}^{i-1}[z_{rj} = k]$ is the number of customers already seated at the k-th table of the r-th restaurant.
   - If $z_{ri} = K_r + 1$, i.e., the i-th customer sits at a new table, increase the number of tables in the r-th restaurant as $K_r = K_r + 1$ and sample the dish for the new table from a multinomial distribution with probabilities:

$$\begin{array}{lcll} p(\phi_{r(K_r+1)} = \varphi_\ell) & = & \frac{m_\ell}{K+\beta_0} & \text{for } \ell = 1, \ldots, L \\ p(\phi_{r(K_r+1)} = \varphi_{L+1}) & = & \frac{\beta_0}{K+\beta_0} & \end{array} \tag{2}$$

     where $K = \sum_{j=1}^{R} K_j$ is the total number of tables in the franchise, $m_\ell = \sum_{j=1}^{R}\sum_{k=1}^{K_j}[\phi_{jk} = \varphi_\ell]$ is the total number of tables serving dish ϕ_ℓ in the franchise, and $\varphi_{L+1} \sim H(\varphi)$ represents a new dish, i.e., the parameters of the new cluster.
   - If $\phi_{rK_r} = \varphi_{L+1}$, i.e., the new table is assigned to a new dish/cluster, increase also the total number of clusters in the franchise $L = L + 1$.

Similarly as for the CRP, although in the process above we have generated the data (customers) for each group (restaurant) sequentially, due to the exchangeability properties of the CRF, the resulting distribution of the data is invariant to the order at which customers are assumed to enter any of the restaurants [3].

3 Multivariate Marked Hawkes process

The second building block of our target model, the CRP+HP model, is the multivariate marked Hawkes process (HP) and accounts for the fact that every user generates data at particular timestamps. For example, every Tweet by a user is characterized by the user that post the tweet, the timestamp of the tweet and the content of the Tweet (the tweet itself). As a consequence, every Tweet (or in general, document) generated by a user may be represented as an event $e := (u, t, \mathbf{x})$, where u is the user, t is the event time, and $\mathbf{x}$ is the mark (in our case the "bag of words") characterizing the event.

The multivariate marked Hawkes process allows us to jointly model the events generated by a group of users, $u \in \{1, \ldots, U\}$. To this end, a sample of a multivariate Hawkes process does not only result in the event time but also in the user performing the event, i.e.,

$$(t, u) \sim Hawkes \begin{pmatrix} \mu_1 + \sum\limits_{i:t_i<t} \kappa_\alpha(t, t_i) \\ \vdots \\ \mu_U + \sum\limits_{i:t_i<t} \kappa_\alpha(t, t_i) \end{pmatrix}, \tag{3}$$

where we are assuming that each user has his own base intensity μ_u and the triggering kernel $\kappa_\alpha(t_i, t_j)$ is parametrized by α for all the users. Note also here that users may cross-excite other users intensities by taking into account the history of events performed by all (or a subset of "connected") the users in the system.

In order to get samples from a multivariate HP, we can use the superposition property of the HP, and first sample the event time from a univariate HP with intensity equal to the sum of the individual intensities of all users, and then sample the user performing the event from a categorical distribution with probabilities proportional to the intensity of each user evaluated at the event time.

4 Clustering event data: CRP + Hawkes process

In the following, we introduce the CRP+HP model, which we will use to cluster event data. The intuition behind the CRP+HP model is ti assume that every observation is a marked event, where the event time is sampled from a HP and the event mark is sampled from an iMM. Importantly, since we would like to jointly model the activity of several users, which are expected to perform similar (in terms of both dynamics and topics) sequences of actions, the (temporal and mark) parameters characterizing every cluster/topic should be shared across users.

In order to account for the clustering of events, we now extend our event representation to also include a latent variable z corresponding to the cluster/topic of the event, which will determine the component in the mixture that the mark $\mathbf{x}$ is sampled from. That is, $e := (u, t, z, \mathbf{x})$, where z is a latent variable corresponding to the cluster (topic) assignment, and the time t, the user u and the mark $\mathbf{x}$ of the event are all observed variables. Since as in the iMM model, we would like to assume *a priori* an infinite number of clusters, we next build a generative process, which resembles the CRF process, to generate samples of a CRP+HP model.

Generative Process If we assume a set of users U and vocabulary W for the content, we can generate N events as follows:

1. Initialize the *total number of tables* $K = 0$ and the *total number of clusters* $L = 0$.
2. For $n = 1, \dots, N$:
   (a) Sample the time t_n and user $u_n \in U$ for the new event, such that $t_n > t_{n-1}$, as

$$(t_n, u_n) \sim Hawkes \begin{pmatrix} \mu_1 + \sum_{i=1}^{n-1} \kappa_{z_i}(t_n, t_i)[u_i = 1] \\ \vdots \\ \mu_U + \sum_{i=1}^{n-1} \kappa_{z_i}(t_n, t_i)[u_i = U] \end{pmatrix} \tag{4}$$

   (b) Sample the table z_n for the new event from a multinomial distribution with probabilities

$$\begin{aligned} \Pr(z_n = k | t_n, u_n, H(t_n), \{\alpha_\ell^{(p)}\}_{\ell=1}^L) &= \tfrac{\lambda_{u_n,k}^{endo}(t_n)}{\lambda_{u_n}^*(t_n)}, \quad \text{for } k = 1, \dots, K \\ \Pr(z_n = K+1) &= \tfrac{\mu_{u_n}}{\lambda_{u_n}^*(t_n)} \end{aligned} \tag{5}$$

   where $\lambda_{u_n,k}^{endo}(t_n) = \sum_{i=1}^{n-1} \kappa_{z_i}(t_n, t_i)[u_i = u_n, z_i = k]$ is the endogenous intensity of user u_n on table k, and $\lambda_{u_n}^*(t_n) = \mu_{u_n} + \sum_{i=1}^{n-1} \kappa_{z_i}(t_n, t_i)[u_i = u_n]$ is the total intensity of user u_n at time t_n.

   (c) If $z_n = K + 1$, assign the new table to a cluster with probability

$$\begin{aligned} \Pr(\phi_{K+1} = \varphi_\ell) &= \tfrac{m_\ell}{K+\beta}, \quad \text{for } \ell = 1, \dots, L \\ \Pr(\phi_{K+1} = \varphi_{L+1}) &= \tfrac{\beta}{K+\beta} \end{aligned} \tag{6}$$

   where $m_\ell = \sum_{k=1}^{K} [\phi_k = \varphi_\ell]$ is the number of tables assigned to cluster ℓ across all users, and $\varphi_{L+1} = \{\alpha_{L+1}, \boldsymbol{\theta}_{L+1}\}$ is the set of parameters of the new cluster $L+1$, which we sample from $\alpha_{L+1} \sim Gamma(\tau_1, \tau_2)$ and $\boldsymbol{\theta}_{L+1} \sim Dirichlet(\eta_0)$.

   Then, increase the number of tables $K = K + 1$ and, if $\phi_{K+1} = \varphi_{L+1}$, increase also the number of clusters $L = L + 1$.

(d) Sample each word in the content $\mathbf{x}_n$ from $x_{n,j} \sim Multinomial(\boldsymbol{\theta}_{z_n})$.

We remark that in the generative process above every cluster (topic) is characterized by the kernel parameters α_ℓ (modeling the dynamics of the cluster) and the mark distribution $\boldsymbol{\theta}_\ell$ (capturing word distribution in every topic).

5 Bayesian inference for the CRP+HP

5.1 Sequential Monte Carlo

Sequential Monte Carlo (SMC) methods are a set of Monte Carlo algorithms that allow for efficient approximation of probability distributions. e.g., posterior distributions using samples (particles). At the base of SMC algorithms, there are Sequential Importance Sampling (SIS) routines, which we review here. As for other Monte Carlo approaches, SIS is applicable when one cannot sample directly from a target distribution $\gamma(\theta_{1:n})$, over a sequence of n variables $z_{1:n}$, but instead one can only sample via an unnormalized related distribution. For a general discussion, assume we would like to perform posterior inference over the parameters $z_{1:n}$ associated to a sequence of observed random variables $x_{1:n}$, that is we are interested in approximating the posterior $\gamma(z_{1:n}) = p(z_{1:n}|x_{1:n})$. Here we assume a first order Markovian model, hence $\gamma(z_n|z_{1:n-1}) = \gamma(z_n|z_{n-1})$. Also, consider the transition probability density $p(z_n|z_{n-1})$ and observational (likelihood) model $p(x_n|z_n)$.

According to SIS, one can draw samples from the proposal distribution $q(z_{1:n}) = q(z_1)\prod_{i=2}^{n} q(z_i|z_{i-1})$, then weight the samples according to the ratio $\frac{\gamma(z_{1:n})}{q(z_{1:n})}$. After normalizing the weights, it is possible to approximate the posterior $p(z_{1:n}|x_{1:n})$ with a Monte Carlo estimate exploiting these weighted samples. In practice, in SIS, each sample is attributed to a *particle*, hence the alternative naming for this approach, particle filtering. At each sampling step the particles estimate the parameters and explore the parameter space. However, if we sample directly from $q(z_{1:n})$ the complexity grows in the order of n. Writing the proposal and weighting recursively allow us to sample sequentially, and thus, perform inference in an online fashion.

More specifically, we can write the true posterior is written as

$$p(z_{1:n}|x_{1:n}) = p(z_{1:n-1}|x_{1:n-1}) \times \frac{p(z_n|z_{n-1})p(x_n|z_n)}{p(x_n|x_{1:n-1})},$$

and thus,

$$p(z_n|z_{1:n-1}, x_{1:n}) = \frac{p(z_n|z_{n-1})p(x_n|z_n)}{p(x_n|z_{n-1})},$$

where $p(x_n|z_{n-1}) = \int p(z_n|z_{n-1})p(x_n|z_n)dz_n$. Then, the recursive formula for the weights for each particle can be obtained as

$$w_n(z_{1:n}) = \frac{\gamma(z_{1:n})}{q(z_{1:n})} = \frac{\gamma(z_{1:n-1})}{q(z_{1:n-1})} \times \frac{\gamma(z_n|z_{n-1})}{q(z_n|z_{n-1})} = w_{n-1}(z_{1:n-1}) \times \frac{\gamma(z_n|z_{n-1})}{q(z_n|z_{n-1})}, \tag{7}$$

where, for conciseness, we denote the posterior distribution $p(z_{1:n}|x_{1:n})$ by $\gamma(z_{1:n})$. For an optimal solution we must choose q of the same form of γ. Although this is not always possible, the proposed framework allows conditional posteriors to be tractable, therefore one can employ Gibbs sampling to approximate the true posterior as defined according to the generatove model. A common choice for the proposal distribution is $q(z_n|z_{n-1}) = p(z_n|x_n, z_{n-1})$.

This algorithm has the disadvantage that the accumulated Monte Carlo error (variance) increases exponentially with n. To diminish this effect, one may rely on a resampling step, in which particles are resampled proportional to their weights. This results in duplication of more weighted particles and elimination of those with low weights. Popular resampling schemes are compared in detail in [1].

5.2 SMC for the CRP+HP

Given a collection of n observed learning actions performed by the users of an online learning site during a time period $[0, T)$, our goal is to infer the learning patterns that these events belong to. To efficiently sample from the posterior distribution, we derive a sequential Monte Carlo (SMC) algorithm that exploits the temporal dependencies in the observed data to sequentially sample the latent variables associated with each learning action. In particular, the posterior distribution $p(z_{1:n}|t_{1:n}, u_{1:n}, q_{1:n}, \{\alpha_\ell^{(p)}\}_{\ell=1}^{L}, \{\mu_u^{(p)}\}_{u=1}^{U})$ is sequentially approximated with a set of $|P|$ particles, which are sampled from a proposal distribution that factorizes as

$$
\begin{aligned}
&q(z_{1:n}|t_{1:n}, u_{1:n}, \mathbf{x}_{1:n}, \{\alpha_\ell^{(p)}\}_{\ell=1}^{L}, \{\mu_u^{(p)}\}_{u=1}^{U}) \\
&= q(z_n|z_{1:n-1}, t_{1:n}, u_{1:n}, \mathbf{x}_{1:n}, \{\alpha_\ell^{(p)}\}_{\ell=1}^{L}, \{\mu_u^{(p)}\}_{u=1}^{U}) q(z_{1:n-1}|t_{1:n-1}, u_{1:n-1}, \mathbf{x}_{1:n-1}, \{\alpha_\ell^{(p)}\}_{\ell=1}^{L}, \{\mu_u^{(p)}\}_{u=1}^{U}),
\end{aligned}
$$

where

$$
\begin{aligned}
&q(z_n|\mathbf{x}_{1:n}, t_{1:n}, u_{1:n}, \{\alpha_\ell^{(p)}\}_{\ell=1}^{L}, \{\mu_u^{(p)}\}_{u=1}^{U}) \\
&= \frac{p(\mathbf{x}_n|\mathbf{x}_{1:n-1}, z_{1:n}) p(z_n|t_{1:n}, u_{1:n}, z_{1:n-1}, \{\alpha_\ell^{(p)}\}_{\ell=1}^{L}, \{\mu_u^{(p)}\}_{u=1}^{U})}{\sum_{(z_n)} p(\mathbf{x}_n|\mathbf{x}_{1:n-1}, z_{1:n}) p(z_n|t_{1:n}, u_{1:n}, z_{1:n-1}, \{\alpha_\ell^{(p)}\}_{\ell=1}^{L}, \{\mu_u^{(p)}\}_{u=1}^{U})}
\end{aligned}
\tag{8}
$$

In the above expression, we can exploit the conjugacy between the multinomial and the Dirichlet distributions to integrate out the word distributions $\boldsymbol{\theta}_\ell$ and obtain the marginal likelihood $p(\mathbf{x}_n|\mathbf{x}_{1:n-1}, z_{1:n})$

$$
p(\mathbf{x}_n|\mathbf{x}_{1:n-1}, z_{1:n}) = \frac{\Gamma\left(C_\ell^{n-1} + \eta_0|W|\right) \prod_{w \in W} \Gamma\left(C_{w,\ell}^{n} + \eta_0\right)}{\Gamma\left(C_\ell^{n} + \eta_0|W|\right) \prod_{w \in W} \Gamma\left(C_{w,\ell}^{n-1} + \eta_0\right)},
$$

where C_ℓ^{n-1} and C_ℓ^{n} are the number of words in the pieces of content (or queries) $\mathbf{x}_{1:n-1}$ and $\mathbf{x}_{1:n}$, respectively, and $C_{w,\ell}^{n-1}$ and $C_{w,\ell}^{n}$ count the number of times that the w appears in queries $\mathbf{x}_{1:n-1}$ and $\mathbf{x}_{1:n}$, respectively.

For each particle p, the importance weight can be iteratively computed as

$$
\mathrm{w}_n^{(p)} = \mathrm{w}_{n-1}^{(p)} p(t_n, u_n|t_{1:n-1}, u_{1:n-1}, b_{1:n-1}^{(p)}, \{\alpha_\ell^{(p)}\}_{\ell=1}^{L}, \{\mu_u^{(p)}\}_{u=1}^{U}) p(\mathbf{x}_n|\mathbf{x}_{1:n-1}, z_{1:n-1}^{(p)}),
\tag{9}
$$

where $p(\mathbf{x}_n|\mathbf{x}_{1:n-1}, z_{1:n}^{(p)}) = \sum_{z_n=1}^{K_+^{(p)}++1} p(\mathbf{x}_n|\mathbf{x}_{1:n-1}, z_{1:n}^{(p)}) p(z_n^{(p)}|t_{1:n}, u_{1:n}, z_{1:n-1}^{(p)})$ and $\mathrm{w}_1^{(p)} = 1/|P|$.

Since the likelihood term depends on the user base intensities μ_u and the kernel parameters $\{\alpha_\ell^{(p)}\}_{\ell=1}^{L}$, which are also unknown, following the literature in SMC devoted to the estimation of a static parameter, we infer these parameters in an online manner by sampling from their posterior given the history of events up to the current sampling time. In particular, we may sample the the base intensity parameter and kernel parameters from their posterior distribution up to, but not including, time t, as

$$
p(\mu_u|H_u(t)) = Gamma\left(a + \sum_{i:t_i<t} [u_i = u], b + t\right)
\tag{10}
$$

and

$$
p(\alpha_\ell|H_\ell(t)) = Gamma\left(\tau_1', \tau_2'\right),
\tag{11}
$$

where

$$
\begin{aligned}
\tau_1' &= \tau_1 + \sum_{i:t_i<t} [\phi_{z_i} = \varphi_\ell] - \sum_{k=1}^{K} [\phi_k = \varphi_l] \quad \text{and} \\
\tau_2' &= \tau_2 + \sum_{i:t_i<t} [\phi_{z_i} = \varphi_\ell] \int_{t_i}^{t} \exp(-(\tau - t_i)) d\tau,
\end{aligned}
\tag{12}
$$

Algorithm 1 summarizes the overall inference procedure, which presents complexity $O(P(U+L+K)+P)$ per learning action i fed to the algorithm, where L and K are the total number of learning patterns and tasks inferred up to the $(i-1)$-th action. Note also that, the for-loop across the particles $p \in P$ can be parallelized, reducing the complexity per learning action to $O(U+L+K+P)$.

Algorithm 1: Inference algorithm for the HDHP

```
Initialize $w_1^{(p)} = 1/|P|$, $K^{(p)} = 0$ and $L^{(p)} = 0$ for all $p \in P$.
for $i = 1, \ldots, n$ do
    for $p \in P$ do
        Sample the kernel parameters $\alpha_\ell^{(p)}$ for $\ell = 1, \ldots, L^{(p)}$ and the user base intensities $\mu_u^{(p)}$ for all $u \in U$ given
          the history of events up to, but not including $t_i$.
        Draw $z_i^{(p)}$ from (8).
        if $z_i^{(p)} = K^{(p)} + 1$ then
            Draw the new table parameters $\phi_{K^{(p)}+1}^{(p)}$ according to (6).
            Increase the number of tasks $K^{(p)} = K^{(p)} + 1$.
            if $\phi_{K^{(p)}+1}^{(p)} = \varphi_{L^{(p)}+1}^{(p)}$ then
                Draw the triggering kernel parameter for the cluster $\alpha_{L^{(p)}+1}^{(p)}$ from the prior.
                Increase the total number of clusters $L^{(p)} = L^{(p)} + 1$.
            end
        end
        Update the particle weight $w_i^{(p)}$ according to (9).
    end
    Normalize particle weights.
    if $\|\boldsymbol{w}_i\|_2^2 < threshold$ then
        Resample particles.
    end
end
```

References

[1] R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. IEEE, 2005.

[2] C. Mavroforakis, I. Valera, and M. G. Rodriguez. Modeling the dynamics of online learning activity. *arXiv preprint arXiv:1610.05775*, 2016.

[3] Y. W. Teh. Dirichlet process. In *Encyclopedia of machine learning*, pages 280–287. Springer, 2011.