User-song model notes

Feb. 13, 2015

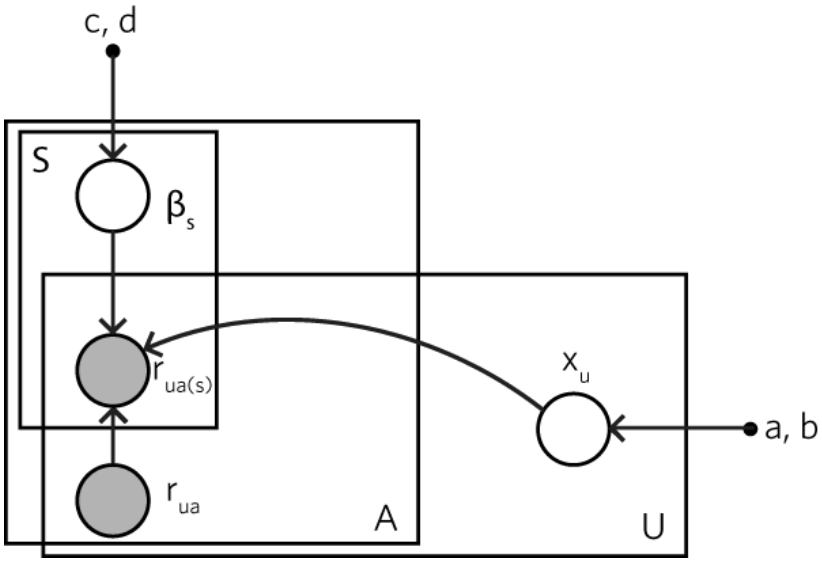

Figure 1: Conditioning on user-artist listen counts.

Motivation

Listening to music is different than reading articles. The artist of a song is of higher importance than the author of a scientific paper for a given user perusing their library. Thinking in terms of a recommendation budget, taking artist-level side information into account should allow us to make better recommendations for both music and articles. Say a user listens to 100 songs: would they rather spread that budget over many songs by one artist, or only one song. Modeling such 'user-artist budgets' is our goal.

We aim to capture this intuition by conditioning how many times a user u has listened to a song $a(s)$ by artist a, denoted $r_{ua(s)}$, on the total number of times a user has listened to songs by that artist, r_{ua} (noting $\sum_s r_{ua(s)} = r_{ua}$).

Generative Process

1. Draw user preferences $x_u \sim \text{Gamma}(a, b)$
2. Draw song-level topics $\beta_s \sim \text{Gamma}(c, d)$
3. Draw user-song count $r_{ua(s)} \mid r_{ua} \sim \text{Mult}(r_{ua},\ x_u^T \beta_s / Z)$

By conditioning the Poisson-distributed user-song count ($r_{ua(s)} \sim \text{Poisson}(x_u^T \beta_s)$) on the user user-artist count r_{ua}, we arrive at a multinomial distribution [1]. Here Z is the normalizing constant, or $Z = \sum_s x_u^T \beta_s$. This multinomial distribution agrees with our intuition, namely that users' artist preferences can be captured in terms of variance.

For example, a user who listens to only one Selena Gomez song 100 times probably doesn't like Selena Gomez that much. However, if that user listens to 10 different Jay Z songs 10 times each, that user probably likes Jay Z a lot. The multinomial will learn to place weight on only one track in the former case, and will be able to weight many tracks in the latter. Both artists have the same number of plays, but we will recommend Jay Z before Selena Gomez.

Extensions

We can generalize this model by thinking of how 'hot' a user is for a certain artist through temperature, denoted T_{ua}. For example, the mere exposure effect might lead people to listen to more top 40 music than they might otherwise enjoy (simply because it is prevalent online and socially).

This would mean modifying the generative process for the song counts to be

$$r_{ua(s)} \sim \text{Poisson}(\exp(x_u^T \beta_s / T_{ua})).$$

Given that we want the rates of the Poisson distributions to be less than one, we then set

$$r_{ua(s)} \sim \text{Poisson}(\exp(\log(x_u^T \beta_s) / T_{ua})).$$

The conditional would then become

$$r_{ua(s)} \mid r_{ua} \sim \text{Mult}(r_{ua},\ \exp(x_u^T \beta_s / T_{ua}) / Z),$$

where Z is again the normalizing constant and b_{ua} is generated from a gamma distribution.

This allows us to interpret the user-artist temperature T_{ua} as the 'temperature' of the system - shifting the weight of the overall distribution (in analogy to the Boltzmann distribution from physics). So a user being 'hot' for an artist such as Jay Z would mean high-entropy parameters in the multinomial, leading to even distribution over Jay's tracks. Likewise, a user being 'cold' for an artist like Selena Gomez would lead to low-entropy parameters in the multinomial, leading to a highly-concentrated user-artist budget in only a few tracks.

Alternatively, one could consider models that account for bias without this energy interpretation:

- by removing the exponential one get Mult(r_{ua}, $(x_u^T \beta_s + T_{ua}) / Z$)) as the conditional
- user-artist level biases: Mult(r_{ua}, $(x_u + b_{ua})^T \beta_s) / Z$)
- as an artist-level correction term, it would look like Mult(r_{ua}, $(x_u^T \beta_s + b_a) / Z$)
- modeling artists in the same space as songs: Mult(r_{ua}, $x_u^T (\beta_s + \beta_a) / Z$).

We might also want to generate user-artist counts (r_{ua}) in the model, rather than simply conditioning on them as we are now. Such iterated models could be quickly tested using the Theano python library.

Alternative distributions

Instead of the conditionally multinomial method of capturing artist-level variance above, we could also user overdispersed Poisson distributions, where the variance is proportional to the mean [2]. Overdispersed Poisson distributions are also known as weighted Poisson distributions. In this case, we would draw user-song counts as

$$r_{ua(s)} \sim \text{OverdispersedPoisson}(x_u^T \beta_s b_{ua}^2, 1/b_{ua}),$$

noting that as variance is proportional to the mean, one of the bias terms would cancel. This would allow us to model user-artist variances in preferences as well. Beta-binomial distributions could be yet another way to model this (i.e. two parameters, one for controlling variance).

Our initial thoughts involved modeling song-level biases generated from a gamma distribution, whose variance is given by the shape and scale parameters multiplied. Cleverly drawing the song-level biases should allow one to capture such a measure of variance, but the number of parameters to be learned is extremely high (diagram).

Scientific article recommendation

We also believe that authors of articles, like artists, are important for recommendations. We could test these models on arXiv user preference data as well.

Inference

We want to minimize the KL divergence between the posterior $p(X,\ \beta)$ and a factorized variational distribution $q(X,\ \beta)$, where X, β, and R denote the latent user factors, song weights, and observed song counts respectively: $X = \{x_u\}_{u=1,\ldots,U}$, $\beta = \{\beta_s\}_{s=1,\ldots,S}$, and $R = \{r_{ua(s)}\}_{u=1,\ldots,U;\ a=1,\ldots,A;\ s=1,\ldots,S}$. Minimizing this KL divergence is accomplished by maximizing the ELBO:

$$\log p(R) = \log \int p(X,\ \beta,\ R)\, d\beta\, dX \ \geq\ E_q[\log p(X,\ \beta,\ R)]\ - E_q[\log q(X, \beta)]\ \equiv L(q)$$

$E_q[\cdot]$ is taken with respect to the variational distribution. We follow BBVI as in [3] to maximize the ELBO.

Our mean-field variational distribution factors as:

$$q(\beta,\ X) = \prod_{s,k} q(\beta_{sk} \mid \lambda_{sk}^{shp},\ \lambda_{sk}^{sca}) \prod_{u,k} q(x_{uk} \mid \gamma_{uk}^{shp},\ \gamma_{uk}^{sca})$$

where β_{sk} and x_{uk} are drawn from gamma distributions with shape and scale parameters. We have $\lambda_i = \{\lambda_{sk}^{shp},\ \lambda_{sk}^{sca},\ \gamma_{uk}^{shp},\ \gamma_{uk}^{sca}\}$. Next we calculate

$$\frac{\partial}{\partial \lambda_{sk}^{shp}} q(\beta_{sk} \mid \lambda_{sk}^{shp},\ \lambda_{sk}^{sca}) \ =\ -\psi(\lambda_{sk}^{shp})\ - \log\ \lambda_{sk}^{sca} + \log \beta_{sk}$$
$$\frac{\partial}{\partial \lambda_{sk}^{sca}} q(\beta_{sk} \mid \lambda_{sk}^{shp},\ \lambda_{sk}^{sca}) \ =\ -\frac{\lambda_{sk}^{shp}}{\lambda_{sk}^{sca}} + \frac{\beta_{sk}}{(\lambda_{sk}^{sca})^2}$$

and find similar expressions for $q(x_{uk} \mid \gamma_{uk}^{shp},\ \gamma_{uk}^{sca})$. Finally, we have

$$grad_{\lambda_{sk}}(L) = (grad_{\lambda_{sk}} q(\beta_{sk} \mid \lambda_{sk}^{shp},\ \lambda_{sk}^{sca}))(\sum_{u=1}^{U} \log p_{\beta_{sk}}(R, x_{uk}) - \log q(\beta_{sk} \mid \lambda_{sk}^{shp},\ \lambda_{sk}^{sca}))$$
$$grad_{\lambda_{uk}}(L) = (grad_{\lambda_{uk}} q(x_{uk} \mid \gamma_{uk}^{shp},\ \gamma_{uk}^{sca}))(\sum_{s=1}^{S} \log p_{x_{uk}}(R, \beta_{sk}) - \log q(x_{uk} \mid \gamma_{uk}^{shp},\ \gamma_{uk}^{sca}))\,.$$

In this expression, we expand $p_{\beta_{sk}}$ as the terms in the joint with β_{sk} (the expression for $p_{x_{uk}}$ is similar but with a sum over songs instead of users):

$$\sum_{u=1}^{U} \log p_{\beta_{sk}}(R, x_{uk}) \ =\ \sum_{u=1}^{U} \log[Mult(r_{ua(s)} \mid r_{ua},\ x_u,\ \beta_s) p(\beta_s \mid \lambda_s^{shp},\ \lambda_s^{sca})]$$

So the total gradient is of size $2SK + 2UK$, for $U,\ S,\ K$ users, songs, and latent factors respectively. Next, we can plug these expressions into the algorithm in Figure 1 in [3]. We first

test this on a subset of the data (10k users, 5k songs), then implement RMSprop and control variates as in [4].

References

1. PK Gopalan, JM Hofman, and DM Blei. “Scalable Recommendation with Poisson Factorization”. In: arXiv (Nov. 2013). URL: http://arxiv.org/abs/1311.1704v3.
2. G Rodriguez, “Models for Count Data With Overdispersion”. URL: http://data.princeton.edu/wws509/notes/c4a.pdf.
3. R Ranganath, S Gerrish, and DM Blei. “Black Box Variational Inference”. In: arXiv (Dec. 2013). URL: http://arxiv.org/pdf/1401.0118v1.pdf
4. R Ranganath, L Tang, L Charlin, and DM Blei. “Deep Exponential Families”. In: arXiv (Nov. 2014). URL: http://arxiv.org/pdf/1411.2581v1.pdf