Ranker

Arthur Breitman

October 2022

Abstract

Ranker is a Bayesian algorithm for ranking items in a set, inspired by the ELO rating system in chess. It assumes each item has a latent score, normally distributed with a latent variance, and that the outcome of a pairwise comparison is a Bernoulli trial with a probability of success given by the logistic function of the difference.

Ranker infers the latent scores of the items and the variance using a variational Bayes approximation which models the posterior as an inverse gamma distributed variance and independent normally distributed scores. We compute the KL-divergence gradient and Hessian in closed-form by approximating the logistic function with the error function. This yields a very fast inference technique, based on the Levenberg-Marquardt algorithm.

Ranker can do more than infer scores and rank items, it can also be used to select the most useful pair of items to compare next. We do this efficiently by computing the gradient of our loss function with respect to the observed trials. The entire algorithm runs in $\mathcal{O}(n^2)$ time, where n is the number of items.

1 Introduction

Ranker is a Bayesian algorithm for ranking items in a set, inspired by the ELO rating system in chess. It assumes each item has a latent score, normally distributed with a latent variance, and that the outcome of a pairwise comparison is a Bernoulli trial with a probability of success given by the logistic function of the difference.

Ranker infers the latent scores of the items and the variance using a variational Bayes approximation which models the posterior as an inverse gamma distributed variance and independent normally distributed scores. We compute the KL-divergence gradient and Hessian in closed-form by approximating the logistic function with the error function. This yields a very fast inference technique, based on the Levenberg-Marquardt algorithm.

Ranker can do more than infer scores and rank items, it can also be used to select the most useful pair of items to compare next. We do this efficiently by computing the gradient of our loss function with respect to the observed trials. The entire algorithm runs in $\mathcal{O}(n^2)$ time, where n is the number of items.

2 Background

Ranker is inspired by the ELO rating system in chess. In chess, each player has a rating which is a positive integer. The rating of a player is a measure of their skill, and it is updated after each game. Each player's rating is updated according to the following formula:

$$R_{new} = R_{old} + K(S - E) \tag{1}$$

where R_{new} and R_{old} are the new and old ratings, K is a constant which determines how much the rating changes, S is the score of the player in the game, and E is the expected score of the player, given the ratings of the two players. The expected score is computed as follows:

$$E = \frac{1}{1 + 10^{-\frac{R_{other} - R_{self}}{400}}} \tag{2}$$

where R_{other} is the rating of the opponent, and R_{self} is the rating of the player.

This is an approximate update rule which is only valid when ratings are close to each other. It corresponds to a gradient descent algorithm with a learning rate of K. The lack of decay in the learning rate is a two-edged sword: it allows the ratings to change quickly, reflecting a player's recent performance, but it also means that the ratings can change wildly. For that reason, K is typically set at 16 for masters and 32 for grandmasters.

A partial Bayesian approach to the ELO rating system is described in BayesElo, using Maximum a Posteriori (MAP) estimation, following the work of [Hun04].

3 Approach

3.1 Model

Ranker assumes that each item has a latent score, normally distributed with a latent variance. The outcome of a pairwise comparison is a Bernoulli trial with a probability of success given by the logistic function of the difference. The model is as follows:

$$\begin{aligned} \nu &\sim \text{InvGamma}(\alpha_h, \beta_h) \\ \forall i, z_i &\sim \mathcal{N}(0, \nu) \\ \forall i < j, o_{i,j} &\sim \text{Binomial}\left(\frac{1}{1 + e^{-(z_i - z_j)}}, m_{i,j}\right) \\ \forall i > j, o_{i,j} &= m_{i,j} - o_{j,i} \end{aligned} \tag{3}$$

where α_h and β_h are hyperparameters, and $\text{o}_{i,j}$ is the outcome of $m_{i,j}$ comparisons between items i and j, where $m_{i,j} = m_{j,i}$ is fixed and given. Note that we use the convention that β_h represents a rate, not a scale. The scale is $1/\beta_h$.

3.1.1 Choice of hyperparameters

We pick concrete values for the hyperparameters $\alpha_h = 1.2$ and $\beta_h = 2$ A handwavy justification for these values follows.

Consider the binomial distribution over 10 stars, obtained by flipping a coin ten times and summing the number of heads. It has a standard deviation of $\sqrt{5/2} \simeq 1.5811$. The difference between two adjacent start ratings is thus $\sqrt{2/5} \simeq 0.6325$. Based on this intuition, we call a difference of $\sqrt{2/5}$ standard deviations a "star" of difference.

Assuming this is an intuitive notion of a "star", what odds should an additional star confer to an item in a heads up match? If those comparisons are hard to judge and very subjective, maybe only 55% of the time? If they're very clear, perhaps over 95% of the time?

Using a logistic rule:

- 55% would correspond to a standard deviation of $\sqrt{5/2}\log(11/9) \simeq 0.317$
- 99% would correspond to a standard deviation of $\sqrt{5/2}\log(19) \simeq 4.656$

None of this is very rigorous, but it gives use some sense of the scale of standard deviations we're dealing with. Clearly 0.0001 and 100 are silly.

These roughly correspond to the 1% and 99% tails of the InverseGamma$(1.2, 2)$ distribution for the variance and justifies our choice of hyperparameters. We are proud to incorporate actual prior knowledge, as opposed to coping out of it by using a so-called "uninformative" prior :-)

3.2 Variational Bayes approximation

We approximate the posterior with a factorized distribution.

$$\begin{aligned} \nu &\sim \text{InvGamma}(\alpha, \beta) \\ \forall i, z_i &\sim \mathcal{N}(\mu_i, \sigma_i) \end{aligned} \tag{4}$$

Note that α and β are parameters of the approximation to the posterior, not the hyperparameters of the model, which are α_h and β_h. Nonetheless, we naturally initialize α and β to α_h and β_h. We initialize μ_i to 0. The case of σ_i is more complicated. Absent any observation, the distribution of z_i is a Student's distribution with infinite variance. We choose to initialize σ_i to minimize the KL divergence with that Student distribution. We discovered empirically (but did not attempt to prove) that the minimum is reached for

$$\sigma_i = \sqrt{\frac{\beta_h}{\alpha_h}} \left(1 + \frac{3}{6\alpha_h + 2\alpha_h^2 + \mathcal{O}(\alpha_h^3)}\right)$$

4 Closed form expression

4.1 KL-divergence

The KL divergence between the true posterior, P, and the variational approximation Q is thus $D_{KL}(Q||P) = -H(Q) + H(Q, P)$, where $H(Q)$ is the entropy of Q and $H(Q, P)$ is the cross entropy of Q and P. The entropy of Q is

$$H(Q) = \int_{\nu=0}^{\infty} \frac{\beta^{-\alpha} \nu^{-\alpha-1} e^{-\frac{1}{\beta\nu}}}{\Gamma(\alpha)} \left(\alpha_h \log(\beta_h) + (\alpha + 1)\log(\nu) + \frac{1}{\beta\nu} + \log(\Gamma(\alpha))\right) d\nu$$

$$+\sum_{i=1}^{n}\int_{z_i=-\infty}^{\infty}\frac{e^{-\frac{1}{2}\left(\frac{z_i-\mu_i}{\sigma_i}\right)^2}}{\sigma_i\sqrt{2\pi}}\left(\frac{1}{2}\log 2\pi+\log\sigma_i+\frac{1}{2}\left(\frac{z_i-\mu_i}{\sigma_i}\right)^2\right)dz_i$$

The cross entropy of Q and P is

$$\begin{aligned}H(Q,P)=&\int_{\nu=0}^{\infty}\frac{\beta^{-\alpha}\nu^{-\alpha-1}e^{-\frac{1}{\beta\nu}}}{\Gamma(\alpha)}\left(\alpha_h\log(\beta_h)+(\alpha+1)\log(\nu)+\frac{1}{\beta\nu}+\log(\Gamma(\alpha))\right)d\nu\\&+\sum_{i=1}^{n}\int_{\nu=0}^{\infty}\int_{z_i=-\infty}^{\infty}\frac{e^{-\frac{1}{2}\left(\frac{z_i-\mu_i}{\sigma_i}\right)^2}}{\sigma_i\sqrt{2\pi}}\left(\frac{1}{2}\log 2\pi+\log(\nu)+\frac{1}{2}\frac{z_i^2}{\nu}\right)dz_i d\nu\\&+\sum_{i=1}^{n}\sum_{j=1}^{n}o_{i,j}\int_{z_\delta=-\infty}^{\infty}\frac{e^{-\frac{1}{2}\left(\frac{z_\delta-(\mu_i-\mu_j)}{\sqrt{\sigma_i^2+\sigma_j^2}}\right)^2}}{\sqrt{\sigma_i^2+\sigma_j^2}\sqrt{2\pi}}\log\left(1+e^{-z_\delta}\right)dz_\delta+Cst\end{aligned}$$

4.2 The logistic-normal integral

As it turns out, we can express most of the integrals above in terms of elementary functions, as well as the polygamma function Ψ. The integral

$$\int_{z_\delta=-\infty}^{\infty}\frac{e^{-\frac{1}{2}\left(\frac{z_\delta-(\mu_i-\mu_j)}{\sqrt{\sigma_i^2+\sigma_j^2}}\right)^2}}{\sqrt{\sigma_i^2+\sigma_j^2}\sqrt{2\pi}}\log\left(1+e^{-z_\delta}\right)dz_\delta$$

is more problematic. A related integral is the logistic-normal integral,

$$\int_{z_\delta=-\infty}^{\infty}\frac{e^{-\frac{1}{2}\left(\frac{z_\delta-(\mu_i-\mu_j)}{\sqrt{\sigma_i^2+\sigma_j^2}}\right)^2}}{\sqrt{\sigma_i^2+\sigma_j^2}\sqrt{2\pi}}\frac{1}{1+e^{-z_\delta}}$$

[Cro09] offers an approximation by replacing the logistic function with an error function:

$$\frac{1}{1+e^{-z}}\simeq\frac{1}{2}+\frac{1}{2}\mathrm{erf}\left(\frac{\sqrt{\pi}}{4}z\right)$$

Given that

$$\frac{d}{dz}\log\left(1+e^{-z}\right) = \frac{1}{1+e^{-z}} - 1$$

we choose to approximate $\log\left(1+e^{-z}\right)$ as

$$\log\left(1+e^{-z}\right) \simeq \int_{\zeta=z}^{\infty}\left(\frac{1}{2} - \frac{1}{2}\mathrm{erf}\left(\frac{\sqrt{\pi}}{4}\zeta\right)\right)\,d\zeta = \frac{2}{\pi}e^{-\frac{\pi z^2}{16}} - \frac{1}{2}z\ \mathrm{erfc}\left(\frac{\sqrt{\pi}}{4}z\right)$$

The difference between the two is maximal for $z=0$ and is $\log(2) - \frac{2}{\pi} \simeq 0.0565$. The difference between the derivatives with respect to z is at most 0.0177.

4.3 Putting it all together

The KL part of the divergence that does not depend on the observations is given by

$$\begin{aligned}
&-\alpha + \frac{\alpha\beta}{\beta_h} + \alpha_h\log(\beta_h) - \frac{1}{2}n(1+\log(2\pi)) + \ln(\beta) + \Gamma(\alpha_h) - \Gamma(\alpha) \\
&+ (\alpha-\alpha_h)\psi(\alpha) - (1+\alpha_h)\ln(\beta) - \left(\sum_{i=0}^{n-1}\ln\sigma_i\right) \\
&+ \frac{1}{2}\left(\alpha\beta\left(\sum_{i=0}^{n-1}\mu_i^2+\sigma_i^2\right) - n(\ln(\beta)+\psi(\alpha)-\log(2\pi))\right)
\end{aligned}$$

Then to account for all observed trials, we add

$$\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} o_{i,j}\cdot\left(e^{-\frac{\mu_\delta^2}{h^2}}\frac{h}{2\sqrt{\pi}} + \mu_\delta g_{\mu_\delta}\right)$$

where

$$\begin{aligned}
\mu_\delta &= \mu_i - \mu_j, \\
\sigma_\delta &= \sqrt{\sigma_i^2+\sigma_j^2}, \\
h &= \sqrt{\frac{16}{\pi}+2\sigma_\delta^2},
\end{aligned}$$

$$g_{\mu_\delta} = -\frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{\mu_\delta}{h}\right)\right).$$

Note that $o_{i,j}$ is always positive, it represents the number of time i beat j. This is not a signed quantity. This is indeed important, i beating j and then j beating i over the course of two trials is not the same as no trial at all.

4.4 Gradient and Hessian

These follow directly from the expression in the previous function, please don't make me type them out.

5 Inference

We use the Levenberg-Marquardt algorithm to minimize the KL divergence with a Newton-CG algorithm. We condition the Hessian using the diagonal.

6 Loss function

6.1 functions

The expected number of inversions is fairly simple to compute analytically from the model. The entropy of the posterior is not because it collapses into determining any single score with as much precisions as possible, at the exclusion of the rest. The sum of variances is reasonable.

6.2 derivative

These are all easily differentiable with respect to the matrix of observations o. In fact we simply maintain the derivative of the mu and $sigma$ with respect to the observations, and this lets us get the derivative of the loss function with respect to the observations by the chain rule.

So we can pick the top 10 observations that marginally improve the loss function the most. We then compute what the expected improvement in the loss with a hypothetical non-infinitesimal observation and pick that.

In fact we can backtrack through possible observations to make using the gradient as a heuristic to search the tree.

Conclusion

This is probably more trouble than it's worth, but it's a fun exercise. The closed form formulas have no business working out so simple with the error function approximation, but they do. The inference is very fast which means this can scale to many items, or can be used for a deep backtracking search.

This could be extended to more sophisticated models of match outcomes, including the possibility of draws, the possibility of more than two players in a match, allowing scores to vary over time and so on.

References

[Hun04] David R. Hunter. "MM algorithms for generalized Bradley-Terry models". In: *The Annals of Statistics* 32.1 (2004), pp. 384–406. DOI: 10.1214/aos/1079120141. URL: https://doi.org/10.1214/aos/1079120141.

[Cro09] Gavin E Crooks. "Logistic approximation to the logistic-normal integral". In: *Technical Report Lawrence Berkeley National Laboratory* (2009).