A Two Level Mixture Model for Bandits

Guowei Sun
Mathematicas
gwsun@math.umd.edu

Cheng Jie
Mathematics
cjie@math.umd.edu

Abstract

We propose a two level mixture model to exploit the similarity between arms in an online manner, where similar arms will be clusterred together to shrink the exploration space. We define a general, flexible bayesian mixture model which enables the usage of state-of-the-art bayesian upper confidence bounds. We prove an upper bound on the regret. We demonstrate that the estimation of cluster and arm parameters can be efficiently solved by a natural extension of the Expectation-Maximization algorithm. Finally, we apply our framework to both continuous and binary simulation experiments.

1 Introduction

In multi-armed bandits problems, one central goal is to utilize structures in arms to 1) solve difficult problems or 2) further minimize regrets. Most work is focused on using known information. Lipshitz bandits[6],[7] assumed lipshitiz reward functions and discretized the arms space to solve uncountably many arms problem. Spectral bandits uses a given similarity graph to represent the pair-wise similarity information on arms, and assumed a smooth function on this graph to extract more information every time an arm is played[9]. Gaussian bandits assumed a correlated gaussian prior on the expected rewards. Therefore, the rewards from one arm will give information about rewards on all arms[3].

There are some works that learns the structure between arms in an online manner. Gentile et al proposed to dynamically cluster the arms on a graph as samples are collected[2].
In our project, we propose a two level mixture model which is capable of utilizing known information and learning certain structure between arms. We assume arms with similar rewards can be clustered together, the expected rewards on each cluster is generated from one common distribution. In our algorithm, we first select a promising cluster, then select an arm within that cluster. We show that this will give better upper bound on the bayes regret, and performs better on simulated experiments compared to popular bandit algorithms.

2 Mixture Models

Mixture models are commonly used to cluster data[1], to track objects in images[10]. It is very powerful in discovering similarities between data points.

2.1 A Two Level Mixture Model

Given K arms, each with expected reward μ_k. Assume the prior on the μ is a mixture model, where each model $f_j(, \alpha_j)$ is a distribution parameterized by α_j. Let π_j to be the probability of μ generated

FALL CMSC858G Course Project (Experts, Bandits and Games 2016), UMD .

from the jth model, the prior distribution would be

$$\mu \sim \sum_{j=1}^{C} \pi_j f_j(\mu, \alpha_j) \tag{1}$$

Introduce a latent variable, or label function $c(\mu) : \mathcal{A} \to \mathcal{C}$ indicating the source of μ. Given the expected rewards μ_k and its source $c(\mu_k) = j$, the mth sample of rewards x_{km} from arm k follow a distribution $g_k(, \theta_k)$, parameterized by θ_k

$$x_{km}|\mu_k, c(\mu_k) = j \sim g_k(x_{km}, \theta_k) \tag{2}$$

Given data matrix X, where the mth sample on kth arm is x_{km}. Denote n_k to be number of samples we have for arm k. The posterior of μ_k given its source would be

$$f(\mu_k|X, c(\mu_k) = j) \sim f_j(\mu_k, \alpha_j) \prod_{m=1}^{n_k} g_k(x_{km}, \theta_k) \tag{3}$$

Uncondition on the label function, the posterior of the expected rewards would be a mixture distribution, with each model given by the conditioned posterior

$$f(\mu|X) \sim \sum_{j=1}^{C} \pi_j f_j(\mu, \alpha_j) \prod_{m=1}^{n_k} g_k(x_{km}, \mu_k, \theta_k) \tag{4}$$

For simplicity, use ϕ_{jk} to denote the posterior distribution of $\mu|c(\mu) = j$. Then from above, then we would get

$$\phi_{jk} = f_j(\mu_k, \alpha_j) \prod_{m=1}^{n_k} g_k(x_{km}, \mu_k, \theta_k) \tag{5}$$

Our method would first estimate all the model parameter α_j and arm parameter μ_k, θ_k from the given data matrix X. Then, treat each model as an arm, first choose a model, then choose the best arm generated from this model. For estimating the parameters, we use the Expectation-Maximization methods to maximize the posterior distribution given the data.

2.2 MAP treatment

Most mixture model estimation problems deal with the standard mixture model, while our model introduce an extra layer of distribution. Here, we maximize our posterior mixture likelihood, and show that it naturally fits into the standard Expectation Maximization method.
For all the rewards $\mu_k, k \in 1, 2, .., K$, their joint posetrior distribution would be the product of their densities

$$f(\mu|X) \sim \prod_{k=1}^{K} [\sum_{j=1}^{C} \pi_j \phi_{jk}] \tag{6}$$

To maximize the posterior likelihood, we can maximize the posterior $\log$ likelihood. Let $l(X) = \log f(\mu|X)$. Introduce a weight to simplify notation,

$$w_{jk} = \frac{\pi_j \phi_{jk}}{\sum\limits_{j=1}^{C} \pi_j \phi_{jk}} \tag{7}$$

To max the posterior density of the above model, both the model parameter α_j, π_j and arm parameter μ_k, θ_k can be estimated from the estimating equation

$$\frac{\partial l}{\partial \alpha_j} = \sum_{k=1}^{K} w_{jk} \frac{\partial \log \phi_{jk}}{\partial \alpha_j} = 0 \tag{8}$$

$$\frac{\partial l}{\partial \mu_k} = \sum_{j=1}^{C} w_{jk} \frac{\partial \log \phi_{jk}}{\partial \mu_k} = 0 \tag{9}$$

$$\frac{\partial l}{\partial \theta_k} = \sum_{j=1}^{C} w_{jk} \frac{\partial \log \phi_{jk}}{\partial \theta_k} = 0 \tag{10}$$

Substitute the expression for ϕ_{jk}, and notice the fact that $\sum_{j=1}^{C} w_{jk} = 1$. The above estimating equations can be simplified as

$$\frac{\partial l}{\partial \alpha_j} = \sum_{k=1}^{K} w_{jk} \frac{\partial \log f_j(\mu_k, \alpha_j)}{\partial \alpha_j} = 0 \tag{11}$$

$$\frac{\partial l}{\partial \theta_k} = \sum_{m=1}^{n_k} \frac{\partial \log g_k(x_{km}, \mu_k, \theta_k)}{\partial \theta_k} = 0 \tag{12}$$

$$\frac{\partial l}{\partial \mu_k} = \sum_{j=1}^{C} w_{jk} \frac{\partial \log f_j(\mu_k, \alpha_j)}{\partial \mu_k} + \sum_{m=1}^{n_k} \frac{\partial \log g_k(x_{km}, \mu_k, \theta_k)}{\partial \mu_k} = 0 \tag{13}$$

Denote $\hat{\mu}, \hat{\alpha}, \hat{\theta}$ to be the estimator from the above estimating equations. For the "arm parameter" θ, the estimator is the Maximum Likelihood Estimator using reward samples from this arm. The μ_k and α_j estimation involves the weights w_{jk}. We can solve this optimization sub-problem by the Expectation Maximization method, where we iteratively update the parameters μ, α and the weights w.

2.3 Computational Complexity

In the above treatment of estimating all parameters in our mixture model, we have to use all samples for all arms at each time step. To update the weights w, we would have K function evaluations. At time t, with a new reward sample from arm a_t. We can use our estimated parameters from the previous step as starting point for the MAP iterative estimating algorithm. In our tests, the convergence would happen within 4 or 5 steps. Therefore, the computational cost in each time step is $O(K)$ for our method.

However, online estimation of mixture model is itself an open problem. Efficient methods has been proposed[5],[8]. In further work, we can hopefully reduce the computational cost to C, rather than K by incorporating the recent advances in mixture model estimation.

3 Two Level Algorithm for Clusters and Arms

With the estimated parameters, we propose a two level algorithm. First, we use a UCB-like index to choose a promixing cluster at each time step, then, we choose an arm inside that cluster.

3.1 Selection of Cluster

For clusters $1, 2, .., C$, we build a confidence interval from the estimated parameters. Write

$$Q_j = Q(\alpha_j, q), \forall j \in 1, 2, .., C \tag{14}$$

With q being some small number. The index Q is the qth quantile of the estimated distribution f. At each time step, we choose the cluster with the maximum Q.

3.2 Selection of Arm

For the selection of arms, we use the posterior distribution of μ_k to build a bayes-ucb index for arm k following the Bayes ucb paper[4]. The bayes ucb index is

$$q_k = Q(1 - \frac{1}{t(\log n)^q, \lambda_k^{t-1}}) \tag{15}$$

We choose the arm with the maximum bayes-ucb index.

Algorithm 1: Two Level Mixture Algorithm for Bandits

```
Running: for $t \in 1, 2, ..., T$ do
    $C_t \leftarrow argmax(Q_j)$
    $a_t \leftarrow argmax(q_k)$, from within $C_t$
    Observe $x_{a_t}$
    Update Estimate $\alpha, \beta$
end
```

4 Numerical Experiments

Our test our algorithm and compare it to usual ucb algorithm on two simulated experiments. One for continuous rewards, where we assume a gaussian mixture prior. The other on binary rewards, where we assume a beta mixture prior.

4.1 Continuous Rewards

4.1.1 Gaussian Mixture bandits: setting up

In this experiment, we generate a problem with 1000 arms. The reward vector μ is simulated from a gaussian mixture model. Using α, β to denote the prior mean and variance in the model. The prior is then

$$f(\mu, \alpha, \beta) \sim \sum_{j=1}^{C} \pi_j f_j(\mu, \alpha_j, \beta_j), f_j \sim \mathcal{N}(\alpha_j, \beta_j^2) \tag{16}$$

Our goal would be to first find the best "model" f^* among the $f_j's$, the proceed to select the best arm generated from this model. For the kth arm, given the reward μ_k, the samples of rewards the algorithm observes follows another gaussian distribution

$$x|(mu_k, c(\mu) = j) \sim g_k(x, \mu_k, \sigma_k), g_k \sim \mathcal{N}(\mu_k, \sigma_k^2) \tag{17}$$

In our simulation experiment, we set $\alpha = \{1, 2, 3, 4, 5\}, \beta = 0.25, \sigma = 0.5$. Then some arms will naturally be closer to each other. Following the EM algorithm, we iteratively update the parameter estimation and the weights. The distribution of the five clusters is illustrated in figure 4.1.1

4.1.2 Gaussian Mixture bandits: Experiment results

We conducted the macro-simulation process of **generating means of the arms → performing bandits algorithm** 30 times. During each simulation process, we record the accumulated regrets at each round, and generate a two-dimensional plot of accumulated-reward against the rounds, which is displayed below: From figure 4.1.2, we see that the first 5000 rounds will create a linear regrets, and this is reasonable because we sample 5 times for each arm at the 5000 exploration rounds. After 5000 rounds, we narrow down our concentration on the cluster which generate the best mean rewards, and our regret at each time is decreasing gradually.

Remark. Notice that we haven't achieve a $\sqrt{T}$ regret yet since we haven't simulated enough samples, but after 10000 rounds it's apparently clear that the instantaneous regret is going to be less than $O(\sqrt{T})$. In addition to that, since we make the line to be relatively bold and the scale of the graph is very large , we didn't see too much variations between each segment, but actually the accumulated regret is not a smooth curve corresponding to the time-horizon.

5 Summary and Future Work

In our project, we proposed a mixture model to discover similarity between arms. We showed that by clustering the arms into clusters, we can find the optimal arm faster by first narrowing down to a promising clsuter. In the furture, we hope to find appropriate online algroithms for the estimation of the parameters to decrease computational load. In our method, all parameters are estimated using collected samples of rewards. If we know in advance some arms are very close, we can encode

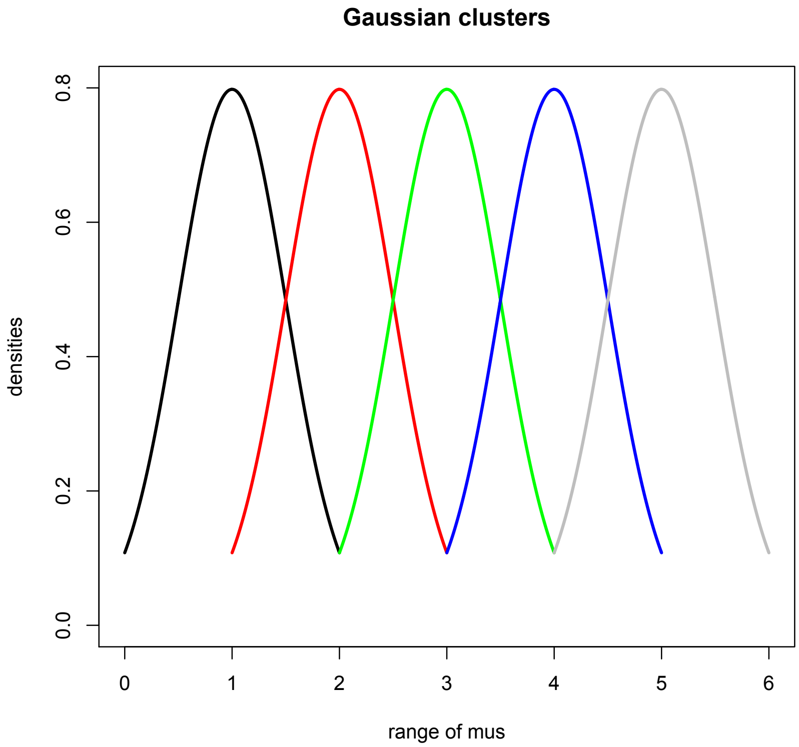

Figure 1: Distribution of five clusters

our information by setting constraints on some parameters inside our model. A model that encodes known information and updates the knowledge using new samples would intuitively outperform our entirely online method.

References

[1] Brian S Everitt. *Finite mixture distributions*. Wiley Online Library, 1981.

[2] Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *ICML*, pages 757–765, 2014.

[3] Junya Honda and Akimichi Takemura. Optimality of thompson sampling for gaussian bandits depends on priors. In *AISTATS*, pages 375–383, 2014.

[4] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On bayesian upper confidence bounds for bandit problems. In *AISTATS*, pages 592–600, 2012.

[5] Dahua Lin. Online learning of nonparametric mixture models via sequential variational approximation. In *Advances in Neural Information Processing Systems*, pages 395–403, 2013.

[6] Stefan Magureanu, Richard Combes, and Alexandre Proutiere. Lipschitz bandits: Regret lower bound and optimal algorithms. In *COLT*, pages 975–999, 2014.

[7] Aleksandrs Slivkins. Contextual bandits with similarity information. *Journal of Machine Learning Research*, 15(1):2533–2568, 2014.

[8] Mingzhou Song and Hongbin Wang. Highly efficient incremental estimation of gaussian mixture models for online data stream clustering. In *Defense and Security*, pages 174–183. International Society for Optics and Photonics, 2005.

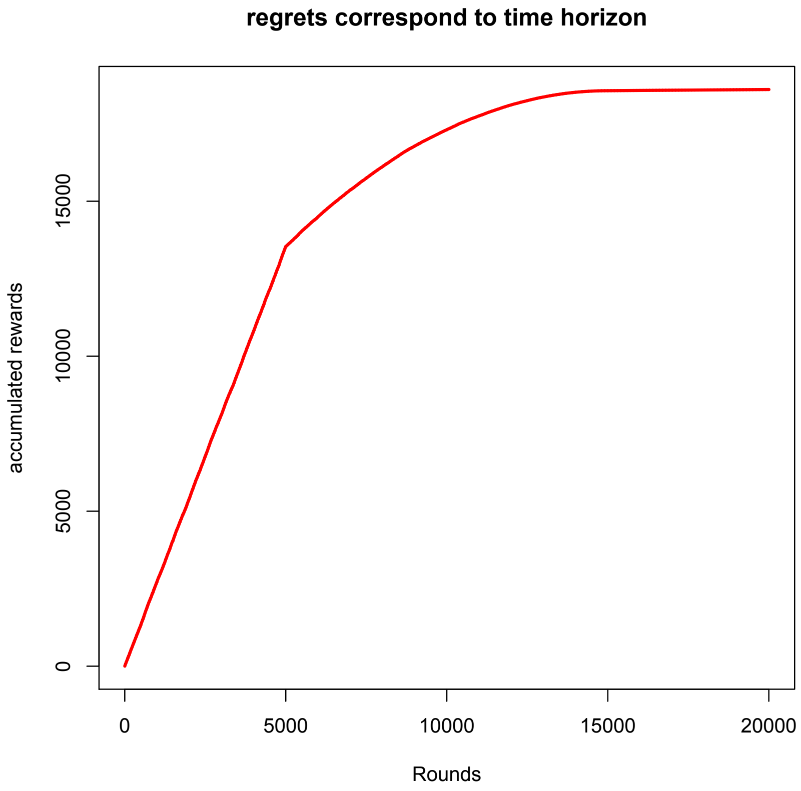

Figure 2: Distribution of five clusters

[9] Michal Valko, Rémi Munos, Branislav Kveton, and Tomas Kocak. Spectral bandits for smooth graph functions. In *ICML*, pages 46–54, 2014.

[10] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE, 2004.