Retrieval of Experiments by Efficient Comparison of Marginal Likelihoods

Sohan Seth[1], John Shawe-Taylor[2], and Samuel Kaski[1,3]

[1] Helsinki Institute for Information Technology HIIT,
Department of Information and Computer Science, Aalto University, Finland
[2] Centre for Computational Statistics and Machine Learning,
University College London, UK
[3] Helsinki Institute for Information Technology HIIT,
Department of Computer Science, University of Helsinki, Finland
{sohan.seth,samuel.kaski}@hiit.fi, j.shawe-taylor@ucl.ac.uk

Abstract. We study the task of retrieving relevant experiments given a query experiment. By experiment, we mean a collection of measurements from a set of 'covariates' and the associated 'outcomes'. While similar experiments can be retrieved by comparing available 'annotations', this approach ignores the valuable information available in the measurements themselves. To incorporate this information in the retrieval task, we suggest employing a retrieval metric that utilizes probabilistic models learned from the measurements. We argue that such a metric is a sensible measure of similarity between two experiments since it permits inclusion of experiment-specific prior knowledge. However, accurate models are often not analytical, and one must resort to storing posterior samples which demands considerable resources. Therefore, we study strategies to select informative posterior samples to reduce the computational load while maintaining the retrieval performance. We demonstrate the efficacy of our approach on simulated data with simple linear regression as the models, and real world datasets.

Keywords: information retrieval, experiments, ranking, classification.

1 Introduction

An experiment is an organized procedure for validating a hypothesis, and usually comprises measurements over a set of variables that are either varied (covariates or independent variables) or studied (outcomes or dependent variables). For example, in the study of genome-wide association, one explores the association between 'traits' (controlled variable) and common genetic variations (response variables), or in the study of functional genomics covariates can be the species, disease state, and cell type, whereas outcome can be microarray measurements.

Traditionally, similar experiments have been retrieved from qualitative assessment of related scientific documents without explicitly handling the experimental data. Recent technological advances have allowed researchers to both acquire measurements in an unprecedented scale throughout the globe, and to release

C.K. Loo et al. (Eds.): ICONIP 2014, Part II, LNCS 8835, pp. 135–142, 2014.

these measurements for public use after curation, e.g., [1]. However, exploring similar experiments still relies on comparing the manual annotations which suffer extensively from variations in terminology, and incompleteness in annotations (see e.g., [2]). The global effort of availing researchers with wealth of data invites the need for sophisticated retrieval systems that look beyond annotations in comparing related experiments to improve accessibility.

The next step toward this goal is to compare the *knowledge* acquired from experimental measurements rather than just annotations. From a Bayesian perspective, one can quantify knowledge as the posterior distribution of parameters given the measurements. The posterior distribution captures both the information content of the measurements, in terms of the *likelihood function*, as well as the experience and expertise of the experimenter in terms of the *prior distribution* over parameters. We study the future scenario where researchers have submitted (Bayesian generative) models learned on their experiment along with measurements and annotations. We explicitly assume that we have access to such database and develop efficient approaches for retrieving relevant experiments. Developing a successful retrieval engine is a first step toward realizing the future scenario.

We suggest the *marginal likelihood* (1) as a similarity metric, where the underlying idea is to evaluate the likelihood of the query experiment on Bayesian models learned from (individual) existing experiments. Here the underlying idea is that an experiment is relevant to a query if models learned from it are good for describing the query data. Bayesian models usually need to be stored as a collection of samples from the posterior distribution since the posterior distribution itself might not be available in closed form. The suggested metric (1) then can be efficiently estimated as the average likelihoods over the posterior samples (2). However, this approach has two issues: storing the posterior samples requires considerable resource, and evaluating each marginal likelihood can be computationally demanding (in particular for latent variable models for which the latent components cannot be integrated out in closed form). This paper deals with selecting *informative* posterior samples to reduce both storage and computational requirements while maintaining the retrieval performance.

We achieve this by approximating the marginal likelihood as a *weighted average* of individual likelihoods over posterior samples (3). The weights are then learned to preserve the relative order of experiments in a training set (section 2.1). This is done while imposing a suitable sparsity constraint which allows us to only consider posterior samples with non-zero weights when computing the likelihood of a query sample, thus reducing the storage and computational burden considerably.

2 Method

Assume a set of experiments $\{\mathcal{E}_d\}_{d=1}^{D}$. Each experiment is defined as a collection of measurements over covariates and outcomes, i.e., $\mathcal{E}_d = \{(\mathbf{x}_{di}, \mathbf{y}_{di})\}_{i=1}^{n_d}$. We assume that each experiment $\mathcal{E}_d$ has been modeled by a model $\mathcal{M}_d$, producing a set

of posterior MCMC samples $\{\theta_{dk}\}_{k=1}^{m_d}$ from each model. Our general objective is to rank the experiments $\mathcal{E}_d$—actually the models $\mathcal{M}_d$ in the database—according to their relevance to a new query experiment $\mathcal{E}_q$ which is not in the database.

We suggest retrieving similar experiments ranked according to the marginal likelihood they produce for the query, i.e. [1],

$$\mathrm{ML}_{q|d} = p(\mathcal{E}_q|\mathcal{E}_d). \tag{1}$$

This metric has been previously discussed in the context of document retrieval where its use is motivated by capturing the user's intent in terms of the likelihood of a set of keywords $\mathcal{E}_q$ being generated by a document $\mathcal{E}_d$ [3]. In the context of document retrieval the marginal likelihood is usually computed by jointly modeling the whole document database. However, we cannot evaluate this metric by modeling multiple experiments jointly, since we explicitly allow experimenters to submit their models to the database (however, query does not need to be modeled). Therefore, we utilize individual models, represented by posterior distributions, $p(\cdot|\mathcal{E}_d) \propto p(\mathcal{E}_d|\cdot)\pi_d(\cdot)$ to evaluate the marginal likelihood as $\mathrm{ML}_{q|d} = \mathbb{E}_{p(\cdot|\mathcal{E}_d)}p(\mathcal{E}_q|\cdot)$, where π_d is the prior information specific to experiment d. The likelihood can be approximated using posterior samples $\{\theta_{dk}\}_{k=1}^{m_d} \sim p(\cdot|\mathcal{E}_d)$ as

$$\widehat{\mathrm{ML}}_{q|d} \approx \frac{1}{m_d}\sum_{k=1}^{m_d} p(\mathcal{E}_q|\theta_{dk}). \tag{2}$$

However, this approach is computationally demanding: even if one has access to a closed form likelihood function without latent components, this scales up as $O(\sum_d m_d n_q p)$ where p is the number of parameters for the model (assuming the models are in the same exponential family). Additionally, if the latent variables cannot be explicitly integrated out then the samples have to computationally approximate $\int p(\mathbf{x}, \mathbf{z}|\theta)\mathrm{d}\mathbf{z}$ as well. The technical contribution of this paper is to address this issue by selecting *fewer* posterior samples that are essential in the retrieval task, i.e., discriminative between experiments. We achieve this by approximating the marginal likelihood as

$$\widetilde{\mathrm{ML}}_{q|d} \approx \frac{1}{m_d}\sum_{k=1}^{m_d} w_{dk}\prod_{i=1}^{n_d} p((\mathbf{x}_{qi}, \mathbf{y}_{qi})|\theta_{dk}) \tag{3}$$

where $\mathbf{w}_d = [w_{d1}, \ldots, w_{dm_d}]$ is a vector of *sparse non-negative weights*. In this way, the posterior samples for which the corresponding weights are zero can be safely ignored. Since we are effectively estimating the *weighted mean* of a set of values, ideally speaking, $\mathbf{w}_d$ should be a *stochastic* vector: positive values that sum to one. However, we observe that even without explicitly imposing this constraint we can achieve favorable performance, and this simplifies the optimization problem considerably.

[1] Marginal likelihood is often used to refer to the model evidence. In our case the model is defined by the data set $\mathcal{E}_d$, and the data in computing the likelihood is the query data $\mathcal{E}_q$. We retrieve the data set for which model evidence is the largest.

2.1 Preserving Ranking of Experiments

To learn the weights for each experiment, we adapt the concept of *learning to rank* which is a well explored research problem in information retrieval [4]. However, while this approach is usually applied for learning a function over document-query pairs, we utilize the concept in learning weights over posterior samples for all experiments ("documents") together. Assume, without loss of generality, that given a query q and two experiments i_1 and i_2 in the database, i_1 ranks higher than i_2, i.e., $\widehat{\mathrm{ML}}_{q|i_1} > \widehat{\mathrm{ML}}_{q|i_2}$. Therefore, while learning the weights $\mathbf{w}_{i_1}$ and $\mathbf{w}_{i_2}$, we need to ensure that

$$\sum_k w_{i_1 k} p(\mathcal{E}_q|\theta_{i_1 k}) > \sum_k w_{i_2 k} p(\mathcal{E}_q|\theta_{i_2 k})$$

i.e., we learn the weights to preserve the relative ranks of the experiments with respect to the unweighted metric. When each experiment in the training set is used as a query q, preserving the relative ranks of each pair $\{i_1, i_2\} \subset \{1, \ldots, D\} \setminus \{q\}$ translates to needing to satisfy $D(D-1)(D-2)$ binary constraints for learning the weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_d$. Fortunately not all of the constraints are usually required since a user is often interested in retrieving only the top (say, top K) experiments rather than all experiments. Therefore, we reformulate our approach and, given a query q, focus on preserving the order of top K experiments. Given any experiment q we select the K closest experiments, $I_q^K = \{i_{j_1}, \ldots, i_{j_K}\}$, and compare them pairwise with the rest of the $(D-2)$ experiments in the database. Intuitively, this preserves the relative orders among the top K experiments I_q^K, and also ensures that these experiments are ranked higher compared to the rest of the $\{1, \ldots, D\} \setminus \{q \cup I_q^K\}$ experiments. This reduces the set of constraints to $KD(D-2)$ where $K \ll D$. Notice that it is certainly feasible to choose different K for different queries.

2.2 Optimization Problem

Satisfying the binary constraints can be formalized as a classification problem $\{(\mathbf{X}_l, y_l)\}_{l=1}^L$ with a highly sparse design matrix $\mathbf{X}$ of dimension $L \times m$, with $L = KD(D-2)$ realizations and $m = \sum_d m_d$ features for learning a combined weight vector $\mathbf{w} = [\mathbf{w}_1, \ldots, \mathbf{w}_d]$, i.e., to satisfy $(\mathbf{X}_l \mathbf{w} + b) y_l > 0$ for all l. Each row of $\mathbf{X}$ belongs to a triplet (q, i_1, i_2), and in that row only the columns associated with posterior samples from i_1 and i_2 are non-zero, and have values $\{p(\mathcal{E}_q|\theta_{i_1 k})\}_{k=1}^{m_{i_1}}$ and $\{-p(\mathcal{E}_q|\theta_{i_2 k})\}_{k=1}^{m_{i_2}}$ respectively. The label associated with this entry is 1 if $\widehat{\mathrm{ML}}_{q|i_1} > \widehat{\mathrm{ML}}_{q|i_2}$, and zero otherwise. An important aspect of this construction is that the label is not absolute, i.e., we can change the sign of a row in the design matrix, i.e., assign the values $\{-p(\mathcal{E}_q|\theta_{i_1 k})\}$ and $\{p(\mathcal{E}_q|\theta_{i_2 k})\}$ to the row instead, and switch the label accordingly. Actually, for each row we randomly pick one of these scenarios to maintain class balance, i.e., we have similar numbers of zeros and ones. Since we are solving a classification problem, each row of the design matrix can be normalized without effecting the class label. This helps

solve scaling issues: Instead of likelihoods p_l, we can classify log likelihoods $\ln p_l$, and compute the normalized entries as $\pm \exp(\ln p_l - \max_l \ln p_l)$. These values are in $[-1, 1]$. We use the library liblinear [5] to solve this optimization problem. We use the logistic cost with l_1 regularization, and set the regularization value to 1.

3 Related Works

If one models the query experiment as well, then there are other possible approaches of evaluating similarity between two experiments. Posterior samples $\{\theta_{dk}\}$ have recently been modeled [6] sequentially with Dirichlet process mixtures of normal distributions using particle filtering. Once this model (over posterior samples) has been learned, the similarity between two experiments can be evaluated through similarity of the cluster assignments of the respective posterior samples. Given models of the query and the existing experiments, one can also evaluate their similarity in terms of probabilistic distances or kernels [7]. However, both these approaches have the limitation that the models have to belong to the same family for the similarity to be defined whereas the proposed approach does not require that. Moreover, the distances or kernels between models are not tailored to assist in the user's task, in our case retrieval.

Another possible approach for measuring similarity between experiments is to model the measurements together in a multi-task learning framework [8]. However, off-the-shelf methods for modeling multiple experiments together utilize the same prior and likelihood for all experiments which restricts the generality, and will not exploit the benefit of the knowledge available at the experimenter's disposal. That said, the true purpose of multi-task learning is to utilize knowledge from similar tasks to improve the learning of a new task, which is fundamentally different than retrieval. Also, treating each experiment or model separately rather than as part of a unified model provides well desired modularity to separate the modeling and retrieval task that can be handled by respective experts which is achieved by the proposed set-up.

A similar problem has been explored before by [9] where the authors aimed at retrieving a single data vector given a query vector. This was done by modeling all data together using latent Dirichlet allocation. Retrieving an experiment given a query experiment, however, is conceptually very different since a single data point cannot capture the experimental variability that one would be interested in which is achieved by the proposed approach. That said, retrieval of experiments as discussed in this article allows one to also query with a single observation to find the closest experiment which could have generated that particular sample. This approach has an intriguing characteristic that it enables assigning different parts of the query experiment to different models.

4 Experiments

We demonstrate the performance of the proposed approach on four real world datasets: landmine [8], computer [10], restaurant [11], and LINCS (described below). The first two are standard in the multi-task learning genre. For landmine,

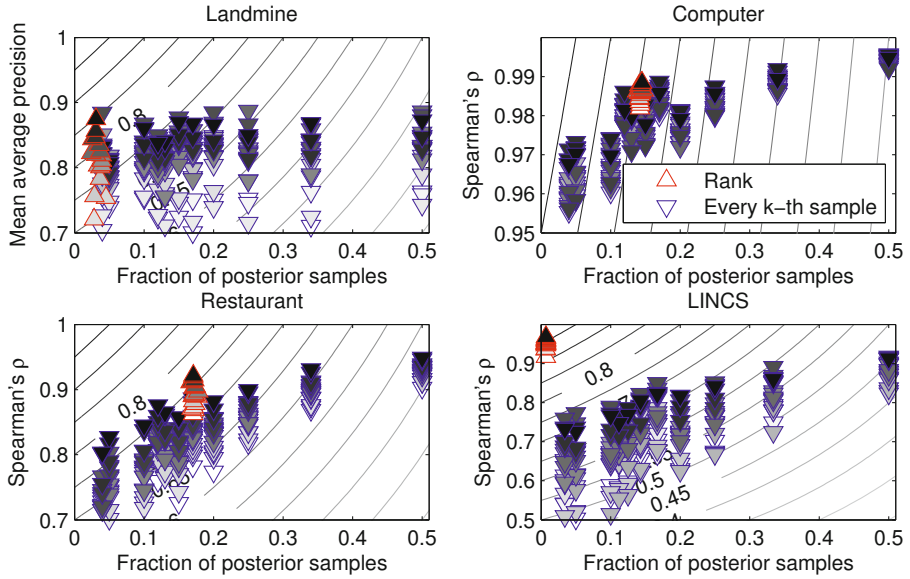

Fig. 1. Comparison of the proposed approach and a simpler metric (evaluating $\widehat{\mathrm{ML}}_{q|d}$ by choosing every k-th posterior samples without any optimization) on real datasets. For landmine we present mean average precision MAP as we have access to labels of each experiment, while for the other two datasets we present the performance compared to $\widehat{\mathrm{ML}}_{q|d}$ estimated with all posterior samples. Each gray shade corresponds to a random partition of the dataset in database and queries. The proposed approach shows improved performance compared to storing every k-th sample for LINCS with respect to (1-sparsity) x retrieval-performance (contours), and performs equally well otherwise.

we have access to class labels of each experiment, and we evaluate the performance of our approach in terms of mean average precision MAP, while on the other two datasets we use correlation with respect to the ranking given by $\widehat{\mathrm{ML}}_{q|d}$ with all posterior samples. We present the results collectively in Fig. 1. For landmine, we train binary probit regression models, while for the other datasets we use normal regression models with non-sparse gamma priors over the weight precisions. For each experiment we generate 100 (1000 for LINCS) posterior samples. For each dataset we randomly split it 3:1 into the database and queries. We observe whether we can preserve retrieval performance after selecting informative posterior samples.

Landmine. The data consist of 29 experiments: each experiment is a classification task for detecting the presence of either landmine (1) or clutter (0) from 9 input features. Each experiment has been collected from either a highly foliated region or a desert-like region. Thus they can be split in two classes (16-13). We observe that this is a relatively simple problem in the sense that the classes are well

separated, and thus a few posterior samples are sufficient for good retrieval performance. Due to the same reason, the proposed approach is able to retain the retrieval performance using only very few posterior samples.

Computer. The data consist of 200 experiments: each experiment is a prediction task of how a student rates 20 computers in the scale 0-10. Each computer is described in 13 binary features. Thus, each experiment $\mathbb{R}^{13} \to \mathbb{R}$ has about 20 samples (some entries missing). Since there are no obvious ground truth labels, we measure how well the proposed approach can reduce the number of posterior samples while preserving rankings. We observe that the problem is relatively simple since even a few posterior samples have been able to preserve the ranking with respect to $\widehat{\mathrm{ML}}_{q|d}$. However, the number of samples stored is larger than in the previous example since there is no clear clustering.

Restaurant. The data consist of 119 experiments: each experiment is a prediction task of how a customer rates 130 restaurants in the scale 1-3. All customers do not rate all available restaurants, and so the number of observations in each experiments varies, from 3-18. We select 7 categorical features for each experiment and binarize them, resulting in a $\mathbb{R}^{22} \to \mathbb{R}$ regression problem. We observe that this problem is more difficult in the sense that performance drops when the number of samples is decreased. However, the proposed approach has been able to collect essential samples to preserve the true rank better.

LINCS. The LINCS (Library of Integrated Network-based Cellular Signatures) data consist of 65 experiments, each measuring post-treatment gene expression values in response to a specific drug[2]. The model for each experiment is a prediction model from the post-treatment gene expression values to drug toxicity: 959 gene expression[3] values have been measured over 26-44 cell lines for each drug, thus the equivalent regression problem is $\mathbb{R}^{959} \to \mathbb{R}$. Drug toxicity values were acquired from CTD[2] (Cancer Target Discovery and Development). We observe that we achieve distinctively better performance over random sampling.

5 Discussion

This paper is intended to be a proof of concept towards a potentially highly useful community effort of extending experiment databanks to include also knowledge of the experimenters in a rigorously reusable form, as models. As of now, this is highly non-standard yet would be beneficial since the experimenter alone is best acquainted with his/her measurements and is able to train the most sensible model by incorporating his/her experience as prior knowledge. Storing models of experiments can, however, be cumbersome since most often they are not expressed in an analytic form. A widely applicable alternative is to store samples of the posterior; we suggested approaches to select the most informative posterior

[2] Personal communication with Dr. Subramanian, Broad Institute.
[3] Originally 978.

samples to store. Notice that posterior samples can be generated also when one has an analytic posterior. We have presented a set of convincing results on simulated data with regression as a task, as well as on standard real datasets.

Acknowledgments. This project is partly supported by the Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, 251170), and the Aalto University MIDE (Multidisciplinary Institute of Digitalisation and Energy) research programme. The calculations presented above were performed using computer resources within the Aalto University School of Science "Science-IT" project. Collection of gene expression values in LINCS data has been supported by the Broad Institute LINCS center 5U54HG006093 from the NIH Common Fund. The authors thank Suleiman Ali Khan for his help with the LINCS experiment preparation.

References

1. Rustici, G., Kolesnikov, N., Brandizi, M., Burdett, T., Dylag, M., Emam, I., Farne, A., Hastings, E., Ison, J., Keays, M., Kurbatova, N., Malone, J., Mani, R., Mupo, A., Pedro Pereira, R., Pilicheva, E., Rung, J., Sharma, A., Tang, Y.A., Ternent, T., Tikhonov, A., Welter, D., Williams, E., Brazma, A., Parkinson, H., Sarkans, U.: ArrayExpress update–trends in database growth and links to data analysis tools. Nucleic Acids Research 41, D987–D990 (2013)
2. Baumgartner Jr., W.A., Cohen, K.B., Fox, L.M., Acquaah-Mensah, G., Hunter, L.: Manual curation is not sufficient for annotation of genomic databases. Bioinformatics 23, i41–i48 (2007)
3. Buntine, W., Lofstrom, J., Perkio, J., Perttu, S., Poroshin, V., Silander, T., Tirri, H., Tuominen, A., Tuulos, V.: A scalable topic-based open source search engine. In: Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence, pp. 228–234 (2004)
4. Burges, C.J.C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.N.: Learning to rank using gradient descent. In: ICML, pp. 89–96 (2005)
5. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. J. Mach. Learn. Res. 9, 1871–1874 (2008)
6. Dutta, R., Seth, S., Kaski, S.: Retrieval of experiments with sequential Dirichlet process mixtures in model space. arXiv:1310.2125 [cs, stat] (2013)
7. Muandet, K., Fukumizu, K., Dinuzzo, F., Schlkopf, B.: Learning from distributions via support measure machines. arXiv e-print 1202.6504 (2012)
8. Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multi-task learning for classification with Dirichlet process priors. Journal of Machine Learning Research 8, 35–63 (2007)
9. Caldas, J., Gehlenborg, N., Faisal, A., Brazma, A., Kaski, S.: Probabilistic retrieval and visualization of biologically relevant microarray experiments. Bioinformatics 12, i145–i153 (2009)
10. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems 19, pp. 41–48. MIT Press, Cambridge (2007)
11. Vargas-Govea, B., González-Serna, J.G., Ponce-Medellín, R.: Effects of relevant contextual features in the performance of a restaurant recommender system. In: Workshop on Context Aware Recommender Systems (CARS) (2011)