



Thompson Sampling Based Active Learning in Probabilistic Programs with Application to Travel Time Estimation

Sondre Glimsdal^(✉) and Ole-Christoffer Granmo^(✉)

Centre for Artificial Intelligence Research, University of Agder, Grimstad, Norway
{sondre.glimsdal,ole.granmo}@uia.no

Abstract. The pertinent problem of Traveling Time Estimation (TTE) is to estimate the travel time, given a start location and a destination, solely based on the coordinates of the points under consideration. This is typically solved by fitting a function based on a sequence of observations. However, it can be expensive or slow to obtain labeled data or measurements to calibrate the estimation function. Active Learning tries to alleviate this problem by actively selecting samples that minimize the total number of samples needed to do accurate inference. Probabilistic Programming Languages (PPL) give us the opportunities to apply powerful Bayesian inference to model problems that involve uncertainties. In this paper we combine Thompson Sampling with Probabilistic Programming to perform Active Learning in the Travel Time Estimation setting, outperforming traditional active learning methods.

Keywords: Thompson Sampling · Distance Estimation · Probabilistic Programming · Active Learning · Travel Time Estimation

1 Introduction

An important part of traveling is estimating travel time. Without knowing the time it will take to travel between two locations it can be difficult to plan ahead and ensure that things go according to plan. Many services already provide good estimates for well-known scenarios such as car travel and public transportation. For these services, estimates are typically based on first determining a route and then adding up the individual components of that route to obtain the total travel time.

However, in many situations, the navigation system may fail to provide adequate information to form a route, leaving it unable to provide travel time estimates. These situations could occur for instance when hiking cross country or when traveling in areas where shortcuts and obstacles that do not appear on maps are frequent, such as in urban city centers. An alternative approach is

to focus on estimating the *true road distance*. While this is an interesting approach, the data required for estimation is significantly harder to gather. Not only does one need a timekeeping device, one also need some way to accurately track velocity. We avoid this by instead focusing on the actual time it takes to travel between two points.

Definition 1. A *Travel Time Estimation Function (TTEF)* is a function $R^d \times R^d \rightarrow R$, where a is the start location, b is the destination, and θ is the parameters learned from the observations.

The challenge is thus to determine θ by observing the actual travel time between locations and generalize from those observations. To minimize this calibration cost, the number of observations should be kept at a minimum. An important task is therefore to gather information in such a manner that each observation maximizes the gain in estimation accuracy. The objective of this paper is to address the calibration of TTEFs using as few data points as possible. We achieve this by formulating the problem as an Active Learning problem.

1.1 Active Learning

Active Learning (AL) has emerged as an effective tool for bridging supervised learning and unsupervised learning [2, 12]. The settings where supervised learning thrive are those abundant with labeled data, for instance sentiment analysis for movie reviews where the reviews typically have been assigned e.g. a star-rating by the reviewer, allowing the collection of large amounts of labeled data [9].

This is in contrast to other fields such as medical imaging, where one often need human experts to manually label the data. In such cases, it becomes pertinent with learning algorithms that maximize the information gained from each labeled example. An active learner operates by carefully selecting the most beneficial example to be labeled, with the result that fewer examples have to be labeled in total, while simultaneously performing as well as a passive learner, i.e., a learner that simply observes the labeled examples.

The active learning paradigm can roughly be divided into two parts based on the nature of the unsupervised examples, it is either *pool-based* where all the examples are available without a label, or *stream-based* where the examples are given as a stream, feeding one example at a time. In our novel variant of the TTE, the data is neither stream based nor pool based, it is instead a hybrid between the two types of AL. In TTE the learner is faced with a stream of pools, where the learner may only select one example from each pool, discarding the rest, as clarified below.

1.2 Active Learning in Travel Time Estimation

We define the data generating process of TTE as follows. An observer is standing on a location $a_{t=1}$ and then has to select a destination from a set or pool of n distinct locations, $D_{t=2} = \{d_1, d_2, \dots, d_n\}$. Once a destination $a_{t=2} \in D_{t=2}$ is

selected the observer travel from $a_{t=1}$ to $a_{t=2}$ and record the travel time δ_1 . This process is then repeated with $a_{t=2}$ as the new starting location. A new destination $a_{t=3}$ needs to be selected, now from $D_{t=3}$, and we obtain δ_2 – the travel time between $a_{t=2}$ and $a_{t=3}$. An important factor that makes TTE more difficult is that the observation δ_t does not only depend on a_t but on a_{t-1} as well.

1.3 Probabilistic Programming

Probabilistic Programming (PP) is an attempt to close the representation gap between the much celebrated probabilistic graphical models (PGM) such as Bayesian Networks and Markov Networks and the more specialized algorithms that are typically represented as a mixture of pseudo code, natural language, and mathematics. The idea is to formulate the entire model, from sample generation to the joint distribution in a unified representation framework, and let the underlying architecture handle the inference. This alleviates the need for highly specialized algorithms and lets the designer focus on designing a correct model, rather than focusing on models that are easy to do inference on. With the advances in computational power, a wide array of PPL have appeared in the literature. In this paper we employ PyMC3 [11], which is built on top of the Theano framework [15].

1.4 Paper Contributions

In this paper, we demonstrate the effectiveness of using Probabilistic Programming to solve the TTE problem, while simultaneously applying Thompson Sampling based Active Learning to minimize the number of observations required. To further investigate the effectiveness of this approach we also show that it performs comparable to traditional baselines for active learning on a well know regression problem [4].

2 Related Work

2.1 Active Learning

The highly effective Query By Committee (QBC) [4,13] algorithm is based on the premise that a committee of unique learners label each potential data point. That is, in a pool-based setting each data point in the pool is labeled by each learner. The next data point to obtain a label for is simply the data point where the learners disagree the most. For the simple case with binary labeled points and two learners, any point where the two learners disagree is considered as the next query point. In cases where the labels are real-valued, an alternative approach is to select the point that is expected to reduce prediction error the most [4]. For real-valued regression problems, the data point that maximizes the variance of the training set after being added is selected [3].

A critical aspect of the QBC algorithm is the disagreement between the learners. In the original work [13] a randomized algorithm was used. However, a more general approach is to train the same algorithm on different subsets of the data, as in query by bagging and query by boosting [8].

Bandit based active learning is a well-explored area of research [1, 5, 10], but this class of approaches is ill-suited to the TTE problems due to the simple fact that they require a pool based approach where one can track the uncertainty for each possible query-point as part of the active learning.

2.2 Distance Estimation

The field of Distance Estimation (DE) has primarily been dominated by the use of parameterized functions of a simple yet effective form. These functions are then calibrated using a set of inter-connected points and their distances [7] by maximizing the Goodness of Fit (GoF) between the observed values and the underlying function. Recently, an Adaptive Tertiary Search (ATS) based method that does not explicitly depend on GoF was proposed [6]. Instead, this method depends on the sign between the estimated distance and the actual, observed distance, and can thus be seen as a form of gradient descent.

To be consistent with previous work we will restrict ourself to the family of Weighted LP functions:

$$\text{W-L}_p(X) = k(\sum |x_i|^p)^{1/p}$$

where k is the linear weight, and $p \in R^+$ denotes the p-norm.

3 Active Learning with Thompson Sampling for Travel Time Estimation

The principle of Thompson Sampling (TS) can be summarized as follows. Given a distribution $\pi(\theta)$ over a parameter θ to be estimated, we sample an instance s from $\pi(\theta)$. We then assume that s is, in fact, the correct underlying value for θ . Thus, we explore by assuming that s is optimal and gather information I_s that we use to update the distribution over $\pi_{t+1}(\theta | I_s)$. Consequently, we also exploit our previous knowledge as the distribution over θ gets sharper towards the optimal value as t increases.

In the context of active learning in a probabilistic program, the objective of TS is to *convince* the maximum a-posteriori (MAP) model M_{map} that the TS sampled model M_{ts} is optimal by selecting the observation $o \in O$ such that the difference between $M_{\text{map}}(o)$ and $M_{\text{ts}}(o)$ is minimized after observing o .

In contrast, QBC is based on generating a committee $M_{\text{map}}^{(i)}, i = 1, 2, \dots, m$ where each MAP estimate is based on a different subset of the data. This inher-

ently means that the quality of an individual committee $M_{\text{map}}^{(i)}$ is worse than the MAP estimate of a TS model M_{map} that employs all available data.

Algorithm 1. The TS-PPL algorithm

Result: A program $P(T)$ based on T observations.
 Set a prior program $P(0)$;
for $t = 1, 2, \dots, T$ **do**
 $M_{\text{map}}(t) = \text{MAP}(P(t-1))$;
 $M_{\text{ts}}(t) \sim P(t-1)$;
 $o(t) = \text{argmax}_{o_i \in O(t)} [M_{\text{map}}(o_i; t) - M_{\text{ts}}(o_i; t)]^2$;
 $y(t) = \text{Query}(o(t))$
 $P(t) = P(t-1) \leftarrow o(t), y(t)$
end

4 Experiments

To demonstrate the efficiency of TS-PPL we apply it to two different problems. First, we investigate the performance for learning real-valued functions as done in [2]. Second, we investigate how it perform in the Travel Time Estimation problem. The metric of interest for the experiments will be the head-to-head results generated from identical experimental data and model. That is, the trials will be identical except the choice of observations to label. The objective function is to minimize the error on a separate hold-out set and thus, the cumulative error E_T is the sum of errors from $t = 0$ to $t = T$. The head-to-head metric between A and B is therefore the fraction of trials where scheme A have a lower cumulative error than B at the reported time-step t , i.e. $\sum_{i=0}^N \mathbb{1}[E_t^{(i)}(A) < E_t^{(i)}(B)]/N$.

4.1 Active Learning of Real-Valued Functions

The objective in a real-valued function is to minimize the generalization error between the learned function and the underlying true function, e.g. the difference in the area under the curve. We will now test TS-PPL with a standard function learning experimental setup [2], that have an underlying true function as shown in Eq. 1, with $z = \frac{x-0.2}{0.4}$, $a = 1, b = -1, c = 0$ and $\epsilon \sim N(0, 0.1^2)$.

$$f(x) = ax^2 + bx + c + \delta \frac{z^3 - 3z}{\sqrt{6}} + \epsilon \quad (1)$$

Algorithm 2. The PP for the function regression problem

$a \sim N(\mu = 1, \sigma = 2)$
 $b \sim N(\mu = 1, \sigma = 2)$
 $c \sim N(\mu = 1, \sigma = 2)$
 $s \sim |N(\mu = 0, \sigma = 10)|$
 $\mu_t = ax^2 + bx + c$
 $y_{\text{obs}} \sim N(\mu = \mu_t, \sigma = s \mid y)$

The available observations are drawn from $N(0.2, 0.4^2)$ and presents a serious challenge for QBC to outperform due to the Signal-To-Noise Ratio (SNR) of $0.4^2/0.3^2 = 1.8$ shift between the underlying function and the test distribution that the candidate points are drawn from. In Table 1 we observe that the standard QBC outperform the TS-PPL algorithm when the assumed function type is approximately correct ($\delta = 0.005$). However, it is outperformed when the difference between the assumed model and the underlying model is large ($\delta = 0.05$).

Table 1. The result of head-to-head comparisons between the different methods based on 5k trials in the function approximation scenario. The data is given in the format X/Y where X is the fractions of wins in head-to-head matches from $t = 20$ and Y is the fraction of wins from $t = 40$.

	δ	arms = 5	arms = 10	arms = 20	arms = 100
TS vs QBC	0.05	0.56/0.50	0.48/0.43	0.55/0.42	0.56/0.53
TS vs Passive	0.05	0.58/0.57	0.69/0.61	0.66/0.64	0.84/0.83
QBC vs Passive	0.05	0.57/0.61	0.65/0.64	0.62/0.65	0.76/0.75
TS vs QBC	0.005	0.49/0.48	0.46/0.45	0.41/0.41	0.51/0.51
TS vs Passive	0.005	0.55/0.64	0.71/0.67	0.72/0.69	0.71/0.72
QBC vs Passive	0.005	0.58/0.62	0.68/0.69	0.77/0.73	0.68/0.71

4.2 Travel Time Estimation

Similar to [6], we conduct the TTE experiments on publicly available data from the TSPLIB Symmetric Traveling Salesman Problem Instances (MP-TESTDATA) [14] with $N=29$. The $O(t) = \{(x_i, y_i)\}^n$ pairs available for observations at time t is drawn from $x_i \sim U(0, x_{\max}), y_i \sim U(0, y_{\max})$ where $x_{\max} = 2300$ and $y_{\max} = 1900$. The purpose is to draw the observations uniformly from the entire dataset.

The oracle computes the travel time from \mathbf{a} to \mathbf{b} , denoted $Q(\mathbf{a}, \mathbf{b})$, as

$$\|\mathbf{a} \rightarrow \mathbf{p}\|_{L_1} + \text{TravelTime}(\mathbf{p}, \mathbf{q}) + \|\mathbf{q} \rightarrow \mathbf{b}\|_{L_1} \quad (2)$$

where \mathbf{p}, \mathbf{q} is the closest points in the dataset to \mathbf{a} and \mathbf{b} respectively and $\text{TravelTime}(\mathbf{p}, \mathbf{q})$ is provided by the dataset.

The PP used is defined as a Bayesian prior over the W- L_p model from [6]. and is as follows:

Algorithm 3. The PP for the Travel Time problem

$$\begin{aligned} k &\sim N(\mu = 1, \sigma = 2) \\ p &\sim N(\mu = 1, \sigma = 2) \\ s &\sim |N(\mu = 0, \sigma = 50)| \\ \mu_t &= k \|l_i - l_{i-1}\|_p \\ y &\sim N(\mu = \mu_t, \sigma = s \mid t_1, \dots, t_n) \end{aligned}$$

Table 2. The result of head-to-head comparisons between the different methods based on 5k trials. The data is given in the format X/Y where X is the fractions of wins in head-to-head matches from $t = 20$ and Y is the fraction of wins from $t = 40$.

	arms = 2	arms = 5	arms = 10
TS-PPL vs QBC	0.65/0.64	0.54/0.58	0.61/0.61
TS-PPL vs Passive	0.55/0.58	0.70/0.71	0.72/0.72
QBC vs Passive	0.45/0.46	0.64/0.57	0.72/0.65
	arms = 20	arms = 50	arms = 100
TS-PPL vs QBC	0.64/0.62	0.66/0.64	0.50/0.57
TS-PPL vs Passive	0.70/0.70	0.63/0.70	0.69/0.66
QBC vs Passive	0.60/0.60	0.57/0.62	0.69/0.60

The results for comparing between OBC and TS-PPL is found in Table 2. From the results, it is quite clear that TS-PPL outperforms QBC as well as Passive for the TTE problem achieving near 10% better results. This indicates that when the problem is not a simple regression problem, anchoring the selection process in the MAP estimate, as done in TS, gives a better trade-off than anchoring in the variance over a committee.

5 Conclusion

We have proposed TS-PPL an effective scheme for performing Active Learning in Probabilistic Programs. We have shown that TS-PPL can be applied to both a standard regression problem and a more complex problem in the Travel Time Estimation problem. Our method significantly outperforms the strong baseline of Query by Committee as well as passive learning for Travel Time Estimation. TS-PPL further gives competitive results in the case of regression.

References

1. Bouneffouf, D., Laroche, R., Urvoy, T., Feraud, R., Allesiardo, R.: Contextual bandit for active learning: active Thompson sampling. In: Loo, C.K., Yap, K.S., Wong, K.W., Teoh, A., Huang, K. (eds.) ICONIP 2014. LNCS, vol. 8834, pp. 405–412. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12637-1_51
2. Burbidge, R., Rowland, J.J., King, R.D.: Active learning for regression based on query by committee. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 209–218. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77226-2_22
3. Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. *J. Artif. Intell. Res.* **4**, 129–145 (1996)
4. Freund, Y., Seung, H.S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. *Mach. Learn.* **28**(2–3), 133–168 (1997)

5. Ganti, R., Gray, A.G.: Building bridges: viewing active learning from the multi-armed bandit lens. arXiv preprint [arXiv:1309.6830](https://arxiv.org/abs/1309.6830) (2013)
6. Havelock, J., Oommen, B.J., Granmo, O.C.: Novel distance estimation methods using “stochastic learning on the line” strategies. *IEEE Access* **6**, 48438–48454 (2018)
7. Love, R.F., Morris, J.G.: Modelling inter-city road distances by mathematical functions. *J. Oper. Res. Soc.* **23**(1), 61–71 (1972)
8. Mamitsuka, N.A.H., et al.: Query learning strategies using boosting and bagging. In: *Machine Learning: Proceedings of the Fifteenth International Conference (ICML 1998)*, vol. 1 (1998)
9. McAuley, J., Yang, A.: Addressing complex and subjective product-related queries with customer reviews. In: *Proceedings of the 25th International Conference on World Wide Web*, pp. 625–635. *International World Wide Web Conferences Steering Committee* (2016)
10. Osugi, T., Kim, D., Scott, S.: Balancing exploration and exploitation: a new algorithm for active machine learning. In: *Fifth IEEE International Conference on Data Mining*, 8-p. IEEE (2005)
11. Salvatier, J., Wiecki, T.V., Fonnesbeck, C.: Probabilistic programming in python using PYMC3. *PeerJ Comput. Sci.* **2**, e55 (2016)
12. Settles, B.: *Active learning literature survey*. Computer Sciences Technical report 1648, University of Wisconsin-Madison (2009)
13. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 287–294. ACM (1992)
14. Skorobohatyj, G.: MP-TESTDATA—the TSPLIB symmetric traveling salesman problem instances. <http://elib.zib.de/pub/mptestdata/tsp/tsplib/tsp/index.html>. Accessed 16 Nov 2015
15. Theano Development Team: Theano: a Python framework for fast computation of mathematical expressions. arXiv e-prints [abs/1605.02688](https://arxiv.org/abs/1605.02688), May 2016. <http://arxiv.org/abs/1605.02688>