The Journal of Open Source Software

Efficiently Learning Relative Similarity Embeddings with Crowdsourcing

Scott Sievert[1], **Robert Nowak**[1], **and Timothy Rogers**[1]

1 University of Wisconsin–Madison

DOI: 10.21105/joss.04517

Software

- Review
- Repository
- Archive

Editor: Andrew Stewart

Reviewers:

- @hoechenberger
- @stain
- @jorgedch

Submitted: 31 May 2022
Published: 17 April 2023

Summary

Social scientists often investigate human reasoning by collecting relative similarity judgements with crowdsourcing services. However, this often requires too many human responses to be practical for large experiments. To address this problem, we introduce software called Salmon, which makes intelligent choices on query selection (aka active machine learning or adaptive sampling) while collecting relative similarity judgments from crowdsourcing participants. Salmon is usable by experimentalists because it requires little to no programming experience and only requires an Amazon AWS account for launching (though a local install is available). Extensive simulations and experiments suggest that Salmon requires 2 to 3 times fewer response than random sampling.

Statement of need

Relative similarity judgments take the form "is item a or b more similar to item h?" These queries work well with human working memory limitations, and have been used successfully to characterize human perceived similarity between faces (Sankaranarayanan et al., 2016), vehicles (Kuma et al., 2019) and shoes (Heim et al., 2015b).

Typically, experimentalists require an inordinate number of human responses (about 10,000) to produce an accurate embedding when making a similarity map in $d = 2$ dimensions of $n = 50$ chemistry molecules (Mason et al., 2019). The number of human responses required will scale like $\mathcal{O}(nd \log n)$, which means that asking about $n = 100$ molecules for $d = 3$ dimensions will likely require about 35,000 responses.

Many "active machine learning" methods have been proposed to reduce the number of queries required (Tamuz et al., 2011; Van Der Maaten & Weinberger, 2012). These show gains, at least offline when computation is not a limitation. However, the online deployment of these algorithms has posed more challenges (Jamieson et al., 2015).

Related work

Systems to deploy active machine learning (ML) algorithms to crowdsourcing audiences include SMART (Chew et al., 2019), NEXT (Jamieson et al., 2015) and Microsoft's Multiworld Testing Decision Service (Agarwal et al., 2016). The most relevant related work, NEXT is capable of serving triplet queries to crowdsourcing participants (Jamieson et al., 2015). In this work the authors concluded that "there is no evidence for gains from adaptive sampling." However, other work has found gains from adaptive sampling when computation is not a priority (Heim et al., 2015b).

Several active algorithms for triplet embedding have been developed (Tamuz et al., 2011; Van Der Maaten & Weinberger, 2012). These algorithms require searching queries and fitting the responses to the underlying noise model. With a naive computation, scoring a single query requires $\mathcal{O}(nd)$ floating point operations (FLOPs), and the embedding typically requires significant computation (Ma et al., 2021; Vankadara et al., 2019), though some work has been done to reduce the amount of computation (Heim et al., 2015a).

Design goals

Salmon's main design goals are below:

1. Generate accurate relative similarity embeddings.
2. Require fewer responses than random sampling to generate an embedding.
3. Allow experimentalists to easily achieve both items above.

One method to achieve goal (2) above is to use an active machine learning (ML) sampling algorithm. This task requires considering how to create a responsive query page with a service to run active ML algorithms. The result is a frontend server that *serves* queries and *receives* answers, and a backend server that *searches* queries and *processes* answers – notably, not the same data flow that NEXT has (Jamieson et al., 2015), though it is common in other systems (Agarwal et al., 2016; Chew et al., 2019).

To verify goal (2), extensive crowdsourcing experiments and simulations have been run, and have compared with the most relevant work (Jamieson et al., 2015). In this, Salmon's architecture required modification of the query search algorithm to circumvent some experimental design issues. With these modifications, we have observed active ML algorithm gains in extensive experiments and simulations. To the best of the author's knowledge, this is a novel achievement in the crowdsourcing context.

Goal (1) is aided by the fact that Salmon integrates a popular deep learning framework, PyTorch (Paszke et al., 2019). This allows for easy customization of the underlying optimization method during both online and offline computation, including by the experimentalist managing Salmon if so desired.

Goal (3) is enabled by a relatively simple launch through Amazon AWS using Amazon Machine Images (AMIs).[1] The AMI for Salmon[2] pulls the latest release of Salmon from GitHub and then launches Salmon. After some other tasks (e.g., opening ports, etc), Salmon is ready be launched. Salmon requires fairly minimal computational resources; all the experiments and simulation were performed with `t3.xlarge` Amazon EC2 instance, which has 4 cores, 16GB of memory and costs about $3.98 per day.

After launch, Salmon can start an experiment with stimuli consisting of text, images, video or HTML strings. It provides a mechanism to monitor an ongoing experiment, which includes the following information:

- **Basic experiment statistics:** number of unique users, launch date, etc.
- **Server performance:** processing time for different endpoints, rate responses received, etc.
- **Client timings,** including response and new query latency.
- **Embedding visualization** and a list of targets in the embedding.

In addition, Salmon provides links to download the responses and configuration. Salmon also supports experiment persistence through downloading and uploading experiments. The embedding that Salmon generates can be downloaded, at least if active samplers are used. Regardless of the sampler used, Salmon can be used to generate the embeddings offline from the downloaded responses.

[1] A local install is available, and only requires Docker. Collection of crowdsourced responses will require running a web server or collecting in-person responses (though a local install may be useful for development).
[2] Details are at https://docs.stsievert.com/salmon/installation

Uses

Salmon has been used by several groups, including psychologists at the University of Wisconsin–Madison and the Louisiana State University.

Acknowledgments

This work has been supported by the SMART Scholarship Program.

References

Agarwal, A., Bird, S., Cozowicz, M., Hoang, L., Langford, J., Lee, S., Li, J., Melamed, D., Oshri, G., Ribas, O., & others. (2016). Making contextual decisions with low technical debt. *arXiv Preprint arXiv:1606.03966*. https://arxiv.org/pdf/1606.03966.pdf

Chew, R., Wenger, M., Kery, C., Nance, J., Richards, K., Hadley, E., & Baumgartner, P. (2019). SMART: An open source data labeling platform for supervised learning. *Journal of Machine Learning Research*, *20*(82), 1–5. http://jmlr.org/papers/v20/18-859.html

Heim, E., Berger, M., Seversky, L. M., & Hauskrecht, M. (2015a). Efficient online relative comparison kernel learning. *Proceedings of the 2015 SIAM International Conference on Data Mining*, 271–279. https://doi.org/10.1137/1.9781611974010.31

Heim, E., Berger, M., Seversky, L., & Hauskrecht, M. (2015b). Active perceptual similarity modeling with auxiliary information. *arXiv Preprint arXiv:1511.02254*. https://arxiv.org/pdf/1511.02254.pdf

Jamieson, K. G., Jain, L., Fernandez, C., Glattard, N. J., & Nowak, R. (2015). NEXT: A System for Real-World Development, Evaluation, and Application of Active Learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 28). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/89ae0fe22c47d374bc9350ef99e01685-Paper.pdf

Kuma, R., Weill, E., Aghdasi, F., & Sriram, P. (2019). Vehicle re-identification: An efficient baseline using triplet embedding. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–9. https://doi.org/10.1109/IJCNN.2019.8852059

Ma, K., Zeng, J., Xiong, J., Xu, Q., Cao, X., Liu, W., & Yao, Y. (2021). Fast stochastic ordinal embedding with variance reduction and adaptive step size. *IEEE Transactions on Knowledge and Data Engineering*, *33*(6), 2467–2478. https://doi.org/10.1109/TKDE.2019.2956700

Mason, B., Rau, M. A., & Nowak, R. (2019). Cognitive task analysis for implicit knowledge about visual representations with similarity learning methods. *Cognitive Science*, *43*. https://doi.org/10.1111/cogs.12744

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

Sankaranarayanan, S., Alavi, A., Castillo, C. D., & Chellappa, R. (2016). Triplet probabilistic embedding for face verification and clustering. *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 1–8. https://doi.org/10.1109/BTAS.2016.7791205

Tamuz, O., Liu, C., Belongie, S., Shamir, O., & Kalai, A. T. (2011). Adaptively learning the crowd kernel. *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 673–680. ISBN: 9781450306195

Van Der Maaten, L., & Weinberger, K. (2012). Stochastic triplet embedding. *2012 IEEE International Workshop on Machine Learning for Signal Processing*, 1–6. https://doi.org/10.1109/MLSP.2012.6349720

Vankadara, L. C., Haghiri, S., Lohaus, M., Wahab, F. U., & Luxburg, U. von. (2019). Insights into ordinal embedding algorithms: A systematic evaluation. *arXiv Preprint arXiv:1912.01666*. https://arxiv.org/abs/1912.01666