Distributed, Real-Time Bayesian Learning in Online Services

Ralf Herbrich
Facebook Inc.
1601 Willow Road
Menlo Park, CA 94025, US
ralf@fb.com

ABSTRACT

The last ten years have seen a tremendous growth in Internet-based online services such as search, advertising, gaming and social networking. Today, it is important to analyze large collections of user interaction data as a first step in building predictive models for these services as well as learn these models in real-time.

One of the biggest challenges in this setting is scale: not only does the sheer scale of data necessitate parallel processing but it also necessitates distributed models; with over 900 million active users at Facebook, any user-specific sets of features in a linear or non-linear model yields models of a size bigger than can be stored in a single system.

In this talk, I will give a hands-on introduction to one of the most versatile tools for handling large collections of data with distributed probabilistic models: the sum-product algorithm for approximate message passing in factor graphs. I will discuss the application of this algorithm for the specific case of generalized linear models and outline the challenges of both approximate and distributed message passing including an in-depth discussion of expectation propagation and Map-Reduce.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*

Keywords

Graphical Models, Distributed Machine Learning, Gaming

1. SUMMARY

Over the past decade, the emergence of online services has changed one of the implicit assumptions of machine learning made thus far, namely that the amount of data available for learning can be processed in real-time sequentially by a single computer. Today, rather than having to be concerned about the lack of data and compensating with sophisticated and computationally intensive methods such as structural priors and regularizers, we live in a world where we have more training data available than can be handled by a single computer.

For example, in a single day, an average of 526 million users generate more than 3.2 billion likes and comments on the Facebook platform alone[1]. Such data-streams, which are often the digitization of people interacting with the world, each other or search results, are no longer generated as a single stream but get recorded and analyzed in a fully distributed manner. For example, it is estimated that Google uses around 1 million servers to drive its platform[2].

This shift in data availability necessitates reconsidering machine learning methods from a scaling-out point of view — *distributed machine learning.* In distributed machine learning, we are given a distributed *set of* data streams producing data of varying rate and are required to successively make a prediction as well as incorporating the data item in a distributed way; for conditional or discriminative models, the data is usually divided into a set of features together with a set of target values that *co-occur.* In this setting, key performance criteria for such systems include the ability to be fault-tolerant, i.e., what fraction of machines can fail to still operate successfully, the ability to be state-less, i.e., how many additional bits need to be communicated per request in order to perform prediction and/or training, and the ability to be adaptive to varying rates of data due to distribution shifts in the data.

In this talk, I will present the *sum-product algorithm* [2] and expectation propagation for approximate inference—one of the key building blocks for distributed machine learning. I will discuss how to map the probabilistic messages to network messages and show that the algorithm can be viewed as a real-time generalization of the famous Map-Reduce framework [1] which is a single-pass message exchange in a tree-approximation to a probabilistic model. I will also give an overview of some distributed learning systems already in use in industry applications including the *TrueSkill* ranking and matchmaking system in Xbox Live and collaborative filtering systems used for content recommendation at Facebook.

2. REFERENCES

[1] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *In Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation*, pages 137–150. USENIX Association, 2004.

[2] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions in Information Theory*, 47:498–519, 1998.

RecSys'12, September 9–13, 2012, Dublin, Ireland.
ACM 978-1-4503-1270-7/12/09.

[1] see `http://newsroom.fb.com/`.

[2] see `http://www.datacenterknowledge.com/`.