Theoretical Computer Science 410 (2009) 1765–1780

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Multiple pass streaming algorithms for learning mixtures of distributions in $\mathbb{R}^d$

Kevin L. Chang*

Yahoo! Labs, Yahoo! Inc, 701 First Avenue, Sunnyvale, CA 94089, United States

ARTICLE INFO

Keywords:
Streaming algorithm
Machine learning
Mixture model
Computational learning theory

ABSTRACT

We present a multiple pass streaming algorithm for learning the density function of a mixture of k uniform distributions over rectangles in $\mathbb{R}^d$, for any $d > 0$. Our learning model is: samples drawn according to the mixture are placed in *arbitrary order* in a data stream that may only be accessed sequentially by an algorithm with a very limited random access memory space. Our algorithm makes $2\ell + 2$ passes, for any $\ell > 0$, and requires memory at most $\tilde{O}(\epsilon^{-2/\ell}k^2d^4 + (4k)^d)$, where ϵ is the tolerable error of the algorithm. This exhibits a strong memory-pass tradeoff in terms of ϵ: a few more passes significantly lower its memory requirements, thus trading one of the two most important resources in streaming computation for the other. Chang and Kannan first considered this problem for $d = 1, 2$.

Our learning algorithm is especially appropriate for situations where massive data sets of samples are available, but computation with such large inputs requires very restricted models of computation.

1. Introduction

The rise of machine learning as an invaluable data analysis paradigm has coincided with the proliferation of massive data sets that stress computer systems in ways that render traditional models of computation inadequate. These two important considerations necessitate the theoretical study of algorithms for machine learning and statistical analysis that respect the resource constraints imposed by massive data set computation.

Of paramount importance is the observation that a large data set will not fit into the main memory of a computer system, but rather must be stored on disk or optical drives. For such data, well-designed memory access patterns are crucial, since access to data requires physical movement within storage devices. An algorithm will thus incur large time penalties for each random access; for large data sets, frequent random access is highly undesirable. Random access can be eliminated and I/O optimized by instead reading the data in a sequential fashion. The **streaming model** addresses these concerns and is popular in the theoretical computer science literature. The first few problems examined in the streaming model include sorting and selection [11] and approximating frequency moments [1]. In the streaming model, data in storage is modeled as a read-only array that can only be accessed sequentially in passes over the entire array. The algorithm may make a few passes over the array and use a small random-access memory space (usually sublinear in size, since one cannot hope to store the entire data set in memory) and may take constant time to process each element of the array. We will refer to this random-access memory space as simply "memory". The important resources to be optimized are therefore passes and memory.

An important class of data mining and learning problems arises from generative clustering models. In these models, k clusters are defined by k probability distributions, $F_1, \ldots, F_k$, over some universe Ω, each of which is given a weight $w_i \geq 0$ such that $\sum_1^k w_i = 1$. If the F_is are density functions, then the mixture of these k distributions is defined by the density

* Tel.: +1 408 349 4699.
E-mail address: klchang@yahoo-inc.com.

0304-3975/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2009.01.043

function $F = \sum_1^k w_i F_i$. The natural interpretation of a point drawn according to the mixture is that distribution F_i is picked with probability w_i, and then a point is drawn according to F_i. We consider the problem of estimating the probability density function of the mixture F given samples drawn according to the mixture.

In this paper, we will study the problem of learning **mixtures of k uniform distributions over axis-aligned rectangles in $\mathbb{R}^d$**, for any $d > 0$. In this case, each F_i is a uniform distribution over some cell in d dimensions $R_i = \left\{x \in \mathbb{R}^d | a_1 \leq x_1 \leq b_1, \ldots, a_d \leq x_d \leq b_d\right\}$ for scalars $a_1, b_1, \ldots, a_d, b_d$. The R_is may intersect in arbitrary ways. Since the R_is are arbitrary, learning the R_is and w_is from a set of samples from the mixture is an ill-defined problem, since different sets of rectangles and weights, when "mixed", can form exactly the same distribution. Therefore, we will learn the density function, rather than the components, of the mixture. The output of the algorithm will be a function G that is an estimate of F.

One motivation behind learning mixtures of uniform distributions over rectangles is that these are among the simplest mixtures, and therefore any theory for learning mixture models in massive data set paradigms should start with this. Furthermore, these mixtures are building blocks for more complicated functions; continuous distribution in $\mathbb{R}^d$ can be approximated as a mixture of sufficiently many uniform distributions over rectangles in $\mathbb{R}^d$. Our algorithm can then be used to learn this mixture.

Our learning and computational model is that samples drawn according to the mixture F are placed in a data stream X, in **arbitrary order**.[1] Learning algorithms are required to be multiple-pass streaming algorithms, as described above. The output of the algorithm will be a function G that is an estimate of F, with error measured by L^1 distance: $\int_{\mathbb{R}^d} |F - G|$. An input parameter to the algorithm will be its probability of failure, $\delta > 0$. The approximation G will in general be *more complex* than simply the density function of a mixture of k uniform distributions.

Chang and Kannan [3] designed pass-efficient algorithms for learning a mixture of k uniform distributions over intervals in $\mathbb{R}$ and axis-aligned rectangles in $\mathbb{R}^2$. This work was subsequently improved by Guha and Mcgregor in [7]. In this paper, we use a similar high level approach, but develop new tools in order to generalize the algorithm to solve problems in an arbitrary dimension.

1.1. Our results

Our main result is a multiple-pass algorithm for learning a mixture of k uniform distributions in $\mathbb{R}^d$ with flexible resource requirements. The number of passes the algorithm may make is a function of an input parameter $\ell > 0$ that is independent of all other variables. The algorithm exhibits the power of multiple passes in the streaming learning model: if the algorithm is allotted just a few more passes and its error is held constant, then its memory requirements drop significantly as a function of ϵ. This is a strong **trade-off** between pass and memory requirements.

We will need the technical assumption that there exists a number $w > 0$, known to the algorithm, such that $F(x) \leq w$ for all $x \in \mathbb{R}^d$ and that all the probability mass of the mixture is contained in $[0, 1]^d$.

The main result of the paper is a $2\ell + 2$ pass algorithm that, with probability at least $1 - \delta$, will learn the mixture's density function to within L^1 distance ϵ and that uses memory at most

$$\tilde{O}\left(\frac{k^2 d^4}{\epsilon^{2/\ell}} + (4k)^d\right).$$

The algorithm requires the data stream to satisfy: $|X| = \tilde{\Omega}\left(\left(\frac{10}{8}\right)^{\ell} \frac{(kd)^{4d+3} w^{4d+2\ell}}{\epsilon^{4\ell d+5\ell}}\right)$.[2]

1.1.1. Discussion of results

We note that the sample complexity and the memory requirements of the algorithm are exponential in the dimension d; we partially justify the former requirement by our massive data set paradigm: we are working in the streaming model precisely because the data set is large. Despite this observation, the result is mostly of theoretical interest, since the sample complexity and memory requirements become unrealistic even for relatively modest values of d and k.

This does not preclude a strong pass-space trade-off for the algorithm, for many settings of the parameters d, k, ϵ. Since the memory requirement is $\tilde{O}(k^2 d^4 \epsilon^{-2/\ell} + (4k)^d)$, the trade-off between passes and memory is most significant in the case where the term involving ϵ dominates the memory requirement for small values of ℓ. This situation occurs when d and k are held constant and the tolerable error ϵ becomes very small. In this case, increasing ℓ will indeed reduce the memory requirement by very large factors. Again, this situation may be mostly of theoretical interest, since such small values of ϵ may not be required by applications.

1.2. Overview of methods

The main action of the algorithm is to learn the locations of the boundaries of the constituent mixture rectangles in $2\ell + 1$ passes. With this knowledge, **Learn(d, k)** can partition the domain into cells such that $F(x)$ is a constant function

[1] Assuming that the data are randomly ordered is not always realistic; for instance if the data were collected from the census, then perhaps it would be ordered by address or some other attribute.

[2] $\tilde{O}(\cdot)$, $\tilde{\Omega}(\cdot)$, $\tilde{\Theta}(\cdot)$ denote asymptotic notation with polylogarithmic factors omitted.