EP-MEANS: An Efficient Nonparametric Clustering of Empirical Probability Distributions

Keith Henderson
Lawrence Livermore National Laboratory
keith@llnl.gov

Brian Gallagher
Lawrence Livermore National Laboratory
bgallagher@llnl.gov

Tina Eliassi-Rad
Rutgers University
eliassi@cs.rutgers.edu

ABSTRACT

Given a collection of m continuous-valued, one-dimensional empirical probability distributions $\{P_1, \ldots, P_m\}$, how can we cluster these distributions efficiently with a nonparametric approach? Such problems arise in many real-world settings where keeping the moments of the distribution is not appropriate, because either some of the moments are not defined or the distributions are heavy-tailed or bi-modal. Examples include mining distributions of inter-arrival times and phone-call lengths. We present an efficient algorithm with a non-parametric model for clustering empirical, one-dimensional, continuous probability distributions. Our algorithm, called EP-MEANS, is based on the Earth Mover's Distance and k-means clustering. We illustrate the utility of EP-MEANS on various data sets and applications. In particular, we demonstrate that EP-MEANS effectively and efficiently clusters probability distributions of mixed and arbitrary shapes, recovering ground-truth clusters exactly in cases where existing methods perform at baseline accuracy. We also demonstrate that EP-MEANS outperforms moment-based classification techniques and discovers useful patterns in a variety of real-world applications.

1. INTRODUCTION

We address the following problem: given a collection of m continuous-valued, one-dimensional empirical probability distributions $\{P_1 \ldots P_m\}$, how can we *cluster* the distributions efficiently with a nonparametric approach? For example, consider airlines and their business models. Each airline operates a number of routes. The distances of these routes comprise an empirical distribution for each carrier. Can we use these distributions to discover a small number of typical business models for airlines? Several approaches are possible. For example, we could assume that the distributions are from some known family of (analytical) distributions, and find the best-fit parameters for each P_i. Then, the distributions can be clustered in the parameter space using some existing spatial clustering technique. However, when the various airlines have different underlying distributional families, then this approach is not suitable.

We postulate that an effective clustering algorithm for empirical, continuous-valued, one-dimensional probability distributions should be (1) *efficient*, (2) *nonparametric*, (3) *empirical*, (4) *distance-based*, and (5) *interpretable*. Let's discuss each of these in turn. **Efficient:** Let n_i be the number of observed values in P_i, and $N = \sum n_i$. Then the algorithm should be subquadratic in N–i.e., an $O(N^2)$ or worse algorithm is unacceptable. **Nonparametric:** Ideally, an effective clustering algorithm does not depend on choices of model size. Our proposed method uses k-means, a parametrized clustering algorithm for which many extensions have been proposed that eliminate the manual model selection. **Empirical:** If it is known that the observed distributions come from a given analytical family, then it makes sense to summarize each P_i by a parameterized best-fit distribution. However, we are interested in a general-purpose algorithm that can successfully cluster distributions without knowing the family information. The algorithm should be agnostic to the types of underlying distributions involved. **Distance-based:** There are a number of dissimilarity measures for distributions, but most of them are not true distance metrics. Examples include the Kullback-Leibler (KL) divergence and the Kolmogorov-Smirnov (KS) statistic. These methods ignore the underlying metric space from which distributions are drawn, and as such can produce poor clusters. **Interpretable:** Given a clustering, we would like to be able to say something meaningful about why the data was clustered in a certain way. In traditional (e.g., spatial) clustering, this is typically achieved by reporting the *centroid* of the cluster. Some techniques, such as multidimensional scaling [8], do not allow for easy interpretation.

We propose a novel algorithm, called EP-MEANS, which is based on two well-known techniques: *earth mover's distance (EMD)* and *k-means clustering*. k-means is a well-studied spatial clustering algorithm that uses expectation-maximization (EM) to compute k centroids and assign each observed point to its closest centroid. EMD is a distance metric on probability distributions. EP-MEANS is a new efficient algorithm, which computes each k-means iteration in $O(N \log(N))$ runtime, $O(N)$ space, and utilizes EMD.

1.1 Motivating Example

Returning to the example of airline-route distances, we collected data from *openflights.org*, which includes routes for hundreds of airlines as 3-tuples: ⟨Route ID, Airline ID, distance⟩. Each route contributes a single observation to the

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
SAC'15 April 13-17, 2015, Salamanca, Spain.
Copyright 2015 ACM 978-1-4503-3196-8/15/04 ...$15.00.
http://dx.doi.org/10.1145/2695664.2695860.

Cluster 1	**Cluster 2**	**Cluster 3**
Local and Regional	Domestic	International
Shenzhen	United	British Airways
Wizz Air	Ryanair	Korean Air
Xiamen	Delta	Emirates
Hellas Jet	American	Qatar Airways
Sichuan	US Airways	Transaero

Table 1: EP-MEANS clustering results on the openflights.org airline-route data. For each cluster, the top five airlines (by route count) are listed.

empirical distribution for its airline (i.e., we do not consider how many flights per day fly each route). Figure 1 highlights the results of EP-MEANS when applied to the airline-route data. Each cluster is labeled by the airline in that cluster with the most routes. The solid lines indicate cluster *centroids* (described in Section 3.2). Dashed lines indicate cluster averages—i.e., the normalized sum of all distributions in the cluster. For a given cluster, the difference between its centroid and its average distribution gives a rough idea of how "good" the cluster is—i.e., how closely its constituents resemble each other. Table 1.1 lists the top five airlines (by route count) in each cluster. Clusters are labeled by hand based on the approximate business models associated with their constituent airlines.

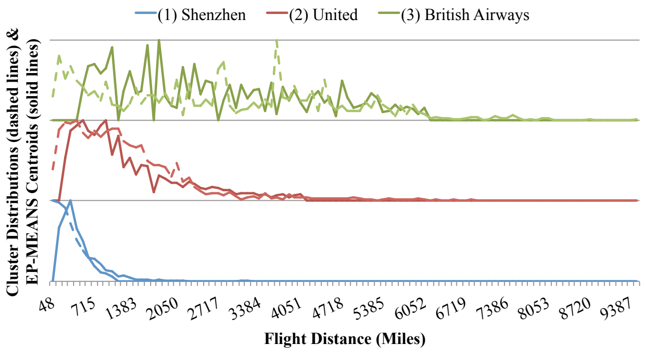

Figure 1: (Best viewed in color.) Results of clustering airlines based on route lengths. Solid lines are cluster centroids by EP-MEANS; dashed lines are average cluster distributions. Clusters are labeled by the busiest (most routes) airline in the cluster. Each distribution is scaled vertically to emphasize shape; the total area under each distribution is one.

The results on our motivating example demonstrate several desirable aspects of EP-MEANS. Examining the airlines included in each cluster, we see that they are roughly grouped by business model. Note that the distributions do not appear to come from the same distributional family, suggesting that analytical distributions may not be appropriate in this application. For instance, Cluster 3 is mostly major international airlines without as much regional service as domestic airlines. The average distribution (shown as a histogram in Figure 1) has a tail of longer flights, This makes sense, since major world cities are expected to be distributed roughly evenly throughout the globe. In contrast, Clusters 1 and 2 include airlines with more regional flights. This leads to increased probability mass in the sub-1000 mile range. Cluster 1 is comprised of short-hop airlines with almost exclusively local and regional service, while Cluster 2 contains domestic airlines with some international flights. Thus, distributional clustering can be a very powerful analysis tool for organizing unstructured data. Not only is it able to discover clusters with a variety of distributional shapes, but it is also interpretable. The centroids and average distributions give us insight into what behaviors the clusters represent.

Contributions. (1) We present EP-MEANS: an efficient, nonparametric, empirical, distance-based, and interpretable algorithm for clustering a collection of m continuous-valued one-dimensional empirical probability distributions. **(2)** Our extensive empirical study demonstrates the utility, efficacy, and effectiveness of EP-MEANS.

The outline of the paper is as follows: background and preliminaries in Section 2, proposed method in Section 3, experiments in Section 4, related work in Section 5, and conclusions in Section 6.

2. BACKGROUND AND PRELIMINARIES

2.1 Background

The primary decision that one must make when clustering distributions is how to represent the distributions and what dissimilarity measure is appropriate. Here we describe two simple approaches that will serve as baselines for our experiments. See Section 5 for more details.

Parameter Clustering. One of the simplest approaches to clustering empirical distributional data $\{P_1, \ldots, P_m\}$ is to fit a known analytical distribution to each P_i. For example, one can assume that each P_i is drawn from a Gaussian distribution. In that case, each distribution can be replaced by a best-fit Gaussian distribution with two parameters: mean and variance. Instances can be represented as points in $\mathbb{R}^2$ and clustered using any spatial clustering method (e.g., k-means). Parameter clustering is simple and can be powerful, but it fails when the distributional family is unknown or hard to express with a small number of parameters. While it has the advantage of using a true distance metric (Euclidian distance), it is not immediately obvious how to scale the dimensions properly. Section 4.2 presents experimental results supporting this claim.

Histogram Binning. A second simple approach is to divide the values of the distribution into B bins, and encode each empirical distribution as a B-dimensional vector of real values. Once these vectors are computed, one can simply apply existing spatial clustering techniques. This approach has several benefits. It is fast and does not require explicit knowledge about the types of distributions in the collection. It also allows one to directly apply spatial clustering algorithms, which have been studied extensively. However, binning has drawbacks that make it inappropriate for effective clustering. First, the number, size, and locations of the bins must be determined. For distributions with long tails, equally-spaced bins may not be effective as they tend to lose information about values near zero. Second, information about how far apart different bins are is lost. Each bin is considered to be an orthogonal direction in $\mathbb{R}^B$, when in reality some bins are near each other and others are not. This can lead to poor clusters (see Sections 4.2 and 5).

2.2 Preliminaries

Earth Mover's Distance (EMD) is a dissimilarity metric that meets all the requirements in Section 1. Given two probability distributions P and Q, the EMD [10] is most easily understood as the total area between their *cumulative distribution functions* (CDFs). Recall that the CDF of a distribution is a nondecreasing function $CDF(x)$ whose value at any real number x is the probability that a draw from the distribution will be less than or equal to x.

$$EMD(P,Q) = \int_{x=0}^{1} |CDF_P^{-1}(x) - CDF_Q^{-1}(x)| \tag{1}$$

We use Equation 1 to compute the EMD between two distributions. If we consider the two distributions as piles of earth, then EMD computes the minimum total distance that earth must be moved to transform one into the other.

We use ***k*-means clustering** with EMD as the distance metric to cluster distributions. We describe model selection and initialization in Section 3.

3. PROPOSED METHOD

We propose EP-MEANS (where EP is short for empirical probability), a novel and efficient algorithm for clustering empirical, real-valued univariate probability distributions. EP-MEANS combines EMD and k-means in a novel, scalable algorithm that does not rely on any *a priori* knowledge about the shapes of the observed distributions.

3.1 Applying k-means to Distributions

EP-MEANS is an efficient combination of k-means and EMD. Distributions $\{P_1 \dots P_m\}$ are the data instances given to k-means, and rather than minimizing the squared Euclidian distance to a centroid point in $\mathbb{R}^n$, EP-MEANS minimizes the squared EMD to a centroid distribution. The choice of centroid, similar to k-means, is one that minimizes within-cluster squared error. We utilize *kmeans++* [2] to initialize cluster centers. See Section 3.4 for how we select the model size k.

EP-MEANS consists of two main steps: centroid computation and distance computation. Centroid computation is straightforward, but distance computation can be slow if not carefully optimized. We first present some technical details on the way EMD is computed, then describe each of these steps in Sections 3.2 and 3.3.

3.2 Centroid Computation

Given a collection of distributions $\{P_1 \dots P_m\}$, their centroid for the purposes of k-means clustering is another empirical distribution Q such that $\sum_{i=1}^{m} (EMD(P_i, Q))^2$ is minimized. The intuition behind computing Q is based on Equation 1. If we consider the CDFs of all P_i's simultaneously, the value of CDF_Q at any given height y is just the mean value of all P_i's at height y.

To illustrate this further, let $r_i = min(\{v|P_i(v) > 0\})$ and $s = min(P_i(r_i))$. So for $0 \leq x < s$, $CDF_{P_i}^{-1}(x) = r_i$. The inverse CDF of Q should also be constant on this interval. If $CDF_Q^{-1}(x) = y$ for $0 \leq x < s$, then the contribution of this interval to $EMD(P_i, Q)$ is just $s(r_i - y)$ by Equation 1. The total contribution of this interval to within-cluster squared error is $s^2 \sum_{i=1}^{m} (r_i - y)^2$, which is minimized by choosing $y = \frac{1}{m}\sum_{i=1}^{m} r_i$.

Applying this logic to the rest of the probability axis, we see that Q should be selected such that $CDF_Q^{-1}(x) = \frac{1}{m}\sum_{i=0}^{m} CDF_{P_i}^{-1}(x)$. Figure 2 shows the centroids for five clusters of distributions on IP traffic data (see Section 4 for details). The distributions and centroids are shown as CDFs. The darker CDFs are the centroids. Observe that for a given centroid, at any height y (cumulative probability) its x-value is the mean of all constituent x-values at y.

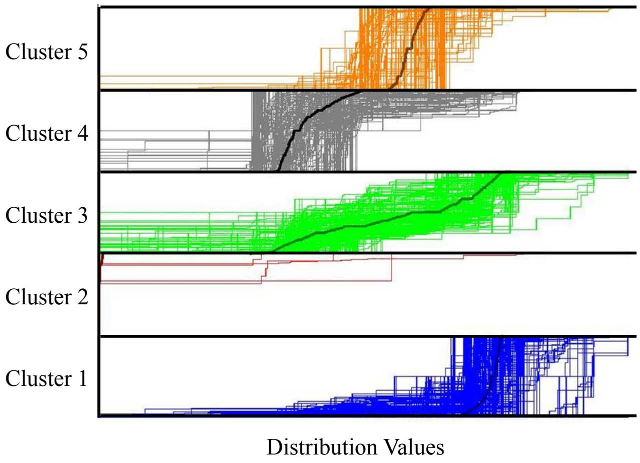

Figure 2: Five EMD centroids on IP traffic data. Each color above is a group of distributions, shown as CDFs. Overlaid are corresponding centroids (in darker lines), also represented as CDFs. The x-value of a centroid at height y is the average of the x-values of its constituents at y. This minimizes total (squared) EMD from all constituent distributions.

Q can be computed in $O(N\log(N))$ time for m distributions and N total observed values. The algorithm is straightforward. Initially, set $CDF_Q^{-1}(0) = \frac{1}{m}\sum_{i=1}^{m} CDF_{P_i}^{-1}(0)$. Then use a scanline to update CDF_Q^{-1} each time the inverse CDF of any P_i increases. This requires sorting all such values, at $O(N\log(N))$ time complexity.

We avoid performing a full sort on every EM iteration by initially sorting each inverse CDF and using a heap to keep track of the next value for each distribution in the cluster. This reduces the per-iteration runtime to $O(N\log(m))$.

3.3 Distance Computation Between Pairs of Distributions

The naïve implementation of distance computation is very similar to centroid computation. Scan through the inverse CDFs of both distributions simultaneously, and at each value for which an inverse CDF increases, compute the area of the rectangle that was just traversed.

This is enough to perform EP-MEANS, and runs in $O(|P| + |Q|)$ time if we do not count the time it takes to sort the CDFs. However, note that in the worst case, the centroid Q has mass at N distinct values. This means that for each of the m distributions, we have to call this function at a cost of $O(N + |P_i|)$. The resulting runtime is $O(mN + N\log(N))$ which may be unacceptable when m is large.

It is possible to compute all distances to a centroid in $O(N\log(N))$ time by applying an optimization to the aforementioned algorithm. The general idea is to do a single scan of the centroid's inverse CDF, updating each P_i's EMD incrementally along the way. To this end, we must split the inverse CDF of each distribution into *intervals* along the probability axis. The intervals associated with a single distribution P and a centroid Q are defined as follows:

1. The first interval starts with $x = 0$; the last interval ends with $x = 1$.

2. A new interval starts at any x in which the value of CDF_P^{-1} changes.

3. A new interval starts at any x where CDF_Q^{-1} starts below CDF_P^{-1} and ends above it.

Since all inverse CDFs are non-decreasing, these intervals can be determined in $O(N \log(N))$ time (after sorting the CDFs) by doing a single scan through Q. To accomplish this, we first construct a map from the set of values $\{y \mid \exists i : P_i(y) > 0\}$ to subsets $\{P_i \mid P_i(y) > 0\}$ that have mass at those values. Then we sort these y values and scan through Q's inverse CDF. At each step, we check to see if Q crossed a y value in our map; and if so, we check each P_i at that y value to determine if it crosses Q at that interval. See Figure 3 for an illustration of intervals. The black inverse CDF corresponds to the centroid, with lots of steps. The red inverse CDF corresponds to a single P_i.

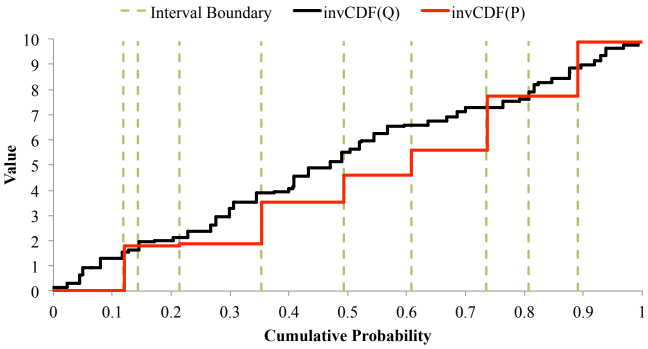

Figure 3: (Best viewed in color.) Intervals for EMD computation. Green lines indicate interval boundaries.

Given our definition of the intervals (enumerated above), the following observation holds:

Observation 1. *Within a given interval, either $CDF_Q^{-1} \geq CDF_P^{-1}$ everywhere, or $CDF_Q^{-1} \leq CDF_P^{-1}$ everywhere.*

Assume without loss of generality that the former (i.e., $CDF_Q^{-1} \geq CDF_P^{-1}$) is true on some interval I of the inverse CDFs. To compute the EMD contribution in I, let us define P_I as the (constant) value of P on I and w_I as the width of I. The EMD contribution on I is equal to the area between the inverse CDFs of P and Q on I. If the area under Q on I is $A_I(Q)$, then the EMD contribution is simply $A_I(Q) - w_I P_I$.[1]

Now let us focus on computing $A_I(Q)$. On I, the inverse CDF of Q takes a number of values $(q_I^1 \ldots q_I^t)$, and each value has a "width" $(x_I^1 \ldots x_I^t)$. Clearly

$$A_I(Q) = \sum_{i=1}^{t} q_I^i x_I^i \tag{2}$$

which can be rewritten as:

$$A_I(Q) = \sum_{J \leq I} \sum_{i=1}^{t} q_J^i x_J^i - \sum_{J<I} \sum_{i=1}^{t} q_J^i x_J^i \tag{3}$$

Here, $J < I$ includes all intervals to the left of interval I. The first term $A_{I+}(Q)$ is simply the total area under CDF_Q^{-1} from 0 to the end of I; and the second term $A_{I-}(Q)$ is the total area under CDF_Q^{-1} from 0 to the beginning of I. So to compute $A_I(Q)$, we simply scan from left to right in the inverse CDF of Q keeping track of the current total area $A_{I+}(Q)$ and the area at the beginning of the current interval $A_{I-}(Q)$. This requires constant space since we can throw away old values of A as we go along. Thus, the final EMD between P and Q is simply:

$$EMD(P, Q) = \sum_{I} |A_{I+}(Q) - A_{I-}(Q) - w_I P_I| \tag{4}$$

The important insight here is that we can process intervals for multiple P_i's simultaneously with only a single scan through Q. Any time we hit the end of an interval for some P_i, we update its EMD as described above and record the current value of $A_{I+}^i(Q)$ for P_i, which we will use next time we see P_i as $A_{J-}^i(Q)$ for the interval J that occurs after I.

Runtime Complexity with Optimization. The initial sorting of CDFs takes $O(N \log(N))$ time but only needs to be performed once and can be re-used in all subsequent iterations of k-means. Scanning through the intervals can be done in $O(N \log(m))$ time by maintaining a heap with $m + 1$ entries. The heap has one entry per P_i (which contains the end of the current interval on P_i) and one entry for Q (which contains the next value of CDF_Q^{-1} to be processed). The total number of intervals for a given P_i is at most $2|P_i|$, since Q can only "catch up" to P_i at most one time for every step in the inverse CDF of P_i (since both are non-decreasing functions). The total number of insertions and removals from the heap is at most $6N$. Q contributes N insertions and N removals, at most. Each P_i can contribute at most $2|P_i|$ insertions (one per interval) and $2|P_i|$ removals. Since the heap is never bigger than $m+1$ elements, the total time to compute all EMDs is $O(N \log(m))$ per iteration, which is potentially a large improvement over the $O(mN)$ naïve approach. Thus, the total runtime of this optimized algorithm is $O(N \log(N) + kTN \log(m))$ to compute centroids and distances, assuming T iterations of k-means.

3.4 Model Selection

To determine the appropriate number of clusters, we adopt a *stability-based model selection* approach introduced in [3]. We use *Variation of Information* (*VI*; see Section 4.2) [7] to define similarity between two clusterings. Specifically, for a given model size k, we define $S(C_1, C_2) = 1-(VI(C_1, C_2)/2 \log(k))$. S is minimized at 0 when mutual information between the clusters is minimized, and maximized at 1 when the clusters are identical. We choose not to correct for chance [13], because computing expected mutual information is computationally prohibitive on very large inputs.

For each k (up to some k_{max}), we compute the mean stability as follows. For parameter $\beta \in (0, 1]$, select $\lceil \beta m \rceil$ distributions from the input (uniformly at random, with replacement) and compute k ep-means clusters on this subset. Using the resulting centroids, assign the full set of input distributions to clusters. Repeat this procedure t times (for some parameter t). Then compute $S(C_i, C_j)$ for each pair of clusterings to determine S_k, which is the average similarity

[1] This in true because we choose intervals for which Observation 1 holds.

(a.k.a. stability) for model size k. The selected k is simply the model size that maximizes S_k. Note that $S_1 = S_m = 1$, so it is necessary to restrict k to a reasonable range. We demonstrate the effectiveness of this approach in Section 4.1.

3.5 Discussion

Other useful optimizations are: (1) do not update the centroid of a cluster that did not change in an iteration; and (2) do not recompute distances to such a centroid. In practice, these reduce runtime significantly (although not asymptotically).

EP-MEANS cannot easily be modified to handle multivariate distributions. First, exact computation of EMD for empirical distributions in two or more dimensions is prohibitive. Second, if the different dimensions have fundamentally different meanings, it may not be clear how to scale them appropriately. Recall that EMD is defined in terms of the underlying distance metric on the observed values. Operating in more than one dimension may result in an underlying metric that makes little sense.

4. EXPERIMENTS

So far, we presented results on our motivating example: the airline route data. Here, we (1) apply EP-MEANS to several distributional clustering problems; (2) present results on synthetic data that demonstrate EP-MEANS's ability to reconstruct ground-truth clusters even when they have complicated analytical forms; (3) show that EP-MEANS can be effective at clustering hosts in IP traffic data; (4) demonstrate the scalability of EP-MEANS on synthetic data; and (5) discuss some of EP-MEANS's limitations.

4.1 Model Selection

We apply EP-MEANS to a pair of synthetic data sets to measure the efficacy of its model selection. For each experiment, we use $\beta = 0.7$ and $t = 5$ (for a total of 20 comparisons per model size). Each data set is comprised of 8 ground-truth clusters with 10 distributions per cluster (i.e., $m = 80$). Each distribution is Gaussian (with mean μ and standard deviation σ). We draw 10,000 samples from each distribution. The ground-truth clusters are as follows. For the first experiment, each cluster has $\sigma = 0.5$ and $\mu = 4i$, where $1 \leq i \leq 8$. For the second experiment, the first four groups have $\sigma = 0.5$ and $\mu = 4i$, where $1 \leq i \leq 4$ and the last four groups have $\sigma = 0.005$ and the same means. Figure 4 shows the results of EP-MEANS with model selection on these data. For the simpler problem, we see a single peak at $k = 8$. For the overlaid problem, we see a secondary peak at $k = 4$ corresponding to the clustering that groups distributions with the same mean but different variances.

Recalling the airline example in Section 1.1, the only model size with stability over 0.8 is $k = 3$. A secondary peak around $k = 8$ presents another potentially useful model. The model with 8 clusters groups the airlines based on less significant differences in their business models. For example, some domestic airlines (e.g., United) provide international service as well, while others (e.g., Southwest) do not. Additionally, Cluster 3 gets split into true international commuter airlines and long-haul, low-cost airlines.

4.2 Synthetic Experiments

Can EP-MEANS recover clusters when we know the underlying groupings in the input data? To answer this, we

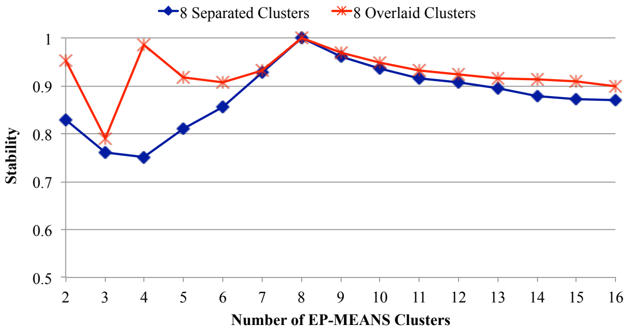

Figure 4: Model selection in synthetic data. Each data set has 8 clusters. When the clusters are overlaid, we see a secondary spike at $k = 4$.

perform the following synthetic experiment. We generate 10 *mixture distributions*. Each mixture is an analytical probability distribution, chosen via a number of randomized steps. First, the number of component distributions for each mixture is chosen between 1 and 5. Each component distribution can be Uniform, Gaussian, Pareto, Exponential, or Beta (chosen uniformly at random). Once a family is selected, the parameters to the component are selected at random within a range. Each component distribution in the mixture is weighted uniformly at random, so each mixture distribution is the weighted sum of each of its component distributions. Then, we scale each mixture distribution so that it has mean 0 and standard deviation 1. Under this setup, the naïve approach of clustering by mean and standard deviation will not perform well.

After generating 10 mixtures, we instantiate $m = 1000$ instances of each mixture. For each instance, we draw d values from its underlying mixture distribution, for a variety of d. Thus, each instance has an empirical, one-dimensional, continuous-valued distribution. Then, we apply EP-MEANS with $k = 10$. For this experiment, we run EP-MEANS 10 times with different initial centroids.

We compare EP-MEANS to two baseline approaches. In the first baseline approach (a.k.a. the moments method), we map each instance to a 4-tuple of *moments*: $(\mu, \sigma, \gamma, \kappa)$ corresponding to the mean, standard deviation, skewness, and kurtosis of the observed empirical distribution. We treat each instance as a point in $\mathbb{R}^4$ corresponding to its moments. We then apply k-means with $k = 10$ and run 10 trials. In the second baseline approach (a.k.a. the bins method), we use fixed-width *bins* to construct a histogram. We compute the minimum and maximum values observed across all distributions, and generate 10 equal-width bins spanning this range. In this case, each instance is treated as a point in $\mathbb{R}^{10}$. We perform 10 rounds of k-means with $k = 10$.

To compare the clusterings, we use the Variation of Information (VI) measure [7], which is related to mutual information. Specifically, it is defined as $VI(C_1, C_2) = H(C_1) + H(C_2) - 2I(C_1, C_2)$, where $H(C)$ is the entropy of C and $I(C_1, C_2)$ is the mutual information between C_1 and C_2. We compute $VI(C_{method}, T)$ where C_{method} are the clusters described above for each method and T is the ground-truth clustering of instances into their distributional families.

VI is bounded below by 0, and achieves this value only when the two clusters are identical. It is bounded above by

$\log_2(m)$, where m is the number of objects being clustered. However, a tighter upper bound is available here because we are forcing the clusterings to have size 10. With that restriction, the upper bound is $2\log_2(10) \approx 6.64$. As a point of reference, if one were to choose a clustering where all instances are in a single cluster C_{all}, then we would have $VI(C_{all}, T) = \log_2(10) \approx 3.32$.

Figure 5 shows the results of this experiment for a variety of choices for d. We report the minimum, mean, and maximum VI values for each method. EP-MEANS dominates the baseline methods, especially as the number of samples increases. Once the number of samples is sufficiently high, the baseline methods fail to improve past the performance of C_{all}. This is most likely due to very large values being generated by some of the long-tailed distributions. For the moments method, very large values in the third and fourth moments may dominate the clustering. For the bins method, these large values force all of the small observed values into a single bin near zero. This causes most of the instances to be grouped into a single cluster.

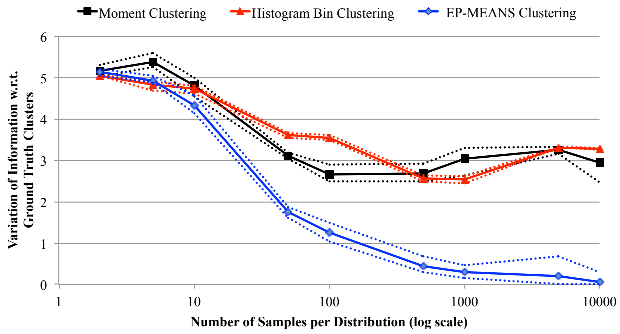

Figure 5: EP-MEANS recovers ground-truth clusters more effectively than clustering based on moments or histogram. The lower the Variation of Information (VI), the better. The solid line is the mean VI value; the dashed lines represent the minimum and maximum VI values.

4.3 IP Traffic Classification

We have shown that EP-MEANS can capture information about the shape of empirical probability distributions, which distributional moments and binning approaches cannot capture. Now we show that this representational advantage translates into a practical advantage in a real-world application: classification of IP network traffic. Specifically, we test the hypotheses that EP-MEANS-based features provide more predictive power in traffic classification than features from moments clustering or histogram clustering.

We test these hypotheses using packet trace data collected from an enterprise IP network over a one hour period. This data set contains ~5K hosts and ~600K network connections. For each host, we use packet signatures to identify the dominant traffic type (peer-to-peer=72%, Web=23%, DNS=5%). We discard hosts with other dominant traffic types (e.g., games, mail, etc) and any host with fewer than 10 communications. We then collect 23 traffic-based features for each connection of each host (e.g., # bytes transferred, # packets, max inter-arrival time between packets, etc). Since each host has many connections, this yields an empirical distribution of values for each traffic feature for each host. We compare three approaches to modeling these distributions: (1) histogram with 10 equal-width bins, (2) moments clustering based on mean and variance of the distributions, and (3) EP-MEANS with 10 distribution clusters, where each individual distribution is represented by a vector of 10 distances (one to each cluster centroid).

We use each of the three representations of the 23 traffic features to predict the traffic type of each host in the network. We use 10-fold cross-validation and the R `randomForest` classifier. Figure 6 shows that EP-MEANS outperforms moments and histogram clustering across nearly all of the single features. EP-MEANS performs either significantly better than (14/23) or equivalent to (9/23) moments clustering across all features. EP-MEANS significantly outperforms histogram clustering for 20 features, ties for one, and performs worse for two. We assess significance using a paired two-tailed t-test with $\alpha = 0.05$. The average performance across all single features is: EP-MEANS =83%, moments clustering=80%, and histogram clustering=76%. These results indicate that: (a) the shape of the feature distributions matters and (b) EP-MEANS is a more effective representation of distributional shape than a histogram of the same size. Moreover, we note that by combining information from all 23 features, moments and histogram clustering are able to make up for their representational disadvantage and close the gap somewhat on EP-MEANS. However, even in this case, EP-MEANS maintains a statistically significant advantage over both.

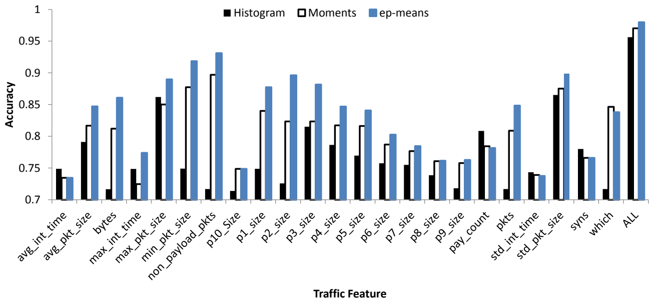

Figure 6: (Best viewed in color.) IP Traffic classification accuracy for three representations of empirical probability distributions based on (1) histogram clustering, (2) moments clustering, and (3) EP-MEANS centroid distances. EP-MEANS is the top performer for both the complete set of features and on average across individual features.

4.4 IP Traffic Characterization

Here we take the same IP traffic data as in the previous section and use EP-MEANS to characterize it. Specifically, we take each observed host (IP address) as an instance; the observed value is the duration of its IP connections. Figure 2 in Section 3.2 depicted the discovered clusters for this data, where each band represented a cluster and the dark CDF is the centroid of that cluster. Another way to look at the centroids is as histograms; see Figure 7. Values of k greater than 5 seem to split these clusters into smaller clusters, which are more focused on given ranges of values.

This experiment is harder to interpret, since we do not have any ground-truth about the observed hosts. However, it illustrates some insights into IP communication patterns. First, note that the cluster centroids do not appear to come from the same distributional family. Cluster 2 seems to consist only of hosts whose connections have almost zero length. These are most likely failed connections. Clusters 1 and 4 appear to be some sort of truncated Gaussian distribution, with the truncation happening in opposite directions. Clusters 3 and 5 seem to have similar mean values, but Cluster 3 appears to have mass uniformly spread across a wide range, while Cluster 5 is more concentrated near the mean.

Note that once these centroids are computed, we can quickly characterize a new host by simply computing the distance from the host to each centroid. This operation is significantly faster than performing k-means iterations. This suggests a method for change detection and anomaly detection.

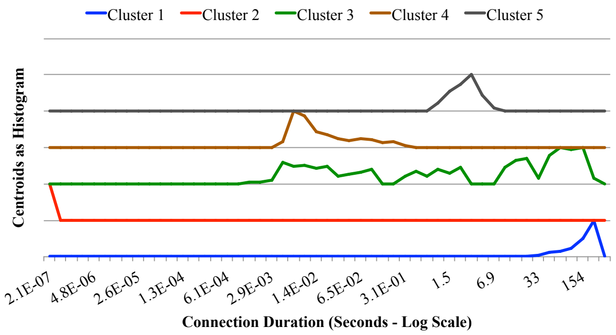

Figure 7: (Best viewed in color.) Centroids for connection duration in observed hosts. The centroids do not appear to come from the same family of distributions. Distributions are scaled up to highlight shape. The total area under each curve is 1.

4.5 Scalability

Using the synthetic distributions from Section 4.2, we analyze the scalability of EP-MEANS. In order to eliminate variations based on the number of EM iterations, we report the time taken to generate initial centroids, compute distances from each distribution to each centroid, and recompute centroids once. Times are averaged over ten trials. Figure 8 shows the runtime of EP-MEANS for a variety of problem sizes. Times are divided by $N \log(N)$, where N is the number of samples across all distributions in each experiment, to demonstrate the $O(N \log(N))$ asymptotic runtime.

5. RELATED WORKS

Many dissimilarity measures are available on empirical probability distributions. We highlight a number of popular approaches and describe their properties as they relate to our requirements here. While this is not an exhaustive list, it captures common shortcomings and illustrates the reasoning behind our choice of EMD as a clustering metric.

Kolmogorov-Smirnov (KS) Statistic. A popular dissimilarity measure on distributions is the KS statistic. This easily computed statistic is the maximum vertical distance between the CDFs of each distribution. For empirical distributions, the CDF is a step function which increases at each value that is observed in the distribution. The KS statistic has an undesirable property that renders it incapable of clustering correctly in even very simple situations. Namely, KS does not take into account distances in the underlying space of values. For example, any two distributions which have all their mass at a single point will have KS statistic of 1 (if the points are distinct), regardless of how far apart the distributions are. Consider three distributions, each with all its mass at values 0, 1, and 100 respectively. KS considers the clustering ((1,2),(3)) equally good as the clustering ((1),(2,3)), although the former is intuitively better.

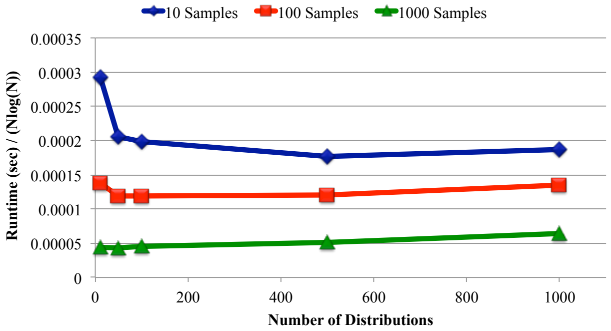

Figure 8: Timing results for EP-MEANS synthetic data. Runtimes are averaged over 10 trials. Displayed runtimes are divided by $N \log(N)$, where N is the total number of observations. Our implementation of EP-MEANS essentially achieves its theoretical runtime complexity.

Kullback-Leibler (KL) Divergence. Another popular dissimilarity measure for distributions is the KL divergence [6]. It measures the increased cost of encoding samples from a target distribution when the encoding is based on a reference distribution. KL divergence has a number of disadvantages in this setting. For empirical distributions, it typically requires binning, which as described above can be undesirable in certain settings. Most importantly, though, KL divergence is similar to the KS statistic in that it ignores the underlying space of values. It treats each point (or bin) in the distribution as an independent value, and as such cannot correctly cluster the example described above.

Density Estimation. In an attempt to provide identical support across all distributions, a possible approach is to smooth the empirical distributions by a technique like Kernel Density Estimation (KDE). In this approach, each observation is replaced by a *kernel*: a distribution (e.g., normal, triangular, etc) centered at the observation and with some scale parameter. The overall distribution is represented as the sum of all the kernels. Unfortunately, KDE does not allow for scalable clustering of distributions. In this work, we present results in which thousands of distributions are clustered, each of which has tens of thousands of observed values. This can lead to very complicated density estimates, and computing distances (and centroids, for an algorithm like k-means) is expensive [11].

Multidimensional Scaling (MDS). While not technically a clustering algorithm, MDS [8] and its variants are often used to visualize large data sets and can aid in clustering. Classical MDS algorithms take as input an m-by-m matrix D of distances between each input entity, and use an $O(m^3)$ algorithm to place the points in d-dimensional space.

The points are arranged such that their distances in $\mathbb{R}^d$ in the computed configuration are as close as possible to their actual distances in D. Faster, approximate algorithms exist that run in $O(r^2m)$ time using r *landmark* points rather than processing all pairwise distances. MDS is useful for visualization and dimensionality reduction, but it has some shortcomings for our application. First, it does not actually place the entities in clusters; instead, it simply assigns them coordinates in $\mathbb{R}^d$. Secondly, it is not obvious how to interpret the dimensions.

Lévy and Lévy-Prokhorov Metrics. Similar to EMD, these metrics [14, 15] are on the space of cumulative distribution functions of one-dimensional random variables. The computational costs of these metrics are not known.

Related Applications of EMD. EMD and other *spatially aware* (i.e., based on the underlying metric space) distances have been applied to clustering of points in space [5, 9]. However, in these cases the clusters of points are treated as distributions and EMD is used to compute/compare these clusters. This is a fundamentally distinct problem from the one we address here, which is efficiently clustering *distributions* rather than clustering points in space.

Indyk and Price [4] propose a promising approximation of multi-dimensional EMD using compressive-sensing techniques. However, their work is related to clustering segments of an image, rather than clustering distributions. This is similar to the point clustering algorithms described above, but with histogram bins instead of continuous-valued points.

Shirdonkar and Jacobs [12] approximate EMD for a multidimensional histogram with B bins in linear time in B. However, the runtime is exponential in the dimension of the histogram, restricting the applications to domains with a small number of dimensions. Our algorithm is restricted to one dimension, but gives exact EMD in linear time in B if the distributions are binned. Additionally, there is no clear path from the algorithm in [12] to execute clustering. For m distributions with B bins, one would need to compute m^2 distances unless there is some way to compute centroids efficiently. No such algorithm is provided. Applegate et al. [1] modify the algorithm of Shirdonkar and Jacobs, but as the authors note it requires m^2 distance computations to perform clustering. In their experiments, they choose to downsample their large dataset due to this runtime complexity. EP-MEANS provides an efficient way to cluster many distributions at once without ever needing to compute pairwise distances.

6. CONCLUSIONS

We presented EP-MEANS, a novel algorithm for clustering univariate, empirical probability distributions. We have demonstrated that EP-MEANS generates useful clusters in a variety of application domains, as well as that it satisfies a number of desirable criteria. Specifically, EP-MEANS is **efficient**: runtime of the algorithm is $O(N\log(N)+kTN\log(m))$ for m distributions, k clusters, T iterations, and N total observed values. EP-MEANS is **nonparametric**: the algorithm does not require binning or quantization; and the number of clusters can be discovered automatically using a number of existing techniques. EP-MEANS is **empirical**: input instances are treated as empirical distributions, rather than assuming they come from some known analytical family. EP-MEANS is **distance-based**: the clusters are based in a true metric space, which can avoid some counterintuitive results. EP-MEANS is **interpretable**: discovered centroids can be visualized and understood in a natural way.

7. ACKNOWLEDGEMENTS

Prepared by LLNL under Contract DE-AC52-07NA27344. LLNL-CONF-665288 and supported in part by NSF CNS-1314603, by DTRA HDTRA1-10-1-0120, and by DAPRA under SMISC Program Agreement No. W911NF-12-C-0028.

8. REFERENCES

[1] D. Applegate, T. Dasu, S. Krishnan, and S. Urbanek. Unsupervised clustering of multidimensional distributions using earth mover distance. In *KDD*, pages 636–644, 2011.

[2] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *SODA*, pages 1027–1035, 2007.

[3] R. Giancarlo and F. Utro. Algorithmic paradigms for stability-based cluster validity and model selection statistical methods, with applications to microarray data analysis. *Theoretical Comp. Sci.*, 428(0):58 – 79, 2012.

[4] P. Indyk and E. Price. K-median clustering, model-based compressive sensing, and sparse recovery for earth mover distance. In *STOC*, pages 627–636, 2011.

[5] S. Jegelka, A. Gretton, B. Schölkopf, B. K. Sriperumbudur, and U. von Luxburg. Generalized clustering via kernel embeddings. In *KI 2009: Advances in AI*, pages 144–152, 2009.

[6] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Math. Stat.*, 22(1):79–86, 1951.

[7] M. Meila. Comparing clusterings by the variation of information. In B. Schölkopf and M. K. Warmuth, editors, *Learning Theory and Kernel Machines*, pages 173–187. 2003.

[8] M. E. Mugavin. Multidimensional scaling: A brief overview. *Nurs. Res.*, 57:64–8, 2008.

[9] P. Raman, J. M. Phillips, and S. Venkatasubramanian. Spatially-aware comparison and consensus for clusterings. *CoRR*, abs/1102.0026, 2011.

[10] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *ICCV*, pages 59–66, 1998.

[11] O. Schwander and F. Nielsen. Learning mixtures by simplifying kernel density estimators. In F. Nielsen and R. Bhatia, editors, *Matrix Information Geometry*, pages 403–426. 2013.

[12] S. Shirdhonkar and D. Jacobs. Approximate earth mover's distance in linear time. In *CVPR*, pages 1–8, 2008.

[13] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *ICML*, pages 1073–1080, 2009.

[14] V. M. Zolotarev. Lévy metric. In M. Hazewinkel, editor, *Encyclopedia of Mathematics*. Springer, 2001.

[15] V. M. Zolotarev. Lévy-Prokhorov metric. In M. Hazewinkel, editor, *Encyclopedia of Mathematics*. Springer, 2001.