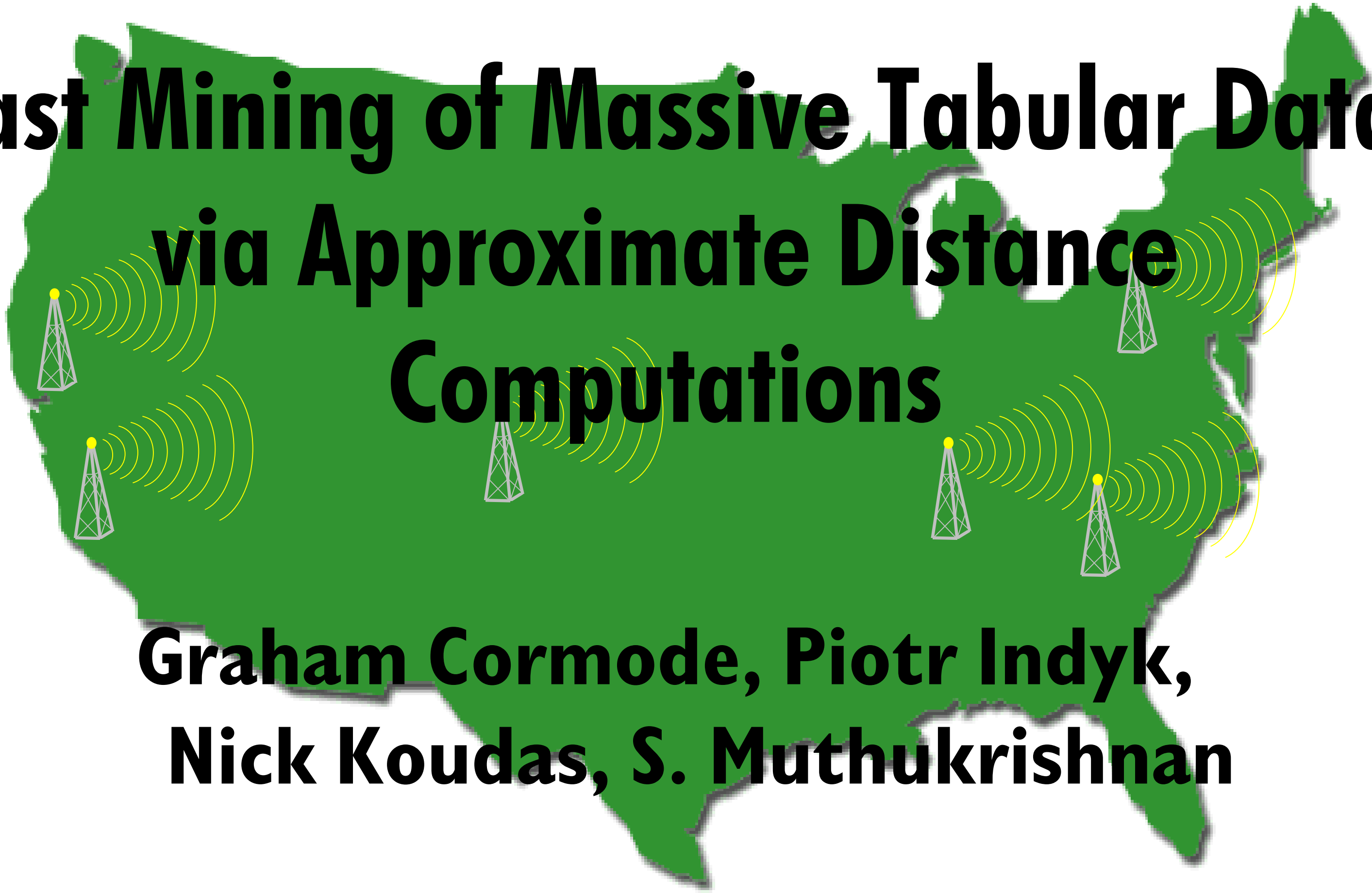
Fast Mining of Massive Tabular Data via Approximate Distance Computations

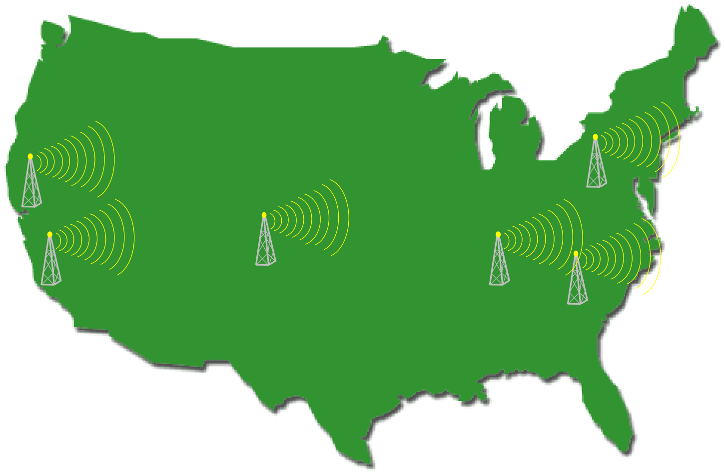


Graham Cormode, Piotr Indyk,
Nick Koudas, S. Muthukrishnan

Tabular Data

Much data is stored in tables:

- Cellphone traffic
- IP traffic between source and destination
- Traditional database tables

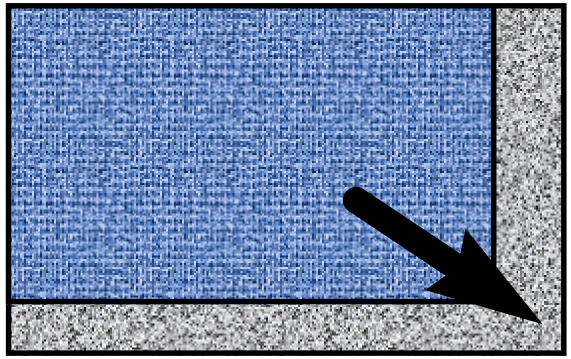


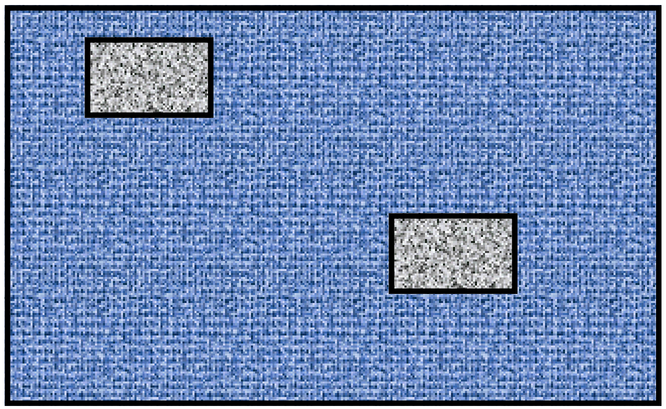
Mining this data presents new challenges to database technology.

Need to find appropriate, efficient comparison methods

Tables are massive

Adding extra rows or columns increases the size by thousands or millions of readings





The objects of interest are *subtables* of the data

eg Compare cellphone traffic of SF with LA

These subtables are also massive!

How to compare subtables?

- L_2 difference of values

  Sum of squares differences: $(\Sigma_i (a_i - b_i)^2)^{1/2}$
- L_1 difference of values

  Sum of absolute differences: $\Sigma_i |a_i - b_i|$
- More generally, L_p difference

  $(\Sigma_i |a_i - b_i|^p)^{1/p}$ $\quad 0 < p \leq 2$

Letting *p* take fractional values may give interesting similarity results

Prior Works

[AFS93], [IKM00] have studied mining 1-dimensional time series under L_2

Efficient mining methods have been studied with k-means, CLARANS [NH94], BIRCH [ZRL96], DBSCAN [EKSX96] CURE [GRS98] etc.

These have focused on minimising the number of comparison operations.

Here, our focus is on reducing the cost of each comparison – an orthogonal goal to prior work. We extend to L_1 and other L_p distances.

Our results

- We consider Lp distance for non-integral *p*
  These often given better results than the traditional L_1, L_2
- We give methods for computing approximations of L_p distances for massive multidimensional data
  These are proven to be accurate and much faster than previous methods
- We demonstrate the applicability of these methods on real network data
  Approximate comparisons can be used to speed up any method that uses comparisons

Sketches for L_p distance

We want to find $(\Sigma_i |a_i - b_i|^p)^{1/p} = ||\mathbf{a} - \mathbf{b}||_p$ for tabular data **a** and **b.**

Main Idea: for subtables of interest **a** and **b** we will find a much smaller *sketch* so that the L_p distance can be found approximately by comparing the two sketches.

[IKM00] gave sketches for L_2. Here we extend this for all (fractional *p*) between 0 and 2.

Main Tool: Stable Distributions

Let X be a random variable distributed with a *stable distribution*. Stable distributions have the property that

$$a_1X_1 + a_2X_2 + a_3X_3 + \dots a_nX_n \sim ||(a_1, a_2, a_3, \dots , a_n)||_pX$$

if $X_1 \dots X_n$ are stable with stability paramater p

The Gaussian distribution is stable with parameter 2

Stable distributions exist and can be simulated for all parameters $0 < p < 2$.

So, let $\mathbf{X} = x_{1,1} \dots x_{m,n}$ be a matrix of values drawn from a stable distribution with parameter p...

Creating Sketches

$$(a_1 \ldots a_n) \bullet \begin{pmatrix} x_{1,1} \ldots x_{m,1} \\ \ldots \\ x_{1,n} \ldots x_{m,n} \end{pmatrix} = (s_1, \ldots s_m)$$ [a *sketch*, s]

$$(b_1 \ldots b_n) \bullet \begin{pmatrix} x_{1,1} \ldots x_{m,1} \\ \ldots \\ x_{1,n} \ldots x_{m,n} \end{pmatrix} = (t_1, \ldots t_m)$$ [a sketch, t]

Then median$(|s_1 - t_1|, |s_2 - t_2|, \ldots, |s_m - t_m|)$/median(X) is an estimator for $\| a - b \|_p$

Can guarantee the accuracy of this process: will be within a factor of $1+\varepsilon$ with probability δ if $m = O(1/\varepsilon^2 \log 1/\delta)$

Efficient Computation

Computing sketches in this way can be time consuming – it relies on a lot of matrix multiplications (one for each entry in the sketch vector)

Computing multiple sketches of data size N can be sped up:

- For a fixed subtable size, M, we can find sketches of all subtables using Fourier transform to do the multiplications in total time O(N log M)
- A sketch for a subtable can be found by summing sketches for subtables that cover the area

Properties of Sketches

- Sketches can be very small

The length of the sketch vector **does not** depend on the size of the subtable that it represents.

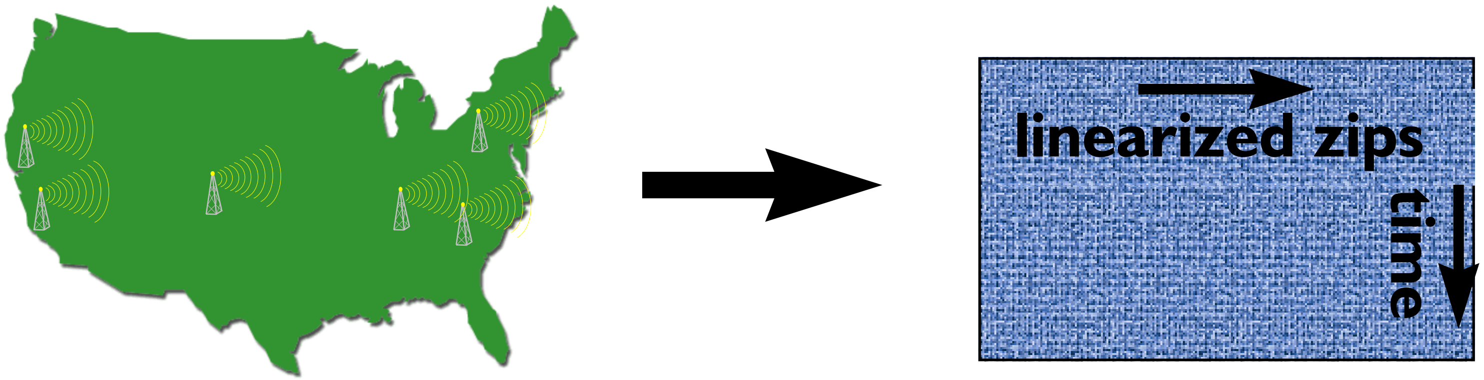
- The accuracy is **guaranteed**

Other methods – coefficients of Fourier Transform, Cosine Transform, Wavelet Transform etc. work only for L_2.
They do not extend to other $_{Lp}$ distances.

- Can be manipulated arithmetically

The sketch of the sum of two subtables is the sum of their sketches.

Experimental Setting



- We took approx 600Mb of call data for a couple of weeks from the AT&T Network
- We also used synthetic data to test finding a known clustering
- Used k-means as the clustering method

Measurements

We define a variety of measurements to test using sketches:

Cumulative accuracy – how accurate in the long run

Average accuracy – how accurate is each comparison

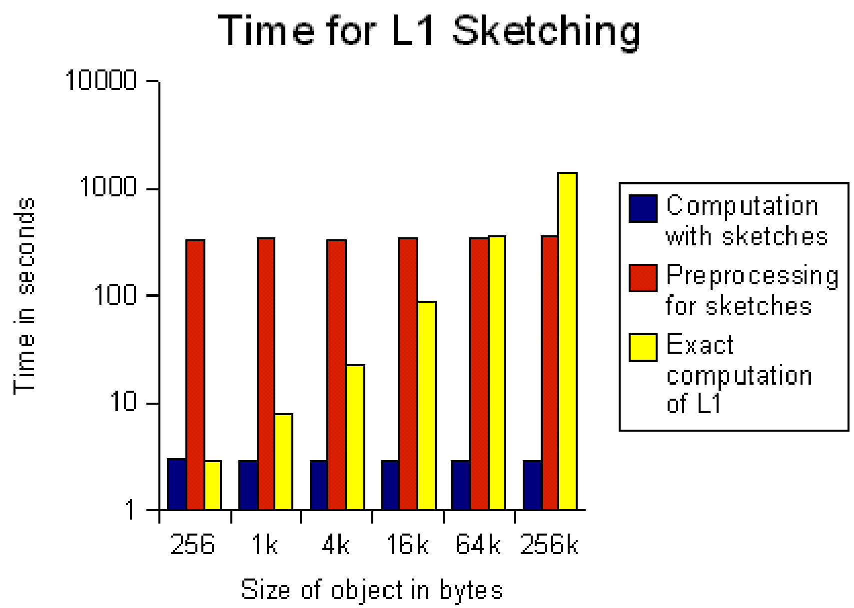
Pairwise comparison – correctly identifying the closest subtable out of two

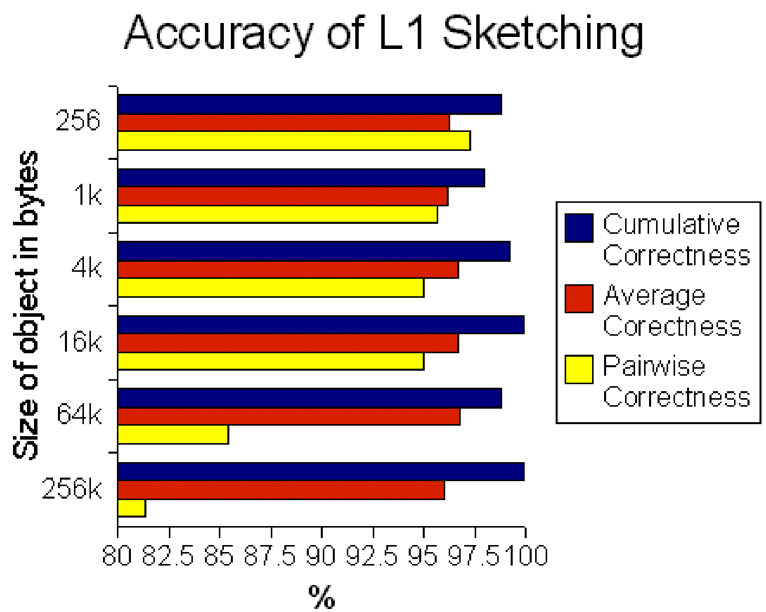
Confusion matrix agreement – compares two clusterings based on the confusion matrix between them

Quality of clustering – how tight is one clustering compared to another

L_1 Tests

We took 20,000 pair of subtables, and compared them using L_1 sketches. The sketch size was less than 1Kb.

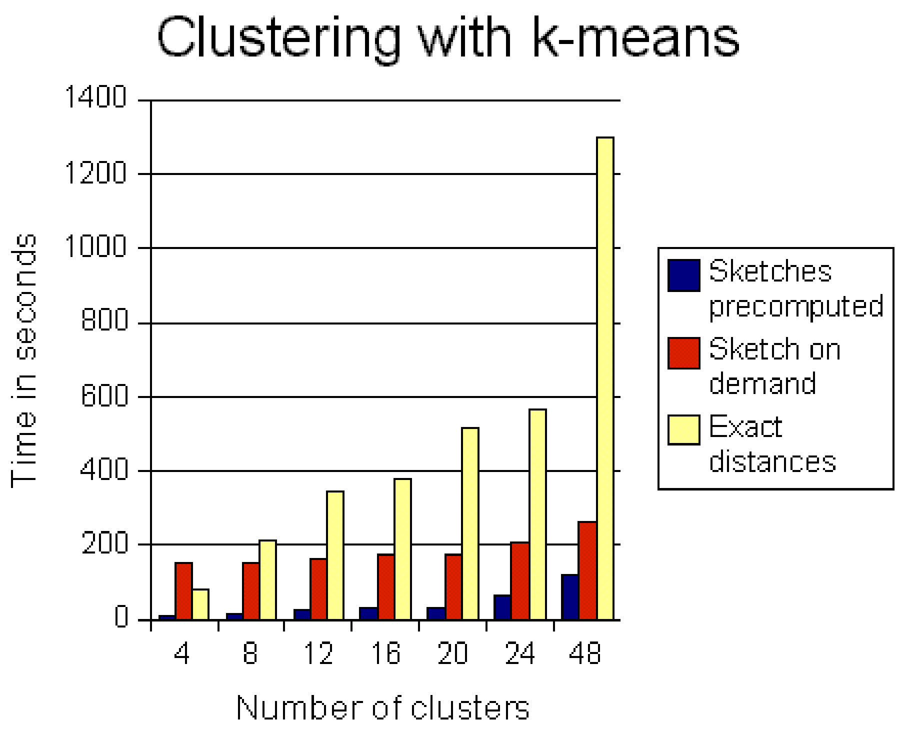




- Sketches are very fast and accurate (can be improved further by increasing sketch size)
- For large enough subtables (>64k) the time saving "buys back" the preprocessing cost of sketch computation

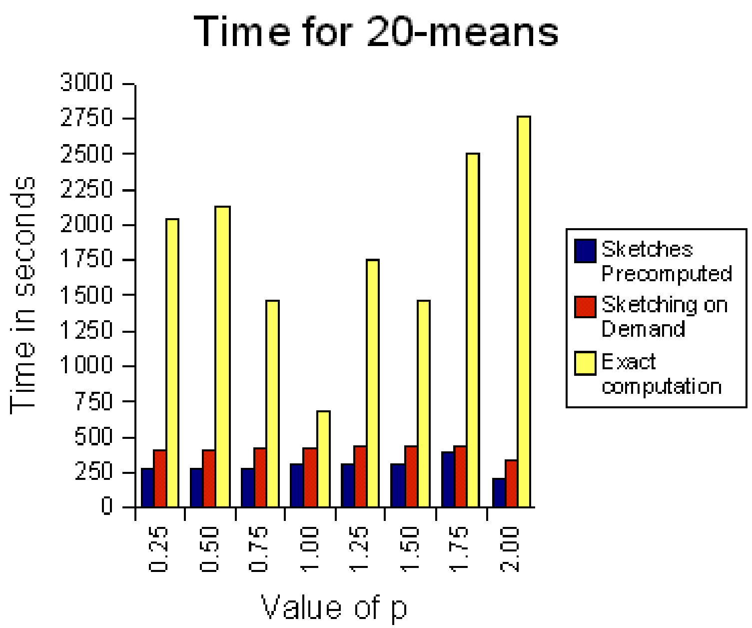
Clustering with k-means

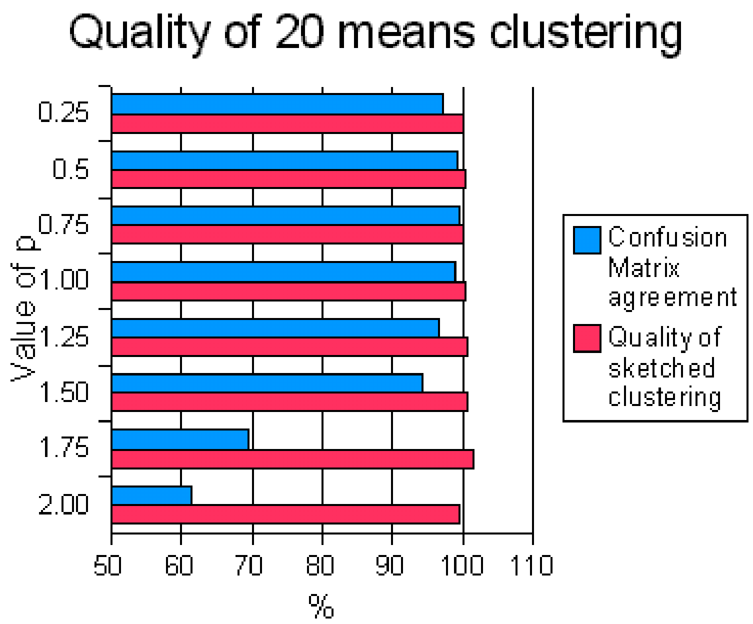
- Sketches are much faster than exact methods, and creating sketches when needed is always faster than exact computation.
- As k increases, the time saving becomes more significant.
- For 8 or more clusters, creating sketches when needed is much faster.



k-means with L_p distances

Varied p from 0.25 to 2.0, and used k = 20 means





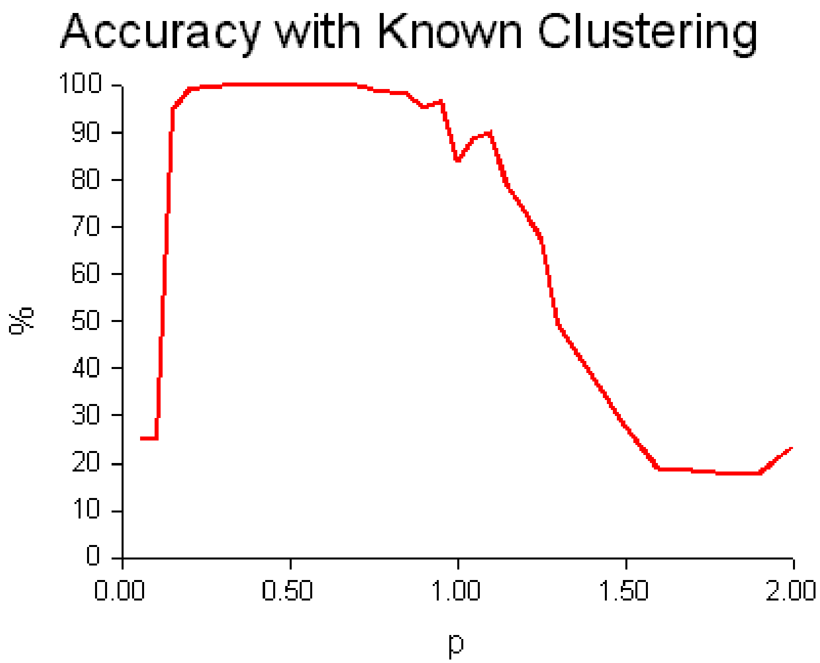
- Using sketches still results in much faster computation

- There is no significant loss of quality from using sketches – in fact, sometimes better!

Varying p

We fixed a known clustering within some synthetic data, and considered the confusion matrix.

The traditional L_2 and L_1 methods didn't find the known clustering



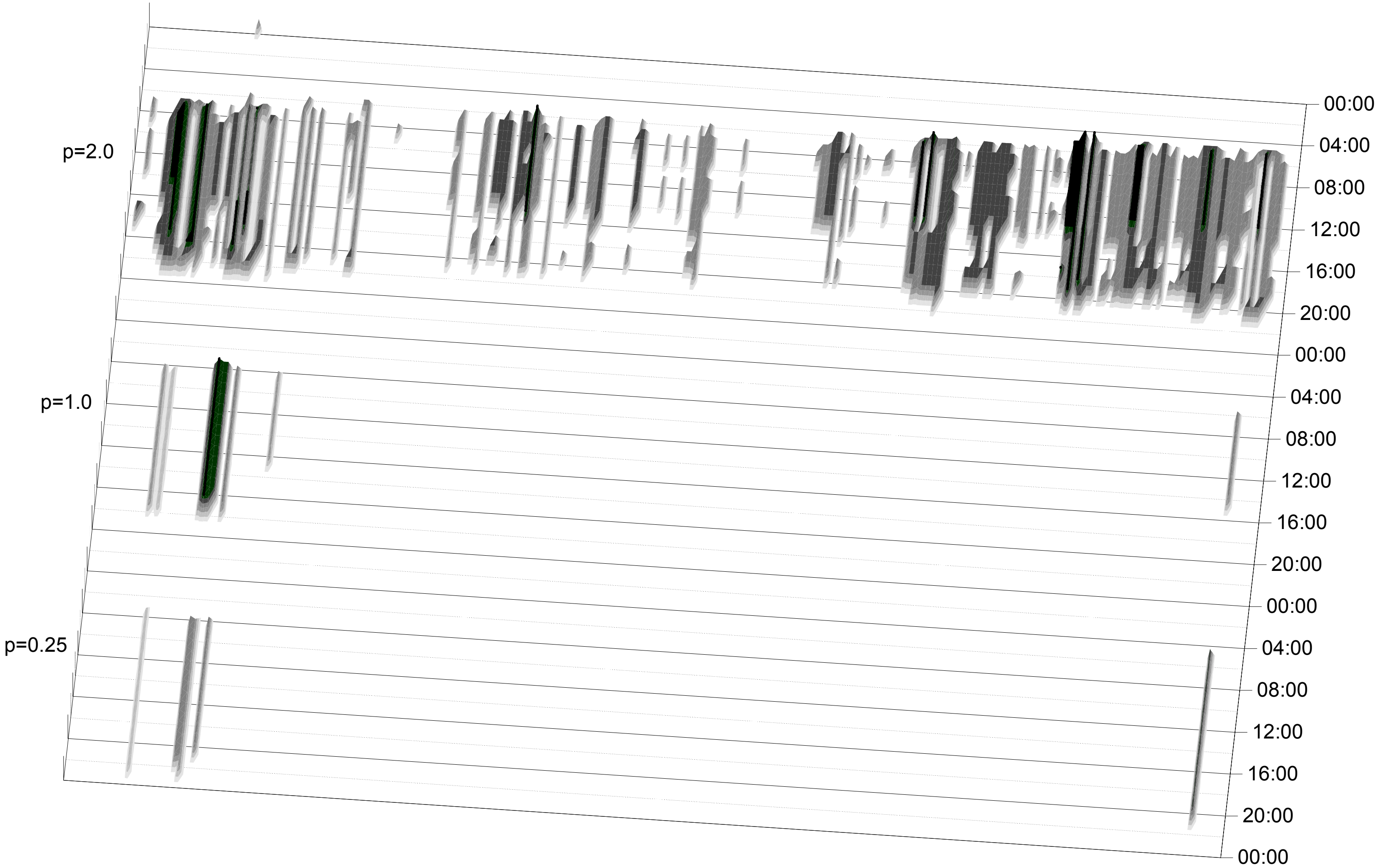
L_2 fails completely: the differences are too large and throw off k-means

L_p for $p<1$ finds the correct clustering

p = 0.5 seems a good value. This dampens the effect of outlier points

Case Study: US Call Data

One day's data clustered under p=2.0, p=1.0, p=0.25



Case study: US Call data

We looked at the call data for the whole US for a single day

- $p = 2$ shows peak activity across the country from 8am - 5pm local time, and activity continues in similar patterns till midnight
- $p = 1$ shows key areas have similar call patterns throughout the day
- $p = 0.25$ brings out a very few locations that have highly similar calling patterns

Conclusions

- The spectrum of Lp distances give different and interesting results for all) $0 < p \leq 2$, not just $p = 1$ and $p = 2$.
- $p < 1$ seems especially interesting, supressing outliers.
- Sketches give an efficient and accurate way of finding Lp distances for arbitrary p
- Sketches are proven accurate and shown to be useful in practice
- Can be used in any application that compares vector, tabular or higher dimensional data