# Efficient Clustering for Large-Scale, Sparse, Discrete Data with Low Fundamental Resolution

Veronika Strnadová-Neeley,
supervised by John R. Gilbert
University of California, Santa Barbara
veronika@cs.ucsb.edu

## ABSTRACT

Scalable algorithm design has become central in the era of large-scale data analysis. My contribution to this line of research is the design of new algorithms for scalable clustering and data reduction, by exploiting inherent low-dimensional structure in the input data to overcome the challenges of significant amounts of missing entries. I demonstrate that, by focusing on a property of the data that we call its *fundamental resolution*, we can improve the efficiency of clustering methods on sparse, discrete-valued data sets.

## 1. INTRODUCTION AND BACKGROUND

The necessity for efficient algorithms in large-scale data analysis has become clear in recent years, as unprecedented scaling of information has sprung up in a variety of domains, from bioinformatics to social networks to signal processing. In many cases, it is no longer sufficient to use even quadratic-time algorithms for such data, and much of recent research has focused on developing efficient methods to analyze vast amounts of information.

Here we focus on scalable clustering algorithms, a form of unsupervised learning that is invaluable in exploratory data analysis [16]. Many successes in the effort to design these algorithms have focused on leveraging an inherent structure in the data, and its structure may be best expressed in various ways. The data may be best described as lying in an inherently low-dimensional Euclidean space, along a low-dimensional manifold, or it may have certain self-repeating, or *fractal* properties. All these structural properties have been explored to some degree in order to design more efficient clustering algorithms.[11, 2, 6, 7, 8, 17]

My work focuses on leveraging a property of large-scale, discrete-valued data that I call its *fundamental resolution*, a concept that can be explained by comparing the images in Figure 1. More pixels produce a clearer image, but only up to a point – we cannot distinguish the leftmost image from that in the middle, even though more pixels are used to render the image in the leftmost position. Similarly, in many large, discrete-valued data sets, the fundamental resolution, rather than the number of data points, determines the extent to which we can distinguish points from one another before the data becomes redundant. If we know a large data set with many missing values has a fundamental resolution, we can more easily single out data points that are noise and fill in missing entries.

Figure 1: Fundamental resolution of an image: the more pixels we use, the clearer it gets, but only up to a point.

In the following, I present efficient algorithms for clustering large-scale, discrete-valued data sets with missing values by leveraging the fundamental resolution of the data. Previously, my collaborators and I have demonstrated that the underlying fundamental resolution of binary-valued genetic mapping data can be used to quickly cluster large genetic mapping data sets. I am now generalizing this clustering approach to large-scale, discrete-valued data, such as that found in the recommender systems domain. Genetic mapping and recommender systems present similar challenges to clustering algorithms, due to the large degree of sparsity and the sheer scale of the input data in these domains.

## 2. RELATED WORK

Much attention has been paid to the intuition that many large-scale data sets lie in an inherently low-dimensional space, which explains the popularity of matrix factorization methods for large scale data analysis. Methods such as principal component analysis rely on an SVD decomposition in order to project a high-dimensional data set into a lower dimensional space [10, 9]. Spectral clustering is another such example, and has been modified in recent years to improve in running time [11]. More recently, the CUR decomposition [12] has gained popularity as a sparse matrix factorization method that is both fast and in some cases more interpretable than a decomposition based on eigenvectors. With matrix factorization approaches, clustering the projected data in the lower-dimensional space often results in better clustering performance. However, my work focuses on data that does not necessarily lie in a low-dimensional Euclidean subspace – many dimensions in the input may be relevant in data with a low fundamental resolution. In addition, there is no clear answer on how to deal with noise and missing entries when factorizing a large data matrix, whereas my work takes these issues into account.

Proceedings of the VLDB 2017 PhD Workshop, August 28, 2017. Munich, Germany. 

Other forms of lower-dimensional inherent structure have also been explored to speed up the clustering of large-scale data. A data set's fractal dimension has been exploited for clustering [8], but this method is not scalable to large data sets. An approach based on exploiting a low fractal dimension and entropy of a data set has been successfully applied to quickly search massive biological data sets.[17] However, here we focus here on efficient clustering, not efficient search.

Older, popular methods such as the well-known DBSCAN[6], algorithm seek to preserve the shape of data, but rely on the input lying in a metric space. In addition, these methods typically require at least quadratic time when the input data lies in three or more dimensions[7], and again do not account for missing values. Popular nonlinear dimensionality reduction methods, such as Laplacian eigenmaps[2], also don't account for missing data and noise, and many such approaches do not scale well.

# 3. EXPLOITING THE FUNDAMENTAL RESOLUTION OF GENETIC MAP DATA

Genetic map data for a homozygous *mapping population* can be represented as a binary matrix $X$, composed of rows $x_u$, where each entry $x_{ui}$ can take on one of two values $a$ or $b$, or it can be missing. [4] In this application domain, errors occur when an entry was erroneously recorded during sequencing – that is, it was flipped from $a$ to $b$ or from $b$ to $a$ – and errors typically occur at a fixed rate $\epsilon$. The goal of genetic mapping is to produce a map of the genome, which shows the correct clustering and ordering of the input $x_u$. Such maps have applications in health, agriculture, and the study of biodiversity.

Producing a genetic map typically requires three stages: 1) Clustering the vectors $x_u$ into linkage groups, 2) Ordering the vectors within each linkage group, and 3) Determining the correct genetic distance between the ordered vectors. In previous work ([13], [14]), we showed that by exploiting the fundamental resolution of genetic map data, we can quickly and accurately cluster the input vectors and reduce them from a large-scale, noisy, and incomplete data set into a small set of *bins* that more accurately represent the genome.

## 3.1 Scalable Clustering

We have shown that, using the well-known *LOD score* similarity that is ubiquitous in genetics, we can design a fast and accurate algorithm to cluster input data for genetic mapping [13]. The LOD score is a logarithm of odds, comparing the likelihood of two vectors being in the same cluster to the likelihood that they were generated by chance:

$$\mathrm{LOD}(x_u, x_v) = \log_{10} \frac{\mathrm{P(data}|x_u \text{ and } x_v \text{ in same linkage group)}}{\mathrm{P(data}|\text{ pure chance)}}$$

Because we have binary data, the denominator in the LOD fraction is simply $(\frac{1}{2})^\eta$, where $\eta$ is the number of entries that are non-missing in both $x_u$ and $x_v$. For example, if $x_u = \begin{bmatrix} a & b & b & - & b \end{bmatrix}$ and $x_v = \begin{bmatrix} a & a & - & - & a \end{bmatrix}$, then the denominator is $(\frac{1}{2})^3$, because the first, second, and last entries are non-missing entries in both $x_u$ and $x_v$. The numerator is a function of the estimated *recombination fraction* of the genetic data, and is explained in more detail in our previous work [13]. The LOD score does not obey the triangle inequality, which together with the presence of errors (flipped entries) and missing values, eliminates the possibility of accurately clustering the data with existing efficient algorithms.

| Dataset | Input Size | BubbleCluster | |
|---|---|---|---|
| | | $F$-score | Time |
| Barley | 64K | 0.9993 | 15 sec |
| Switchgrass | 113K | 0.9745 | 8.9 min |
| Switchgrass | 548K | 0.9894 | 1.9 hrs |
| Wheat | 1.582M | N/A | 1.22 hrs |

Table 1: Clustering performance on Barley, Switchgrass, and Wheat from the Joint Genome Institute using BubbleCluster. State-of-the-art mapping tools are unable to cluster data sets at this scale. (Table originally appeared in [13])

Our algorithm BubbleCluster, which resembles the DBSCAN method [6], utilizes the LOD score to efficiently cluster the data. First, we build a sketch of the clustering by linking together points that exceed a high LOD threshold, which only occurs for vectors with many matching binary values. Then, points with more missing values are linked to the point in the skeleton attaining the highest LOD score. The fundamental resolution limits the number of unique input vectors and thus as the data size grows, it is more likely that enough high-quality points exist to build the skeleton and accurately place the remaining points.

BubbleCluster allows for efficient clustering of genetic map input data into linkage groups. As Table 1 shows, our method achieved both high precision and recall (expressed as the $F$-score) on real genetic data. It was also the first method to successfully cluster genetic map data at large scales, including the grand challenge hexaploid bread wheat genome [3], and outperformed state-of-the-art mapping tools in terms of clustering performance on simulated data [13].

To further aide the efficiency and accuracy of the genetic mapping process, we introduced a fast data reduction method that quickly converts the large-scale, noisy, and incomplete input data into a small-scale, more accurate and more complete set of points which more clearly represent the genetic map. I will next describe this data reduction method, based on the fundamental resolution of the genetic mapping input data.

## 3.2 Efficient Data Reduction

Genetic map data has a fundamental resolution that is linear in the dimensionality of the input vectors. We can leverage this property of the data to efficiently reduce it to a much smaller and more accurate set of vectors we call *bins*, that represent positions along the genetic map as illustrated in Figure 2. The data reduction process uses a recursive bisection method to quickly reduce the input vectors within each linkage group to bins [14].

The binary nature of the data limits the number of possible unique input points to $2^n$, where $n$ is the dimensionality of the data. However, the fundamental resolution of the data limits this number much further – the fundamental resolution of a genetic map is equivalent to the number of possible unique positions on the map. With a homozygous mapping population (binary data), this number is $O(kn)$, where $k$ is the number of linkage groups (clusters) [14]. Therefore, when the number of input points is much larger than the fundamental resolution, many points must be identical, helping us filter out errors. Furthermore, the large data set size allows us to fill in missing data if we can cluster together points that belong to the same unique map position.

If we know two points belong in the same position on the

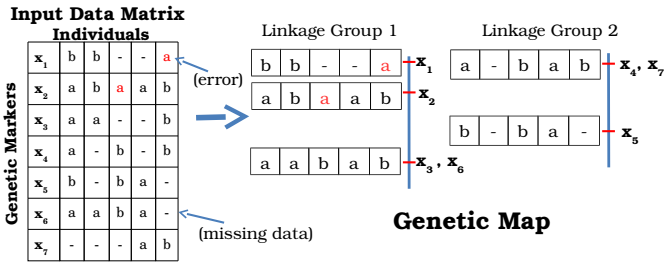


Figure 2: The fundamental resolution of genetic map data allows us to reduce the large-scale, incomplete and noisy input to a more complete set of representative vectors that more clearly describe positions along the genetic map.

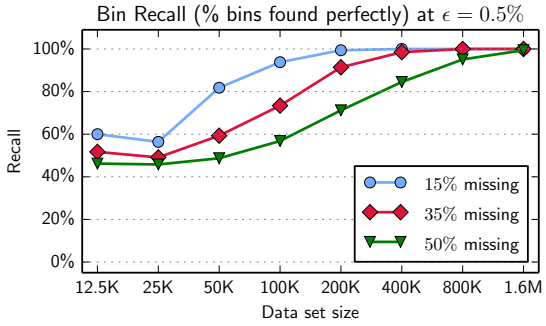


Figure 3: As the data size grows, more map position vectors (bins) are recovered perfectly, while lower missing rates allow our algorithm to recover the bins with less data.

genetic map, we can infer which values are errors and what the missing data should be. In Figure 2, for example, $x_3$ and $x_6$ belong to the same position, so we can fill in the missing data in both $x_3$ and $x_6$ based on each other's values. A similar idea applies to errors – the more points belong to the same position, the more clear it becomes which values are errors, as long as $\epsilon$ is fairly low.

We designed an algorithm that uses recursive bisection to quickly clusters together points in the same genetic map position. At each step, we use a maximum a posteriori (MAP) estimate of $\epsilon$ in order to find the best dimension along which to split the point set. The algorithm returns both an estimated error rate and a set of bins that represent unique positions on the genetic map. We showed that the number of bins and the error rate is consistent with existing real-world maps of wheat and barley[14].

We also simulated genetic map data with realistic error rates and a variety of missing data rates. Our algorithm scaled linearly with data set size for all tested missing rates and error rates in synthetic data. As Figure 3 shows, the bin *recall*, or fraction of bins we can recover perfectly, improves at each missing rate with more data. Note that although an error rate of 0.5% seems low, it is actually much higher than encountered in practice.

# 4. GENERALIZING TO DISCRETE-VALUED DATA WITH MISSING VALUES

Next, I will describe the last portion of my thesis work, clustering large discrete-valued data sets with missing values. One example of such data is found in the Recommender Systems (RS) domain, and much of the experimentation of these clustering methods will be on RS data.

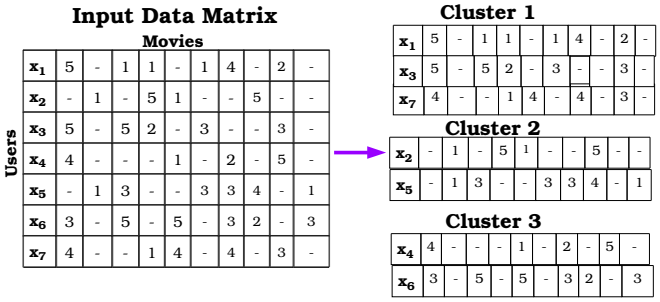


Figure 4: Fundamental resolution in Recommender Systems data equates to finding the vectors that best represent user sub-groups with low disagreement rates.

My hypothesis is that a fundamental resolution exists in many discrete-valued data sets, and can be exploited to efficiently cluster these data at large scales. In the RS domain, we often have a discrete-valued input matrix with many missing values as shown in Figure 4, where an entry $X_{ui}$ represents the rating that user $u$ gave to item $i$, with ratings typically taking values on a discrete scale. The methods for clustering and reducing the binary genetic map data can be adapted to this more general setting.

The fundamental resolution in RS data can be expressed as the number of unique user sub-groups that rate items very similarly. Recently, Christakopoulou et al.[5]have shown that utilizing user clusters to improve prediction of rating values and recommend better items is an extremely effective approach. An efficient, accurate clustering method for RS data has the potential to enhance such approaches to rating prediction as well as the *top-n* recommendation problem [1].

## 4.1 The LiRA Similarity Score for Recommender Systems

We have developed a statistical score analogous to the LOD score for RS data called *LiRa*, based on a likelihood ratio. We assume that RS data has a fundamental resolution, and thus users (rows of the input matrix) can be clustered into groups with vectors representing the rating trends of each group. LiRa compares the likelihood of observing the values in two user vectors $x_u$ and $x_v$ assuming the users are in the same cluster, to the likelihood of observing the same data by chance, based on differences in their rating values:

$$\text{LiRa}(x_u, x_v) = \log_{10} \frac{p(\text{differences in } x_u \text{ and } x_v | \text{ same cluster })}{p(\text{differences in } x_u \text{ and } x_v | \text{ pure chance})} \quad (1)$$

LiRa generalizes the LOD score by assuming that differences in two discrete-valued vectors from the same cluster follow a particular multinomial distribution, which is used to compute the numerator. The LiRa score is useful in the RS setting because it leverages more data to make a more accurate judgment of similarity.

We have shown that using the LiRa score to find nearest neighbors in a $k$-nearest neighbors approach outperforms the popular and widely used Pearson and Cosine similarity scores in terms of rating prediction error [15]. I am currently expanding on the clustering model used to compute the likelihood of users belonging to the same cluster in the LiRa score, which will be useful for efficient data reduction in the discrete-value setting.

## 4.2 Generalizing Efficient Data Reduction to the Discrete-Valued Domain

As noted previously, my goal is to generalize my previous work on efficient clustering and data reduction in genetic map data to the more general setting of large-scale, discrete-valued data with missing values and noise. The LiRa score from section 4.1 is the first step in this direction, and can be used to produce an initial clustering of the input using a thresholding scheme similar to the BubbleCluster algorithm. The threshold LiRa score within which points will belong to the same cluster will rely on the clustering model used to represent the data. For RS data, a working model is already presented in previous work [15].

After an initial fast clustering, I hope to generalize the data reduction stage to discrete-valued data also. Here, future work involves more precisely defining the point at which the fundamental resolution has been reached. In RS data, the idea is to cluster together users who have very similar rating patterns. One possibility is to only cluster together users if the distribution of their rating differences follows a clustering model. For example, in Cluster 1 in Figure 4, only one pair of ratings for the same item differs significantly: $x_{13} = 1$ and $x_{33} = 5$, giving a rating difference of 4. The remaining ratings are all close together. The recursive bisection method for data reduction in genetic mapping can be modified to this more general case, by dividing user clusters with large differences in rating values, based on MAP estimates of the proportion of each rating difference.

I am currently formalizing the notion of fundamental resolution in the general case, and experimenting with the best clustering model for RS data. As the final piece to my thesis, I hope to demonstrate that the efficient clustering and data reduction methods can be applied to more general data sets, and will be useful in the RS domain for rating prediction.

# 5. CONCLUSION

I have shown that the concept of *fundamental resolution* can be exploited to design efficient and accurate clustering algorithms in the genetic mapping domain, and I am extending this concept to the more general setting of discrete-valued data. The methods presented here are useful for applications with large-scale, discrete input data with many missing values, such as that found in the Recommender Systems domain. Future directions beyond my thesis work include exploring the connection between fractal dimension and fundamental resolution, as well as defining new clustering models for data sets with a low fundamental resolution.

# 6. ACKNOWLEDGMENTS

This work is supported by the Applied Mathematics Program of the DOE Office of Advanced Scientific Computing Research under contract number DE-AC02-05CH11231 and by NSF Award CCF-1637564.

# 7. REFERENCES

[1] D. C. Anastasiu, E. Christakopoulou, S. Smith, M. Sharma, and G. Karypis. Big data and recommender systems. 2016.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[3] J. A. Chapman, M. Mascher, A. Buluç, K. Barry, E. Georganas, A. Session, V. Strnadova, J. Jenkins, S. Sehgal, L. Oliker, et al. A whole-genome shotgun approach for assembling and anchoring the hexaploid bread wheat genome. *Genome biology*, 16(1):26, 2015.

[4] J. Cheema and J. Dicks. Computational approaches and software tools for genetic linkage map estimation in plants. *Briefings in bioinformatics*, 10(6):595–608, 2009.

[5] E. Christakopoulou and G. Karypis. Local item-item models for top-n recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 67–74. ACM, 2016.

[6] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[7] J. Gan and Y. Tao. Dbscan revisited: mis-claim, un-fixability, and approximation. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 519–530. ACM, 2015.

[8] A. Gionis, A. Hinneburg, S. Papadimitriou, and P. Tsaparas. Dimension induced clustering. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 51–60. ACM, 2005.

[9] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[10] J. Leskovec, A. Rajaraman, and J. D. Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[11] F. Lin and W. W. Cohen. Power iteration clustering. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 655–662, 2010.

[12] M. W. Mahoney and P. Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

[13] V. Strnadová, A. Buluç, J. Chapman, J. R. Gilbert, J. Gonzalez, S. Jegelka, D. Rokhsar, and L. Oliker. Efficient and accurate clustering for large-scale genetic mapping. In *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, pages 3–10. IEEE, 2014.

[14] V. Strnadová-Neeley, A. Buluç, J. Chapman, J. R. Gilbert, J. Gonzalez, and L. Oliker. Efficient data reduction for large-scale genetic mapping. In *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 126–135. ACM, 2015.

[15] V. Strnadová-Neeley, A. Buluç, J. R. Gilbert, L. Oliker, and W. Ouyang. Lira: A new likelihood-based similarity score for collaborative filtering. *arXiv preprint arXiv:1608.08646*, 2016.

[16] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[17] Y. W. Yu, N. M. Daniels, D. C. Danko, and B. Berger. Entropy-scaling search of massive biological data. *Cell systems*, 1(2):130–140, 2015.