PLASMA-HD: Probing the LAttice Structure and MAkeup of High-dimensional Data

David Fuhry, Yang Zhang, Venu Satuluri*, Arnab Nandi, Srinivasan Parthasarathy†
Department of Computer Science and Engineering
The Ohio State University
2015 Neil Avenue,Columbus, OH 43210-1277
Contact:{fuhry,srini}@cse.ohio-state.edu

ABSTRACT

Rapidly making sense of, analyzing, and extracting useful information from large and complex data is a grand challenge. A user tasked with meeting this challenge is often befuddled with questions on where and how to begin to understand the relevant characteristics of such data. Real-world problem scenarios often involve scalability limitations and time constraints.

In this paper we present an incremental interactive data analysis system as a step to address this challenge. This system builds on recent progress in the fields of interactive data exploration, locality sensitive hashing, knowledge caching, and graph visualization. Using visual clues based on rapid incremental estimates, a user is provided a multi-level capability to probe and interrogate the intrinsic structure of data. Throughout the interactive process, the output of previous probes can be used to construct increasingly tight coherence estimates across the parameter space, providing strong hints to the user about promising analysis steps to perform next.

We present examples, interactive scenarios, and experimental results on several synthetic and real-world datasets which show the effectiveness and efficiency of our approach. The implications of this work are quite broad and can impact fields ranging from top-k algorithms to data clustering and from manifold learning to similarity search.

1. INTRODUCTION

Our capability for collecting and storing data has far outstripped our ability to *efficiently explore and subsequently analyze* such data-stores. While database technology has provided us with the basic tools for accessing and manipulating such large and complex data-stores, the issue of how to help human end-users in making sense of such data in order to glean actionable insights has become a pressing issue.

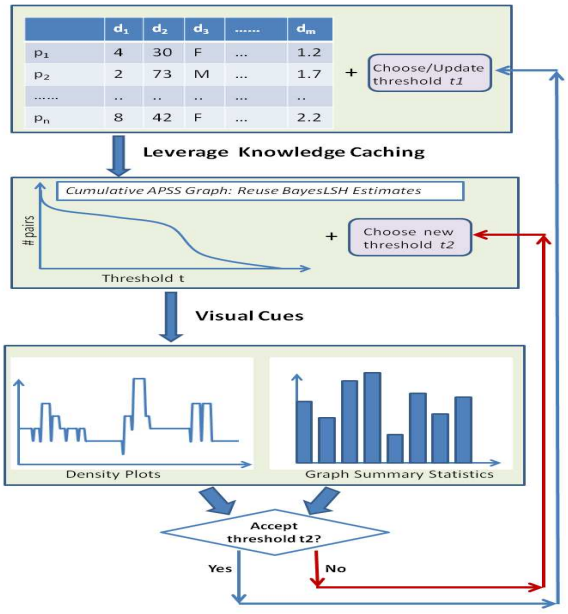


Figure 1: **PLASMA-HD Workflow**

An end-user is often overwhelmed with the data size and complexity and often does not have a clear path forward to understanding and probing the relevant characteristics of the data (e.g. attribute and entity interactions, intrinsic structure, data connectivity and clusterability) in order to make progress. The ability to efficiently explore, and browse data is thus absolutely essential for downstream analytics (e.g. parameter tuning) to mine actionable knowledge. An important consideration here in the context of big data is cognitive overload motivating the use of simple, visual cues to enhance one's ability to fuse and query data.

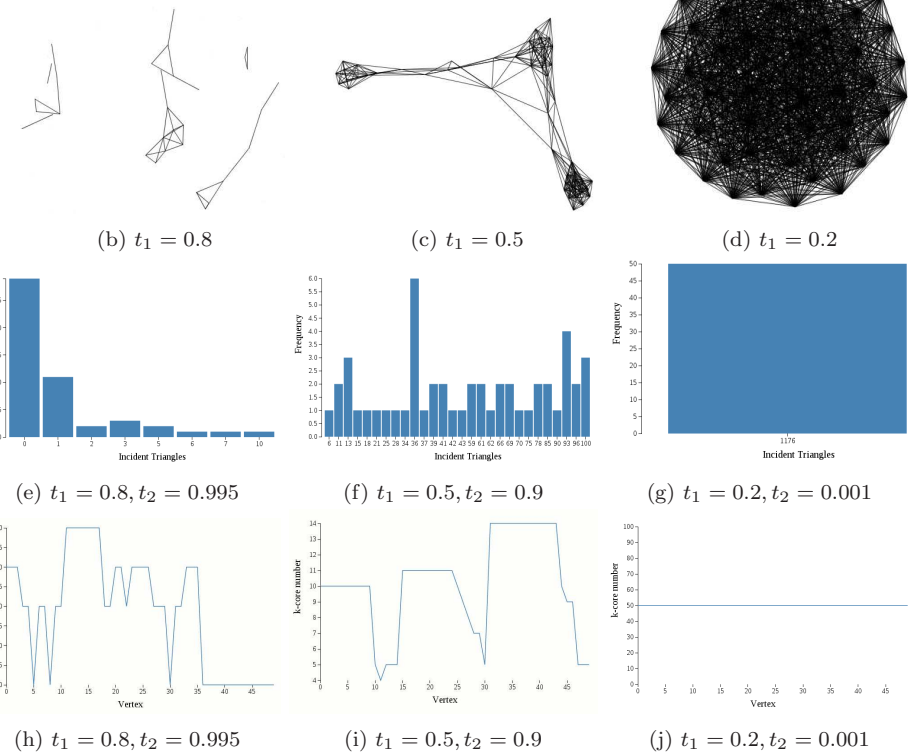
In this effort we present a system for Probing the LAttice Structure and MAkeup of High-dimensional Data (PLASMA-HD). For an arbitrary dataset, and given a similarity or distance measure-of-interest, PLASMA-HD enables an end-user to interactively explore the intrinsic connectivity or clusterability of a dataset under different threshold criteria. PLASMA-HD employs and enhances, a recently proposed idea called Bayesian Locality Sensitive Hashing (BayesLSH)[10], for efficiently estimating connectivity structure among database entities, at a given similarity or distance threshold. The crucial enhancement we propose in this paper, involves leveraging a form of knowledge caching[3, 7] wherein BayesLSH estimates at a given threshold level can be re-used to esti-

*Now at Twitter.
†To whom correspondence must be addressed.

Proceedings of the VLDB Endowment, Vol. 6, No. 12

	d_1	d_2	d_3
p_1	.07	.26	.96
p_2	.51	.56	.64
p_3	.20	.94	.30
p_4	.34	.15	.93
⋮			
p_{50}	.12	.98	.13

(a) Input D: 50 records



(b) $t_1 = 0.8$ (c) $t_1 = 0.5$ (d) $t_1 = 0.2$

(e) $t_1 = 0.8, t_2 = 0.995$ (f) $t_1 = 0.5, t_2 = 0.9$ (g) $t_1 = 0.2, t_2 = 0.001$

(h) $t_1 = 0.8, t_2 = 0.995$ (i) $t_1 = 0.5, t_2 = 0.9$ (j) $t_1 = 0.2, t_2 = 0.001$

Figure 2: PLASMA-HD overview on toy dataset. Of networks (b-d), only (c) with $t_1 = 0.5$ is community structure clear.

mate connectivity structure at other threshold levels. By leveraging the above transformation and converting a high dimensional dataset into a graphical (dimensionless) representation, PLASMA-HD then takes advantage of recent advances in graph and sub-graph visualization to provide end-users with relevant visual cues to understand the intrinsic structure of the data they are examining.

To convey a dataset's essential structural information estimated in a typical PLASMA-HD session we introduce the **Cumulative APSS Graph** which shows the number of similar pairs [1] as the similarity threshold is varied. The main utility of this visualization is that when the user studies the data at one similarity threshold, we can compute and display bounded estimates of the number of pairs at other thresholds not directly being studied. In this way the user is guided towards discovery of the most interesting data characteristics. An example scenario is presented in Section 2.2.

A typical PLASMA-HD workflow is depicted in Figure 1. Given a similarity-measure of interest, a typical user will probe the data D (alternatively, a sample of the data) with threshold t_1 to generate a Cumulative APSS Graph by leveraging BayesLSH[10]. This graph estimate can then be further probed by the user, **without accessing** D (ensuring interactive response times), through a combination of knowledge cached estimates and dimensionless *visual cues* (see Section 2.3) to determine the next threshold with which to probe the data. Unlike what is done typically the selection of the next threshold is not pre-canned but based on re-using estimates generated by the BayesLSH procedure. Once the new threshold is determined one can iterate through the process again (now using this new threshold to probe D).

[1]We also implemented a variant for triangles.

2. PLASMA-HD

In this section we discuss the intellectual engine of PLASMA-HD, i.e. mechanisms for exploring the underlying coherency and structure of high dimensional data built on top of a recently proposed idea for locality sensitive hashing and all-pairs similarity search. Specifically, we discuss how previously accumulated knowledge (obtained via intrinsic exploration or prior analysis) can be effectively exploited for future queries by providing appropriate systems support and algorithmic hooks to manage such knowledge. Our ideas for knowledge caching leverage prior work in incremental data mining [7, 8]. Finally, we discuss how the resulting knowledge can be interactively explored through visual cues drawn from recent advances in graph visualization[13, 12].

2.1 Enhancing BayesLSH

The all pairs similarity search problem – where one seeks to find all pairs of vectors whose similarity score (alternatively distance) exceeds a certain user-defined threshold – is a good starting point since it offers important insight on relevant characteristics-of-interest of our problem at hand. For example, Figure 2 shows a toy dataset and networks where all pairs of data points whose similarity is $\geq t_1$ are connected with an edge. $t_1 = 0.5$ as shown in 2c is one threshold for which good community structure is revealed for this dataset.

As noted by several researchers [2, 6] the all-pairs problem finds use not just in our specific use case, but also in a host of applications from query refinement for web search to collaborative filtering for advertisements and from duplicate

detection of documents to coalition fraud detection. Fundamental challenges in the context of this problem are: i) the choice of distance measure (not addressed in this paper); ii) the scale and dimensionality of the data – computing all pairs similarity naively is prohibitively expensive even for a single threshold value; and iii) a lack of guidance for selecting the threshold.

We address the latter two concerns by expanding upon a recently proposed idea for similarity search called BayesLSH[10]. BayesLSH adopts a principled Bayesian approach on top of Locality Sensitive Hashing (LSH) [1], to reason about and estimate the probability that a particular pair of objects will meet the user-specified threshold. Unpromising pairs are quickly pruned away based on these estimates realizing significant performance gains for a single threshold. Unlike most space-partitioning or space-filling curve approaches [5], sparse records are handled elegantly. For a candidate pair, a number of hashes n are incrementally computed. Let m be the number of hashes which match between the pair. A candidate pair is pruned when the probability that the similarity is greater than the threshold given computed hashes $M(m,n)$ becomes less than ϵ:

$$Pr(S \geq t|M(m,n)) < \epsilon \quad (1)$$

Besides pruning due to Equation 1, the other possibility is that the probability that the similarity is greater than the threshold becomes sufficiently large. A candidate pair is retained when the probability that the similarity estimate is accurate to within δ of the true similarity becomes greater than $1-\gamma$:

$$Pr(|\hat{s}(x,y) - s(x,y)| \geq \delta) < \gamma \quad (2)$$

Parameters δ and γ are user-specified.

Our main hypothesis is that its current avatar[10] the algorithm fails to retain very useful information across runs, which if retained, can help one make progress on the two challenges listed above. We memoize relevant information on hash match-sets and probability estimates (for each candidate pair evaluated) so as to generate a Cumulative APSS Graph described in Section 1.

Specifically, a candidate pair is evaluated until either Equation 1 or Equation 2 is satisfied. Before proceeding to the next candidate pair we log the maximum a posteriori similarity estimate of the pair given n, the number of hashes and m, the number of matching hashes, and the estimate variance. At the end of each query the cumulative distribution function of similarity estimates is updated. Plotting this distribution gives a useful hint to the user as to the number of pairs to expect at different thresholds. Based on comparisons with the ground truth, this heuristic is successful in pruning a fraction of the invalid candidates – the development of a more aggressive pruning mechanism is something we are currently looking into.

As described above the inference engine within the algorithm can be modified to compute such a histogram and estimate with reasonable accuracy. Examples for our test dataset d1 are shown in Figure 3 (best viewed in color), where the red line is the (initially unknown) ground truth number of pairs and other lines show estimates from the all-pairs algorithm run at user-selected thresholds. Error bars show slight increased uncertainty above - and more significant uncertainty below - the specified threshold due to concentration and pruning, respectively.

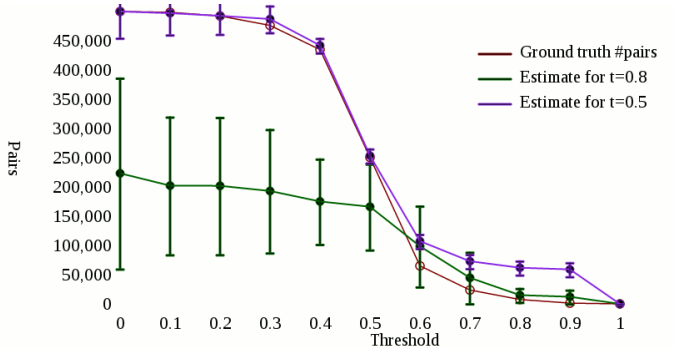


Figure 3: d1 with user selection of t_1 as 0.8 followed by 0.5.

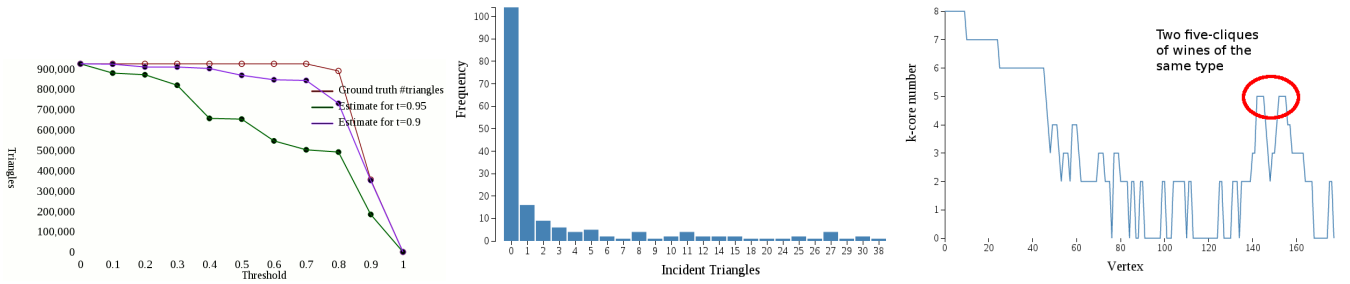
This data-driven histogram can potentially guide the user to pick the next threshold to evaluate, thereby avoiding a pre-canned data-independent protocol for threshold selection. Additionally, with each subsequent iteration on the dataset (with different thresholds) the accuracy of the histogram estimate will improve (this is provable). The memoization can also be viewed as a *knowledge cache*, enabling one to speed up subsequent iterations of the algorithm by re-using previously computed and memoized information. In a nutshell, previously computed hash match sets can be re-used to refine the priors and estimates the algorithm currently uses to prune and concentrate candidate pairs.

To ensure interactive response times PLASMA-HD relies on a combination of data sampling, fast similarity estimation through BayesLSH and knowledge cached estimates, coupled with statistical bounds (for accuracy estimates) for determining subsequent threshold probes. The above coupled tightly with dimensionless visual aids and correlated visual plots ensures a novel approach for the intrinsic interrogation of big data.

2.2 Interactive Scenario

Figure 3 shows two steps of an interactive scenario described below. The user begins with a blank canvas; suppose they first choose a similarity threshold of $t_1 = 0.8$. The system computes an estimate for the number of pairs at 0.8 as well as estimates at other thresholds and renders the green line. The user does not see the (dark red) ground truth line but we note that the green line is accurate at upper thresholds around 0.8. The user then notices the "knee" in steepness of the green line about threshold 0.5, and investigating it, selects a new similarity threshold of $t_1 = 0.5$. The system then computes the estimate for the 0.5 threshold and renders the purple line, which is much more accurate across lower similarity thresholds. Combining the upper threshold estimates for 0.8 (green) and the lower for 0.5 (purple), a close approximation to ground truth is obtained in just two steps. The 0.8 and 0.5 estimates as shown take 0.7 and 1.5 seconds to generate, respectively. The brute-force alternative, iteratively computing a pair-count estimate for each $0.0, 0.1, \ldots, 1.0$ threshold value, takes a total of 13.3 seconds. The interactive approach yields an 83% time savings, which can be even more significant for larger and more complex datasets.

To guide the user we provide error bars for each estimate, giving the user a feel for the parameter space. We next discuss augmenting these simple visualizations with enhanced visual cues and summary graph statistics.



(a) Triangle count, $t_1 = 0.9, 0.95$ (b) Triangle histogram, $t_1 = t_2 = 0.992$ (c) Triangle density plot, $t_1 = t_2 = 0.99$

Figure 4: wine dataset triangle count and visual cues

2.3 Visual Cues and Interaction

A limitation of Cumulative APSS Graph is that it cannot convey some data coherency characteristics such as density and clusterability. In addition running an all-pairs computation can be expensive lower thresholds (see Section 2.2).

To provide fast interactive insight into data coherency at different thresholds we provide some *visual cues* driven solely by the knowledge cache without requiring further access to source data D. Once all-pairs is run at a threshold t_1, these visualizations can be generated repeatedly for any threshold t_2 as shown in Figure 1.

The currently implemented visual cues are 1) triangle vertex cover histogram and 2) triangle density plot shown in Figs. 4b and 4c for the wine dataset from UCI machine learning repository. Since triangles are closely related to clusterability [11], the histogram of the number of triangles incident on each vertex gives the user an estimate of how clusterable the data is. The triangle density plot [12, 13] method visualizes cohesive subgraphs (dense subcomponents) within a large graph. The density plot is the clique distribution of the graph and flat peaks in the plot indicate potential cliques.

3. AUDIENCE EXPERIENCE

We plan a demonstration that leads audience members through various elements of PLASMA-HD. Starting with candidate datasets drawn from text, social media, image processing, biomedical and WWW application domains users will be allowed to explore the different interactive elements of the proposed system. Currently two popular metrics of similarity are supported (e.g. Jaccard, Cosine) and we hope to have other similarity metrics in time for the live demonstration. In its current avatar PLASMA-HD is implemented as both a standalone system operating on a laptop or tablet, as well as a web-based service working off a back-end server, where one may upload one's own data. Users will have the option to choose between these two options.

Users will experience the workflow presented in Figure 1 and will come across the various exemplar visual cues discussed in this extended abstract as well as some additional ones that we lacked space to describe. Explicit feedback from audience members will be sought to rate and evaluate the system along the traditional axes of quality, efficiency and usability. While we do plan to have some pre-canned workflow experience for time-constrained audience members, the more inquisitive members of the audience will have the opportunity to probe various datasets in a relaxed manner.

4. POTENTIAL IMPACT

The ability to probe the intrinsic structure of data can have broad scientific appeal. Within the database community a tool like PLASMA-HD can help with NN [9] and Reverse NN [4] searches as well as help with identifying good parameters for indexing – especially clustered indexing. A fundamental challenge for large scale clustering algorithms is to determine K the number of clusters that truly model the data. The PLASMA-HD system through its interactive guidance and visual cues may provide some guidance on this – particularly through its density plots. Similarly, a fundamental pre-processing step for a number of learning algorithms (e.g. manifold learning) is to first derive a nearest neighbor graph. Selecting the threshold for identifying such an NN Graph can be accomplished through PLASMA-HD.

Acknowledgements: This work was supported by Google grant "Guided and Interactive Exploration of Large Datasets" and NSF SoCS grant IIS-1111118.

5. REFERENCES

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, Jan. 2008.

[2] R. J. Bayardo, Y. Ma, and R. Srikant. Scaling up all pairs similarity search. In *WWW'07*, 2007.

[3] A. Ghoting and S. Parthasarathy. Knowledge-conscious exploratory data clustering. Technical report, KDD, 2006.

[4] F. Korn and S. Muthukrishnan. Influence sets based on reverse nearest neighbor queries. In *SIGMOD'00*, SIGMOD '00, pages 201–212, New York, NY, USA, 2000. ACM.

[5] M. D. Lieberman, J. Sankaranarayanan, and H. Samet. A fast similarity join algorithm using graphics processing units. In *ICDE '08*, 2008.

[6] A. Metwally and C. Faloutsos. V-smart-join: a scalable mapreduce framework for all-pair similarity joins of multisets and vectors. *PVLDB*, 5(8):704–715, Apr. 2012.

[7] B. Nag, P. M. Deshpande, and D. J. DeWitt. Using a knowledge cache for interactive discovery of association rules. In *KDD'99*, pages 244–253, New York, NY, USA, 1999. ACM.

[8] S. Parthasarathy, M. J. Zaki, M. Ogihara, and S. Dwarkadas. Incremental and interactive sequence mining. In *CIKM'99*, pages 251–258, New York, NY, USA, 1999. ACM.

[9] N. Roussopoulos, S. Kelley, and F. Vincent. Nearest neighbor queries. *SIGMOD Rec.*, 24(2):71–79, May 1995.

[10] V. Satuluri and S. Parthasarathy. Bayesian locality sensitive hashing for fast similarity search. *PVLDB*, 5(5), Jan. 2012.

[11] C. E. Tsourakakis, U. Kang, G. L. Miller, and C. Faloutsos. Doulion: counting triangles in massive graphs with a coin. In *KDD'09*, pages 837–846, New York, NY, USA, 2009. ACM.

[12] N. Wang, S. Parthasarathy, K.-L. Tan, and A. K. H. Tung. Csv: visualizing and mining cohesive subgraphs. In *SIGMOD'08*, pages 445–458, New York, NY, USA, 2008. ACM.

[13] Y. Zhang and S. Parthasarathy. Extracting analyzing and visualizing triangle k-core motifs within networks. In *ICDE'12*.