Brief Announcement: Hamming Distance Completeness and Sparse Matrix Multiplication

Daniel Graf
Department of Computer Science, ETH Zürich, Switzerland
daniel.graf@inf.ethz.ch

Karim Labib
Department of Computer Science, ETH Zürich, Switzerland
labibk@student.ethz.ch

Przemysław Uznański
Department of Computer Science, ETH Zürich, Switzerland
przemyslaw.uznanski@inf.ethz.ch

Abstract

We show that a broad class of $(+, \diamond)$ vector products (for binary integer functions $\diamond$) are equivalent under one-to-polylog reductions to the computation of the Hamming distance. Examples include: the dominance product, the threshold product and ℓ_{2p+1} distances for constant p. Our results imply equivalence (up to $\operatorname{poly}\log n$ factors) between complexity of computation of All Pairs: Hamming Distances, ℓ_{2p+1} Distances, Dominance Products and Threshold Products. As a consequence, Yuster's (SODA'09) algorithm improves not only Matoušek's (IPL'91), but also the results of Indyk, Lewenstein, Lipsky and Porat (ICALP'04) and Min, Kao and Zhu (COCOON'09). Furthermore, our reductions apply to the pattern matching setting, showing equivalence (up to $\operatorname{poly}\log n$ factors) between pattern matching under Hamming Distance, ℓ_{2p+1} Distance, Dominance Product and Threshold Product, with current best upperbounds due to results of Abrahamson (SICOMP'87), Amir and Farach (Ann. Math. Artif. Intell.'91), Atallah and Duket (IPL'11), Clifford, Clifford and Iliopoulous (CPM'05) and Amir, Lipsky, Porat and Umanski (CPM'05). The resulting algorithms for ℓ_{2p+1} Pattern Matching and All Pairs ℓ_{2p+1}, for $2p + 1 = 3, 5, 7, \ldots$ are new.

Additionally, we show that the complexity of AllPairsHammingDistances (and thus of other aforementioned AllPairs- problems) is within $\operatorname{poly}\log n$ from the time it takes to multiply matrices $n \times (n \cdot d)$ and $(n \cdot d) \times n$, each with $(n \cdot d)$ non-zero entries. This means that the current upperbounds by Yuster (SODA'09) cannot be improved without improving the sparse matrix multiplication algorithm by Yuster and Zwick (ACM TALG'05) and vice versa.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases fine-grained complexity, matrix multiplication, high dimensional geometry, pattern matching

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.109

1 Introduction

Many classical algorithmic problems received new attention when formulated as algebraic problems. In pattern matching we can define a similarity score between two strings and ask for this score between the pattern $\mathbf{P}$ of length m and every m-substring of the text $\mathbf{T}$ of length $n \geq m$. For example, scores of Hamming distance or L_1 distance between numerical strings generalize the classical pattern matching. All those problems share an additive structure, i.e. for an input pattern $\mathbf{P}$ and text $\mathbf{T}$, the score vector $\mathbf{O}$ is such that $\mathbf{O}[i] = \sum_j \mathbf{P}[j] \diamond \mathbf{T}[i + j]$

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 109; pp. 109:1–109:4
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 Summary of different score functions and the corresponding problems. $\mathbf{1}[\varphi]$ is 1 iff φ and 0 otherwise.

Name	Score function	Pattern Matching problem	All Pairs problem
Hamming	$\mathbf{1}[x \neq y]$	$\mathbf{O}[i] = \lvert\{j : \mathbf{P}[j] \neq \mathbf{T}[i+j]\}\rvert$	$O[i][j] = \lvert\{k : \mathbf{A}_i[k] \neq \mathbf{B}_j[k]\}\rvert$
Dominance	$\mathbf{1}[x \leq y]$	$\mathbf{O}[i] = \lvert\{j : \mathbf{P}[j] \leq \mathbf{T}[i+j]\}\rvert$	$O[i][j] = \lvert\{k : \mathbf{A}_i[k] \leq \mathbf{B}_j[k]\}\rvert$
δ-Threshold	$\mathbf{1}[\lvert x-y\rvert \geq \delta]$	$\mathbf{O}[i] = \lvert\{j : \lvert\mathbf{P}[j] - \mathbf{T}[i+j]\rvert > \delta\}\rvert$	$O[i][j] = \lvert\{k : \lvert\mathbf{A}_i[k] - \mathbf{B}_j[k]\rvert > \delta\}\rvert$
ℓ_1 distance	$\lvert x-y\rvert$	$\mathbf{O}[i] = \sum_j \lvert\mathbf{P}[j] - \mathbf{T}[i+j]\rvert$	$O[i][j] = \sum_{k=1}^{n} \lvert\mathbf{A}_i[k] - \mathbf{B}_j[k]\rvert$

for some binary function $\diamond$. Just as those pattern matching generalizations are based on *convolution*, there is a family of problems based on *matrix multiplication*, varying in flavour according to the vector product used. There, we are given two matrices A and B and the output is the matrix $O[i][j] = \sum_k A[i][k] \diamond B[k][j]$. This is equivalent to the computation of all pairwise $(+,\diamond)$-vector products for two vector families, the so called AllPairs- problems. For a certain class of score functions, pattern matching generalizations admit independently algorithms of identical complexity $\mathcal{O}(n\sqrt{m \log m})$ (c.f. [1–3, 8]). For the same score functions, the best algorithms for corresponding AllPairs- problems are of complexity $\mathcal{O}(n^{(\omega+3)/2})$ or similar (c.f. [6, 8, 11]).

Our contribution:

We show that for a wide class of $(+,\diamond)$ products, the corresponding problems are of (almost) equivalent hardness. This class includes Hamming distance or Dominance, but also any piecewise polynomial function of two variables (for appropriate definition of piecewise polynomiality, c.f. Definition 2) excluding certain degenerate forms (e.g. polynomials). Thus we should not expect the problems based on $(+,\diamond)$ products to be significantly harder to compute than e.g. ones based on Hamming distance. The reduction applies both to Pattern Matching setting and to All Pairs- setting alike. We refer to Table 1 for a summary of considered problems and to Figure 1 for a summary of the old and new reductions. It implies that Yuster's [11] improvement to the exponent of AllPairsDominanceProducts applies to all other AllPairs- problems considered here. Additionally, any tradeoffs between vectors dimension and runtime (c.f. [5, 8]), or input sparsity and runtime (c.f. [4, 9, 10]) translates between problems. Additionally, we link the complexity of AllPairsHammingDistances (and thus to other AllPairs- problems) to one of a sparse rectangular matrix multiplication (c.f. Theorem 4): an instance of APHam can be expanded to an instance of sparse matrix multiplication of rectangular matrices, and any matrix multiplication instance with those parameters can be contracted back to APHam. It is interesting to observe that applying the fastest existing sparse matrix multiplication algorithm (c.f. [12]) to the resulting instance results in the same runtime as solving APHam directly.

2 Preliminaries

For vectors $\mathbf{A}, \mathbf{B}$ and matrices $\mathcal{A}, \mathcal{B}$, we denote the $(+,\diamond)$ vector product as $\text{vprod}(\diamond, \mathbf{A}, \mathbf{B}) \stackrel{\text{def}}{=} \sum_i \mathbf{A}[i] \diamond \mathbf{B}[i]$, the $(+,\diamond)$ convolution as $\text{conv}(\diamond, \mathbf{A}, \mathbf{B}) = \mathbf{C}$ where $\mathbf{C}[k] = \sum_{i+j=k} \mathbf{A}[i] \diamond \mathbf{B}[j]$ and the $(+,\diamond)$ matrix product as $\text{mprod}(\diamond, \mathcal{A}, \mathcal{B}) = \mathcal{C}$ where $\mathcal{C}[i,j] = \sum_k \mathcal{A}[i,k] \diamond \mathcal{B}[k,j]$.

Thus, e.g. defining $\mathsf{Ham}(x,y) \stackrel{\text{def}}{=} \mathbf{1}[x \neq y]$, then $\text{vprod}(\mathsf{Ham}, \cdot, \cdot)$, $\text{conv}(\mathsf{Ham}, \cdot, \cdot)$ and $\text{mprod}(\mathsf{Ham}, \cdot, \cdot)$ correspond to Hamming Distance between vectors, HamPM and APHam.

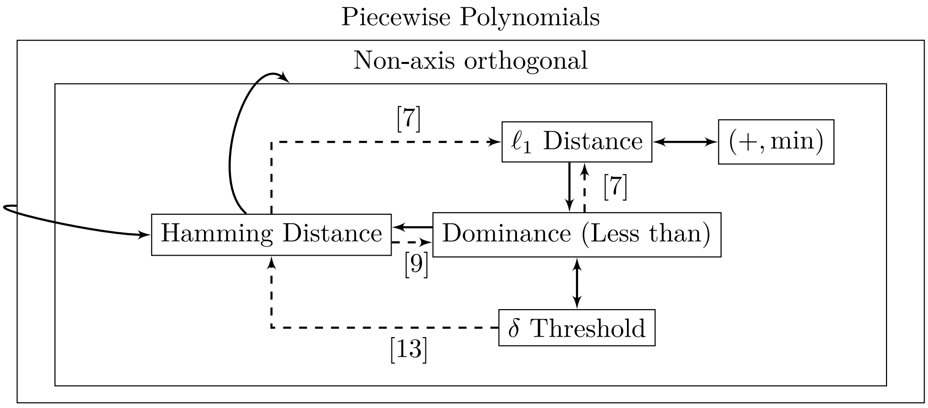

Figure 1 Existing and new reductions between problems, together with problem classes.

▶ **Definition 1.** We say that $\diamond$ reduces preserving linearity to instances of $\Box_1, \ldots, \Box_K$, if there are functions $f_1, \ldots, f_K$ and $g_1, \ldots, g_K$ and coefficients $\alpha_1, \ldots, \alpha_K$, such that for any x, y:[1] $x \diamond y = \sum_i \alpha_i \cdot \big(f_i(x) \Box_i\ g_i(y)\big)$.

Given Definition 1, we have for any vectors $\mathbf{A}, \mathbf{B}$ and matrices $\mathcal{A}, \mathcal{B}$: $\textsc{vprod}(\diamond, \mathbf{A}, \mathbf{B}) = \sum_i \alpha_i \cdot \textsc{vprod}(\Box_i, f_i(\mathbf{A}), g_i(\mathbf{B}))$, $\textsc{conv}(\diamond, \mathbf{A}, \mathbf{B}) = \sum_i \alpha_i \cdot \textsc{conv}(\Box_i, f_i(\mathbf{A}), g_i(\mathbf{B}))$ and $\textsc{mprod}(\diamond, \mathcal{A}, \mathcal{B}) = \sum_i \alpha_i \cdot \textsc{mprod}(\Box_i, f_i(\mathcal{A}), g_i(\mathcal{B}))$, where $f(\mathbf{A})$ and $f(\mathcal{A})$ denotes a coordinate-wise application of f to vector $\mathbf{A}$ and matrix $\mathcal{A}$, respectively.

3 Main results

▶ Remark. We assume that all input values and coefficients are integers bounded in absolute value by $\text{poly}(n)$.

▶ **Definition 2.** For integers A, B, C and polynomial $P(x, y)$ we say that the function $P(x, y) \cdot \mathbf{1}[Ax + By + C > 0]$ is *halfplane polynomial*. We call a sum of halfplane polynomial functions a *piecewise polynomial*. We say that a function is *axis-orthogonal piecewise polynomial*, if it is piecewise polynomial and for every i, $A_i = 0$ or $B_i = 0$.

Observe that $\mathsf{Ham}(x, y) = \mathbf{1}[x > y] + \mathbf{1}[x < y]$, $\max(x, y) = x \cdot \mathbf{1}[x \ge y] + y \cdot \mathbf{1}[x < y]$, $|x - y|^{2p+1} = (x - y)^{2p+1} \cdot \mathbf{1}[x > y] + (y - x)^{2p+1} \cdot \mathbf{1}[x < y]$, and $\mathsf{Thr}_\delta(x, y) \stackrel{\text{def}}{=} \mathbf{1}[|x - y| \ge \delta] = \mathbf{1}[x \le y - \delta] + \mathbf{1}[x \ge y + \delta]$.

▶ **Theorem 3.** *Let $\diamond$ be a piecewise polynomial of constant degree and* $\text{poly}\log n$ *number of summands.*

- *If $\diamond$ is axis orthogonal, then $\diamond$ is "easy": $(+, \diamond)$ convolution takes $\widetilde{O}(n)$ time, $(+, \diamond)$ matrix multiplication takes $\widetilde{O}(n^\omega)$ time.*
- *Otherwise, $\diamond$ is* Hamming distance complete*: under one-to-polylog reductions, $(+, \diamond)$ product is equivalent to Hamming distance, $(+, \diamond)$ convolution is equivalent to* HamPM *and $(+, \diamond)$ matrix multiplication is equivalent to* APHam.

[1] For the sake of simplicity, we are omitting in the definition the post-processing function necessary e.g. $(\,\cdot\,)^{1/p}$ for L_p norms.

▶ **Theorem 4.** *The time complexity of* APHam *on* n *vectors of dimension* d *is (under randomized Las Vegas reductions) within* $\text{poly}\log n$ *from time it takes to multiply matrices* $n \times (n \cdot d)$ *and* $(n \cdot d) \times n$*, each with* $(n \cdot d)$ *non-zero entries.*

References

1. Amihood Amir, Ohad Lipsky, Ely Porat, and Julia Umanski. Approximate matching in the L_1 metric. In *CPM*, pages 91–103, 2005. doi:10.1007/11496656_9.
2. Mikhail J. Atallah and Timothy W. Duket. Pattern matching in the hamming distance with thresholds. *Inf. Process. Lett.*, 111(14):674–677, 2011. doi:10.1016/j.ipl.2011.04.004.
3. Peter Clifford, Raphaël Clifford, and Costas S. Iliopoulos. Faster algorithms for δ,γ-matching and related problems. In *CPM*, pages 68–78, 2005. doi:10.1007/11496656_7.
4. Ran Duan and Seth Pettie. Fast algorithms for (max, min)-matrix multiplication and bottleneck shortest paths. In *SODA*, pages 384–391, 2009. URL: http://dl.acm.org/citation.cfm?id=1496770.1496813.
5. Omer Gold and Micha Sharir. Dominance product and high-dimensional closest pair under L_∞. In *ISAAC*, pages 39:1–39:12, 2017. doi:10.4230/LIPIcs.ISAAC.2017.39.
6. Piotr Indyk, Moshe Lewenstein, Ohad Lipsky, and Ely Porat. Closest pair problems in very high dimensions. In *ICALP*, pages 782–792, 2004. doi:10.1007/978-3-540-27836-8_66.
7. Ohad Lipsky and Ely Porat. L_1 pattern matching lower bound. *Inf. Process. Lett.*, 105(4):141–143, 2008. doi:10.1016/j.ipl.2007.08.011.
8. Kerui Min, Ming-Yang Kao, and Hong Zhu. The closest pair problem under the Hamming metric. In *COCOON*, pages 205–214, 2009. doi:10.1007/978-3-642-02882-3_21.
9. Virginia Vassilevska. *Efficient algorithms for path problems in weighted graphs.* PhD thesis, Carnegie Mellon University, 2008.
10. Virginia Vassilevska, Ryan Williams, and Raphael Yuster. All pairs bottleneck paths and max-min matrix products in truly subcubic time. *Theory of Computing*, 5(1):173–189, 2009. doi:10.4086/toc.2009.v005a009.
11. Raphael Yuster. Efficient algorithms on sets of permutations, dominance, and real-weighted APSP. In *SODA*, pages 950–957, 2009. URL: http://dl.acm.org/citation.cfm?id=1496770.1496873.
12. Raphael Yuster and Uri Zwick. Fast sparse matrix multiplication. *ACM Trans. Algorithms*, 1(1):2–13, 2005. doi:10.1145/1077464.1077466.
13. Peng Zhang and Mikhail J. Atallah. On approximate pattern matching with thresholds. *Inf. Process. Lett.*, 123:21–26, 2017. doi:10.1016/j.ipl.2017.03.001.