Frequent Knot Discovery

Floris Geerts
Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh, UK
fgeerts@inf.ed.ac.uk

February 7, 2008

Abstract

We explore the possibility of applying the framework of frequent pattern mining to a class of continuous objects appearing in nature, namely knots. We introduce the frequent knot mining problem and present a solution. The key observation is that a database consisting of knots can be transformed into a transactional database. This observation is based on the Prime Decomposition Theorem of knots.

1 Introduction

Many algorithms have recently been developed for mining frequent patterns. Traditionally, these patterns consist of subsets of attributes in a relational database [1]. Recently, other patterns have been mined, such as trees [27] and graphs [13, 14, 25, 10]. However, most objects appearing in nature lack the discrete character of graph and trees. In this paper we explore the possibility of applying the framework of frequent pattern mining to a class of continuous objects appearing in nature, namely knots. A knot can be thought of as a piece of rope (where the rope has zero thicknes) which forms a loop in three-dimensional Euclidean space $\mathbb{R}^3$. Figure 1 shows an example of a knot known as the Trefoil knot.

The history of knots dates back to the late 1800's when Lord Kelvin suggested that atoms where knots in an invisible and frictionless fluid. Since then, theoretical properties of knots are extensively studied in mathematics [3]. In physics, knot invariants (e.g., the Jones polynomial) are used in statistical physics [12] and knots also appear in the context of quantum gravity [2]. Recently, knots showed up as building blocks for future quantum computers [15].

In biology, knots are used to characterize topoisomerase enzymes [22] and in polymer science, physical properties of long ring polymers, such as DNA, gels and rubbers are related to properties of knots [4, 5, 24]. It is shown that knots are present in such polymers with probability one when the polymers are long enough [21]. There is also much interest in developing artificial knotted biopolymers as building blocks for DNA-based computing [18].

The study of knotted polymers is done both by using experimentally obtained knots and by using knots obtained by numerical approaches based on self-avoiding random-walk simulations. Examples of questions one would like to answer in these studies are what is the probability of having a certain knot in polymers of a certain length [19], and whether the knots appear tight or loose in the knotted polymers [11].

In this article we consider the *frequent knot mining problem* which can be stated as follows: Given a collection of knots, find all subknots which appear frequently in this collection.

We believe that finding frequent subknots in a large collection of real or simulated knotted polymers, will contribute to a deeper understanding of the statistical properties of knotted polymers in $\mathbb{R}^3$. This article reports a first attempt for solving the frequent knot mining problem.

The solution presented in this article consists of three steps:

1. Encoding of knots in transactions;
2. Mining these transactions; and finally,
3. Decoding of the frequent itemsets into knots.

The article is organized as follows: In Section 2, definitions are given and the frequent knot mining problem is stated formally. The encoding (decoding) of knots (transactions) into transactions (knots) is described in Section 3 . In Section 4, we present the KnotMiner algorithm for mining frequent knots. Finally, conclusions are drawn in Section 5.

2 Preliminaries

A *knot* K can be thought of as a piece of rope (where the rope has zero thickness) which forms a loop in three-dimensional Euclidean space $\mathbb{R}^3$. Two knots K and K' are equivalent, or in symbols $K \equiv K'$, if they can be transformed into each other without cutting and pasting the ropes. We will only consider so-called tame knots. These are knots which are equivalent to piecewise linear knots, i.e.,

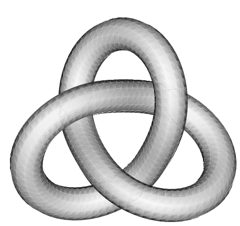

Figure 1: Trefoil knot.

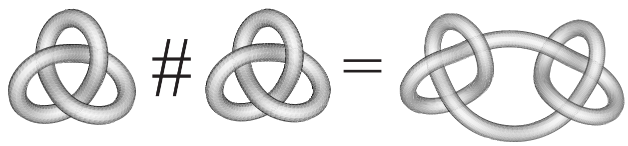

Figure 2: Sum of two knots.

knots consisting of a finite number of straight lines. A knot is trivial if it is equivalent to a rope which forms a circle in a plane in $\mathbb{R}^3$.

A knot can be finitely represented by a *knot diagram.* The knot diagram of a knot K is a connected undirected planar graph, which correspond to a (generic) projection of K onto a plane. Vertices in a knot diagram correspond to places where the projection of the knot intersects, and each edge adjacent to a vertex is labelled as an undercrossing or overcrossing, whichever is the case. Given a knot diagram consisting of n vertices one can find in time polynomial in n a piecewise linear knot such that its z-projection gives the original diagram [8]. Two knot diagrams can be transformed into each other using the so-called Reidemeister moves if and only if they represent equivalent knots [3].

The *connected sum* of two knots K_1 and K_2 is formed by removing a small piece of rope from both knots and then connecting the four endpoints by two new pieces of rope in such a way that no new crossings are introduced, the result being a a single knot, which is denoted by $K = K_1 \# K_2$. This operation is illustrated in Figure 2. The connected sum $K_1 \# K_2$ is equivalent to $K_2 \# K_1$ and $(K_1 \# K_2) \# K_3$ is equivalent to $K_1 \# (K_2 \# K_3)$. The connected sum of a knot K and the trivial knot is equivalent to K [3].

A knot is called *prime* if for any decomposition as a connected sum, one of the factors is the trivial knot. There are infinitely many prime knots.

Theorem 1 (Prime decomposition Theorem [3, 17]) *Every knot K can be decomposed as a connected sum of nontrivial prime knots. If $K \equiv K_1 \# K_2 \# \cdots \# K_m$ and $K \equiv L_1 \# L_2 \# \cdots \# L_n$, where K_i and L_i are nontrivial prime knots, then $m = n$, and after reordering each K_i is equivalent to L_i.*

This theorem motivates the following definition. Let

$$K = K_1 \# K_2 \# \cdots \# K_p$$

and let

$$L = L_1 \# L_2 \# \cdots \# L_q.$$

Then, K is a *subknot* of L, or $K \preceq L$, if for any $i = 1, \ldots, p$ we have that

$$|\{j \mid K_j \equiv K_i\}| \leq |\{j \mid L_j \equiv K_i\}|.$$

A *knot datatabase* $\mathcal{D}$ is a finite collection of knots. The *support* of a knot K in $\mathcal{D}$ is defined as

$$\mathsf{supp}(K) = |\{L \in \mathcal{D} \mid K \preceq L\}|.$$

The *frequent knot mining problem* can be stated as follows: Given a knot database $\mathcal{D}$ and a threshold value $\sigma \in \mathbb{N}$, find all knots K in $\mathcal{D}$ such that $\mathsf{supp}(K) > \sigma$.

For completeness, we also state the frequent itemset mining problem. A *transaction database* $\mathcal{T}$ is a finite collection of k-tuples in $\mathbb{N}^k$. The *support* of an itemset I in $\mathcal{T}$ is defined as

$$\mathsf{supp}(I) = |\{J \in \mathcal{T} \mid \forall \ell : (I)_\ell \leq (J)_\ell\}|,$$

where $(I)_\ell$ (resp. $(J)_\ell$) denotes the ℓth component of I (resp. J). The *frequent itemset mining problem* is then: Given a transaction database $\mathcal{T}$ and a threshold value $\sigma \in \mathbb{N}$, find all itemsets I in $\mathcal{T}$ such that $\mathsf{supp}(I) > \sigma$.

3 From Knot Databases to Transaction Databases

In this section we show how to transform a knot database $\mathcal{D}$ into a transactional database. We assume that the knots in $\mathcal{D}$ are represented by knot diagrams.

We start by computing for each knot K in $\mathcal{D}$ its prime decomposition. Schubert [17] gives an algorithm computing this decomposition. The running time is at worst exponential in the number of vertices in the knot diagram. In this way, we obtain a set $\mathsf{primes}(\mathcal{D})$ consisting of knot diagrams for all prime knots occurring in $\mathcal{D}$. Two different knot diagrams in $\mathsf{primes}(\mathcal{D})$ can represent the same prime knot, so we have to eliminate duplicates. There exists an algorithm for testing whether two knot diagrams represent equivalent knots [9, 23]. However, at present, the complexity of this algorithm is not known. From here on, we assume that $\mathsf{primes}(\mathcal{D})$ does not contain duplicates and order it arbitrarily.

We now define a mapping, denoted by encode, from knots in a knot database to elements in a transaction database. Let $\mathcal{D}$ be a knot database, and K a knot in $\mathcal{D}$. Then,

$$\mathsf{encode}(K) = (n_1, \ldots, n_p),$$

with $p = |\mathsf{primes}(\mathcal{D})|$ and n_i is the number of times the prime knot corresponding to the ith knot diagram in $\mathsf{primes}(\mathcal{D})$ appears in the prime decomposition of K. Clearly, $\mathsf{encode}(\mathcal{D})$ is a transaction database consisting of $|\mathsf{primes}(\mathcal{D})|$ attributes.

Given a set of knots $\mathcal{K} = \{K_1, \ldots, K_p\}$, we now define the mapping, denoted by decode, from itemsets of a transaction database $\mathcal{T} \subset \mathbb{N}^p$ to knots in $\mathbb{R}^3$. Let $t \in \mathcal{T}$ and let $(t)_{i_1,\ldots,i_k} = (m_1, \ldots, m_k)$ be a k-itemset. Then,

$$\mathsf{decode}(m_1, \ldots, m_k) = \underbrace{K_{i_1} \# \cdots \# K_{i_1}}_{m_1 \text{ times}} \# \cdots \# \underbrace{K_{i_k} \# \cdots \# K_{i_k}}_{m_k \text{ times}}.$$

For any knot K, we have that $K \equiv \mathsf{decode}(\mathsf{encode}(K))$.

4 Algorithm

We now present The KnotMiner algorithm for computing the frequent knots in a knot database.

Algorithm 1: KnotMiner

Input: knot database $\mathcal{D}, \sigma$
Output: All knots K such that $\mathsf{supp}(K) > \sigma$.
1: Compute $\mathcal{T} := \mathsf{encode}(\mathcal{D})$
2: Compute the set $\mathcal{F}$ of frequent itemsets in $\mathcal{T}$
3: Output $\mathsf{decode}(\mathcal{F})$.

The first and last step in KnotMiner are already fully explained in Section III. For the second step one can either transform $\mathcal{T}$ into a binary transaction database and use a standard mining algorithm like Apriori [1], Eclat [26] or FP-growth [7]. Alternatively one can mine $\mathcal{T}$ directly using algorithms presented in [20] and [16]. The following result is immediate.

Theorem 2 *The KnotMiner algorithm works correctly.*

5 Concluding Remarks and Future Work

In this article we introduced the frequent knot mining problem and proposed the KnotMiner algorithm to solve it. Currently, there exists no implementation of KnotMiner. This is mainly due to the complex algorithms needed for the encoding of a knot database into a transactional databases.

However, recent research indicates that the knot decomposition of knotted polymers can be obtained by "Coulomb decomposition", which is a technique where polymers are brought into an equilibrium state using Coulomb interactions [6]. We hope to apply this technique on simulated knotted polymers and hence obtain an implementation of KnotMiner, specifically aimed for mining knotted polymer databases.

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo. Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining*, pages 307–328, 1996.

[2] J. Baez, editor. *Knots and Quantum Gravity.* Oxford U. Press, 1994.

[3] G. Burde, H. Zieschang, and B. Amend. *Knots.* Walter de Gruyter, 2002.

[4] P.-G. de Gennes. *Scaling concepts in polymer physics.* Cornell U. Press, 1979.

[5] M. Doi and S.F. Edwards. *The theory of polymer dynamics*. Clarendon Press, 1986.

[6] P.G. Dommersnes, Y. Kantor, and M. Kardar. Knots in charged polymers. *Phys. Rev. E*, 66, 2002.

[7] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 2003. To appear.

[8] J. Hass, J.C. Lagarias, and N. Pippenger. The computational complexity of knot and link problems. *Journal of the ACM*, 46(2):185–211, 1999.

[9] G. Hemion. *The Classification of Knots and* 3*-Dimensional Spaces*. Oxford U. Press, 1992.

[10] A. Inokuchi, T. Washio, and H. Motoda. Complete mining of frequent patterns from graphs: Mining graph data. *Machine Learning*, 50(3):321–354, 2003.

[11] V. Katritch and et al. Tightness of random knotting. *Phys. Rev. E*, 61(5):5545–5549, 2000.

[12] L.H. Kauffman. *Knots and Physics*, volume 1 of *on Knots and Everything*. World Scientific, 1991.

[13] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 313–320. IEEE Computer Society, 2001.

[14] M. Kuramochi and G. Karypis. Frequent geometric subgraphs. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, pages 258–265. IEEE Computer Society, 2002.

[15] S.J. Lomonaco. *Quantum Computation: A Grand Mathematical Challenge for the Twenty-First Century and the Millennium*, volume 58 of *Proceedings of Symposia in Applied Mathematics*. AMS, 2002.

[16] B. Possas, W. Meira, M. Carvalho, and R. Resende. Using quantitative information for efficient association rule generation. *SIGMOD Record*, 29(4), 2000.

[17] H. Schubert. Die eindeutige zerlegbarkeit einse knoten in primknoten. *S.-B. Heidelberger Akad. Wiss. Math.-Nat.*, 3:57–104, 1949.

[18] N.C. Seeman. New motifs in dna nanotechnology. *Nanotechnology*, 9:257–273, 1998.

[19] M.K. Shimamura and T. Deguchi. Characteristic length of random knotting for cylindrical self-avoiding polygons. *Phys. Lett. A*, 274:184–191, 2000.

[20] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *Proceedings of the ACM-SIGMOD 1996 Conference on Management of Data*, pages 1–12. ACM Press, 1996.

[21] D. W. Sumners and S. G. Whittington. Knots in self-avoiding walks. *Journal of Physics A: Mathematical and General*, 21(7):1689–1694, 1988.

[22] D.W. Sumners. Lifting the curtain: Using topology to probe the hidden actions of enzymes. *Notices of the AMS*, 42(5):528–537, 1995.

[23] F. Waldhausen. Recent results on sufficiently large 3-manifolds. In *Proceedings of the Symposium on Pure Mathematics*, volume 32, pages 21–38. AMS, 1978.

[24] S.G. Whittington, D.W. Sumners, and T. Lodge, editors. *Topology and Geometry in Polymer Science*, volume 103 of *The IMA Volumes in Mathematics and its Applications*. Springer, 1998.

[25] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, pages 721–724. IEEE Computer Society, 2002.

[26] M.J. Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(2):372–390, 2000.

[27] M.J. Zaki. Efficiently mining frequent trees in a forest. In *Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 71–80. ACM Press, 2002.