Approximation algorithms for confidence bands for time series

Nikolaj Tatti[0000−0002−2087−5360]

University of Helsinki, Finland, nikolaj.tatti@helsinki.fi

Abstract. Confidence intervals are a standard technique for analyzing data. When applied to time series, confidence intervals are computed for each time point separately. Alternatively, we can compute confidence bands, where we are required to find the smallest area enveloping k time series, where k is a user parameter. Confidence bands can be then used to detect abnormal time series, not just individual observations within the time series. We will show that despite being an **NP**-hard problem it is possible to find optimal confidence band for some k. We do this by considering a different problem: discovering regularized bands, where we minimize the envelope area minus the number of included time series weighted by a parameter α. Unlike normal confidence bands we can solve the problem exactly by using a minimum cut. By varying α we can obtain solutions for various k. If we have a constraint k for which we cannot find appropriate α, we demonstrate a simple algorithm that yields $\mathcal{O}(\sqrt{n})$ approximation guarantee by connecting the problem to a minimum k-union problem. This connection also implies that we cannot approximate the problem better than $\mathcal{O}\left(n^{1/4}\right)$ under some (mild) assumptions. Finally, we consider a variant where instead of minimizing the area we minimize the maximum width. Here, we demonstrate a simple 2-approximation algorithm and show that we cannot achieve better approximation guarantee.

1 Introduction

Confidence intervals are a common tool to summarize the underlying distribution, and to indicate outlier behaviour. In this paper we will study the problem of computing confidence intervals for time series.

Korpela et al. [11] proposed a notion for computing confidence intervals: instead of computing point-wise confidence intervals, the authors propose computing confidence bands. More formally, given n time series T, we are asked to find k time series $U \subseteq T$ that minimize the envelope area, that is, the sum $\sum_i \left(\max_{t \in U} t(i)\right) - \left(\min_{t \in U} t(i)\right)$. The benefit, as argued by Korpela et al. [11], of using confidence bands instead of point-wise confidence intervals is better family-wise error control: if we were to use point-wise intervals we can only say that a time series *at some fixed point* is an outlier and require a correction for multiple testing (such as Bonferroni correction) if we want to state with a certain probability that the *whole* time series is normal.

In this paper we investigate the approximation algorithms for finding confidence bands. While Korpela et al. [11] proved that finding the optimal confidence band is an **NP**-hard problem, they did not provide any approximation algorithms nor any inapproximability results.

We will first show that despite being an **NP**-hard problem, we can solve the problem for *some* k. We do this by considering a different problem, where instead of having a hard constraint we have an objective function that prefers selecting time series as long as they do not increase the envelope area too much. The objective depends on the parameter α, larger values of α allow more increase in the envelope area. We will show that this problem can be solved exactly in polynomial time and that each α correspond to a certain value of k. We will show that there are at most $n+1$ of such bands, and that we can discover all of them in polynomial time by varying α.

Next, we provide a simple algorithm for approximating confidence bands by connecting the problem to the weighted k-MinUnion problem. We will provide a variant of an algorithm by Chlamtáč et al. [2] that yields $\sqrt{n}+1$ guarantee. We also argue that—under certain conjecture—we cannot approximate the problem better than $\mathcal{O}\big(n^{1/4}\big)$.

Finally, we consider a variant of the problem where instead of minimizing the envelope area, we minimize the width of the envelope, that is, we minimize the maximum difference between the envelope boundaries. We show that a simple algorithm can achieve 2-approximation. This approximation provides interesting contrast to the inapproximability results when minimizing the envelope area. Surprisingly this guarantee is tight: we will also show that the there is no polynomial-time algorithm with smaller guarantee unless $\mathbf{P} = \mathbf{NP}$.

The remainder of the paper is organized as follows. We define the optimization problems formally in Section 2. We solve the regularized band problem in Section 3, approximate minimization of envelope area in Section 4, and approximate minimization of envelope width in Section 5. Section 6 is devoted to the related work. We present our experiments in Section 7 and conclude with discussion in Section 8.

2 Preliminaries and problem definitions

Assume that we are given time series T with each time series $f : D \to \mathbb{R}$ mapping from domain D to a real number. We will often write $n = |T|$ to be the number of given time series, and $m = |D|$ to mean the size of the domain.

Given a set of time series T, we define the upper and lower *envelopes* as

$$ub(T, i) = \max_{t \in T} t(i) \quad \text{and} \quad \ell b(T, i) = \min_{t \in T} t(i) \quad .$$

Our main goal is to find k time series that minimize the envelope area.

*Problem 1 (*SumBand*).* Given a set of n time series $T = (t_1, \ldots, t_n)$, an integer $k \leq n$, and a time series $x \in T$ find k time series $U \subseteq T$ containing x minimizing

$$s_1(U) = \sum_i ub(U, i) - \ell b(U, i) \quad .$$

We will refer to U as *confidence bands*.

Note that the we also require that we must specify at least one sequence $x \in T$ that must be included in the input whereas the original definition of the problem given by Korpela et al. [11] did not require specifying x. As we will see later, this requirement simplifies the computational problem. On the other hand, if we do not have x at hand, then we can either test every $t \in T$ as x, or we can use the mean or the median of T. We will use the latter option as it does not increase the computational complexity and at the same time is a reasonable assumption. Note that in this case most likely $x \notin T$, so we define $T' = T \cup \{x\}$, increase $k' = k + 1$, and solve SumBand for T' and k' instead.

We can easily show that the area function $s_1(\cdot)$ is a submodular function for all non-empty subsets, that is,

$$s_1(U \cup \{t\}) - s_1(U) \leq s_1(W \cup \{t\}) - s_1(W)\,,$$

where $U \supseteq W \neq \emptyset$. In other words, adding t to a larger set U increases the cost less than adding t to W.

We also consider a variant of SumBand where instead of minimizing the area of the envelope, we will minimize the maximum width.

*Problem 2 (*InfBand*).* Given a set of n time series $T = (t_1, \ldots, t_n)$, an integer $k \leq n$, and a time series $x \in T$, find k time series $U \subseteq T$ containing x minimizing

$$s_\infty(U) = \max_i ub(U, i) - \ell b(U, i) \quad .$$

We will show that we can 2-approximate InfBand and that the ratio is tight.

Finally, we consider a regularized version of SumBand, where instead of requiring that the set has a minimum size k, we add a term $-\alpha|U|$ into the objective function. In other words, we will favor larger sets as long as the area $s_1(U)$ does not increase too much.

*Problem 3 (*RegBand*).* Given a set of n time series $T = (t_1, \ldots, t_n)$, a number $\alpha > 0$, and a time series $x \in T$, find a subset $U \subseteq T$ containing x minimizing

$$s_{reg}(U; \alpha) = s_1(U) - \alpha|U| \quad .$$

In case of ties, use $|U|$ as a tie-breaker, preferring larger values.

We refer to the solutions of RegBand as regularized bands. It turns out that RegBand can be solved in polynomial time. Moreover, the solutions we obtain from RegBand will be useful for approximating SumBand.

3 Regularized bands

In this section we will list useful properties of of RegBand, show how can we solve RegBand in polynomial time for a single α, and finally demonstrate how we can discover *all* regularized bands by varying α.

3.1 Properties of regularized bands

Our first observation is that the output of RegBand also solves SumBand for certain size constraints.

Proposition 1. *Assume time series T and $\alpha > 0$. Let U be a solution to* RegBand(α). *Then U is also a solution for* SumBand *with $k = |U|$.*

The proof of this proposition is trivial and is omitted.

Our next observation is that the solutions to RegBand form a chain.

Proposition 2. *Assume time series T and $0 < \alpha < \beta$. Let V be a solution to* RegBand(T, α) *and let U be a solution to* RegBand(T, β). *Then $V \subseteq U$.*

Proof. Assume otherwise. Let $W = V \setminus U$. Due to the optimality of V,

$$0 \geq s_{reg}(V; \alpha) - s_{reg}(V \cap U; \alpha) = s_1(V) - s_1(V \cap U) - \alpha |W| \quad .$$

Since s_1 is a submodular function, we have

$$s_1(V) - s_1(V \cap U) = s_1(W \cup (V \cap U)) - s_1(V \cap U) \geq s_1(W \cup U) - s_1(U) \quad .$$

Combining these inequalities leads to

$$\begin{aligned} 0 &\geq s_1(W \cup U) - s_1(U) - \alpha |W| \\ &\geq s_1(W \cup U) - s_1(U) - \beta |W| \\ &= s_{reg}(W \cup U; \beta) - s_{reg}(U; \beta) \, , \end{aligned}$$

which contradicts the optimality of U. □

This property is particularly useful as it allows clean visualization: the envelopes resulting from different values of α will not intersect. Moreover, it allows us to stored all regularized bands by simply storing, per each time series, the index of the largest confidence band containing the time series.

Interestingly, this result does not hold for SumBand.

Example 1. Consider 4 constant time series $t_1 = 0$, $t_2 = -1$ and $t_3 = t_4 = 2$. Set the seed time series $x = t_1$. Then the solution for SumBand with $k = 2$ is $\{t_1, t_2\}$ and the solution SumBand with $k = 3$ is $\{t_1, t_3, t_4\}$.

3.2 Computing regularized band for a single α

Our next step is to solve RegBand in polynomial time. Note that since $s_1(\cdot)$ is submodular, then so is $s_{reg}(\cdot)$. Minimizing submodular function is solvable in polynomial-time [15]. Solving RegBand using a generic solver for minimizing submodular functions is slow, so instead we will solve the problem by reducing it to a minimum cut problem. In such a problem, we are given a weighted directed graph $G = (V, E, W)$, two nodes, say $\theta, \eta \in V$, and ask to partition V into $X \cup Y$ such that $\theta \in X$ and $\eta \in Y$ minimizing the total weight of edges from X to Y.

In order to define G we need several definitions. Assume we are given n time series T, a real number α and a seed time series $x \in T$. Let m be the size of the domain. For $i \in [m]$, we define $p_i = \{t_j(i) \mid j \in [n]\}$ to be the set (with no duplicates) sorted, smallest values first. In other words, p_{ij} is the jth smallest distinct observed value in T at i. Let P be the collection of all p_i.

We also define c_{ij} to be the number of time series at i smaller than or equal to p_{ij}, that is, $c_{ij} = |\{\ell \in [n] \mid t_\ell(i) \leq p_{ij}\}|$. We also write $c_{i0} = 0$.

We are now ready to define our graph. We define a weighted directed graph $G = (V, E, W)$ as follows. The nodes V have three sets A, B, and C. The set A has $|P|$ nodes, a node $a_{ij} \in A$ corresponding to each entry $p_{ij} \in P$. The set $B = \{b_j\}$ has n nodes, and the set C has two nodes, θ and η. Here, θ acts as a source node and η acts as a terminal node.

The edges and the weights are as follows: For each $a_{ij} \in A$ such that $p_{ij} > x(i)$, we add an edge $(a_{i(j-1)}, a_{ij})$ with the weight

$$w(a_{i(j-1)}, a_{ij}) = n - c_{i(j-1)} + \frac{m}{\alpha}(p_{i(j-1)} - x(i)) \quad .$$

For each $a_{ij} \in A$ such that $p_{ij} < x(i)$, we add an edge $(a_{i(j+1)}, a_{ij})$ with the weight

$$w(a_{i(j+1)}, a_{ij}) = c_{ij} + \frac{m}{\alpha}(x(i) - p_{i(j+1)}) \quad .$$

For each $a_{ij} \in A$ such that $p_{ij} = x(i)$, we add an edge (θ, a_{ij}) with the weight ∞. For each $i \in [m]$ and $\ell = |p_i|$, we add two edges $(a_{i\ell}, \eta)$ and (a_{i1}, η) with the weights

$$w(a_{i\ell}, \eta) = \frac{m}{\alpha}(p_{i\ell} - x(i)) \quad \text{and} \quad w(a_{i1}, \eta) = \frac{m}{\alpha}(x(i) - p_{i1}) \quad .$$

In addition, for each $i \in [m]$, $\ell \in [n]$, let j be such that $p_{ij} = t_\ell(i)$ and define two edges (a_{ij}, b_ℓ) and (b_ℓ, a_{ij}) with the weights,

$$w(a_{ij}, b_\ell) = 1 \qquad\qquad w(b_\ell, a_{ij}) = \infty \quad .$$

Our next proposition states the minimum cut of G also minimizes RegBand.

Proposition 3. *Let X, Y be a (θ, η)-cut of G with the optimal cost. Define $f(i) = \min_j \{p_{ij} \mid a_{ij} \in X\}$ and $g(i) = \max_j \{p_{ij} \mid a_{ij} \in X\}$.*

Then the cost of the cut is equal to

$$nm - m|\{j \mid b_j \in X\}| + \frac{m}{\alpha} \sum_i g(i) - f(i) \quad .$$

Moreover, if $b_\ell \in X$, then $g(i) \leq b_\ell(i) \leq f(i)$, for all i.

Proof. The last claim follows immediately as otherwise there is a cross-edge with infinite cost making the cut suboptimal.

Define $u(i) = \arg\min_j \{p_{ij} \mid a_{ij} \in X\}$ and $v(i) = \arg\max_j \{p_{ij} \mid a_{ij} \in X\}$ to be the indices yielding f and g. Define also

$$d_i = |\{j \mid u(i) \leq t_j(i) \leq v(i)\}| = c_{iv(i)} - c_{i(u(i)-1)}$$

to be the number of time series between $u(i)$ and $v(i)$ at i.

Note that $a_{ij} \in X$ whenever $u(i) \leq j \leq v(j)$ as otherwise we can move a_{ij} to X and decrease the cost.

The cut consists of the cross-edges originating from $a_{iv(i)}$ and $a_{iu(i)}$, and cross-edges between A and B. The cost of the former is equal to

$$\begin{aligned}\sum_i n - c_{iv(i)} + m\frac{p_{iv(i)} - x(i)}{\alpha} + c_{i(u(i)-1)} + m\frac{x(i) - p_{iu(i)}}{\alpha} \\ = nm + \sum \frac{m}{\alpha}(g(i) - f(i)) - \sum_i d_i\end{aligned}$$

while the cost of the latter is

$$\begin{aligned}\sum_i |\{j \mid a_{ij} \in X, b_j \notin X\}| &= \sum_i |\{j \mid u(i) \leq t_j(i) \leq v(i) \in X, b_j \notin X\}| \\ &= \sum_i d_i - |\{j \mid u(i) \leq t_j(i) \leq v(i) \in X, b_j \in X\}| \\ &= \sum_i d_i - m|\{j \mid b_j \in X\}| \quad .\end{aligned}$$

Combining the two equations proves the claim. □

Corollary 1. *Let U' be the solution to* RegBand(α). *Let (X, Y) be a minimum (θ, η)-cut of G. Set $U = \{t_\ell \mid b_\ell \in X\}$. Then $s_{reg}(U; \alpha) = s_{reg}(U'; \alpha)$.*

Proof. Proposition 3 states that the cost of the minimum cut is $nm + \frac{m}{\alpha} s_{reg}(U; \alpha)$.

Construct a cut (X', Y') from U' by setting X' to be the nodes from A and B that correspond to the time series U'. The proof of Proposition 3 now states that the cut is equal to $nm + \frac{m}{\alpha} s_{reg}(U'; \alpha)$.

The optimality of (X, Y) proves the claim. □

We may encounter a pathological case, where we have multiple cuts with the same optimal cost. RegBand requires that in such case we use largest solution. This can be enforced by modifying the weights: first scale the weights so that they are all multiples of $nm + 1$, then add 1 to the weight of each (θ, α_{ij}). The cut with the modified graph yields the largest band with the optimal cost.

The constructed graph G has $\mathcal{O}(nm)$ nodes and $\mathcal{O}(nm)$ edges. Consequently, we can compute the minimum cut in $\mathcal{O}\big((nm)^2\big)$ time [13]. In practice, solving minimum cut is much faster.

3.3 Computing all regularized bands

Now that we have a method for solving RegBand(α) for a fixed α, we would like to find solutions for all α. Note that Proposition 2 states that we can have at most $n + 1$ different bands.

We can enumerate the bands with the divide-and-conquer approach given in Algorithm 1. Here, we are given two, already discovered, regularized bands

Algorithm 1: EnumReg(U, V) finds all regularized bands between U and V

$\gamma \leftarrow \frac{s_1(V)-s_1(U)}{|V|-|U|} - \frac{\Delta}{n^2}$;
$W \leftarrow$ solution to RegBand(γ);
if $U \neq W$ **then**
  report W;
  EnumReg(U, W); EnumReg(W, V);

$U \subsetneq V$, and we try to find a middle band W with $U \subsetneq W \subsetneq V$. If W exists, we recurse on both sides. To enumerate all bands, we start with EnumReg$(\{x\}, V)$.

The following proposition proves the correctness of the algorithm: during each split we will always find a new band if such exist.

Proposition 4. *Assume time series T with n time series. Let $\{U_i\}$ be all the possible regularized confidence bands ordered using inclusion. Define*

$$\Delta = \min \{|t(i) - u(i)| \mid t, u \in T, i, t(i) \neq u(i)\} \quad .$$

Let $i < j$ be two integers and define

$$\gamma = \frac{s_1(U_j) - s_1(U_i)}{|U_j| - |U_i|} - \frac{\Delta}{n^2} \quad .$$

Let U_ℓ be the solution for RegBand(γ). *Then $i \leq \ell < j$. If $j > i+1$, then $i < \ell$, otherwise $\ell = i$.*

For simplicity, let us define $f(x, y) = \frac{s_1(U_y)-s_1(U_x)}{|U_y|-|U_x|}$.

In order to prove the result we need the following technical lemma.

Lemma 1. *Assume time series T with n time series. Let $\{U_i\}$ be all the possible regularized confidence bands ordered using inclusion. Let $\alpha > 0$. Let U_i be the solution for* RegBand(α). *Then $f(i-1, i) \leq \alpha < f(i, i+1)$.*

Proof. Due to the optimality of U_i,

$$s_1(U_i) - \alpha|U_i| = s_{reg}(U_i; \alpha) < s_1(U_{i+1}) - \alpha|U_{i+1}| \quad .$$

Solving for α gives us the right-hand side of the claim. Similarly,

$$s_1(U_i) - \alpha|U_i| = s_{reg}(U_i; \alpha) \leq s_1(U_{i-1}) - \alpha|U_{i-1}| \quad .$$

Solving for α gives us the left-hand side of the claim. □

Proof (of Proposition 4). It is straightforward to see that Lemma 1 implies that $f(a, b) \leq f(x, y)$ for $a \leq x$ and $b \leq y$. Moreover, the equality holds only if $x = a$ and $y = b$, in other cases $f(a, b) + \frac{\Delta}{n^2} \leq f(x, y)$.

If $\ell \geq j$, then Lemma 1 states that $f(i,j) \leq f(\ell-1,\ell) \leq \gamma$, which contradicts the definition of γ. Thus $\ell < j$.

Since $f(i,j) - f(i-1,i) \geq \frac{\Delta}{n^2}$, we have $f(i-1,i) \leq \gamma$. If $\ell < i$, then Lemma 1 states that $\gamma < f(i-1,i)$, which is a contradiction. Thus, $\ell \geq i$.

If $j = i+1$, then immediately $\ell = i$.

Assume that $j > i+1$. Since $f(i,j) - f(i,i+1) \geq \frac{\Delta}{n^2}$, we have $f(i,i+1) \leq \gamma$. If $\ell = i$, then according to Lemma 1 $\gamma < f(i,i+1)$, which is a contradiction. Thus, $\ell > i$. □

Lemma 1 reveals an illuminating property of regularized bands, namely each band minimizes the ratio of additional envelope area and the number of new time series.

Proposition 5. *Let U be a regularized band. Define $g(X) = \frac{s_1(X) - s_1(U)}{|X| - |U|}$. Let $V \supsetneq U$ be the adjacent regularized band. Then $g(V) = \min_{X \supsetneq U} g(X)$.*

Proof. Let $O = \arg\min_{X \supsetneq U} g(X)$, and set $\beta = g(O)$. We will prove that $g(V) \leq \beta$. Let $W = \textsc{RegBand}(\beta)$. Let α be the parameter for which $U = \textsc{RegBand}(\alpha)$. Assume that $\alpha \geq \beta$. We can rewrite the equality $\beta = g(O)$ as

$$0 = s_{reg}(O;\beta) - s_{reg}(U;\beta) \geq s_{reg}(O;\alpha) - s_{reg}(U;\alpha)\,,$$

which violates the optimality of U. Thus $\alpha < \beta$. Proposition 2 states that $U \subseteq W$. Moreover, due to submodularity,

$$s_{reg}(O \cup W;\beta) - s_{reg}(W;\beta) \leq s_{reg}(O \cup U;\beta) - s_{reg}(U;\beta) = 0,$$

which due to the optimality of W implies that $O \subseteq W$. Thus $W \neq U$ and $V \subseteq W$. Lemma 1, possibly applied multiple times, shows that $g(V) \leq g(W) \leq \beta$. □

Proposition 2 states that there are at most $n+1$ bands. Queries done by EnumReg yield the same band at most twice. Thus, EnumReg performs at most $\mathcal{O}(n)$ queries, yielding computational complexity of $\mathcal{O}\left(n^3 m^2\right)$. In practice, EnumReg is faster: the number of bands is significantly smaller than n and the minimum cut solver scales significantly better than $\mathcal{O}\left(n^2 m^2\right)$. Moreover, we can further improve the performance with the following observation: Proposition 2 states that when processing $\textsc{EnumReg}(U,V)$, the bands will be between U and V. Hence, we can ignore the time series that are outside V, and we can safely replace U with its envelope $\ell b(U)$ and $ub(U)$.[1]

4 Discovering confidence bands minimizing s_1

In this section, we will study SumBand. Korpela et al. [11] showed that the problem is **NP**-hard. We will argue that we can approximate the problem and establish a (likely) lower bound for the approximation guarantee.

[1] We need to make sure that the envelope is always selected. This can be done by connecting θ to the envelope with edges of infinite weight.

Algorithm 2: FindSum(T, k, x), approximates SumBand

```
{B_i} ← EnumReg({x}, T);
j ← largest index for which |B_j| ≤ k;
if |B_j| ≤ k − √n then W ← B_{j+1} \ B_j else W ← T \ B_j;
U ← B_j;
greedily add k − |U| entries from W to U, minimizing s_1 at each step;
return U;
```

As a starting point, note that SumBand is an instance of k-MinUnion, weighted minimum k-union problem. In k-MinUnion we are given n sets over a universe with weighted points, and ask to select k sets minimizing the weighted union. In our case, the universe is the set P described in Section 3, the weights are the distances between adjacent points, and a set consists of all the points between a time series and x.

The *unweighted* k-MinUnion problem has several approximation algorithms: a simple algorithm achieving $\mathcal{O}(\sqrt{n})$ guarantee by Chlamtáč et al. [2] and a algorithm achieving lower approximation guarantee of $\mathcal{O}(n^{1/4})$ by Chlamtáč et al. [3]. We will use the former algorithm due to its simplicity and the fact that it can be easily adopted to handle weights.

The pseudo-code for the algorithm is given in Algorithm 2. The algorithm first looks for the largest possible regularized band, say B_j, whose size at most k. The remaining time series are then selected greedily from a set of candidates W. The set W depends on how many additional time series is needed: if we need at most $\sqrt{n}$ additional time series, we set W to be the remaining time series $T \setminus B_j$, otherwise we select the time series from the next regularized band, that is, we set $W = B_{j+1} \setminus B_j$.

Proposition 6. FindSum *yields* $\sqrt{n} + 1$ *approximation guarantee.*

Proof. Let O be the optimal solution for SumBand(k), and let $r = s_1(O)$. Let U be the output of FindSum. Assume that $B_j \neq O$, as otherwise we are done. We split the proof in two cases.

First, assume that $|B_j| \leq k - \sqrt{n}$. Since s_1 is submodular we have $s_1(O \cup B_j) - s_1(B_j) \leq s_1(O) - s_1(\{x\}) = r$, leading to

$$\frac{s_1(B_{j+1}) - s_1(B_j)}{|B_{j+1}| - |B_j|} \leq \frac{s_1(O \cup B_j) - s_1(B_j)}{|O \cup B_j| - |B_j|} \leq \frac{r}{k - |B_j|} \leq \frac{r}{\sqrt{n}},$$

where the first inequality is due to Proposition 5. Rearranging the terms gives us

$$s_1(B_{j+1}) - s_1(B_j) \leq \frac{r(|B_{j+1}| - |B_j|)}{\sqrt{n}} \leq r\frac{n}{\sqrt{n}} = r\sqrt{n} \quad .$$

Finally,

$$\begin{aligned} s_1(U) &= s_1(B_j) + (s_1(U) - s_1(B_j)) \\ &\leq s_1(B_j) + (s_1(B_{j+1}) - s_1(B_j)) \leq s_1(B_j) + r\sqrt{n} \leq r(1 + \sqrt{n}), \end{aligned}$$

where the last inequality is implied by Proposition 1 and the fact that $|B_j| \leq k$.

Assume that $|B_j| > k - \sqrt{n}$, and let $q = k - |B_j|$. Note that $q < \sqrt{n}$. Let $c_1, \ldots, c_q$ be the additional time series added to U. Write $U_i = B_j \cup \{c_1, \ldots, c_i\}$.

Let c'_i be the closest time series to x outside U_{i-1}. Note that $s_1(\{x\} \cup c'_i) - s_1(\{x\}) = s_1(\{x\} \cup c'_i) \leq r$ for $i = 1, \ldots, q$ as otherwise r has to be larger. In addition, Proposition 1 and $|B_j| \leq k$ imply that $s_1(B_j) \leq r$. Consequently,

$$\begin{aligned} s_1(U_q) &= s_1(B_j) + \sum_{i=1}^{q} s_1(U_i) - s_1(U_{i-1}) \\ &\leq s_1(B_j) + \sum_{i=1}^{q} s_1(\{x\} \cup c'_i) - s_1(\{x\}) \leq (1 + \sqrt{n})r, \end{aligned}$$

where the first inequality is due to the submodularity of s_1. □

FindSum resembles greatly the algorithm given by Chlamtáč et al. [2] but has few technical differences: we select B_j as our starting point whereas the algorithm by Chlamtáč et al. [2] constructs the starting set by iteratively finding and adding sets with the smallest average area, $s_1(X)/|X|$, that is, solving the problem given in Proposition 5.[2] Such sets can be found with a linear program. Proposition 5 implies that both approaches result in the same set B_j but our approach is faster.[3] Moreover, this modification allows us to prove a tighter approximation guarantee: the authors prove that their algorithm yields $2\sqrt{n}$ guarantee whereas we show that we can achieve $\sqrt{n}+1$ guarantee. Additionally, we select additional time series iteratively by selecting those time series that result in the smallest increase of the current area, whereas the original algorithm would simply select time series that are closest to $\{x\}$.

Chlamtáč et al. [3] argued that under some mild but technical conjecture there is no polynomial-time algorithm that can approximate k-MinUnion better than $\mathcal{O}\big(n^{1/4}\big)$. Next we will show that we can reduce k-MinUnion to SumBand while preserving approximation.

Proposition 7. *If there is an $f(n)$-approximation polynomial-time algorithm for* SumBand*, then there is an $f(n+1)$-approximation polynomial-time algorithm for k-*MinUnion*.*

Proof. Assume that we are given an instance of k-MinUnion with n sets $\mathcal{S} = (S_1, \ldots, S_n)$. Let $D = \bigcup_i S_i$ be the union of all S_i.

Define T containing $n+1$ time series over the domain D. The first n time series correspond to the sets S_i, that is, given $i \in D$, we set $t_j(i) = 1$ if $i \in S_j$, and 0 otherwise. The remaining single time series, named x, is set to be 0.

[2] The original algorithm is described using set/graph terminology but we use our terminology to describe the differences.

[3] The computational complexity of the state-of-the-art linear program solver is $\mathcal{O}\big((nm)^{2.37} \log(nm/\delta)\big)$, where δ is the relative accuracy [4]. We may need to solve $\mathcal{O}(n)$ such problems, leading to a total time of $\mathcal{O}\big(n(nm)^{2.37} \log(nm/\delta)\big)$.

Assume that we have an algorithm estimating SumBand$(T, x, k+1)$, and let U be the output of this algorithm. Note that since $x \in U$, we have $\ell b(U, i) = 0$.

Let $\mathcal{V}$ be the subset of $\mathcal{S}$ corresponding to the non-zero time series in U. Let $C = \bigcup_{S \in \mathcal{V}} S$ be the union of sets in $\mathcal{V}$. Since $\ell b(U, i) = 0$, and $ub(U, i) = 1$ if and only if $i \in C$, we have $s_1(U) = |C|$. □

The above result implies that unless the conjecture suggested by Chlamtáč et al. [3] is false, we cannot approximate SumBand better than $\mathcal{O}(n^{1/4})$. This proposition holds even if we replace $s_1(\cdot)$ with an ℓ_p^p norm, $\sum_i |t(i) - u(i)|^p$, where $1 \leq p < \infty$, or any norm that reduces to hamming distance if t is a binary sequence and u is 0. Interestingly, we will show in the next section that we can achieve a tighter approximation if we use s_∞.

5 Discovering confidence bands minimizing s_∞

In this section we consider the problem InfBand. Namely, we will show that a straightforward algorithm 2-approximates the problem, and more interestingly we show that the guarantee is tight.

The algorithm for InfBand(T, x, k) is simple: we select k time series that are closest to x according to the norm $\|t(i) - x(i)\|_\infty = \max_i |t(i) - x(i)|$. We will refer to this algorithm as FindInf.

It turns out that this simple algorithm yields 2-approximation guarantee.

Proposition 8. FindInf *yields 2-approximation for* InfBand.

Proof. Let U be the optimal solution for InfBand. Let V be the result produced by FindInf. Define $c = \max_{t \in V} \|t - x\|_\infty$. Then

$$c = \max_{t \in V} \|t - x\|_\infty \leq \max_{t \in U} \|u - x\|_\infty \leq s_\infty(U)\,,$$

where the first inequality holds since V contains the closest time series and the second inequality holds since $x \in U$.

Let i be the index such that $s_\infty(V) = ub(V, i) - \ell b(V, i)$. Then

$$s_\infty(V) = ub(V, i) - \ell b(V, i) = (ub(V, i) - t(i)) + (t(i) - \ell b(V, i)) \leq 2c \quad .$$

Thus, $s_\infty(V) \leq 2c \leq 2s_\infty(U)$, proving the claim. □

While FindInf is trivial, surprisingly it achieves the best possible approximation guarantee for a polynomial-time algorithm.

Proposition 9. *There is no polynomial-time algorithm for* InfBand *that yields* $\alpha < 2$ *approximation guarantee unless* $\boldsymbol{P = NP}$.

Proof. To prove the claim we will show that we can solve k-Clique in polynomial time if we can α-approximate InfBand with $\alpha < 2$. Since k-Clique is an **NP**-complete problem, this is a contradiction unless $\mathbf{P} = \mathbf{NP}$.

The goal of k-Clique is given a graph $G = (V, E)$ with n nodes and m edges to detect whether there is a k-clique, a fully connected subgraph with k nodes, in G. We can safely assume that G has no nodes that are fully-connected.

Fix an order for nodes $V = (v_1, \ldots, v_n)$ and let F be all the edges that are not in E, that is, $F = \{(v_x, v_y) \mid (v_x, v_y) \notin E, x < y\}$.

Next, we will define an instance of InfBand. The set of time series $T = (t_1, \ldots, t_n) \cup \{x\}$ consists of n time series t_i corresponding to the node v_i, and a single time series x which we will use a seed. We set the domain to be F. Each time series t_i maps an element of $e = (v_x, v_y) \in F$ to an integer,

$$t_i(e) = 1, \text{ if } i = x, \quad t_i(e) = -1, \text{ if } i = y, \quad t_i(e) = 0, \text{ otherwise} \quad .$$

We also set $x = 0$. First note that since $t_i(e)$ is an integer between -1 and 1, the score $s_\infty(U)$ is either 0, 1, or 2 for any $U \subseteq T$.

Since we do not have any fully-connected nodes in G, there is no non-zero t_i in T. Since $x \in U$ for any solution of InfBand, then $s_\infty(U) = 0$ implies $|U| = 1$.

Let $W \subseteq V$ be a subset of nodes, and let U be the corresponding time series. We claim that $s_\infty(U) = 1$ if and only if W is a clique. To prove the claim, first observe that if $v_i, v_j \in W$ such that $e = (v_i, v_j) \in F$, then $t_i(e) = 1$ and $t_j(e) = -1$, thus $s_\infty(U) = 2$. On the other hand, if W is a clique, then for every $t_i, t_j \in U$ and $e \in F$ such that $t_i(e) \neq 0$, we have $t_j(e) = 0$ since otherwise $(v_i, v_j) \notin E$. Thus, $s_\infty(U) = 1$ if and only if W is a clique.

Let O be the solution for InfBand$(T, k + 1, x)$. Note that $s_\infty(O) = 1$ if and only if G has a k-clique, and $s_\infty(O) = 2$ otherwise.

Let S be the output of α-approximation algorithm. Since $k > 1$, we know that $s_\infty(O)$ is either 1 or 2. If $s_\infty(O) = 2$, then $s_\infty(S) = 2$. If $s_\infty(O) = 1$, then $s_\infty(S) \leq \alpha s_\infty(O) < 2 \times 1$. Thus, $s_\infty(S) = 1$. In summary, $s_\infty(O) = s_\infty(S)$.

We have shown that $s_\infty(S) = 1$ if and only if G has a k-clique. This allows us to detect k-clique in G in polynomial time proving our claim. □

6 Related work

Confidence bands are envelopes for which confidence intervals of individual points hold simultaneously. Davison and Hinkley [5], Mandel and Betensky [12] proposed a non-parametric approach for finding simultaneous confidence intervals. Here, time series are ordered based on its *maximum* value, and α-confidence intervals are obtained by removing $\alpha/2$ portions from each tail. Note that unlike SumBand and InfBand this definition is not symmetric: if we flip the sign of time series we may get a different interval.

There is a strong parallel between finding regularized bands and finding dense subgraphs. Proposition 5 states that the inner-most regularized band has the smallest average envelope area, or alternatively it has the highest ratio of time series per envelope area. A related graph-theoretical concept is a dense subgraph, a subgraph H of a given subgraph G with the largest ratio $|E(H)|/|V(H)|$. The method proposed by Goldberg [7] for finding dense subgraphs in polynomial time is based on maximizing $|E(H)| - \alpha|V(H)|$ and selecting α to be as small

as possible without having an empty solution. Moreover, Tatti [16] extended the notion of dense subgraphs to density-friendly core decomposition, which essentially consists of the subgraphs minimizing $|E(H)| - \alpha|V(H)|$ for various values of α, the algorithm for finding the decomposition is similar to the algorithm for enumerating all regularized bands. In addition, Tsourakakis [17] extended the notion of dense subgraphs to triangle-density and hypergraphs, and also used minimum cut to find the solutions. As pointed out in Section 4 is that we can view time series as sets of points in P. In fact, the minimum cut used in Section 3 share some similarities with the minimum cut proposed by Tsourakakis [17]. Finally, the algorithm proposed by Korpela et al. [11] to find confidence bands resembles the algorithm by Charikar [1] for approximating the densest subgraph: in the former we delete the time series that reduce the envelope area the most while in the latter we delete vertices that have the smallest degree.

We assume that we are given a seed time series x. If such series is not given then we need to test every $t \in T$ as a seed. If we consider a special case of $k = 2$, then the problem of finding regularized band reduces to the closest pair problem: find two time series with the smallest distance: a well-studied problem in computational geometry. A classic approach by Dietzfelbinger et al. [6], Khuller and Matias [10], Rabin [14] allows to solve the closest pair problem in $\mathcal{O}(n)$ time but the analysis treats the size of the domain, m, as a constant; otherwise, the computational complexity has an exponential factor in m and can be only used for very small values of m. For large values of m, Indyk et al. [9] proposed an algorithm for solving the closest pair problem minimizing $s_1(\cdot)$ in $\mathcal{O}\big(n^{2.687}\big)$ time and minimizing $s_\infty(\cdot)$ in $\mathcal{O}\big(n^{2.687}\log\Delta\big)$ time, where Δ is the width of the envelope of the whole data.

7 Experimental evaluation

In this section we describe our experimental evaluation.

We implemented EnumReg and FindSum using C++ and used a laptop with Intel Core i5 (2.3GHz) to conduct our experiments.[4] As a baseline we used the algorithm by Korpela et al. [11], which we will call Peel. We implemented Peel also with C++, and modified it to make sure that the seed time series x is always included. Finally, we implemented FindInf with Python. In all algorithms we used the median as the seed time series.

Datasets: We used 4 real-world datasets as benchmark datasets. The first dataset, *Milan*, consists of monthly averages of maximum daily temperatures in Milan between the years 1763–2007.[5] The second dataset, *Power*, consists of hourly power consumption (variable `global_active_power`) of a single household over almost 4 years, a single time series representing a day.[6] Our last 2 datasets *ECG-normal* and *ECG-pvc* are heart beat data [8]. We used MLII

[4] The code is available at `https://version.helsinki.fi/DACS`

[5] `https://www.ncdc.noaa.gov/`

[6] `http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption`

Table 1: Basic characteristics of the datasets and performance measures of the algorithms. Here, n stands for the number of time series, m stands for the domain size, $|B_1|$ is the size of the smallest non-trivial regularized band, $|\mathcal{B}|$ is the number of regularized bands, and time is the required time to execute EnumReg in seconds. The scores s_1 for the algorithms FindSum, FindInf, and Peel are normalized with the envelope area of the whole data and multiplied by 100.

						s_1 for $k = \lfloor 0.9n \rfloor$			s_1 for $k = \lfloor 0.95n \rfloor$		
Dataset	n	m	$\lvert B_1 \rvert$	$\lvert \mathcal{B} \rvert$	Time	Sum	Peel	Inf	Sum	Peel	Inf
Milan	245	12	209	17	0.03	70.34	72.49	74.1	75.31	76.99	78.45
Power	1 417	24	1 102	56	3.68	70.94	72.83	77.06	78.89	81.17	82.31
ECG-normal	1 507	253	1 289	72	39.44	51.72	52	72.97	57.22	57.51	73.15
ECG-pvc	520	253	484	19	6.67	80.28	80.02	91.97	83.84	83.92	95.69

Table 2: Scores s_∞ of discovered confidence bands. The scores are normalized with the envelope width of the whole data and multiplied by 100.

	s_∞ for $k = \lfloor 0.9n \rfloor$			s_∞ for $k = \lfloor 0.95n \rfloor$		
Dataset	Sum	Peel	Inf	Sum	Peel	Inf
Milan	72.54	79.78	67.35	78.04	79.78	73.95
Power	79.08	82.16	73.13	82.16	98.71	79.08
ECG-normal	64.78	64.78	54.81	65.64	64.78	57.39
ECG-pvc	93.24	93.24	66.41	93.24	93.24	81.9

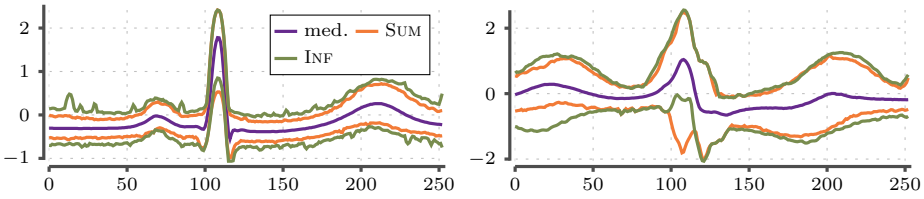

Fig. 1: Envelopes for *ECG-normal* (left) and *ECG-pvc* (right) and $k = \lfloor 0.9n \rfloor$.

data of a single patient (id 106) from the MIT-BIH arrhythmia database,[7] and split the measurements into normal beats (*ECG-normal*) and abnormal beats with premature ventricular contraction (*ECG-pvc*). Each time series represent measurements between −300ms and 400ms around each beat. The sizes of the datasets are given in Table 1.

Results: First let us consider EnumReg. From the results given Table 1 we see that the number of distinct regularized bands $|\mathcal{B}|$ is low: about 4%–7% of n, the number of time series. Having so few bands in practice reduces the computational cost of EnumReg since the algorithm tests at most $2|\mathcal{B}|$ values

[7] `https://physionet.org/content/mitdb/1.0.0/`

of α Interestingly, the smallest non-trivial band B_1 is typically large, containing about 70%–90% of the time series. Note that Proposition 5 states that B_1 has the smallest ratio of $s_1(B_1)/|B_1|$. For our benchmark datasets, B_1 is large suggesting that most time series are equally far away from the median while the remaining the time series exhibit outlier behaviour.

The algorithms are fast for our datasets: Table 1 show that EnumReg requires at most 40 seconds. Additional steps required by FindSum are negligible, completing in less than a second. The baseline algorithm is also fast, requiring less than a second to complete.

Let us now compare FindSum against Peel. We compared the obtained areas by both algorithms with $k = \lfloor 0.9n \rfloor$ and $k = \lfloor 0.95n \rfloor$. We see from the results in Table 1, that FindSum performs slightly better than Peel. The improvement in score is modest, 1%–2%. We conjecture that in practice Peel is close to the optimal, so any improvements are subtle. Interestingly, enough Peel performs better than FindSum for *ECG-pvc* and $\gamma = 0.1$. The reason for this is that the inner band B_1 contains more than 90% of the time series. In such a case FindSum will reduce to a simple greedy method, starting from $\{x\}$. Additional testing revealed that Peel outperforms FindSum when $k \leq |B_1|$ about 50%–90%, depending on the dataset, suggesting that whenever $k \leq |B_1|$ it is probably better to run both algorithms and select the better envelope.

Next let us compare FindInf against the other methods. The results in Tables 1–2 show that FindInf yields inferior s_1 scores but superior s_∞ scores. This is expected as FindInf optimizes s_∞ while FindSum and Peel optimize s_1. The differences are further highlighted in the envelopes for *ECG* datasets shown in Figure 1: FindInf yields larger envelopes but provides a tighter bound under the peak (R wave).

8 Concluding remarks

In this paper we consider the approximation algorithms for discovering confidence bands. Namely, we proposed a practical algorithm that approximates SumBand with a guarantee of $\mathcal{O}(n^{1/2})$. We also argued that the lower bound for the guarantee is most likely $\mathcal{O}(n^{1/4})$. In addition, we showed that we can 2-approximate InfBand, a variant of SumBand problem, with a simple algorithm and that the guarantee is tight.

Our experiments showed that FindSum outperforms the original baseline method for large values of k, that is, as long as k is larger than the smallest regularized band. Our experiments suggest that this condition usually holds, if we are interested, say in, 90%–95% confidence.

Interesting future line of work is to study the case for time series with multiple modes, that is, a case where instead of a single seed time series, we are given a set of time series, and we are asked to find confidence bands around each seed.

References

[1] Charikar, M.: Greedy approximation algorithms for finding dense components in a graph. APPROX (2000)

[2] Chlamtáč, E., Dinitz, M., Konrad, C., Kortsarz, G., Rabanca, G.: The densest k-subhypergraph problem. SIAM Journal on Discrete Mathematics 32(2), 1458–1477 (2018)

[3] Chlamtáč, E., Dinitz, M., Makarychev, Y.: Minimizing the union: Tight approximations for small set bipartite vertex expansion. In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 881–899. SIAM (2017)

[4] Cohen, M.B., Lee, Y.T., Song, Z.: Solving linear programs in the current matrix multiplication time. Journal of the ACM (JACM) 68(1), 1–39 (2021)

[5] Davison, A.C., Hinkley, D.V.: Bootstrap methods and their application. Cambridge university press (1997)

[6] Dietzfelbinger, M., Hagerup, T., Katajainen, J., Penttonen, M.: A reliable randomized algorithm for the closest-pair problem. Journal of Algorithms 25(1), 19–51 (1997)

[7] Goldberg, A.V.: Finding a maximum density subgraph. University of California Berkeley Technical report (1984)

[8] Goldberger, A.L., Amaral, L.A., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E.: Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. circulation 101(23), e215–e220 (2000)

[9] Indyk, P., Lewenstein, M., Lipsky, O., Porat, E.: Closest pair problems in very high dimensions. In: International Colloquium on Automata, Languages, and Programming. pp. 782–792. Springer (2004)

[10] Khuller, S., Matias, Y.: A simple randomized sieve algorithm for the closest-pair problem. Information and Computation 118(1), 34–37 (1995)

[11] Korpela, J., Puolamäki, K., Gionis, A.: Confidence bands for time series data. Data mining and knowledge discovery 28(5), 1530–1553 (2014)

[12] Mandel, M., Betensky, R.A.: Simultaneous confidence intervals based on the percentile bootstrap approach. Computational statistics & data analysis 52(4), 2158–2165 (2008)

[13] Orlin, J.B.: Max flows in $O(nm)$ time, or better. In: Proceedings of the forty-fifth annual ACM symposium on Theory of computing. pp. 765–774 (2013)

[14] Rabin, M.O.: Probabilistic algorithms. In: Traub, J.F. (ed.) Algorithms and Complexity: New Directions and Recent Results. Academic Press New York (1976)

[15] Schrijver, A.: A combinatorial algorithm minimizing submodular functions in strongly polynomial time. Journal of Combinatorial Theory, Series B 80(2), 346–355 (2000)

[16] Tatti, N.: Density-friendly graph decomposition. ACM Transactions on Knowledge Discovery from Data (TKDD) 13(5), 1–29 (2019)

[17] Tsourakakis, C.: The k-clique densest subgraph problem. In: Proceedings of the 24th international conference on world wide web. pp. 1122–1132 (2015)