

Anomaly detection and classification for streaming data using PDEs*

Bilal Abbasi[†] Jeff Calder[‡] Adam M. Oberman[§]

March 17, 2017

Abstract

Nondominated sorting, also called Pareto Depth Analysis (PDA), is widely used in multi-objective optimization and has recently found important applications in multi-criteria anomaly detection. Recently, a partial differential equation (PDE) continuum limit was discovered for nondominated sorting leading to a very fast approximate sorting algorithm called *PDE*-based ranking. We propose in this paper a fast real-time streaming version of the PDA algorithm for anomaly detection that exploits the computational advantages of PDE continuum limits. Furthermore, we derive new PDE continuum limits for sorting points within their nondominated layers and show how the new PDEs can be used to classify anomalies based on which criterion was more significantly violated. We also prove statistical convergence rates for PDE-based ranking, and present the results of numerical experiments with both synthetic and real data.

1 Introduction

Sorting, or ordering of multivariate data is an important and challenging problem in many fields of computational science. Since there is no canonical linear ordering for multivariate data, many different notions of ordering have been proposed in the literature [25], and the problem is very much application dependent.

In the context of multiobjective optimization, ordering by dominance relations has achieved prominence. A general multiobjective optimization problem involves finding among a set of feasible solutions those that minimize a collection of objectives. One feasible solution is said to *dominate* another if it gives a smaller value for *every* objective. The collection of feasible solutions that are not dominated by any other solution are called *Pareto-optimal* or *nondominated*. In the database community the Pareto-optimal solutions are called the *skyline* of the dataset [24].

The notion of Pareto-optimality is widely used in evolutionary algorithms for multiobjective optimization [29], such as the Nondominated Sorting Genetic Algorithm (NSGA-II) [11], the Strength Pareto Evolutionary Algorithm (SPEA) [32, 33], and the Pareto envelope-based selection algorithm (PESA) [10], among many others (see [16] for a survey). Central to many

*The second author was supported by NSF grant DMS-1500829.

[†]Department of Mathematics and Statistics, McGill University. (bilal.abbasi.ba@gmail.com)

[‡]School of Mathematics, University of Minnesota. (jcalder@umn.edu)

[§]Department of Mathematics and Statistics, McGill University. (adam.oberman@mcgill.ca)

of these algorithms is the assignment of a fitness to each feasible solution based on sorting all the feasible solutions via dominance.

The NSGA-II algorithm assigns its fitness level via *nondominated sorting*, sometimes called *Pareto Depth Analysis* (PDA), which arranges the feasible solutions into layers by repeatedly peeling off the Pareto-optimal solutions. Nondominated sorting has also found applications in gene selection and ranking [18], anomaly detection [21, 22], and multiquery image retrieval [20]. As it turns out, nondominated sorting is equivalent to the *longest chain problem*, which has a long history in combinatorics and probability [2, 17, 30].

Due to the wide use of NSGA-II, there has been significant interest in fast algorithms for nondominated sorting [11, 23, 15]. Recently, Calder et al. [6] established a continuum limit for nondominated sorting that corresponds to solving a Hamilton-Jacobi partial differential equation (HJE). This result shows that there is a simple asymptotic structure underlying nondominated sorting, and this opens the door to extremely fast algorithms based on exploiting this structure. Calder et al. [7] recently proposed a sublinear algorithm for approximate nondominated sorting called *PDE-based ranking* that is based on estimating the distribution of the data and solving the HJE numerically.

The purpose of this paper is twofold. First, we show how to use PDE-based ranking to significantly improve the performance of algorithms that are based on nondominated sorting. To illustrate this in a concrete setting, we propose a new real-time version of the multi-criteria PDA anomaly detection algorithm from [22] that uses PDE-based ranking in place of nondominated sorting. The computational complexity is reduced by an order of magnitude (from quadratic to linear), and this allows the model to be updated in real-time upon the acquisition of each additional data sample. We also prove in Theorem 2 a statistical convergence rate for the PDE-based ranking continuum approximation.

Second, we present a new partial differential equation (PDE) continuum limit for ordering solutions *within* the layers generated by nondominated sorting in two dimensions. This new continuum limit allows us to efficiently explore the tradeoff between multiple objectives. In the context of multi-criteria anomaly detection, we show how to use this PDE continuum limit to classify anomalies based on which criterion is more significantly violated. We give a derivation of these new continuum limits and present a convergence analysis. In both cases, we trade exact algorithms of high computational complexity for fast approximate algorithms that are convergent, meaning that the error in the approximation goes to zero as the sample size grows.

This paper is organized as follows. In Section 2 we review the continuum limit for nondominated sorting from [6], and present the PDA multicriteria anomaly detection algorithm from [22]. In Section 3, we derive two new PDE continuum limits for ordering points within Pareto fronts, in Section 4 we construct fast upwind schemes for solving these PDEs numerically. In Section 5, we present our fast PDE-based anomaly detection and classification algorithm in the context of streaming data, and in Section 6 we present the results of numerical experiments for both real and synthetic data streams.

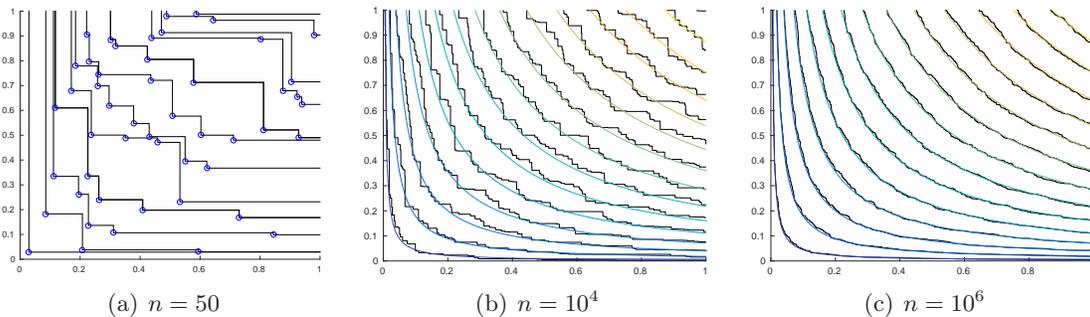


Figure 1: Illustration of nondominated sorting of *i.i.d.* random variables X_1, \dots, X_n drawn from the uniform distribution on $[0, 1]^2$.

2 Previous work

2.1 Nondominated sorting

Nondominated sorting arranges a set of points in \mathbb{R}^d into layers by repeatedly peeling off the coordinatewise minimal points. The coordinatewise partial order on \mathbb{R}^d is defined by

$$x \leq y \iff \forall i \ x_i \leq y_i \quad (x, y \in \mathbb{R}^d).$$

Let $S_n = \{X_1, \dots, X_n\} \subset \mathbb{R}^d$ be a collection of n points in \mathbb{R}^d . We say a point $X_i \in S_n$ is minimal (or nondominated), if no other non-identical point in S_n is smaller with respect to the coordinatewise partial order \leq . The first nondominated layer, denoted \mathcal{F}_1 , consists of the minimal points from S_n . The second nondominated layer, denoted \mathcal{F}_2 , consists of the minimal points from $S_n \setminus \mathcal{F}_1$ and the k^{th} layer is defined recursively by

$$\mathcal{F}_k = \text{Minimal points from } S_n \setminus (\mathcal{F}_1 \cup \dots \cup \mathcal{F}_{k-1}).$$

Nondominated sorting refers to the process of arranging the set S_n into the nondominated layers $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \dots$, which are also called *Pareto fronts*. See Figure 1 for a demonstration of nondominated sorting applied to random points. The index of the Pareto front that a point $X_i \in S_n$ lies on is often called the *Pareto depth* or *Pareto rank* of X_i , and provides the fitness score for the NSGA-II algorithm. In the context of multiobjective optimization, d represents the number of objectives.

The original nondominated sorting algorithm proposed in [11] requires $O(dn^2)$ memory and operations. The quadratic memory complexity in n renders the algorithm intractable for even moderate n . Jensen [23] proposed an algorithm with asymptotic complexity of $O(n(\log n)^{d-1})$ as $n \rightarrow \infty$. The two dimensional version of Jensen’s algorithm was discovered independently in the combinatorics community by Felsner and Wernisch [14]. Fortin et al. [15] recently made some improvements to Jensen’s algorithm regarding its treatment of points with identical coordinates. The exponential complexity of the Jensen-Fortin algorithm with respect to d suggests it may not be useful for high dimensional problems. However, recent numerical results have suggested a better asymptotic complexity as $d \rightarrow \infty$ with n fixed [22]. We also mention there are several other notable approaches to nondominated sorting [28, 8, 13].

Calder et al. [6] discovered a Hamilton-Jacobi equation continuum limit for nondominated sorting. The result applies in the setting where $S_n = \{X_1, \dots, X_n\}$ is a sequence of *i.i.d.* random variables with probability density f on the unit box $(0, 1)^d$. Define the *Pareto depth*

function $U_n : S_n \rightarrow \mathbb{N}$ associated with nondominated sorting of S_n by $U_n(X_i) = k$ if and only if $X_i \in \mathcal{F}_k$. The following continuum limit was established in [6, 4].

Theorem 1 (HJE Continuum Limit). *With probability one*

$$n^{-\frac{1}{d}}U_n \longrightarrow C_d u \quad \text{uniformly on } [0, 1]^d \text{ as } n \rightarrow \infty$$

where $C_d > 0$ is a constant depending only on d , and $u \in C^{0, \frac{1}{d}}([0, 1]^d)$ is the unique nondecreasing viscosity solution of the Hamilton-Jacobi equation

$$\begin{cases} u_{x_1} \cdots u_{x_d} = f & \text{in } (0, 1)^d \\ u = 0 & \text{on } \partial(0, 1)^d \setminus (0, 1)^d. \end{cases} \quad (1)$$

Here $u_{x_i} := \frac{\partial u}{\partial x_i}$ denotes the partial derivative of u with respect to x_i , and by nondecreasing we mean that $u_{x_i} \geq 0$ for all i . We note that when $f = 1$ is the uniform density, the solution of (1) is

$$u(x) = d(x_1 \cdots x_d)^{\frac{1}{d}}. \quad (2)$$

Figure 1 gives an illustration of this continuum limit when X_1, \dots, X_n are independent and uniformly distributed. The continuum limit in Theorem 1 states that the Pareto fronts converge to the level sets of the viscosity solution u of (1). While the value of C_d is not needed for sorting, we should mention that it is known only in dimension $d = 2$, in which case $C_2 = 1$ [26, 31].

Calder et al. [7] proposed a fast algorithm for approximate nondominated sorting called *PDE-based ranking* that is based on estimating the density function f from a small subset of the data X_1, \dots, X_n and then solving (1) numerically. PDE-based ranking can drastically reduce the computation time of nondominated sorting in low dimensions ($d = 2, 3$) while maintaining very high sorting accuracy.

Let us say a few words about viscosity solutions. Hamilton-Jacobi equations like (1) generally do not admit classical solutions (i.e., continuously differentiable solutions) due to the possibility of crossing characteristics. There are, however, infinitely many functions u that are differentiable almost everywhere and satisfy (1) at each point of differentiability. The notion of viscosity solution selects from among these infinitely many feasible solutions the one that is ‘physically correct’ for a *very* wide range of problems. The viscosity solution is correct in this context because it captures the continuum limit of the Pareto depth function [6].

The notion of viscosity solution is based on the maximum principle and enjoys very strong stability properties. It is a notion of weak solution that allows merely continuous functions to be solutions of a fully nonlinear PDE. While viscosity solutions may not possess the derivatives appearing in the equation in the classical sense, the reader will not lose much in the way of understanding by assuming that u is continuously differentiable. In the context of viscosity solutions, the maximum principle is used to prove a comparison principle, which says that subsolutions lie below supersolutions provided their boundary conditions do as well. The reason the notion of viscosity solution is correct in this context is that nondominated sorting also obeys a comparison principle; namely, if we introduce new points into our data set (i.e., we increase the point density), then the Pareto depth function increases as well. For more details on viscosity solutions we refer the reader to [1].

2.2 Anomaly detection

To illustrate the computational advantages of PDE-based ranking, we consider a concrete application of nondominated sorting to anomaly detection [22]. Anomaly detection refers to the problem of detecting patterns in data that deviate from the expected behavior. It is an important and challenging problem with a wide array of applications, including computer intrusion detection, video surveillance, credit card fraud, and biometrics [19, 9]. Many anomaly detection algorithms rely on the availability of a measure of distance (or similarity) between data samples, and look for anomalies by finding samples that are far from their nearest neighbors (see [22] and references therein). These algorithms are usually called *similarity-based*, and are widely used due to their simplicity and robustness.

In contrast, feature-based algorithms seek to embed the data into a relatively low dimensional Euclidean space and make use of the ambient Euclidean (or other) distance to detect anomalies. Techniques used for feature-based algorithms include support vector machines (SVM), clustering, neural networks, and statistical approaches based on density estimation [9]. In this paper we consider *similarity-based* approaches.

In many applications, multiple measures of similarity may be required to detect certain types of anomalies. For example, when tracking pedestrians in video surveillance, one criterion may correspond to differences in individual walking speeds, while another might correspond to differences in the shapes of trajectories. Using multiple criteria allows one to detect a wider range of anomalies than could be obtained from a single criterion alone.

Hsiao et al. [22] proposed an algorithm for multi-criteria anomaly detection that integrates the information from multiple similarity measures via nondominated sorting (or Pareto Depth Analysis (PDA)). Suppose we have a training set consisting of N objects Y_1, \dots, Y_N and d measures of similarity c_1, \dots, c_d for comparing these objects. Without loss of generality, we assume $0 \leq c_i(\cdot, \cdot) \leq 1$ —a lower score indicates the objects are more similar with respect to the i^{th} criteria. The training phase of the algorithm consists of computing the $n := \binom{N}{2}$ dyads

$$X_{i,j} = (c_1(Y_i, Y_j), \dots, c_d(Y_i, Y_j)) \in [0, 1]^d, \quad (3)$$

and constructing the Pareto depth function U_n by applying nondominated sorting to the n points $\{X_{i,j}\}_{i,j=1}^N$. Recall that $U_n(X_{i,j}) = k$ if and only if $X_{i,j}$ belongs to the k^{th} Pareto front.

The testing phase of the algorithm receives a new object Y and compares it to all training samples to create N new dyads Z_1, \dots, Z_N given by

$$Z_j = (c_1(Y, Y_j), \dots, c_d(Y, Y_j)).$$

Fix a number k , and let $I \subset \{1, \dots, N\}$ denote the indices of training samples that are among the k nearest neighbors of Y with respect to at least one similarity measure c_i . The anomaly score for Y is

$$\nu = \frac{1}{|I|} \sum_{j \in I} U_n(Z_j), \quad (4)$$

and Y is declared an anomaly when ν is larger than a predefined threshold $\rho > 0$. We note that it is possible to allow different values of k for each criterion. The idea is that nominal samples should be close to many of their nearest neighbors from the training data in one or many similarities, and thus the dyads Z_1, \dots, Z_N will lie on earlier Pareto fronts. An anomalous sample should be far from its nearest neighbors in the training set in many or all

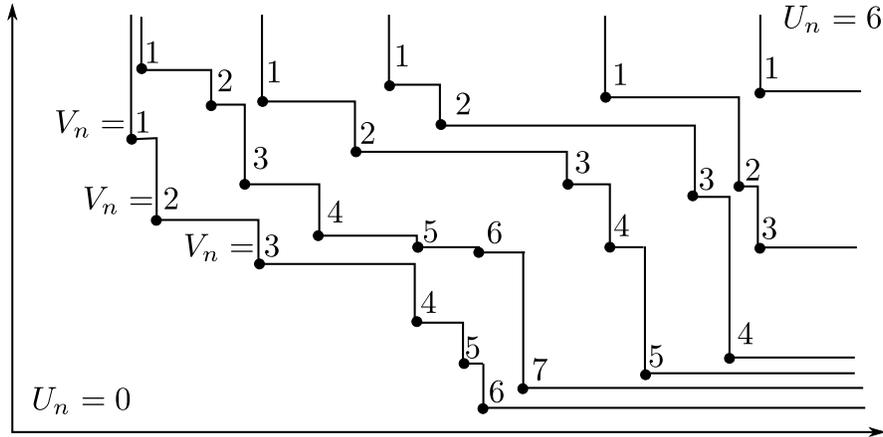


Figure 2: Illustration of the function V_n that orders the points within each nondominated layer.

of the similarities, and the dyads will consequently fall on deeper fronts. The PDA anomaly detection algorithm has been validated on real and synthetic data in [21, 22] and has been shown to achieve state of the art results for integrating information from multiple similarities.

3 New PDE continuum limits

The Hamilton-Jacobi Equation (HJE) continuum limit (1) gives information about which Pareto front a sample lies on. It is also important in applications to know where a sample lies within its Pareto front, as this gives information about the trade-off between the multiple objectives. We present here some new PDE continuum limits for ordering of points *within* Pareto fronts in dimension two.

Suppose $d = 2$ and let $S_n = \{X_1, \dots, X_n\}$ be a sequence of *i.i.d.* random variables with continuous density f on $(0, 1)^2$. Apply nondominated sorting to S_n and then order the points within each Pareto front by x_1 -coordinate. This defines a function $V_n : S_n \rightarrow \mathbb{N}$ given by

$$V_n(X_i) := \text{Index of } X_i \text{ within its Pareto front.} \quad (5)$$

Figure 2 gives an illustration of V_n .

By the continuum limit of nondominated sorting (Theorem 1), there are on the order of $n^{\frac{1}{2}}$ Pareto fronts. Since there are n points in total, each front should have on the order of $n^{\frac{1}{2}}$ points. Therefore let us suppose that

$$n^{-\frac{1}{2}}V_n \longrightarrow v \quad \text{as } n \rightarrow \infty$$

uniformly with probability one, where $v : [0, 1]^2 \rightarrow \mathbb{R}$ is continuously differentiable.

Fix a large value of n and consider a point $x \in (0, 1)^2$. Fix $\varepsilon > 0$ and let

$$y = x + \varepsilon \nabla^\perp u(x),$$

where $\nabla^\perp u = (u_{x_2}, -u_{x_1})$. Since $\nabla^\perp u$ is tangent to the level set $\{u = u(x)\}$, we have $u(y) \approx u(x)$, i.e., x and y are roughly on the same Pareto front. Let A denote the rectangle

whose diagonal is the line segment from x to y , and let L_n denote the number of points on the Pareto front passing through x and y that fall within A . See Figure 3 for an illustration of the setup. Then we have

$$\varepsilon \nabla v(x) \cdot \nabla^\perp u(x) \approx v(y) - v(x) \approx n^{-\frac{1}{2}}(V_n(y) - V_n(x)) = n^{-\frac{1}{2}}L_n. \quad (6)$$

Here, $\nabla v = (v_{x_1}, v_{x_2})$ denotes the gradient of v . When $\varepsilon > 0$ is small, the random variables within A are approximately uniformly distributed within A . Furthermore, as illustrated in Figure 3, we can scale A to the unit box $[0, 1]^2$ without changing the partial ordering within A . Hence, it is reasonable to conjecture that $L_n \sim c\sqrt{m}$ as $n \rightarrow \infty$, where $c > 0$ is a universal constant and m is the number of samples falling in A . While the value of c is not needed for sorting (since we perform a normalization in (9) below), a simple scaling argument suggests that $c = 1$, so we will take $L_n \sim \sqrt{m}$. By the law of large numbers

$$m \sim n \int_A f dx \approx n|A|f(x) = n\varepsilon^2 u_{x_1} u_{x_2} f(x),$$

since the side lengths of A are $|x_1 - y_1| = \varepsilon u_{x_2}$ and $|x_2 - y_2| = u_{x_1}$. Combining this with $u_{x_1} u_{x_2} = f$, (6) and $L_n \sim \sqrt{m}$ yields

$$\varepsilon \nabla v(x) \cdot \nabla^\perp u(x) \approx f(x)\varepsilon.$$

Hence this simple heuristic argument suggests that v satisfies the linear transport equation

$$\boxed{\begin{cases} \nabla v \cdot \nabla^\perp u = f & \text{in } (0, 1)^2 \\ v = 0 & \text{on } \{x_2 = 1\}. \end{cases}} \quad (7)$$

Recall u is the viscosity solution of (1). When $f = 1$ we have $u(x) = 2\sqrt{x_1 x_2}$. If we plug this into (7) and look for a separable solution of the form $v(x) = f_1(x_1)f_2(x_2)$ we find that when $f = 1$

$$v(x) = -\log(x_2)\sqrt{x_1 x_2}. \quad (8)$$

Since each Pareto front has in general a different number of points, the values of V_n within different fronts are difficult to compare. Therefore it is natural to consider the following normalization:

$$W_n(X_i) := \frac{V_n(X_i)}{\#\mathcal{F}(X_i)}, \quad (9)$$

where $\mathcal{F}(X_i)$ denotes the Pareto front that X_i belongs to. The quantity $W_n(X_i)$ is an index between 0 and 1 that gives information about where the point X_i falls within its Pareto front. The arguments above suggest that

$W_n \rightarrow w$ uniformly with probability one, where

$$w(x) = \frac{v(x)}{v(1, \psi(u(x)))}, \quad (10)$$

and ψ is the inverse of $x_2 \mapsto u(1, x_2)$. In other words, we are normalizing $v(x)$ by the asymptotic number of points on the front to which x belongs.

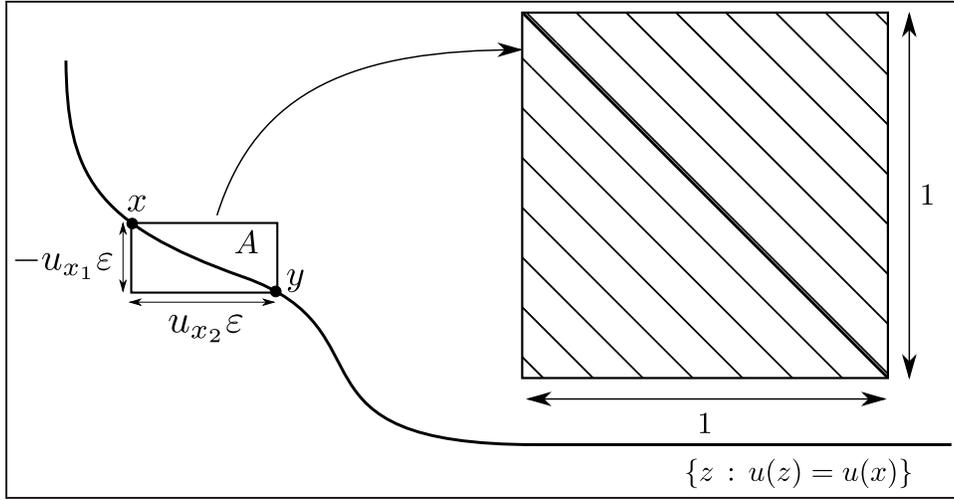


Figure 3: A depiction of some quantities from the derivation of the new continuum limit PDE.

The expression in (10) is difficult to work with numerically. We will instead derive a PDE for w . Differentiating (10) yields

$$v \nabla w = w \nabla v - w^2 v_{x_2} \psi'(u) \nabla u.$$

Take the dot product of both sides with $\nabla^\perp u$ and recall (7) to find that

$$v \nabla w \cdot \nabla^\perp u = w \nabla v \cdot \nabla^\perp u = w f.$$

Since $w = 1$ on $\{x_1 = 1\}$, w can be characterized as the solution of the following transport equation

$$\boxed{\begin{cases} v \nabla w \cdot \nabla^\perp u = w f & \text{in } (0, 1)^2 \\ w = 1 & \text{on } \{x_1 = 1\}. \end{cases}} \quad (11)$$

We note that it would seem equally reasonable to have chosen the boundary condition $w = 0$ on $\{x_2 = 1\}$ instead. However, in this case it is easy to verify that $w = v$ would solve (11), so the solution is not uniquely determined by the boundary condition $w = 0$ on $\{x_2 = 1\}$. This issue arises numerically as well. Indeed, we have found experimentally that if we solve (11) numerically with an upwind scheme and the boundary condition $w = 0$ on $\{x_2 = 1\}$ we find the solver automatically computes v instead of w , and it is unclear how to select the correct solution without changing the boundary condition to $w = 1$ on $\{x_1 = 1\}$. It is impossible to specify both boundary conditions simultaneously since the characteristic curves, which are the level curves of u , flow through both boundaries.

Note that when $f = 1$ we have $u(x) = 2\sqrt{x_1 x_2}$ and $v(x) = -\log(x_2)\sqrt{x_1 x_2}$. Plugging these into (11) and using the method characteristics we find that for $f = 1$

$$w(x) = \frac{\log(x_2)}{\log(x_1) + \log(x_2)}. \quad (12)$$

See Figure 4 for a comparison of V_n and W_n to their continuum limits (7) and (11), respectively. While the arguments in this section are not rigorous, we present a convergence

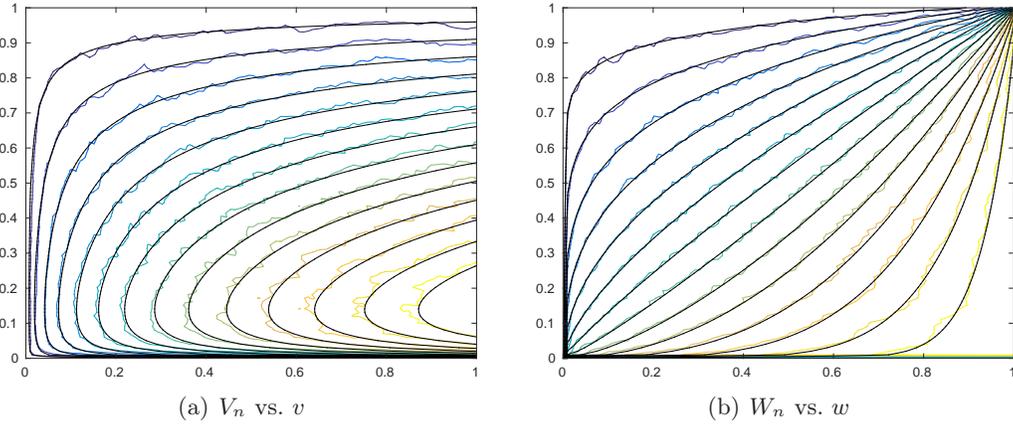


Figure 4: Results of a simulation comparing the level sets of V_n and W_n to the solutions of their continuum limits PDEs (7) and (11).

analysis in Section 3.1 that gives very strong numerical evidence for their validity. A rigorous proof is the subject of current investigation and is far outside the scope of this paper.

On a more technical note, since u is not necessarily differentiable the velocity $\nabla^\perp u$ in the transport equations (7) and (11) is not well-defined. To make sense of these PDE rigorously, we can instead write them in divergence form, since

$$-\operatorname{div}(u\nabla^\perp v) = \nabla v \cdot \nabla^\perp u.$$

This suggests that it is possible to prove existence and uniqueness of weak solutions (defined via integration by parts) of (7) in the Sobolev space H^1 , under the requirement that u is merely continuous. Such results are outside the scope of this paper, and we intend to pursue them in a future work.

3.1 Convergence analysis

We present here a convergence analysis for the continuum limits (7) and (11) in the case that $f \equiv 1$, i.e., the samples are independent and uniformly distributed on the unit box $[0, 1]^2$. In this case we can solve all three PDEs (1), (7), and (11) in closed form using the formulas (2), (8), and (12), respectively.

We performed a convergence analysis by drawing X_1, \dots, X_n independent and uniformly distributed on $[0, 1]^2$ and computing V_n and W_n according to their definitions (5) and (9), respectively. We measured the discrepancy with the continuum limits in the ℓ_1 and ℓ_∞ norms, computed by

$$\|v - n^{-\frac{1}{2}}V_n\|_{\ell_1} := \frac{1}{n} \sum_{i=1}^n |v(X_i) - n^{-\frac{1}{2}}V_n(X_i)|.$$

and

$$\|v - n^{-\frac{1}{2}}V_n\|_{\ell_\infty} := \max_{1 \leq i \leq n} |v(X_i) - n^{-\frac{1}{2}}V_n(X_i)|,$$

respectively. The definitions of $\|w - W_n\|_{\ell_\infty}$ and $\|w - W_n\|_{\ell_1}$ are similar. Figure 5 shows the errors for a single realization and various values of n ranging from $n = 10^2$ to $n = 10^8$. Each

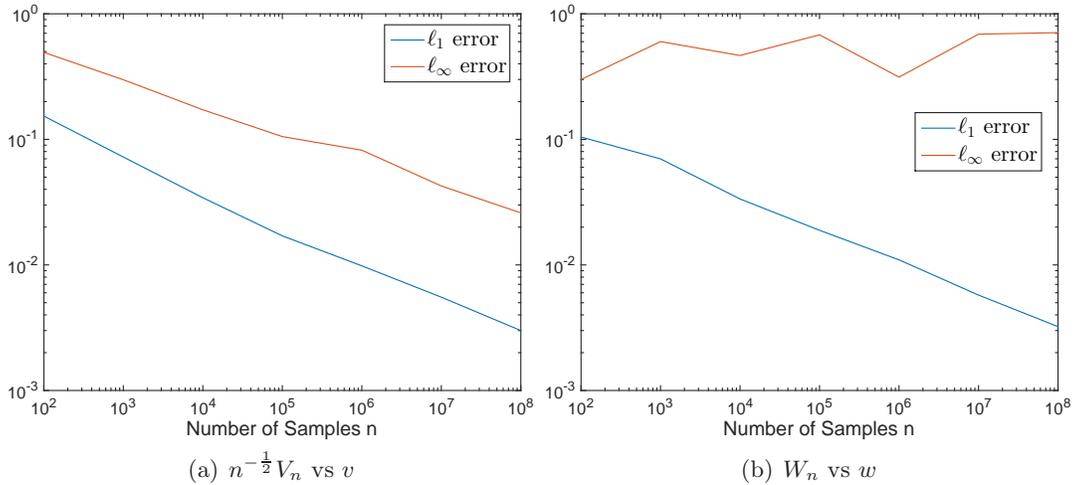


Figure 5: Convergence analysis of conjectured continuum limits. In (a), the error is consistent with $O(n^{-\alpha})$ where $\alpha \approx 0.3$ for the ℓ_1 norm, and $\alpha \approx 0.2$ for the ℓ_∞ norm. In (b), the error for the ℓ_1 norm is consistent with $O(n^{-\alpha})$ where $\alpha \approx 0.25$. Since w is discontinuous at $x = (1, 1)$, the convergence $W_n \rightarrow w$ cannot be uniform (i.e., in the ℓ_∞ norm).

of the errors is observed to be converging to zero at a rate of $O(n^{-\alpha})$ where $\alpha \approx 0.25$, except for $\|w - W_n\|_{\ell_\infty}$. Upon closer inspection, the function $w(x) = \log(x_2)/(\log(x_1) + \log(x_2))$ is discontinuous at $x = (1, 1)$, hence uniform convergence is impossible. This discontinuity reflects the fact that near $x = (1, 1)$ the Pareto fronts cut off an infinitesimal portion of the top corner of the box, and hence W_n transitions from 0 to 1 over an infinitesimally short distance.

4 Numerical schemes

We address in this section the problems of solving the PDEs (1), (7), (11) numerically. Since each PDE involves the solution of the previous PDEs, they must be solved in the order (1)-(7)-(11). We solve (1) numerically using the (formally) first order accurate upwind scheme presented in [5]. The scheme is similar to fast sweeping, but only requires sweeping the grid exactly once, and thus has linear complexity in the number of grid points.

Here, we show how to construct upwind finite difference schemes for the transport PDEs (7) and (11). Fix a grid resolution $h > 0$ and for $U \subset \mathbb{R}^2$ define $U_h := U \cap (h\mathbb{Z}^2)$. We will solve the transport equations on the grid $[0, 1]_h^2$. Both PDEs are degenerate when $\nabla u = 0$, which can only happen when f vanishes (since $u_{x_1} u_{x_2} = f$). To avoid this degeneracy, we numerically precondition the density by replacing f with $f + h^2$ before solving (1) numerically.

The transport equation (7) can be written out as

$$u_{x_2} v_{x_1} - u_{x_1} v_{x_2} = f \quad \text{on } (0, 1)^2$$

with $v(x_1, 1) = 0$. The unknown function is v ; u is obtained by solving (1). Since $u_{x_1} \geq 0$ and $u_{x_2} \geq 0$ the coefficients of v_{x_1} and v_{x_2} have opposite signs. Thus an upwind scheme will use either (A) backward differences for v_{x_1} and forward differences for v_{x_2} , or (B) vice-versa. The choice depends on the direction we want information to propagate. Since our boundary

condition is $v = 0$ on $\{x_2 = 1\}$ and information flows along Pareto fronts in the positive x_1 direction and negative x_2 direction, the correct choice for an upwind scheme is (A) backward differences in v_{x_1} and forward differences in x_2 , which effectively forces the scheme to look backwards along the current Pareto front (or level set of u).

Thus, our upwind scheme for (7) is

$$\boxed{\begin{cases} u_{x_2} D_1^- v_h - u_{x_1} D_2^+ v_h = f & \text{in } (0, 1)_h^2, \\ v_h(0, x_2) = v_h(x_1, 1) = 0. \end{cases}} \quad (13)$$

where $v_h : [0, 1]_h^2 \rightarrow \mathbb{R}$ is the numerical solution, and D_i^\pm are the finite differences defined by

$$D_i^\pm v(x) := \pm \frac{v(x \pm h e_i) - v(x)}{h},$$

where $e_1 = (1, 0)$ and $e_2 = (0, 1)$. At each grid point, (13) is linear equation that is readily solved for $v_h(x)$ in terms of $v_h(x - h e_1)$ and $v_h(x + h e_2)$ to obtain

$$v_h(x) = \frac{u_{x_2}(x)v_h(x - h e_1) + u_{x_1}(x)v_h(x + h e_2) + h f(x)}{u_{x_1} + u_{x_2}}. \quad (14)$$

The numerical solution v_h is computed by sweeping the grid $(0, 1)_h^2$ exactly once in the upwind direction $(1, -1)$ starting with the boundary condition $v_h(0, x_2) = v_h(x_1, 1) = 0$. We note that the boundary condition $v_h(0, x_2) = 0$ is just for numerical convenience, and is not used directly by the scheme, since $u_{x_2}(0, x_2) = 0$ so on the line $x_1 = 0$ so the solution depends only on $v_h(x + h e_2)$ when $x_1 = 0$. When computing $v_h(x)$, we replace u_{x_1} and u_{x_2} in (14) by first order finite differences of the solution u_h of (1) on the same grid. Our numerical experiments suggest that the scheme is not sensitive to the choice of discretization of u_{x_1} and u_{x_2} . We note that convergence of the scheme (13) is a classical result when $u \in C^1$, however, u is in general only Hölder continuous. We leave the analysis of the scheme for $u \notin C^1$ to future work.

We now consider the second transport equation (11). If we again write the PDE out we have

$$v u_{x_2} w_{x_1} - v u_{x_1} w_{x_2} = w f \quad \text{in } (0, 1)^2$$

with boundary condition $w(1, x_2) = w(x_1, 0) = 1$. Since $v u_{x_2} \geq 0$ and $v u_{x_1} \geq 0$ we have the same upwind choices as before. However, here we want information to propagate from the boundary where $x_1 = 1$ or $x_2 = 0$ backwards along Pareto fronts (level sets of u) in the direction $(-1, 1)$. Thus we use forward differences for w_{x_1} and backward differences for w_{x_2} , and our upwind scheme for (15) has the form

$$\boxed{\begin{cases} v_h u_{x_2} D_1^+ w_h - v_h u_{x_1} D_2^- w_h = w_h f & \text{in } (0, 1)_h^2, \\ w_h(1, x_2) = w_h(x_1, 0) = 1. \end{cases}} \quad (15)$$

This is a linear equation that can be solved for $w_h(x)$ in terms of $w_h(x + h e_1)$ and $w_h(x - h e_2)$ to obtain

$$w_h(x) = \frac{v_h(x) u_{x_2}(x) w_h(x + h e_1) + v_h(x) u_{x_1}(x) w_h(x - h e_2)}{h f(x) + v_h(x) u_{x_2}(x) + v_h(x) u_{x_1}(x)}. \quad (16)$$

This scheme can also be solved in a fast sweeping pattern visiting each grid point exactly once, and thus has linear complexity. Note that the boundary condition $w_h(x_1, 0) = 1$ is not directly

used by the scheme, since $u_{x_1}(x_1, 0) = 0$. We impose the condition simply for convenience. We note here as well that convergence of the scheme is classical when $u \in C^1$, and we leave the case of $u \notin C^1$ to future work.

5 Real-time anomaly detection

We propose here a modification of the PDA multicriteria anomaly detection algorithm [22] to the setting of online streaming data. Suppose we have a stream of possibly nonstationary data $\{Y_t\}_{t \in \mathbb{N}}$, and d measures of similarity c_1, \dots, c_d for comparing data samples. As before we suppose that $0 \leq c_i(\cdot, \cdot) \leq 1$. In the streaming setting, we observe the data Y_t sequentially and must determine whether Y_t is an anomaly based only on the previous history $\{Y_s : s < t\}$. Due to memory and computational constraints, it may not be feasible to use this entire history, especially when t is large. Therefore, we fix $T \geq 1$ and consider the windowed history

$$H_t = \{Y_s : t - T \leq s \leq t - 1\}. \quad (17)$$

We use the history H_t as training data in order to determine whether Y_t is an anomaly. Even without memory constraints, only the recent history H_t can be considered reliable when the data is nonstationary. As before, we define dyads

$$X_{r,s} = (c_1(Y_r, Y_s), \dots, c_d(Y_r, Y_s)) \in [0, 1]^d$$

corresponding to every pair (Y_r, Y_s) of the data stream. If we use the PDA anomaly detection algorithm with exact nondominated sorting, then we would need to store in memory all of the $n := \binom{T}{2} = O(T^2)$ dyads corresponding to pairs from the history H_t . Since the addition of a single new sample can potentially affect the arrangement of *all* the Pareto fronts, re-training the model when new samples are acquired requires applying nondominated sorting to *all* $O(T^2)$ dyads, which has complexity slightly worse than $O(T^2)$ for memory and operations. This makes it impossible to update the model frequently in the streaming setting without considering some type of approximation to the sorting.

Using PDE-based ranking we can reduce this complexity to $O(T)$. We keep a running estimate of the marginal distribution of the dyads using the following kernel density estimator

$$f_t(x) = \frac{1}{nh^d} \sum_{t-T \leq r < s \leq t-1} K\left(\frac{x - X_{r,s}}{h}\right). \quad (18)$$

Although there are $O(T^2)$ terms in the sum above, the density estimation $f_t(x)$ can be updated recursively in $O(T)$ time by writing $f_t(x) = f_{t-1}(x) + g_t(x)$ where

$$g_t(x) = \frac{1}{nh^d} \sum_{s=t-T}^{t-1} K\left(\frac{x - X_{s,t}}{h}\right) - K\left(\frac{x - X_{(t-T-1),s}}{h}\right).$$

In our experiments, we use a simple histogram estimator, which is a special case of (18). We then compute an approximation U_t of the Pareto depth function by solving the HJE (1) numerically using the estimated density f_t . By Theorem 1 the continuum approximation of the anomaly score is

$$\nu_t = \frac{1}{|I_t|} \sum_{s \in I_t} U_t(X_{s,t}), \quad (19)$$

where $I_t \subset \{t-T, \dots, t-1\}$ denotes the indices of samples from the history H_t that are among the k nearest neighbors of Y_t with respect to c_i for at least one $i \in \{1, \dots, d\}$. We declare Y_t anomalous if ν_t is greater than a predefined threshold ρ . The steps above work in arbitrary dimension $d \geq 2$ as well.

For the anomaly classification, we specialize to the case of $d = 2$. If Y_t is declared an anomaly, we then solve the transport equations (7) and (11) using the schemes (13) and (15), respectively, to obtain $W_t := w_h$. We define the *anomaly classification score*

$$\mu_t = \frac{1}{|I_t|} \sum_{s \in I_t} W_t(X_{s,t}), \quad (20)$$

and we declare Y_t a c_1 -anomaly if $\mu_t > 0.5$, and a c_2 -anomaly if $\mu_t < 0.5$. The idea is that if a sample is a c_1 -anomaly, then the first coordinate of the dyads $c_1(Y_s, Y_t)$ should be larger on average than in the training set, which is our windowed history. Therefore, the dyads corresponding to Y_t will be on average further to the right along each Pareto front. This corresponds to a front index larger than 0.5 on average. The situation is similar for c_2 anomalies, except that the dyads will be biased towards the left side of the fronts. See Algorithm 1 for a summary of our algorithm in pseudocode.

Algorithm 1 PDE-based online anomaly detection

- 1: **Given:** $\rho > 0$ and $T \in \mathbb{N}$
 - 2: $f_T \leftarrow (18)$ {Initialize density}
 - 3: **for** $t = T + 1 \rightarrow \infty$ **do**
 - 4: $f_t \leftarrow f_{t-1} + g_t$ {Update density estimation}
 - 5: $U_t \leftarrow (1)$ {Solve HJE continuum limit}
 - 6: $\nu_t \leftarrow (19)$ {Compute anomaly score}
 - 7: **if** $\nu_t > \rho$ **then**
 - 8: Declare Y_t to be anomalous
 - 9: $W_t \leftarrow (15)$ {Solve transport equations}
 - 10: $\mu_t \leftarrow (20)$ {Compute anomaly classification score}
 - 11: **if** $\mu_t > 0.5$ **then** Y_t is a c_1 -anomaly
 - 12: **if** $\mu_t \leq 0.5$ **then** Y_t is a c_2 -anomaly
 - 13: **end if**
 - 14: **end for**
-

There are some obvious modifications we could make to improve the performance of the algorithm. First, the continuum limit PDE need not be solved at every iteration, and could instead be solved only periodically, or whenever the density estimation f_t has substantially changed. Second, to keep track of a larger history without incurring additional costs, the history H_t could contain T elements equally (or randomly) spaced among the previous αT samples, where $\alpha \gg 1$. The algorithm would remain otherwise unchanged and the complexity of each iteration remains $O(T)$.

Since the PDE-based anomaly detection algorithm is based on continuum approximations, it is natural to seek a quantification of the approximation error. If we assume the dyads are *i.i.d.* we can prove the following convergence rate.

Theorem 2 (Convergence Rate). *Let X_1, \dots, X_n be i.i.d. with a Lipschitz continuous probability density $f : [0, 1]^2 \rightarrow [m, \infty)$, where $m > 0$. For $h > 0$ let \hat{u}_h denote the numerical solution of (1) obtained via estimating f from X_1, \dots, X_n via a histogram aligned to the grid of spacing h on $[0, 1]^2$. Then there exist constants $C_1, C_2 > 0$ such that*

$$\max_{[0,1]_h^2} |\hat{u}_h - u| \leq C_1 \sqrt{h} \quad (21)$$

holds with probability at least $1 - \exp(-C_2 nh^5 - 2 \log(h))$, where u is the viscosity solution of (1).

Theorem 2 suggests we should choose h as a function of n so that $nh^5 \gg \log(h^{-1})$. In particular, if we choose $h = h(n) \rightarrow 0^+$ so that

$$\lim_{n \rightarrow \infty} \frac{nh^5}{\log(n)} = \infty,$$

then by the Borel-Cantelli Lemma $\hat{u}_h \rightarrow u$ almost surely and uniformly on $[0, 1]^2$ as $n \rightarrow \infty$. We also note that Theorem 2 extends easily to higher dimensions $d \geq 3$. In this case the same convergence rate (21) holds with probability at least

$$1 - \exp(-C_2 nh^{2d+1} - d \log(h)).$$

Proof of Theorem 2. Let X_1, \dots, X_n be independent and identically distributed random variables on $[0, 1]^2$ with Lipschitz continuous density $f : [0, 1]^2 \rightarrow [0, \infty)$. Recall that f is assumed to be positive on $[0, 1]^2$, i.e., there exists $m > 0$ such that $f \geq m$. Let $h := 1/K > 0$ be the grid resolution for solving (1) numerically and for estimating the density f with a histogram estimator, where $K \in \mathbb{N}$. For $1 \leq i \leq k$ and $1 \leq j \leq K$ let

$$B_{ij} = [(i-1)h, ih) \times [(j-1)h, jh)$$

denote the grid cell corresponding to (i, j) , and let N_{ij} denote the number of samples from X_1, \dots, X_n falling in B_{ij} . Then N_{ij} is a Binomial random variable with parameters n and

$$p_{ij} = \int_{B_{ij}} f(x) dx.$$

By the Chernoff-Hoeffding bound (see, e.g., [12]) we have

$$\mathbb{P}(|N_{ij} - \mathbb{E}N_{ij}| \geq t) \leq \exp\left(\frac{-2t^2}{n}\right) \quad (22)$$

for all $t \geq 0$. Let $x_{ij} = (ih, jh)$ denote the grid points. The histogram estimation of f at grid point x_{ij} is given by

$$\hat{f}_h(x_{ij}) := \frac{N_{ij}}{n|B_{ij}|} = \frac{N_{ij}}{nh^2}.$$

Combining this with (22) we have

$$\mathbb{P}\left(\left|\hat{f}_h(x_{ij}) - \mathbb{E}\hat{f}_h(x_{ij})\right| \geq t\right) \leq \exp(-2nh^4t^2) \quad (23)$$

Since $\mathbb{E}\widehat{f}_h(x_{ij}) = p_{ij}/h^2$ we have

$$\left|f(x_{ij}) - \mathbb{E}\widehat{f}_h(x_{ij})\right| = \frac{1}{h^2} \left| \int_{B_{ij}} f(x_{ij}) - f(x) dx \right| \leq \frac{1}{h^2} \int_{B_{ij}} |f(x_{ij}) - f(x)| dx \leq Ch, \quad (24)$$

due to the fact that f is Lipschitz. Here, C depends on the Lipschitz constant of f , which is defined by

$$\text{Lip}(f) = \sup_{x \neq y} \frac{|f(x) - f(y)|}{|x - y|}.$$

It follows from (24) that

$$\begin{aligned} |f(x_{ij}) - \widehat{f}_h(x_{ij})| &\leq \left| \widehat{f}_h(x_{ij}) - \mathbb{E}\widehat{f}_h(x_{ij}) \right| + \left| f(x_{ij}) - \mathbb{E}\widehat{f}_h(x_{ij}) \right| \\ &\leq \left| \widehat{f}_h(x_{ij}) - \mathbb{E}\widehat{f}_h(x_{ij}) \right| + Ch. \end{aligned}$$

Combining this with (23) and the union bound, there exists $C_1 > 0$ such that

$$\mathbb{P} \left(\|\widehat{f}_h - f\|_\infty \geq \lambda \right) \leq \exp \left(-2nh^4(\lambda - C_1h)^2 - 2\log(h) \right), \quad (25)$$

for all $\lambda > C_1h$, where

$$\|u - v\|_\infty := \max_{x_{ij} \in [0,1]_h^2} |u(x_{ij}) - v(x_{ij})|.$$

Let u_h and \widehat{u}_h denote the numerical solutions of (1) on the grid of spacing $h > 0$ computed with f and \widehat{f}_h on the right hand side, respectively. Standard maximum principle arguments (see [5]) yield

$$\|\widehat{u}_h - u_h\|_\infty \leq C\|\widehat{f}_h - f\|_\infty, \quad (26)$$

where the constant C depends on the lower bound $m > 0$ on f . By [5, Theorem 1,2], there exists a constant $C > 0$ such that

$$\|u_h - u\|_\infty \leq C\sqrt{h}.$$

Combining this with (26) we have

$$\|\widehat{u}_h - u\|_\infty \leq C_2(\|\widehat{f}_h - f\|_\infty + \sqrt{h}),$$

for some $C_2 > 0$. By (25)

$$\mathbb{P} \left(\|\widehat{u}_h - u\|_\infty \geq C_2(\lambda + \sqrt{h}) \right) \leq \exp \left(-2nh^4(\lambda - C_1h)^2 - 2\log(h) \right).$$

Setting $\lambda = C\sqrt{h}$ for large enough C we have

$$\mathbb{P} \left(\|\widehat{u}_h - u\|_\infty \geq C_3\sqrt{h} \right) \leq \exp \left(-C_4nh^5 - 2\log(h) \right),$$

for all $0 < h \leq 1$. □

6 Numerical results

We present several experiments that provide numerical evidence supporting the above arguments and outlining the effectiveness of our algorithm. The first two experiments were performed using synthetic data from [21, 22]. The streaming experiments consist of 1500 total samples with a window history of $T = 500$. To underscore the adaptive nature of our algorithm, each of these experiments incurs a significant trend change in the middle of the stream. The third and final experiment was performed with a real pedestrian trajectory data set from a video surveillance problem.

To evaluate the performance of the streaming algorithm we use a Receiver Operating Characteristic (ROC) curve and its resulting Area Under the Curve (AUC). We consider how the AUC varies with time as the algorithm takes in points from a stream. When changing the trend in the simulated streams, we also accordingly change the data used to generate the ROC curves, thereby giving us an appropriate method to visualize the learning aspect of the algorithm. In each simulated data stream we evaluate both the anomaly detection and anomaly classification. The results presented below represent the average of 20 trials. All PDEs were solved on a 100×100 grid.

In each experiment, we compare our continuum limits against the exact sorting PDA algorithm from [21, 22] and we see little to no difference in anomaly detection performance. The PDE-approximations reduce the complexity by an order of magnitude—from $O(T^2)$ to $O(T)$. To give an idea of the difference in CPU time, each trial in the experiments below takes 27 seconds to process with the PDE-approximations, compared to 413 seconds with exact sorting. If we increase the stream length and data history T by a factor of 3, the PDE-approximations take 160 seconds, while the exact sorting PDA algorithm takes over 9.3 hours.

6.1 Uniformly distributed data

The first experiment conducted with synthetic data took *i.i.d.* uniform samples on $[0, 1]^2$ to be nominal, and uniform samples from the region $[0, 1.1]^2 \setminus [0, 1]^2$ to be anomalous. Halfway through the stream the nominal region was changed to the box $[0, 2]^2$, and the corresponding anomalous region was changed to $[0, 2.2]^2 \setminus [0, 2]^2$. The two similarity criteria were simply taken to be the component-wise differences $|\Delta x_1|$ and $|\Delta x_2|$, respectively. The nearest neighbour parameters were chosen as $k_1 = 6, k_2 = 7$. At each time step in the simulated stream there was a 0.05 probability of drawing from the anomalous region.

Figure 6(a) shows the resulting AUCs at each time step. As expected, one can see a significant drop in the AUC of the anomaly detection at the mid-point when the trend is changed. We observe a sharp recovery of the AUC of the anomaly detection once the training history H_t contains a significant number of samples from the new distribution. This illustrates how the algorithm can quickly and efficiently learn a new trend in the data. Note that the AUC for the anomaly classification remains unchanged throughout the experiment because the classification of the anomalies in the new trend are the same with respect to the old trend.

6.2 Categorical data

For the second experiment, we used the synthetic categorical data from [22]. Each sample consists of 2 groups of categorical data A_1 and A_2 . Each group is comprised of 20 different

attributes, where each attribute can assume a different number of values. The number of possible values for the j^{th} attribute of the i^{th} group, denoted $n_{i,j}$, is chosen uniformly at random between 6 and 10. Each attribute is then assigned a categorical distribution with parameters $p_1, \dots, p_{n_{i,j}}$ which are in turn drawn from a Dirichlet distribution with parameters $\alpha_1, \dots, \alpha_{n_{i,j}}$.

The nominal distribution is characterized by setting $\alpha_1 = 5$ and $\alpha_k = 1$ for every $k \neq 1$ for every attribute. This forces a bias towards attributes assuming the value one. For the anomalous distribution we set $\alpha_k = 1$ for every k , so that no attribute has a bias towards assuming any particular value. Halfway through the stream the nominal distributions were changed so that for every attribute, the parameters of the categorical distribution were drawn from a Dirichlet distribution with parameters $\alpha_2 = 5$ and $\alpha_k = 1$ for every $k \neq 2$. This shifts the nominal bias towards the value two. The anomalous distribution was unchanged.

To generate a nominal sample, we draw from the nominal distribution for each group. To generate an anomalous sample, we randomly choose a group with probability 0.5 and draw from the anomalous distribution for that group, and nominal distribution for the other. At each time step in the stream there was a 0.05 probability of drawing an anomalous sample.

The similarity between samples was computed between respective groups using the Inverse Occurrence Frequency (IOF) measure presented in [3]. The Goodall2 and Overlap metrics gave similar performance. The nearest neighbour parameters were chosen as $k_1 = k_2 = 10$. Figure 6(b) shows the resulting AUCs at each time step. Similar to the previous experiment we observe a drop in the AUC of the anomaly detection and a recovery thereafter. We also observe a similar drop in anomaly classification and the corresponding recovery. In contrast to the previous example, the new anomalies are anomalies in both criteria with respect to the old trend, so that the classification has no bias towards a specific criteria.

6.3 Pedestrian trajectories

Our third experiment consisted of data from a real pedestrian trajectory data set [27], with over 100,000 trajectories. The first similarity criterion used to compare trajectories was their difference in shape, given by the ℓ_2 -distance between interpolated trajectories. The second was their difference in walking speed, given by the ℓ_2 -distance between the velocity histograms of each trajectory.

As a preliminary experiment, we tested the anomaly detection and anomaly classification on 1666 trajectories from a single day. These trajectories were hand-labelled as normal or anomalous by [22]. In each experiment the training set consisted of 500 trajectories randomly drawn from a total of 1666 trajectories that day. The mean AUCs of the PDE-based and exact sorting based algorithms were 0.9274 ± 0.0085 and 0.9363 ± 0.0072 , respectively and the ROC curves are shown in Figure 7(a). We observe very little difference between the exact sorting and the PDE-approximations. We cannot present quantitative results for anomaly classification in this setting as there is no ground truth labeled data available. Along with some normal trajectories, we also plotted some anomalous trajectories with their respective classification scores in Figure 7(b,c).

Finally, we applied the PDE-based streaming anomaly detection algorithm to a large portion of the pedestrian dataset, spanning over several days of data. Figure 6(c) shows the AUC as a function of artificial time for a simulated stream consisting of 15,000 trajectories with an initial training set of 400 randomly drawn trajectories. The small labeled portion of the

dataset (approx. 1000 trajectories) was used to generate the ROC curves.

7 Conclusion

In this paper, we showed how to use some recently discovered PDE continuum limits for non-dominated sorting to perform anomaly detection and classification in real-time in a streaming setting. The classification is performed using new PDE continuum limits for ordering within the nondominated layers. We proved convergence rates for the continuum approximations and presented the results of numerical experiments with synthetic and real data that show our algorithm can adapt quickly and efficiently to a changing data stream. Although we focused in this paper on the anomaly detection problem, the ideas are not restricted to this context. Indeed, nondominated sorting is widely used in multiobjective optimization, and the ideas in this paper potentially apply to any such application, leaving many interesting problems for future work.

In particular, we outline below some directions for future work that we are currently investigating.

1. The arguments used in Section 3 to derive the new PDE continuum limits (7) and (11) for sorting points within layers are not rigorous. We are currently investigating a rigorous proof of these conjectured continuum limits.
2. The upwind finite difference schemes for the transport equations (7) and (11) presented in Section 4 are provably convergent only when $u \in C^1$, which is not generally true, and only when we assume the exact values of u_{x_1} and u_{x_2} are used in the scheme. It would be interesting to prove convergence of these new upwind schemes without the assumption that $u \in C^1$, and under the more realistic condition that u_{x_1} and u_{x_2} are replaced by their finite difference approximations in the schemes for (7) and (11).
3. The PDEs for sorting points within fronts were presented in only $d = 2$ dimensions here. It would be interesting to extend these results to higher dimensions. In $d \geq 3$ dimensions, there is no canonical linear ordering of the points within each front. Instead, we can consider nondominated sorting of the points within each front under a partial order that “forgets” about one of the coordinates x_k (that is, we project to \mathbb{R}^{d-1} by removing the x_k coordinate and apply the usual nondominated sorting). This is akin to sorting the points within each front with respect to the x_k direction. Thus, we have d different ways to sort the points along each Pareto front, and similar arguments can be used to derive d different continuum limit PDEs for sorting functions v^1, \dots, v^k of the form

$$\prod_{i \neq k} \nabla v^k \cdot \nabla_{k,i}^\perp u = u_{x_k}^{d-2} f \quad \text{in } (0, 1)^d,$$

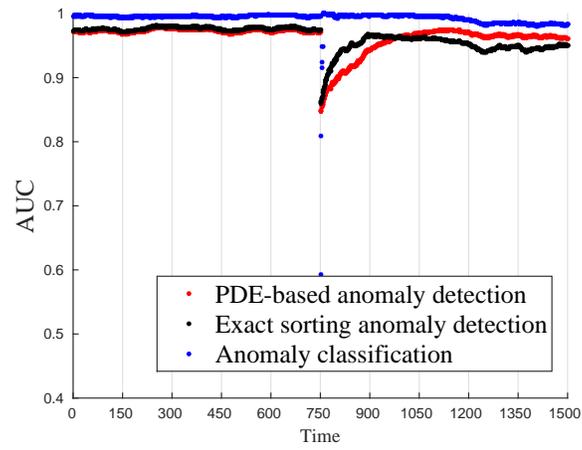
with boundary condition $v^k = 0$ on $\{x_k = 1\}$. Here, $\nabla_{k,i}^\perp u := u_{x_k} e_i - u_{x_i} e_k$, and e_1, \dots, e_d are the standard basis vectors in \mathbb{R}^d . We are currently investigating applications of these continuum PDEs, as well as a rigorous proof in dimensions $d \geq 3$.

References

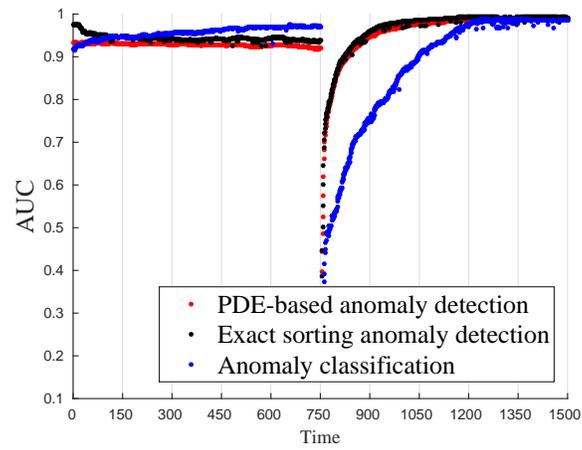
- [1] M. Bardi and I. Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Springer, 1997.
- [2] B. Bollobás and P. Winkler. The longest chain among random points in Euclidean space. *Proceedings of the American Mathematical Society*, 103(2):347–353, 1988.
- [3] L. Boriah, V. Chandola, and V. Kumar. Similarity measures for categorical data: A comparative evaluation. *red*, 30(2):3, 2008.
- [4] J. Calder. A direct verification argument for the Hamilton-Jacobi equation continuum limit of nondominated sorting. *arXiv preprint:1508.01565*, 2015.
- [5] J. Calder. Numerical schemes and rates of convergence for the Hamilton-Jacobi equation continuum limit of nondominated sorting. *arXiv preprint:1508.01557*, 2015.
- [6] J. Calder, S. Esedoğlu, and A. O. Hero. A Hamilton-Jacobi equation for the continuum limit of non-dominated sorting. *SIAM Journal on Mathematical Analysis*, 46(1):603–638, 2014.
- [7] J. Calder, S. Esedoğlu, and A. O. Hero. A PDE-based approach to non-dominated sorting. *SIAM Journal on Numerical Analysis*, 53(1):82–104, 2015.
- [8] Y. Cao. Pareto Sort. [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/1725> 2007.
- [9] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.
- [10] D. W. Corne, J. D. Knowles, and M. J. Oates. The pareto envelope-based selection algorithm for multiobjective optimization. In *Parallel Problem Solving from Nature PPSN VI*, pages 839–848. Springer, 2000.
- [11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [12] D. P. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009.
- [13] H. Fang, Q. Wang, Y.-C. Tu, and M. F. Horstemeyer. An efficient non-dominated sorting method for evolutionary algorithms. *Evolutionary computation*, 16(3):355–384, 2008.
- [14] S. Felsner and L. Wernisch. Maximum k-chains in planar point sets: combinatorial structure and algorithms. *SIAM Journal on Computing*, 28(1):192–209, 1999.
- [15] F.-A. Fortin, S. Grenier, and M. Parizeau. Generalizing the improved run-time complexity algorithm for non-dominated sorting. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, pages 615–622. ACM, 2013.

- [16] A. Ghosh and S. Dehuri. Evolutionary algorithms for multi-criterion optimization: a survey. *International Journal of Computing & Information Sciences*, 2(1):38–57, 2004.
- [17] J. Hammersley. A few seedlings of research. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 345–394, 1972.
- [18] A. Hero and G. Fleury. Posterior Pareto front analysis for gene filtering. In *Proceedings of the Workshop on Genomic Signal Processing and Statistics (GENSIPS)*, 2002.
- [19] V. J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.
- [20] K.-J. Hsiao, J. Calder, and A. O. Hero. Pareto-depth for multiple-query image retrieval. *IEEE Transactions on Image Processing*, 24(2):583–594, 2015.
- [21] K.-J. Hsiao, K. Xu, J. Calder, and A. Hero. Multi-criteria anomaly detection using Pareto Depth Analysis. In *Advances in Neural Information Processing Systems 25*, pages 854–862. 2012.
- [22] K.-J. Hsiao, K. Xu, J. Calder, and A. O. Hero. Multi-criteria similarity-based anomaly detection using Pareto Depth Analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 27(6):1307–1321, 2016.
- [23] M. T. Jensen. Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms. *IEEE Transactions on Evolutionary Computation*, 7(5):503–515, 2003.
- [24] D. Kossmann, F. Ramsak, S. Rost, et al. Shooting stars in the sky: an online algorithm for skyline queries. In *Proceedings of the 28th International Conference on Very Large Data Bases*, pages 275–286, 2002.
- [25] R. Y. Liu, J. M. Parelius, and K. Singh. Multivariate analysis by data depth: descriptive statistics, graphics and inference. *The Annals of Statistics*, 27(3):783–858, 1999.
- [26] B. F. Logan and L. A. Shepp. A variational problem for random Young tableaux. *Advances in Mathematics*, 26(2):206–222, 1977.
- [27] B. Majecka. Statistical models of pedestrian behaviour in the forum. *Master’s thesis, School of Informatics, University of Edinburgh*, 2009.
- [28] C. Shi, M. Chen, and Z. Shi. A fast nondominated sorting algorithm. In *Neural Networks and Brain, 2005. ICNN&B’05. International Conference on*, volume 3, pages 1605–1610. IEEE, 2005.
- [29] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.
- [30] S. Ulam. Monte Carlo calculations in problems of mathematical physics. *Modern Mathematics for the Engineers*, pages 261–281, 1961.
- [31] A. Vershik and S. Kerov. Asymptotics of the Plancherel measure of the symmetric group and the limiting form of Young tables. *Soviet Doklady Mathematics*, 18(527-531):38, 1977.

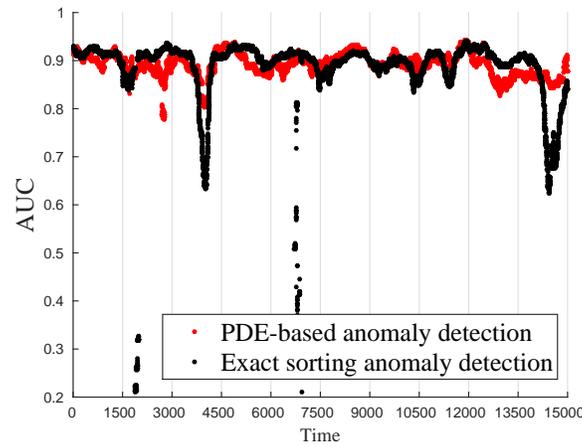
- [32] E. Zitzler, M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm, 2001.
- [33] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *evolutionary computation, IEEE transactions on*, 3(4):257–271, 1999.



(a)

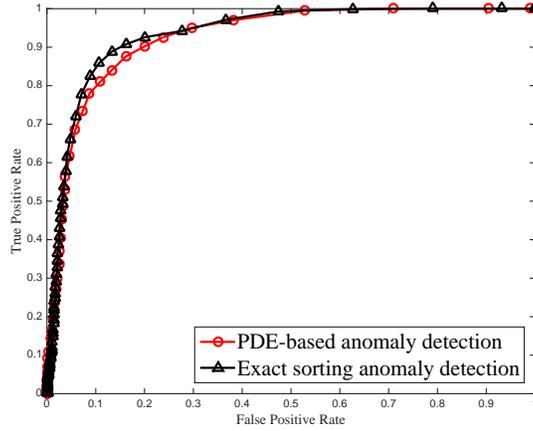


(b)



(c)

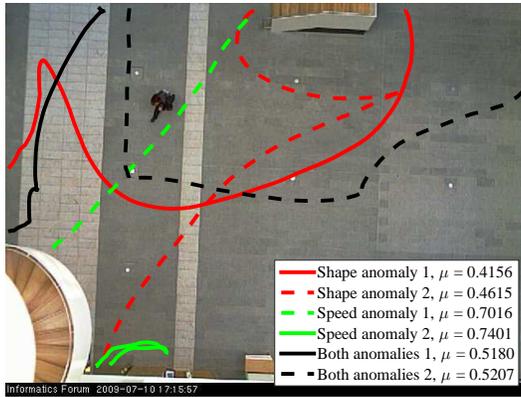
Figure 6: Results of simulated data stream with (a) uniformly distributed data and (b) categorical data. In (c) we show the AUCs for the pedestrian trajectories dataset.



(a)



(b)



(c)

Figure 7: Pedestrian experiment in a stationary setting. (a) ROC curves of PDE-based and exact sorting anomaly detection, (b) trajectories classified as normal, and (c) some anomalous trajectories with their classifications.