

Nazarbayev University

School of Engineering and Digital Sciences
Computer Science

Property Testing for Linear Separability: Theory, Implementation, and Empirical Evaluation

Merey Bissenbin

Supervisor: Meiram Murzabulatov

Astana, Kazakhstan

March, 2026

Abstract

Linear separability is a fundamental concept in machine learning that underlies many classification models and geometric decision procedures. While exact deterministic tests for linear separability are well studied and practical, they can become computationally expensive on large or high-dimensional datasets. In many practical settings, however, an exact answer is not necessary; it is often sufficient to determine whether data are linearly separable or far from separable.

Property testing provides a framework for this type of approximate decision-making by allowing algorithms to inspect only a small fraction of the input. Although property testers for linear separability have been studied theoretically and can achieve sublinear time complexity, their practical behavior and empirical relationship to exact methods remain relatively underexplored.

This thesis presents an empirical study of linear separability testing from a property testing perspective. It compares efficient deterministic algorithms with both two-sided and one-sided sublinear testers. In addition, it develops a practical LP-based implementation of a one-sided tester motivated by VC-dimension theory and investigates its empirical behavior across multiple data regimes.

The evaluation includes both synthetic and real-world datasets. For two-dimensional synthetic data, we construct instances with exactly controlled distance to linear separability. In higher dimensions, we generate non-separable instances by perturbing an ε -fraction of points from an initially separable dataset, which guarantees that the resulting dataset is at most ε -far from separable, although the exact distance may be smaller. This setting differs from the exact formulation of a property testing problem but provides controlled empirical evaluation.

Experimental results show that sublinear property testers can achieve high accuracy while requiring substantially less computation time than exact baselines on larger datasets, especially when instances are clearly separable or clearly nonseparable. Their reliability decreases when the data are very close to separable, particularly at separation distances below 0.01. The thesis identifies the conditions under which property-testing-based methods are useful as fast pre-screening tools and evaluates their empirical performance relative to theoretical guarantees.

Contents

Abstract	i
1 Introduction	1
1.1 Research Questions	2
1.2 Contributions	2
1.3 Organization of the Thesis	3
2 Background and Literature Review	3
2.1 Linear Separability	3
2.2 Deterministic Methods for Testing Separability	4
2.3 VC Dimension and PAC Learning	5
2.4 Property Testing	5
2.5 Testing Linear Separability Using LP-Type Testers	5
2.6 Empirical Context and Real Datasets	6
2.7 Gaps Filled by This Thesis	6
3 Problem Formulation and Preliminaries	7
3.1 Labeled Datasets and Halfspaces	7
3.2 Distance to Linear Separability	7
3.3 Property Testing Model	8
3.4 Why Sublinear Testing Is Meaningful Here	8
4 Deterministic and Sublinear Algorithms	8
4.1 Overview	8
4.2 Full LP Feasibility Baseline	9
4.3 Sphere / Minimum Covering Ball Baseline	9
4.4 Two-Sided Linear Separability Tester	9
4.5 One-Sided Linear Separability Tester	10
5 Sample Complexity Analysis of the One-Sided Tester	11
6 Experimental Methodology	12
6.1 Implementation and Platform	12
6.2 LP Solving and Numerical Tolerances	13
6.3 Compared Methods	13
6.4 Sampling Parameters	13
6.5 Synthetic Data Generation	13
6.6 Real-World Datasets	14
6.7 Evaluation Metrics	14

6.8	Sphere Algorithm Settings	15
7	Experimental Results	15
7.1	Experiment I: Exact Distance to Separability in Two Dimensions	15
7.2	Experiment II: High-Dimensional Synthetic Data	16
7.3	Experiment III: Real-World Benchmarks	19
7.4	Summary	22
8	Discussion	23
9	Conclusion and Future Work	24

List of Figures

1	Accuracy versus true distance for different ε values. The dashed vertical line in each plot corresponds to the chosen ε threshold.	16
2	Flip-noise runtime scaling at fixed $\varepsilon = 0.05$. Top row: runtime versus n for fixed dimensions. Bottom row: runtime versus d for fixed dataset sizes.	17
3	Feature-noise runtime scaling at fixed $\varepsilon = 0.05$. Top row: runtime versus n for fixed dimensions. Bottom row: runtime versus d for fixed dataset sizes.	18
4	Sphere-style synthetic generation runtime scaling at fixed $\varepsilon = 0.05$. Top row: runtime versus n for fixed dimensions. Bottom row: runtime versus d for fixed dataset sizes.	18

List of Tables

1	Selected results for Experiment I: accuracy versus true distance and ε for different dataset sizes.	15
2	Selected sign-flip results for Experiment II.	17
3	Selected feature-noise results for Experiment II.	17
4	Comparison of algorithms on CoverType class pairs for $\varepsilon = 0.5$ (sample size 1080).	19
5	Comparison of algorithms on CoverType class pairs for $\varepsilon = 0.05$ (sample size 10800).	20
6	Comparison of algorithms on MNIST one-vs-all tasks for $\varepsilon = 0.5$ (sample size 15680).	21
7	Comparison of algorithms on MNIST one-vs-all tasks for $\varepsilon = 0.2$ (sample size 39200).	21

1 Introduction

Linear separability—the existence of a hyperplane that separates two labeled classes—is a central concept in machine learning and computational geometry. It underlies widely used models such as the perceptron [1], support vector machines [2], and linear classifiers, and it plays an important role in exploratory data analysis, model selection, and feature engineering. Understanding whether data are linearly separable can provide valuable insight into the complexity of a classification task and the suitability of different learning models [3].

There have been numerous attempts to determine whether a dataset can be separated into two classes by a hyperplane. Classical deterministic methods view testing linear separability as a feasibility problem in computational geometry or linear programming. When all objects are considered simultaneously, such methods can determine with certainty whether the dataset is separable. Their reliability and theoretical foundations remain central to the field, but the increasing size and dimension of modern datasets can make even optimized exact methods costly. In many machine-learning workflows, an exact separability certificate is not always required at the earliest stage of analysis. A fast indication that the data are clearly separable, clearly far from separable, or too close to the boundary for sampling alone can already guide downstream choices. This motivates approximate decision-making methods that trade some reliability for much lower input access. Property testing gives a rigorous framework for such methods: the goal is to distinguish objects that satisfy a property from objects that are ε -far from satisfying it. From this perspective, testing linear separability can also be studied as a property testing problem.

Recent theoretical work has investigated this perspective by formulating testing linear separability as an LP-type problem and proposing sublinear testers with formal guarantees [4]. These guarantees are attractive, but they do not by themselves determine whether the testers are useful in numerical practice. In particular, it remains important to understand how sublinear testers compare empirically with deterministic solvers, how they behave near the separation threshold, and whether their theoretical savings translate into meaningful speedups on large realistic datasets.

The main purpose of this thesis is therefore to explore linear separability testing from a property testing point of view and to identify the circumstances under which sublinear testers can complement deterministic exact testing algorithms. The study considers both two-sided and one-sided property testers, compares them with deterministic baseline algorithms for testing linear separability, and evaluates a practical LP-based implementation of a one-sided tester motivated by VC-dimension theory.

The real use case emphasized in this thesis is fast pre-screening in large-scale machine-learning workflows. Before running a full LP feasibility test, a costly geometric separa-

bility routine, or a more expensive model-selection process, a practitioner may want to know whether a dataset appears clearly separable, clearly nonseparable, or close enough to the boundary that exact analysis is still needed. A sublinear tester is not intended to replace exact solvers in every case. Instead, it can act as an early diagnostic tool that reduces computation on easy large instances and flags difficult near-separable instances for more careful treatment.

The property testers are evaluated on several synthetic and real-world datasets. On two-dimensional synthetic datasets, the experiments use controlled distances to linear separability, allowing tester performance to be studied across different separability regimes. In higher dimensions, nonseparable datasets are produced by perturbing an ε -proportion of points from an initially separable dataset. As a consequence, the resulting dataset is at most ε -far from being separable, although it is not necessarily exactly ε -far. This construction is nevertheless consistent with the proximity model used in property testing and supports systematic evaluation at scale.

1.1 Research Questions

This thesis is guided by the following research questions:

- RQ1.** Under what conditions can property-testing-based methods serve as an effective alternative to exact deterministic algorithms for determining if a dataset can be separated by means of a hyperplane?
- RQ2.** How do two-sided and one-sided sublinear testers differ in terms of runtime and empirical reliability?
- RQ3.** What factors affect the performance of sublinear testers? In other words, how does the runtime/reliability trade-off change with the number of elements in the input dataset, dimension d , the value of the proximity parameter ε , and the degree of closeness to separability?
- RQ4.** In situations where violations are rare and difficult to find based upon random sampling alone (i.e., near-separable instances), how do randomized testers behave?

These four research questions motivate both the theoretical discussions and experiments described in later sections of this thesis.

1.2 Contributions

Contributions of this thesis include:

1. An empirical investigation of the testing of linear separability via property testing.

2. A comparison among efficient deterministic algorithms and both two-sided and one-sided sublinear testers on synthetic and real-world datasets.
3. Implementation and empirical assessment of a practical LP-based one-sided tester motivated by VC-dimension theory.
4. Analysis of how sublinear property testers change their performance as functions of varying data regimes, including near-separable instances, noisy datasets, and high-dimensional data regimes.
5. Identification of specific practical conditions under which sublinear testers are faster than efficient deterministic testing algorithms while remaining empirically reliable relative to exact full-data baselines.

1.3 Organization of the Thesis

The rest of this thesis is outlined below. [Section 2](#) reviews background information related to linear separability, deterministic testing algorithms for linear separability, VC-dimension theory, and property testing. [Section 3](#) defines the formal problems involved in this thesis along with relevant notation. [Section 4](#) identifies and studies deterministic and sampling-based algorithms for solving the problems defined in [Section 3](#). [Section 5](#) develops the theoretical foundation for why a specific one-sided tester should work. [Section 6](#) specifies experimental methodologies, including datasets and implementation details. [Section 7](#) reports empirical results obtained on synthetic and real-world datasets. [Section 8](#) discusses implications and limitations associated with observed trade-offs. Finally, [Section 9](#) provides conclusions to this thesis along with suggestions for further research.

2 Background and Literature Review

2.1 Linear Separability

Linear separability asks whether two subsets $A, B \subset \mathbb{R}^d$ can be separated by a single hyperplane. Mathematically, A and B are linearly separable when there exist (w, b) , with $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$, such that

$$w^\top x + b > 0 \quad \forall x \in A, \quad w^\top x + b < 0 \quad \forall x \in B.$$

This geometric property underlies traditional linear classifiers, including the perceptron and support vector machines [1, 2], and remains a useful diagnostic for classification complexity [3]. A linearly separable dataset suggests that a simple linear decision rule

may be sufficient, while a nonseparable or nearly separable dataset may indicate label noise, feature overlap, insufficient representation, or the need for a nonlinear model. Recent studies also use separability to interpret learned representations in deep networks [5], and related questions arise in kernel-induced feature spaces [6].

The same question can be viewed through convex geometry. By the hyperplane separation theorem, two finite labeled point sets are linearly separable if and only if their convex hulls do not intersect. This equivalence connects separability testing to convex-hull computation, intersection testing, linear programming, and other geometric optimization problems. It is powerful in low dimension, but it also illustrates why exact separability can become expensive: convex-hull and full-feasibility formulations tend to use the entire dataset.

2.2 Deterministic Methods for Testing Separability

Exact methods for testing linear separability are usually based on either computational geometry or linear programming. In low-dimensional settings, one can compute convex hulls and test whether the two hulls intersect. For points in $\mathbb{R}^2$, Andrew’s monotone chain algorithm gives a simple and efficient convex-hull construction [7], and specialized two-dimensional procedures can approach near-linear time [8]. These geometric approaches are attractive when dimension is small and the structure of the input is favorable, but their direct use becomes less appealing as d grows.

The most general exact formulation is linear programming feasibility. Given labeled data (x_i, y_i) , with $y_i \in \{\pm 1\}$, one seeks (w, b) satisfying

$$y_i(w^\top x_i + b) \geq 1 \quad \text{for all } i.$$

If this system is feasible, the dataset is linearly separable; otherwise, it is not. Linear programming is supported by mature theory and practical solvers, from the simplex tradition [9] to polynomial-time and interior-point methods [10–14]. This makes full LP feasibility a natural reference method in empirical work. Its limitation in the present setting is that it uses all n constraints, so runtime and memory may become significant for very large datasets or repeated exploratory analysis.

Several specialized deterministic alternatives have also been proposed. Elizondo surveys classical linear separability tests and highlights their different algorithmic assumptions [15]. Low-dimensional randomized LP algorithms, including the work of Megiddo, Seidel, and Clarkson, show that geometric feasibility can be solved very efficiently when d is fixed or small [16–18]. LP-type frameworks further abstract these ideas and explain why many geometric optimization problems admit randomized incremental algorithms [19]. Minimum enclosing ball methods, beginning with geometric location work and Welzl’s randomized smallest enclosing ball algorithm [20, 21], motivate sphere or

minimum-covering-ball approaches to separability [22]. These methods are important baselines, but they are not automatically superior in the empirical regime studied here: some depend strongly on dimension, some require careful numerical treatment, and most still process the full dataset.

2.3 VC Dimension and PAC Learning

The VC dimension of the family of all halfspaces in $\mathbb{R}^d$ equals $d + 1$ [23, 24]. Through the Sauer–Shelah lemma [25, 26], this finite combinatorial dimension bounds the number of distinct labelings that halfspaces can induce on a finite sample. Standard PAC-learning results then imply that the number of observations needed for small error and high confidence scales with d and the desired error parameters [27].

For this thesis, VC theory is relevant not because the goal is to train a classifier, but because it explains when a sample can reveal whether any halfspace is consistent with the data. If every halfspace makes error at least ε , then a sufficiently large random sample is unlikely to appear perfectly separable. This connection supports the one-sided tester analyzed later. At the same time, the theory is worst-case: it does not predict exact runtime, numerical stability, or the sample size that works best for a particular dataset.

2.4 Property Testing

Property testing provides a framework for approximate decisions using sublinear access to the input [28–30]. Unlike classical exact algorithms, property testers distinguish objects that satisfy a property from objects that are ε -far from satisfying it. This shift is especially relevant for exploratory data analysis: the tester may inspect only a small fraction of the dataset and still provide useful evidence about whether exact separability analysis is worth the cost.

Learning complexity often implies testing complexity for natural concept classes [29]. Related sampling ideas appear in geometric and visual property testing, where uniform samples can be powerful but also have clear limitations [31–33]. This is an important caution for linear separability: sampling works well when violations are common, but it can fail when nonseparability is caused by rare or localized violations.

2.5 Testing Linear Separability Using LP-Type Testers

Many geometric decision problems, including linear separability, can be treated as LP-type problems. This connection allows a randomized tester to solve a reduced LP on a sampled subset of constraints rather than solving the full feasibility problem. Epstein and Silwal’s LP-type property tester selects $O(d/\varepsilon)$ constraints, solves the induced reduced

problem, and then verifies the candidate solution on additional random constraints [4]. Its query complexity depends on d and ε , rather than directly on the full dataset size n .

The theoretical advantage is clear: when n is very large and $d/\varepsilon \ll n$, a sublinear tester may examine far fewer constraints than an exact solver. The practical comparison is more subtle. The two-sided tester can reject after finding an infeasible sampled problem or a violated verification constraint, but it may also be affected by numerical instability in sampled LPs. A one-sided tester that only checks sampled feasibility preserves perfect completeness for separable data, because every subset of a separable dataset is separable; however, it may accept a nonseparable dataset if the sample misses rare violations. Thus, the central empirical question is not whether exact methods are theoretically sound, but when a sampling-based method is reliable enough to be useful as a pre-screening tool.

2.6 Empirical Context and Real Datasets

Empirical evaluation is necessary because the observed value of a tester depends on dataset size, dimension d , distance from separability, and solver overhead. MNIST provides a high-dimensional benchmark for one-versus-all digit tasks [34], while CoverType provides large tabular class-pair tasks from cartographic variables [35]. These datasets are not used here to claim that linear separators are the best classifiers for the tasks. Rather, they provide realistic settings in which the cost of exact separability testing can be compared with the cost and reliability of sublinear tests.

2.7 Gaps Filled by This Thesis

The literature contains strong foundations for linear separability, linear programming, VC theory, LP-type optimization, and property testing. What remains less developed is an empirical comparison that places sublinear separability testers beside practical deterministic baselines. More specifically, there are few empirical studies regarding:

- runtime/reliability trade-offs when using high-dimensional and noisy datasets,
- how the value of the proximity parameter ε affects actual performance,
- behavior of one-sided and two-sided testers around the threshold of separability, and
- usefulness of LP-type testing as a fast pre-screening step in machine-learning workflows.

This thesis addresses that gap through a joint theoretical and experimental study. Its goal is to clarify when property-testing-based approaches are practical for detecting linear separability, when exact solvers remain necessary, and how one-sided and two-sided testers differ in empirical behavior.

3 Problem Formulation and Preliminaries

3.1 Labeled Datasets and Halfspaces

We consider a finite labeled dataset

$$S = \{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \{\pm 1\}.$$

Let $A = \{x_i : y_i = +1\}$ and $B = \{x_i : y_i = -1\}$. We denote by $\mathcal{H}$ the family of all linear separators (halfspaces), where each $h \in \mathcal{H}$ has the form

$$h(x) = \text{sign}(w^\top x + b)$$

for some $(w, b) \in \mathbb{R}^d \times \mathbb{R}$.

Definition 3.1 (Linear separability). *The dataset S is linearly separable if there exists $h \in \mathcal{H}$ such that $h(x_i) = y_i$ for every $i \in \{1, \dots, n\}$.*

Equivalently, linear separability can be written as the feasibility of the system

$$y_i(w^\top x_i + b) \geq 1, \quad i = 1, \dots, n.$$

The margin value 1 is simply a normalization, since any strictly separating hyperplane can be rescaled.

3.2 Distance to Linear Separability

To formulate property testing, we need a notion of distance from separability.

Definition 3.2 (Empirical error of a halfspace). *For $h \in \mathcal{H}$, define its empirical error on S by*

$$\text{err}_S(h) = \Pr_{(x,y) \sim S} [h(x) \neq y] = \frac{1}{n} |\{i : h(x_i) \neq y_i\}|,$$

where the probability is with respect to the uniform distribution on the dataset.

Definition 3.3 (ε -far from separable). *The dataset S is ε -far from linearly separable if $\text{err}_S(h) \geq \varepsilon$ for every $h \in \mathcal{H}$.*

In two dimensions, one can sometimes compute the exact distance to separability by enumerating candidate separating lines supported by data points. In higher dimensions, however, exact computation becomes substantially more difficult, and practical studies typically rely on controlled data generators or surrogate estimates.

3.3 Property Testing Model

In the property testing framework, the goal is not to determine exact distance but to distinguish:

- datasets that are linearly separable, and
- datasets that are ε -far from linearly separable.

A tester is allowed only limited access to the input, typically through random queries to the dataset.

Definition 3.4 ((ε, δ) tester). *A randomized algorithm is an (ε, δ) property tester for linear separability if:*

- (i) *it accepts every linearly separable dataset with probability at least $1 - \delta$, and*
- (ii) *it rejects every dataset that is ε -far from linearly separable with probability at least $1 - \delta$.*

If the tester always accepts separable datasets, it is said to have one-sided error. Otherwise, if both types of mistakes are possible, it has two-sided error.

3.4 Why Sublinear Testing Is Meaningful Here

The attraction of property testing is that the complexity of the tester may depend on d and ε rather than on the full dataset size n . For large datasets, this creates the possibility of substantial computational savings. In exploratory machine learning pipelines, such savings may be more important than obtaining an exact decision on every instance.

4 Deterministic and Sublinear Algorithms

4.1 Overview

We compare four methods:

1. a full LP feasibility test over all constraints,
2. a deterministic sphere / minimum covering ball (MCB) algorithm,
3. a two-sided linear feasibility tester following the LP-type property testing paradigm, and
4. a one-sided tester based on random sampling and LP feasibility on the sample.

4.2 Full LP Feasibility Baseline

The most direct exact baseline solves the feasibility system

$$y_i(w^\top x_i + b) \geq 1, \quad i = 1, \dots, n$$

on the full dataset. If the linear program is feasible, the dataset is declared linearly separable; otherwise, it is not. This method serves as the reference decision procedure throughout the experimental study whenever full-data LP remains computationally feasible.

4.3 Sphere / Minimum Covering Ball Baseline

A second deterministic baseline is the sphere or MCB-based separability tester [22]. Its geometric idea is to transform the problem into one involving the minimum covering ball of appropriately defined difference vectors. Intuitively, if the transformed set encloses the origin in a certain way, then the original classes are not linearly separable.

Algorithm 1: Sphere-Model Linear Separability Tester (MCB-Based)

Input : Point sets $A, B \subset \mathbb{R}^d$.

Output : **LS** or **NLS**.

- 1 Compute the geometric difference set $C \leftarrow A \ominus B := \{a - b : a \in A, b \in B\}$
 - 2 Construct a model sphere centered at the origin with radius r_0
 - 3 Project each point of C onto the model sphere along the ray from the origin
 - 4 Compute the minimum covering ball of the projected point set
 - 5 **if** *the resulting radius is strictly smaller than r_0* **then**
 - 6 | output **LS**
 - 7 **else**
 - 8 | output **NLS**
 - 9 **end**
-

4.4 Two-Sided Linear Separability Tester

The two-sided tester follows the LP-type property testing paradigm [4]. It samples a subset of constraints, solves the induced reduced LP, and verifies the candidate solution on additional random samples.

Algorithm 2: Linear Feasibility Tester (Two-Sided)

Input : Dimension d , proximity parameter ε , query access to constraint set S .

Output : ACCEPT or REJECT.

```
1  $r \leftarrow \lceil 10d/\varepsilon \rceil$ 
2 Sample a set  $R$  of  $r$  constraints uniformly at random from  $S$ 
3 Form the reduced feasibility LP induced by  $R$  and solve it
4 if the reduced LP is infeasible then
5   | output REJECT
6 end
7 Obtain a candidate solution  $(w, b)$  from the reduced LP
8  $T \leftarrow \lceil 2/\varepsilon \rceil$ 
9 for  $t \leftarrow 1$  to  $T$  do
10  | draw a fresh random constraint from  $S$ 
11  | if the candidate solution violates that constraint then
12  |   | output REJECT
13  | end
14 end
15 output ACCEPT
```

This tester may make both kinds of mistakes: it can occasionally reject a separable dataset due to random sampling or numerical effects, and it can occasionally accept a nonseparable dataset that is not sufficiently far from separable.

4.5 One-Sided Linear Separability Tester

The one-sided tester modifies the LP-type approach in a simple way: instead of verifying a candidate separator on additional samples, it only checks whether a random subset is linearly separable. This preserves perfect completeness, because every subset of a linearly separable dataset is itself linearly separable.

Algorithm 3: One-Sided Linear Feasibility Tester

Input : Dimension d , proximity parameter ε , query access to constraints of S .

Output : ACCEPT or REJECT.

```
1  $r \leftarrow \lceil 10d/\varepsilon \rceil$ 
2 Draw a uniform random sample  $R$  of size  $r$  from  $S$ 
3 Construct the feasibility LP: find  $(w, b)$  such that  $y_i(w^\top x_i + b) \geq 1$  for all sampled
  points
4 if the sampled LP is feasible then
5   | output ACCEPT
6 else
7   | output REJECT
8 end
```

This tester never rejects a truly separable dataset, but it may accept a nonseparable

dataset if the sample happens to miss the witness of nonseparability.

5 Sample Complexity Analysis of the One-Sided Tester

Let $S = \{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \{\pm 1\}$ be a labeled dataset, and let $\mathcal{H}$ denote the set of all halfspaces in $\mathbb{R}^d$. The tester from [Algorithm 3](#) samples r points uniformly at random and accepts if and only if the sampled subset is linearly separable. The following proposition is a standard VC-dimension consequence, stated here to make the One-Sided Tester self-contained.

Proposition 5.1 (VC-based one-shot tester for linear separability). *There exists an absolute constant $c > 0$ such that, if*

$$r \geq \frac{c}{\varepsilon} \left((d+1) \ln\left(\frac{1}{\varepsilon}\right) + \ln\left(\frac{1}{\delta}\right) \right), \quad (1)$$

then the one-sided tester is an (ε, δ) property tester for linear separability. In particular:

- **Completeness:** *if S is linearly separable, the tester accepts with probability 1.*
- **Soundness:** *if S is ε -far from linearly separable, the tester rejects with probability at least $1 - \delta$.*

A convenient explicit choice for $\delta = 1/3$ is

$$r \geq \frac{2}{\varepsilon} \left[(d+1) \ln\left(\frac{4e}{\varepsilon}\right) + \ln 3 \right]. \quad (2)$$

Proof. We prove completeness and soundness separately.

Completeness. If S is linearly separable, then there exists some halfspace $h \in \mathcal{H}$ that classifies every point in S correctly. Every subset of S is therefore also separable by the same h . Hence the tester always accepts.

Soundness. Assume that S is ε -far from linearly separable. Then every halfspace $h \in \mathcal{H}$ has empirical error at least ε on S .

Fix a halfspace $h \in \mathcal{H}$. A single uniformly random point from S is classified correctly by h with probability at most $1 - \varepsilon$. Therefore the probability that h makes zero errors on a random sample of size r is at most

$$(1 - \varepsilon)^r \leq e^{-\varepsilon r}.$$

Now we cannot union bound directly over all halfspaces because there are infinitely many. Instead, for any fixed sample of size r , we consider only the distinct labelings

induced on that sample by $\mathcal{H}$. Let $\Pi_{\mathcal{H}}(r)$ denote the growth function of halfspaces on samples of size r . Since the VC dimension of halfspaces in $\mathbb{R}^d$ is $d + 1$, the Sauer–Shelah lemma gives

$$\Pi_{\mathcal{H}}(r) \leq \sum_{i=0}^{d+1} \binom{r}{i} \leq \left(\frac{er}{d+1} \right)^{d+1}.$$

Applying the union bound over the realizable labelings on the sample yields

$$\Pr[\text{tester accepts}] \leq \Pi_{\mathcal{H}}(r) e^{-\varepsilon r} \leq \left(\frac{er}{d+1} \right)^{d+1} e^{-\varepsilon r}.$$

Thus it suffices to choose r such that the right-hand side is at most δ . A standard rearrangement yields the condition stated in (1), and the explicit choice in (2) is a convenient sufficient bound when $\delta = 1/3$. $\square$

Remark 5.2. *The same bound applies whether sampling is performed with or without replacement. Sampling without replacement can only decrease the chance of drawing an entirely error-free sample for a bad classifier.*

Remark 5.3. *The sample complexity scales as*

$$r = \Theta\left(\frac{1}{\varepsilon} \left(d \log(1/\varepsilon) + \log(1/\delta) \right)\right).$$

The extra logarithmic factor in $1/\varepsilon$ arises from the growth-function bound over realizable labelings.

The proposition records the standard VC-based guarantee behind the one-sided tester. In practice, the implementation used in this thesis adopts the simpler heuristic sample size $r = \lceil 10d/\varepsilon \rceil$, which is considerably smaller than the worst-case theoretical upper bound but performed well empirically in the experiments. This gap between worst-case theory and observed behavior is one of the empirical observations studied in the thesis.

6 Experimental Methodology

6.1 Implementation and Platform

All experiments are implemented in MATLAB (R2025a) and executed on a machine equipped with an AMD Ryzen 9 5900X CPU (12 cores / 24 threads), 32 GB RAM, and an NVIDIA RTX 3080 GPU with 12 GB memory, running Ubuntu 24.04.3 LTS. The GPU is not required by the algorithms themselves, but it is part of the experimental workstation used for the study.

6.2 LP Solving and Numerical Tolerances

All linear programs are solved using MATLAB `linprog`. Because separability decisions can be numerically sensitive near the boundary between feasible and infeasible instances, we keep solver tolerances fixed across all LP-based methods. This makes the comparison between algorithms as fair and reproducible as possible.

6.3 Compared Methods

We compare the following methods:

- **Full LP feasibility:** exact reference method on the full dataset.
- **Sphere / MCB algorithm:** deterministic geometric baseline.
- **Two-sided linear feasibility tester:** randomized LP-type tester with verification.
- **One-sided tester:** randomized tester based only on sampled LP feasibility.

For the sampling-based methods, each configuration is repeated over $R = 100$ independent trials unless stated otherwise. We report average per-trial runtime and empirical agreement with the full-LP decision.

6.4 Sampling Parameters

Unless stated otherwise, the sampling-based testers use

$$r = \left\lceil \frac{10d}{\varepsilon} \right\rceil$$

as the primary sample size. Epstein and Silwal’s work provides these sample sizes [4]. The two-sided tester additionally uses a verification sample size

$$T = \left\lceil \frac{2}{\varepsilon} \right\rceil.$$

We evaluate multiple values of ε , typically drawn from

$$\varepsilon \in \{0.5, 0.4, 0.3, 0.2, 0.1, 0.05\}.$$

6.5 Synthetic Data Generation

For low-dimensional experiments, we generate labeled point sets in $\mathbb{R}^2$ and compute the exact distance to linear separability. Starting from a linearly separable configuration, we introduce controlled perturbations such as label flips. Because the dimension is low,

candidate separating lines can be enumerated to determine the minimum number of labels that must be changed to restore separability.

We generate points x_i in $\mathbb{R}^d$, sample a random separating hyperplane, assign clean labels accordingly, and then independently flip each label with a prescribed probability. This creates nonseparable datasets whose difficulty can be controlled through the noise level.

We also consider feature perturbations in which a clean, initially separable dataset is modified via additive Gaussian noise,

$$x_i \leftarrow x_i + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \sigma^2 I_d),$$

while labels are kept fixed. This creates subtle violations of separability that are often harder to detect from small samples.

A third synthetic generator follows the style of the sphere / overlap construction used in prior deterministic separability work [22]. In this setting, overlap is induced geometrically, but the exact distance to separability is not directly controlled.

6.6 Real-World Datasets

We use selected class pairs from the CoverType dataset to build binary classification tasks [35]. For a chosen pair of classes (a, b) , we retain only points from those two classes and relabel them as -1 and $+1$.

We also evaluate on MNIST one-vs-all tasks [34]. For each digit $k \in \{0, \dots, 9\}$, we form a binary dataset in which digit k is the positive class and all other digits form the negative class.

6.7 Evaluation Metrics

We report the following metrics:

- **Runtime:** average per-trial runtime for randomized methods; single execution time for deterministic methods.
- **Agreement:** fraction of randomized trials whose decision matches the full-LP baseline.
- **Distance surrogate:** for real-world datasets where exact distance is unknown, we use a soft-margin linear SVM over a grid of regularization values to obtain a practical upper bound on the empirical distance to separability.

6.8 Sphere Algorithm Settings

For the sphere / MCB tester, we use the reference implementation from the original authors and fix three numerical tolerances throughout the experiments:

$$\text{epsTol} = 10^{-4}, \quad \text{epsTol2} = 10^{-7}, \quad \text{epsTol3} = 10^{-6}.$$

These values were chosen after preliminary runs to balance stability and runtime. All sphere / MCB experiments use single precision in order to support the largest tested instances.

7 Experimental Results

7.1 Experiment I: Exact Distance to Separability in Two Dimensions

The first experiment evaluates the empirical behavior of the randomized testers when the true distance to separability is known exactly. This allows us to examine how the choice of ε affects performance near the separability boundary.

Size	True distance	ε	Sample size	One-Sided Accuracy (%)	Two-Sided Accuracy (%)
10000	0.0096	0.5	40	26	50
10000	0.0096	0.25	80	55	65
10000	0.0096	0.05	400	98	98
20000	0.00425	0.5	40	11	25
20000	0.00425	0.25	80	27	39
20000	0.00425	0.05	400	83	86
50000	0.01016	0.5	40	28	30
50000	0.01016	0.25	80	53	60
50000	0.01016	0.05	400	96	99
100000	0.00611	0.5	40	24	38
100000	0.00611	0.25	80	49	51
100000	0.00611	0.05	400	93	97

Table 1: Selected results for Experiment I: accuracy versus true distance and ε for different dataset sizes.

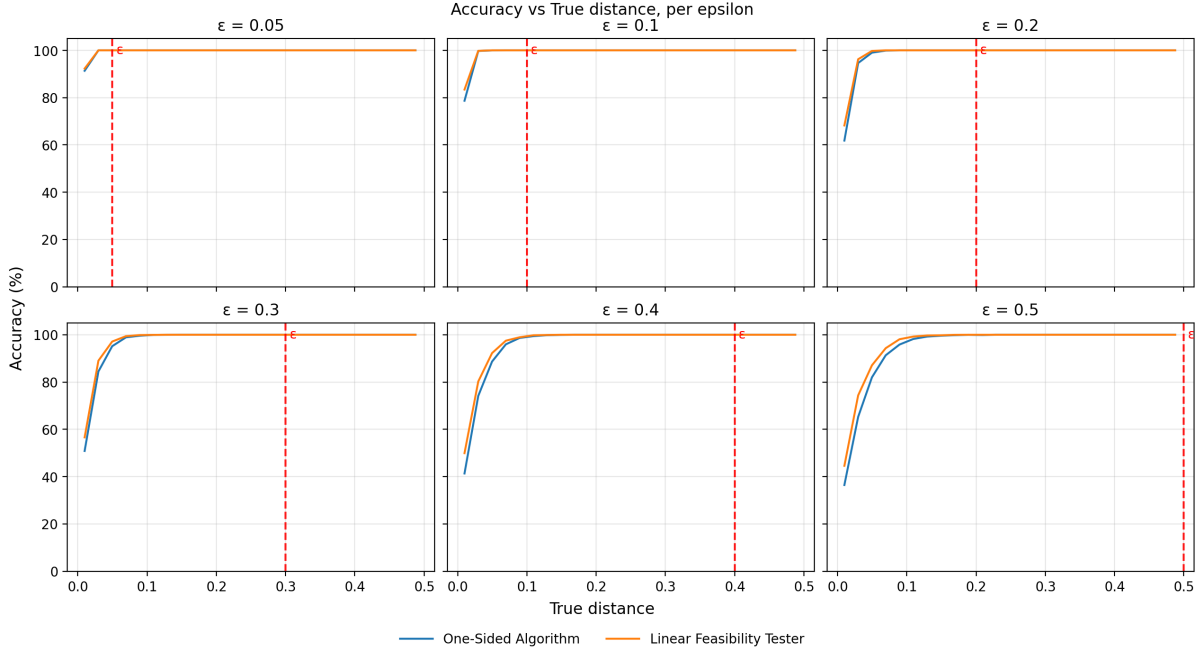


Figure 1: Accuracy versus true distance for different ϵ values. The dashed vertical line in each plot corresponds to the chosen ϵ threshold.

Interpretation. A clear trend appears in these results: as ϵ decreases and the sample size increases, the accuracy of the property tester improves. This effect is especially visible in near-separable cases, where a violation is rare enough to be missed by smaller samples. In practice, the testers perform better than the worst-case theoretical perspective might suggest. For example, when the measured distance is about 0.00425 for $n = 20,000$, the one-sided tester raises its detection accuracy from 11% to 83% when ϵ decreases from 0.5 to 0.05. The two-sided tester followed a similar pattern.

These trends follow directly from the property-testing viewpoint: when ϵ is large, the tester inspects a small subset of all constraints and can miss evidence of nonseparability. As ϵ decreases, the sampled set grows, improving reliability.

7.2 Experiment II: High-Dimensional Synthetic Data

The second experiment evaluates scalability and robustness as both dimension and dataset size increase. We compare deterministic baselines and randomized testers under sign-flip noise, feature noise, and sphere-style synthetic generation.

Dimension	Size	Alpha	ϵ	One-Sided Accuracy (%)	Two-Sided Accuracy (%)
10	10000	0.99	0.5	74	77
10	10000	0.99	0.2	98	98
10	10000	0.99	0.05	100	100
40	20000	0.99	0.5	100	100
40	20000	0.99	0.2	100	100
40	20000	0.99	0.05	100	100

Table 2: Selected sign-flip results for Experiment II.

Dimension	Size	Sigma	ϵ	One-Sided Accuracy (%)	Two-Sided Accuracy (%)
10	10000	0.01	0.5	0	28
10	10000	0.01	0.2	0	25
10	10000	0.01	0.05	91	94
40	20000	0.01	0.5	0	13
40	20000	0.01	0.2	0	20
40	20000	0.01	0.05	74	79

Table 3: Selected feature-noise results for Experiment II.

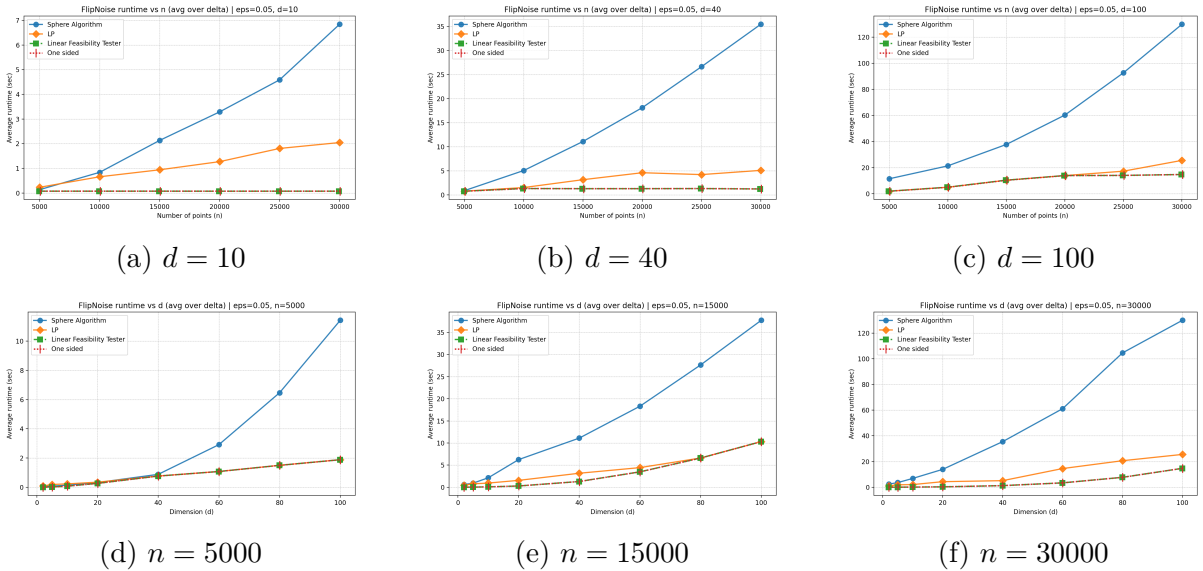
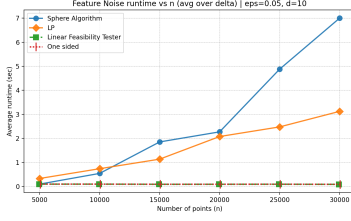
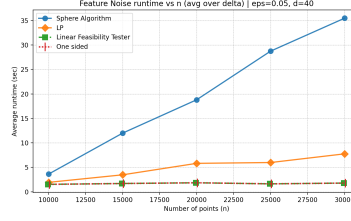


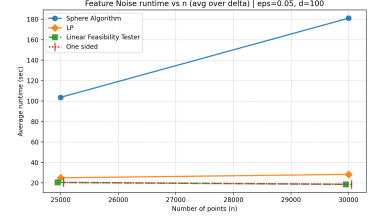
Figure 2: Flip-noise runtime scaling at fixed $\epsilon = 0.05$. Top row: runtime versus n for fixed dimensions. Bottom row: runtime versus d for fixed dataset sizes.



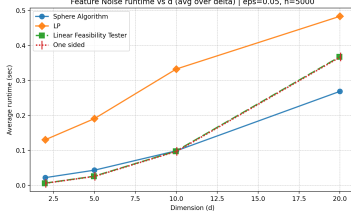
(a) $d = 10$



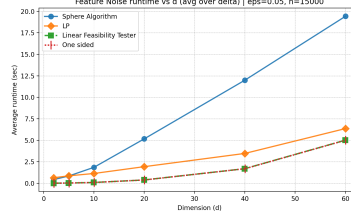
(b) $d = 40$



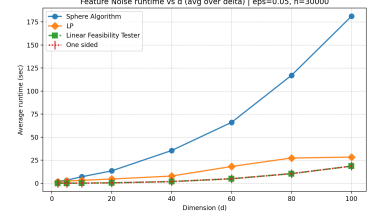
(c) $d = 100$



(d) $n = 5000$

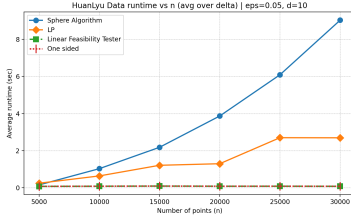


(e) $n = 15000$

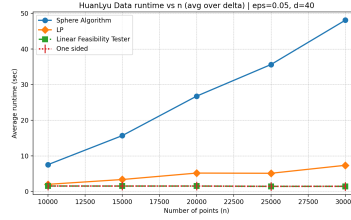


(f) $n = 30000$

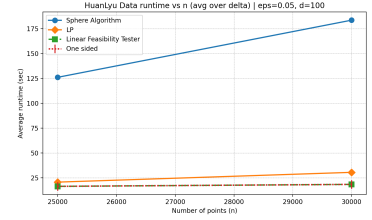
Figure 3: Feature-noise runtime scaling at fixed $\varepsilon = 0.05$. Top row: runtime versus n for fixed dimensions. Bottom row: runtime versus d for fixed dataset sizes.



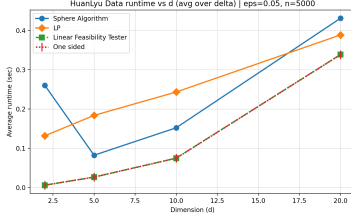
(a) $d = 10$



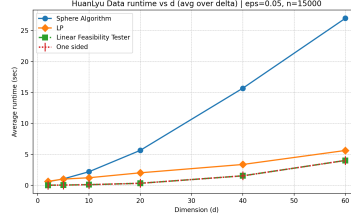
(b) $d = 40$



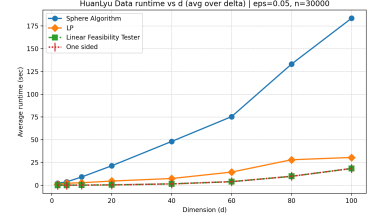
(c) $d = 100$



(d) $n = 5000$



(e) $n = 15000$



(f) $n = 30000$

Figure 4: Sphere-style synthetic generation runtime scaling at fixed $\varepsilon = 0.05$. Top row: runtime versus n for fixed dimensions. Bottom row: runtime versus d for fixed dataset sizes.

Interpretation. Across all synthetic datasets, the randomized testers show weak dependence on the total number of rows n in the full dataset. Each tester examines only a sample of $O(d/\varepsilon)$ constraints. Deterministic baselines, however, use the entire dataset. Therefore, these methods become more costly as either n or d grows. Property tester runtime for each testing round grows with dimension d , due to the increase in sample size and the additional computational costs associated with creating and solving smaller LPs.

Results from the figures and tables indicate that LP-based property testers are generally $3\text{--}5\times$ faster than deterministic methods for $\varepsilon = 0.05$. At that sample size, they also achieve the highest accuracy levels (79%–100%). The experimental results suggest that feature noise is more difficult than flip noise because small samples often do not detect subtle violations introduced by Gaussian perturbation. However, when ε is reduced to 0.05, performance improves substantially. This highlights an important practical point: the harder it is to observe evidence of nonseparability, the more conservative the choice of ε should be. Some parameter configurations are omitted because the required sample size exceeds the full dataset size, in which case the sublinear testing regime no longer applies.

7.3 Experiment III: Real-World Benchmarks

The third experiment evaluates the testers on real-world data. MNIST is a collection of handwritten digit images, while CoverType is a tabular dataset. As in the high-dimensional synthetic experiment, the true distance to separability cannot be computed efficiently, so it is approximated using a soft-margin SVM surrogate.

Class Pair	Size	One-Sided Tester		Two-Sided Tester		Full LP	Sphere Algorithm	Distance (Upper Bound)
		Time	Accuracy	Time	Accuracy			
3 vs 4	38501	0.0224	100	0.0224	100	1.776	6.886	0.050103
3 vs 5	45247	0.0149	100	0.0149	100	1.330	22.647	0.021791
3 vs 6	53121	0.0445	100	0.0445	100	6.970	50.112	0.245195
3 vs 7	56264	0.0050	100	0.0059	93	0.346	78.692	0.000000
7 vs 4	23257	0.0022	100	0.0026	100	0.044	3.132	0.000000
7 vs 5	30003	0.0097	0	0.0103	13	0.937	19.110	0.000233
7 vs 6	37877	0.0040	100	0.0046	94	0.159	33.911	0.000000
5 vs 6	26860	0.0199	100	0.0199	100	0.661	9.039	0.036895
4 vs 6	20114	0.0237	100	0.0237	100	0.824	3.243	0.049170

Table 4: Comparison of algorithms on CoverType class pairs for $\varepsilon = 0.5$ (sample size 1080).

Class Pair	Size	One-Sided Tester		Two-Sided Tester		Full LP Time	Sphere Algorithm Time	Distance (Upper Bound)
		Time	Accuracy	Time	Accuracy			
3 vs 4	38501	0.3815	100	0.3815	100	1.776	6.886	0.050103
3 vs 5	45247	0.2366	100	0.2366	100	1.330	22.647	0.021791
3 vs 6	53121	0.6507	100	0.6507	100	6.970	50.112	0.245195
3 vs 7	56264	0.0630	100	0.0640	88	0.346	78.692	0.000000
7 vs 4	23257	0.0192	100	0.0199	100	0.044	3.132	0.000000
7 vs 5	30003	0.3133	75	0.3135	78	0.937	19.110	0.000233
7 vs 6	37877	0.0395	100	0.0404	89	0.159	33.911	0.000000
5 vs 6	26860	0.2613	100	0.2613	100	0.661	9.039	0.036895
4 vs 6	20114	0.4328	100	0.4328	100	0.824	3.243	0.049170

Table 5: Comparison of algorithms on CoverType class pairs for $\varepsilon = 0.05$ (sample size 10800).

There is an apparent relationship between the surrogate distance and tester reliability found within the CoverType results. There are nine class pairs which can be grouped into three main groups.

The first category includes class pairs with a moderate to high surrogate distance upper bound: 3 vs 6, 3 vs 4, 4 vs 6, 5 vs 6, and 3 vs 5. Both testers classified each of these pairs perfectly at both $\varepsilon = 0.5$ and $\varepsilon = 0.05$. Since the violation sets are large enough that even a small sample contains many violations, the randomized testers obtain a significant time advantage. The one-sided tester took only around a few hundredths of a second, whereas the full LP took from a few tenths of a second to several seconds depending on the class pair. The sphere algorithm was slower than the full LP in these cases.

The second category includes linearly separable pairs: 3 vs 7, 7 vs 4, and 7 vs 6. The full LP confirmed that each of these pairs is linearly separable, and their surrogate distance upper bound is zero. As expected, the one-sided tester accepted each of these pairs correctly at both ε values. However, the two-sided tester falsely rejected some of these pairs, especially 3 vs 7 and 7 vs 6. This behavior is consistent with numerical instability in the sampled LP subproblems and shows the practical advantage of one-sided completeness on separable data.

The third category represents the most difficult case. That is, the surrogate distance between the classes 7 and 5 is very small. Although the full LP was correct in accepting all instances of this pair, neither the one-sided nor the two-sided tester agreed with the full LP at $\varepsilon = 0.5$. Although they improved marginally at $\varepsilon = 0.05$, this pair remained a challenge for both testers. The reason is that violations are extremely rare, so random

samples often miss them entirely.

Overall, the randomized testers are faster than the deterministic baselines on the CoverType dataset. Depending on the class pair, the speedup relative to the full LP ranges from modest to very large.

Digit	Size	One-Sided Tester		Two-Sided Tester		Full LP	Sphere Algorithm	Distance (Upper Bound)
		Time	Accuracy	Time	Accuracy			
0	70000	16.8342	0	16.8358	14	131.09	4478.17	0.0063714
1	70000	8.8476	0	8.8491	17	133.37	5083.46	0.0069
2	70000	21.4819	100	21.4819	100	87.52	4348.20	0.017071
3	70000	20.3018	100	20.3018	100	146.11	3991.75	0.020986
4	70000	37.4646	80	37.4649	83	83.41	4358.21	0.013786
5	70000	21.8749	100	21.8749	100	96.34	3715.42	0.020114
6	70000	17.1600	17	17.1614	26	65.97	4411.04	0.010871
7	70000	19.0062	90	19.0063	91	69.82	4432.01	0.013686
8	70000	19.7050	100	19.7050	100	100.56	3715.33	0.0364
9	70000	15.4688	100	15.4688	100	98.72	4113.27	0.032386

Table 6: Comparison of algorithms on MNIST one-vs-all tasks for $\varepsilon = 0.5$ (sample size 15680).

Digit	Size	One-Sided Tester		Two-Sided Tester		Full LP	Sphere Algorithm	Distance (Upper Bound)
		Time	Accuracy	Time	Accuracy			
0	70000	57.8655	100	57.8655	100	131.09	4478.17	0.0063714
1	70000	86.4109	100	86.4109	100	133.37	5083.46	0.0069
2	70000	59.6656	100	59.6656	100	87.52	4348.20	0.017071
3	70000	66.7923	100	66.7923	100	146.11	3991.75	0.020986
4	70000	57.8092	100	57.8092	100	83.41	4358.21	0.013786
5	70000	59.9925	100	59.9925	100	96.34	3715.42	0.020114
6	70000	32.2181	100	32.2181	100	65.97	4411.04	0.010871
7	70000	44.4905	100	44.4905	100	69.82	4432.01	0.013686
8	70000	57.1702	100	57.1702	100	100.56	3715.33	0.0364
9	70000	48.4336	100	48.4336	100	98.72	4113.27	0.032386

Table 7: Comparison of algorithms on MNIST one-vs-all tasks for $\varepsilon = 0.2$ (sample size 39200).

The MNIST tests examine the methods in a high-dimensional setting ($d = 784$ and

$n = 70,000$). Consistent with the nature of handwritten digit recognition, none of the one-versus-all problems is separable under the full linear programming formulation.

When ε is set to 0.5, classification quality varies substantially with the surrogate distance. Digits 8 and 9 have the largest distances among the ten digits, and both testers classify them correctly. Digits 2, 3, and 5 have the next largest distances and are also classified correctly by both testers. Their lack of separability is strong enough that samples with 15,680 points capture violations adequately.

Classification results decline for digits 4 and 7 because they have lower surrogate distances. The weakest results occur for digits 0, 1, and 6. Digits 0 and 1 have especially low surrogate distances; therefore the one-sided tester accepts these tasks in 100 percent of trials when $\varepsilon = 0.5$. The two-sided tester performs marginally better. These classification errors are consistent with rare violations: when the violating set is small, the sample may not contain enough evidence to prove nonseparability.

Reducing ε to 0.2 increases the sample size to 39,200, which is more than half of the MNIST dataset. For all ten tasks, both testers achieve 100 percent correct classifications. This indicates that the earlier poor classification rates were due to undersampling rather than a fundamental limitation of the testing procedures.

At $\varepsilon = 0.5$, the testers are faster than the full LP and orders of magnitude faster than the sphere algorithm. When ε decreases to 0.2, the runtime advantage becomes smaller because the sampled problem is closer in size to the original problem. This illustrates the primary trade-off: reducing ε improves reliability but decreases the efficiency benefits associated with sampling.

On MNIST, both the one-sided and two-sided testers exhibit similar running times and similarly high accuracy levels. The additional verification step added to the two-sided tester adds minimal overhead when considering the cost associated with solving a large-dimensional LP over sampled data.

7.4 Summary

Randomized testers are most effective when the data are clearly nonseparable or linearly separable. In those regimes, they often agree with the exact baseline while using much smaller sampled subproblems. These results appear in multiple CoverType pairs, several MNIST one-vs-all tasks (such as 2, 3, 5, 8, and 9), and the synthetic sign-flip experiments.

The main difficult regime is near-separability. Instances such as CoverType 7 vs 5 and MNIST digits 0 and 1 at $\varepsilon = 0.5$ show that random samples may miss rare violations. In these cases, reliable decisions require either a larger sample size, equivalently a smaller value of ε , an adaptive strategy for finding rare violations, or an exact solver.

In terms of efficiency, the randomized testers are frequently much faster than their

deterministic counterparts, especially on large datasets and when ε is not too small. An important practical feature of the one-sided tester is that it will never reject a perfectly separable dataset. On the other hand, the two-sided tester can sometimes reject a perfectly separable dataset due to numerical instability.

8 Discussion

The experimental evidence suggests that property testing works particularly well in large-scale settings where the data are either clearly separable or clearly inseparable. In those cases, a relatively small random sample often contains enough information to make a correct decision, resulting in significant runtime savings.

The primary failure mode is the same near-separable regime identified in the results. The test can accept a dataset that an exact full-data method rejects when the sampled subset misses the few points responsible for nonseparability. This failure mode is not merely an implementation issue; it is a direct consequence of the property-testing paradigm.

The one-sided tester has an important practical advantage over the two-sided tester: it guarantees that it never rejects a truly separable dataset. That makes it particularly suitable for pre-screening workflows where falsely rejecting a perfectly separable dataset would send an easy instance to a more expensive solver or suggest unnecessary model complexity. The experiments demonstrate that this guarantee does not come at a large empirical cost relative to the two-sided tester. In many situations both testers behave similarly, with sample size being the major factor determining performance.

The parameter ε should be viewed not only as a theoretical measure of how far from separability a dataset must be before the testers reject it, but also as an adjustable tuning parameter. Larger ε gives smaller samples and therefore quicker decisions, but at the expense of potentially lower reliability near the separability boundary. Smaller ε provides better accuracy but increases runtime. The experiments suggest that it is often reasonable to start with $\varepsilon \approx 0.1$ and reduce it when near-separable structure is expected.

This study has three major limitations. First, since the implementation depends on MATLAB and its built-in numerical linear programming solver, practical performance may vary across software environments and LP backends. Second, exact distance to separability is computed only in two dimensions; in higher dimensions, the experiments rely on controlled generators and surrogate measures. Third, some empirical conclusions depend on the sample heuristic $r = \lceil 10d/\varepsilon \rceil$, which performs reasonably well in practice but is not the only possible sampling rule.

9 Conclusion and Future Work

This thesis studied linear separability testing from a property testing perspective. We compared exact deterministic methods with both two-sided and one-sided sublinear testers, and we evaluated a practical LP-based one-sided tester motivated by VC-dimension theory.

The empirical results show that property-testing-based methods can provide a practical and scalable pre-screening approach for linear separability testing in many settings. On several high-dimensional and large-scale instances, the randomized testers achieve substantial speedups over exact baselines while maintaining high empirical accuracy. The results also show that theoretical worst-case sample bounds can be conservative: in practice, the simpler heuristic sample size $r = \lceil 10d/\varepsilon \rceil$ performs well across a wide range of the tested datasets.

From a practical perspective, the one-sided tester is especially attractive because it never rejects separable data. More generally, the study shows that sublinear testing is not merely of theoretical interest; it can serve as a fast pre-screening tool in exploratory machine-learning workflows before running full LP feasibility tests, sphere-based methods, or more expensive model-selection procedures. Near-separable datasets remain the main exception, and those cases should be handled with smaller ε , larger samples, adaptive sampling, or exact solvers.

Several directions remain open for future work. One promising direction is to design adaptive strategies for selecting ε based on observed data difficulty. Another is to extend the experimental study to additional datasets and other noise models. It would also be interesting to investigate alternative reduced-sample optimization procedures, different verification strategies, and implementations in environments beyond MATLAB. Finally, tighter empirical and theoretical connections between distance-to-separability estimation and property testing remain an important direction for further research.

References

- [1] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. doi: 10.1037/h0042519. URL <https://pubmed.ncbi.nlm.nih.gov/13602029/>.
- [2] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995. doi: 10.1007/BF00994018.
- [3] David A. Elizondo, Ralph Birkenhead, Matías Gómez, Noelia García Rubio, and Esteban Alfaro Cortés. Linear separability and classification complexity. *Expert Systems with Applications*, 39(9):7796–7807, 2012. doi: 10.1016/j.eswa.2012.01.090. URL <https://doi.org/10.1016/j.eswa.2012.01.090>.
- [4] Rogers Epstein and Sandeep Silwal. Property testing of lp-type problems. In *47th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 168 of *LIPIcs*, pages 98:1–98:18. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi: 10.4230/LIPIcs.ICALP.2020.98.
- [5] Chao Zhang, Xinyu Chen, Wensheng Li, Lixue Liu, Wei Wu, and Dacheng Tao. Understanding deep neural networks via linear separability of hidden layers, 2023. URL <https://arxiv.org/abs/2307.13962>.
- [6] Degang Chen, Qiang He, and Xizhao Wang. On linear separability of data sets in feature space. *Neurocomputing*, 70(13):2441–2448, 2007. ISSN 0925-2312. doi: 10.1016/j.neucom.2006.12.002. URL <https://www.sciencedirect.com/science/article/pii/S0925231206005200>. Selected papers from the 3rd International Conference on Development and Learning (ICDL 2004) Time series prediction competition: the CATS benchmark.
- [7] Alex M. Andrew. Another efficient algorithm for convex hulls in two dimensions. *Information Processing Letters*, 9(5):216–219, 1979. doi: 10.1016/0020-0190(79)90072-3. URL [https://doi.org/10.1016/0020-0190\(79\)90072-3](https://doi.org/10.1016/0020-0190(79)90072-3).
- [8] Sylvain Contassot-Vivier and David Elizondo. A near linear algorithm for testing linear separability in two dimensions. In Chrisina Jayne, Shigang Yue, and Lazaros Iliadis, editors, *Engineering Applications of Neural Networks*, volume 311 of *Communications in Computer and Information Science*, pages 114–124. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-32909-8. doi: 10.1007/978-3-642-32909-8_12.
- [9] George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963.

- [10] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984. doi: 10.1007/BF02579150.
- [11] Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*, volume 13 of *Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994. doi: 10.1137/1.9781611970791.
- [12] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.
- [13] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [14] Robert J. Vanderbei. *Linear Programming: Foundations and Extensions*. Springer, 5 edition, 2020. doi: 10.1007/978-3-030-39415-8.
- [15] D. Elizondo. The linear separability problem: Some testing methods. *IEEE Transactions on Neural Networks*, 17(2):330–344, 2006. doi: 10.1109/TNN.2005.860871.
- [16] Nimrod Megiddo. Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. *SIAM Journal on Computing*, 12(4):759–776, 1983.
- [17] Raimund Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete & Computational Geometry*, 6(5):423–434, 1991. doi: 10.1007/BF02574699.
- [18] Kenneth L. Clarkson. Las vegas algorithms for linear and integer programming when the dimension is small. *Journal of the ACM*, 42(2):488–499, 1995. doi: 10.1145/201019.201036.
- [19] Jiří Matoušek, Micha Sharir, and Emo Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4–5):498–516, 1996. doi: 10.1007/BF01940877.
- [20] Jack Elzinga and Donald W. Hearn. Geometrical solutions for some minimax location problems. *Transportation Science*, 6(4):379–394, 1972.
- [21] Emo Welzl. Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science*, volume 555 of *Lecture Notes in Computer Science*, pages 359–370. Springer, 1991. doi: 10.1007/BFb0038202.
- [22] Shuiming Zhong, Huan Lyu, Xiaoxiang Lu, Baowei Wang, and Dingcheng Wang. A new sufficient & necessary condition for testing linear separability between two sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4160–4173, 2024. doi: 10.1109/TPAMI.2024.3356661.

- [23] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971. doi: 10.1137/1116025. URL <https://doi.org/10.1137/1116025>.
- [24] Vladimir Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.
- [25] Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972. doi: 10.1016/0097-3165(72)90019-2.
- [26] Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972. doi: 10.2140/pjm.1972.41.247.
- [27] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989. doi: 10.1145/76359.76371. URL <https://doi.org/10.1145/76359.76371>.
- [28] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996. doi: 10.1137/S0097539793255151. URL <https://doi.org/10.1137/S0097539793255151>.
- [29] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998. doi: 10.1145/285055.285060. URL <https://doi.org/10.1145/285055.285060>.
- [30] Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.
- [31] Sofya Raskhodnikova. Approximate testing of visual properties. In Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, editors, *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, volume 2764 of *Lecture Notes in Computer Science*, pages 370–381. Springer, 2003. doi: 10.1007/978-3-540-45198-3_31. URL https://doi.org/10.1007/978-3-540-45198-3_31.
- [32] Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. The power and limitations of uniform samples in testing properties of figures. *Algorithmica*, 81(3):1247–1266, March 2019. doi: 10.1007/s00453-018-0467-9.
- [33] Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. Tolerant testers of image properties. *ACM Transactions on Algorithms*, 18(4):37:1–37:39, 2022. doi: 10.1145/3531527.

- [34] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.
- [35] Jock A. Blackard and Denis J. Dean. Covertypes. UCI Machine Learning Repository, 1999. URL <https://archive.ics.uci.edu/ml/datasets/Covertypes>.