Learnability

Learnability and Language Acquisition

Alexander Clark

Department of Computer Science
Royal Holloway, University of London

Wednesday

Outline

Computational learning theory

Two problems

- Information theoretic problems
- Computational complexity problems

Learning theory should help!

- Unfortunately, it has often been distorted and used to stifle research rather than guide it.
- If X is mathematically impossible, then don't try doing X . . .
- If you have a program that seems to do X, then it is probably actually doing something else *Y*

Learning is hard!

Two classes of problems: information theoretic and complexity theoretic.

Information theory

Is there in principle enough information in the input?
The “classic” POS argument

Computational Complexity

Can we use this information to construct our hypothesis?

Outline

Wexler (1999)

The strongest most central arguments for innateness thus continue to be the arguments from APS and learnability theory. . . . The basic results of the field include the demonstration that without serious constraints on the nature of human grammar, no possible learning mechanism can in fact learn the class of human grammars.

Mathematical Characterization of Learning Problems

- Using formal models to study learning problems permits one to study the formal limits of learnability under specified assumptions concerning the learning situation.
- The assumptions define the nature of the learning process and the data on the basis of which learning is achieved.
- These involve idealizing the learning situation to facilitate formal modeling.
- The results which one can prove in a model depend on the way in which one defines the object to be learned, what learning consists in, and the evidence available to the learner.

Constraints on models

To offer insight into human language acquisition:

- a formal model of language learning should be sufficiently inclusive to allow a learner to acquire any natural language from a realistic set of data: since this is what happens.
- It must be restrictive enough to rule out types of learning that do not occur (such as acquisition of a mature adult grammar of a language L from too small a data set).
- The model must strike a balance between rendering the learning process impossibly difficult and making it trivially easy.

Formal models of language learning

Pinker (1979)

Put forward various conditions for a model of language learning.

1. Learnability condition
2. Equipotentiality condition
3. Time condition
4. Input conditions
5. Cognitive condition
6. Developmental condition

Formal models of language learning

Pinker (1979)

Put forward various conditions for a model of language learning.

1. Learnability condition
2. Equipotentiality condition
3. Time condition
4. Input conditions
5. Cognitive condition
6. Developmental condition

- Mathematical models tend to ignore the last one.
- Missing an evolutionary criterion?

Pinker (1979)

There are two reasons why formal models of language learning are likely to contribute to our understanding of how children learn to speak, even if none of the models I will discuss satisfies all of our six criteria. First of all, a theory that is powerful enough to account for the fact of language acquisition may be a more promising first approximation of an ultimately viable theory than one that is able to describe the course of language acquisition, which has been the traditional focus of developmental psycholinguistics. As the reader shall see, the Learnability criterion is extraordinarily stringent, and it becomes quite obvious when a theory cannot pass it.

. . .

The second potential benefit of formal models is the explicitness that they force on the theorist, which in turn can clarify many conceptual and substantive issues that have preoccupied the field.

The Value of Formal Learning Models

- It is reasonable to ask whether formal learning models are useful in understanding human language acquisition.
- Isn't it the case that knowledge of this process depends entirely on the psychological and biological facts of acquisition?
- The formal study of grammar induction can clarify the sort of data required, and the nature of the learning biases that must be assumed in order for the class of natural languages to be learnable within the constraints of time and data available to human learners.
- Such learnability results establish basic conditions of adequacy that theories of grammar and language acquisition must satisfy.

Analogy with aerodynamics

Idealisation

Many different models:
the true situation is very complicated; we need to idealise we can idealise in different ways

- Make it harder (pessimistic assumptions)
- Make it easier (optimistic assumptions)

We end up with a spectrum of results.

- Positive results: under some assumptions, certains classes of languages can be learned.
- Negative results: under some circumstances, learning is impossible

Results and assumptions

Informative

Positive results under pessimistic assumptions
Negative results under optimistic assumptions
Not many of these.

Results and assumptions

Informative

Positive results under pessimistic assumptions
Negative results under optimistic assumptions
Not many of these.

Uninformative

Positive results under optimistic assumptions
Negative results under pessimistic assumptions
We have lots of these.

Basic mathematical assumptions

Machine learning and grammatical inference study learning in many different domains.
Typically we have:

- A class of objects (sentences)
- A class of concepts (languages/grammars)
- A source of information
- Some constraints
- Criteria for learning

Idealisation: A single level

Natural languages have different levels. At the very least we have morphology and syntax. We conflate all of these into a single task: learning a formal language.

- Given a finite alphabet Σ, we want to learn a language $L \subset \Sigma^*$.
- Later we will also look at learning a probability distribution over strings.

Idealisation: A single level

Natural languages have different levels. At the very least we have morphology and syntax. We conflate all of these into a single task: learning a formal language.

- Given a finite alphabet Σ, we want to learn a language $L \subset \Sigma^*$.
- Later we will also look at learning a probability distribution over strings.
- Σ could be a set of phonemes, or characters, or a set of words, or a set of lexical categories (POS tags)
- The language could be the set of well-formed sentences, or the set of words that obey the phonotactics of the language .

Idealisation: A single level

Natural languages have different levels. At the very least we have morphology and syntax. We conflate all of these into a single task: learning a formal language.

- Given a finite alphabet Σ, we want to learn a language $L \subset \Sigma^*$.
- Later we will also look at learning a probability distribution over strings.
- Σ could be a set of phonemes, or characters, or a set of words, or a set of lexical categories (POS tags)
- The language could be the set of well-formed sentences, or the set of words that obey the phonotactics of the language .

Use syntax to be concrete.

Weak versus strong learning of syntax

- On a weak view of learning, the learner who acquires a language can identify its string set.
- On a strong view, the learner acquires the formal representation of a grammar that assigns syntactic structures to the string set of the language that it generates.
- This difference corresponds to the distinction between weak and strong generative capacity (E-language/I-language)
- We don't know exactly what the right set of structural descriptions are, so this leads to a version of the theory-internal APS.

Syntax and Semantics

What is L?

Classic view

The set of grammatical sentences
Syntactically well formed sentences
Includes “colorless green ideas sleep furiously”

Primary Linguistic Data

Consists of semantically well formed utterances (largely)
Thus contains information about syntactic/semantic dependencies

Syntax and Semantics

What is L?

Classic view

The set of grammatical sentences
Syntactically well formed sentences
Includes “colorless green ideas sleep furiously”

Primary Linguistic Data

Consists of semantically well formed utterances (largely)
Thus contains information about syntactic/semantic dependencies

If you can learn the set of syntactically and semantically well-formed utterances, then you will have implicitly learned the dependencies.

Machine Learning

Machine Learning definition; Tom Mitchell

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

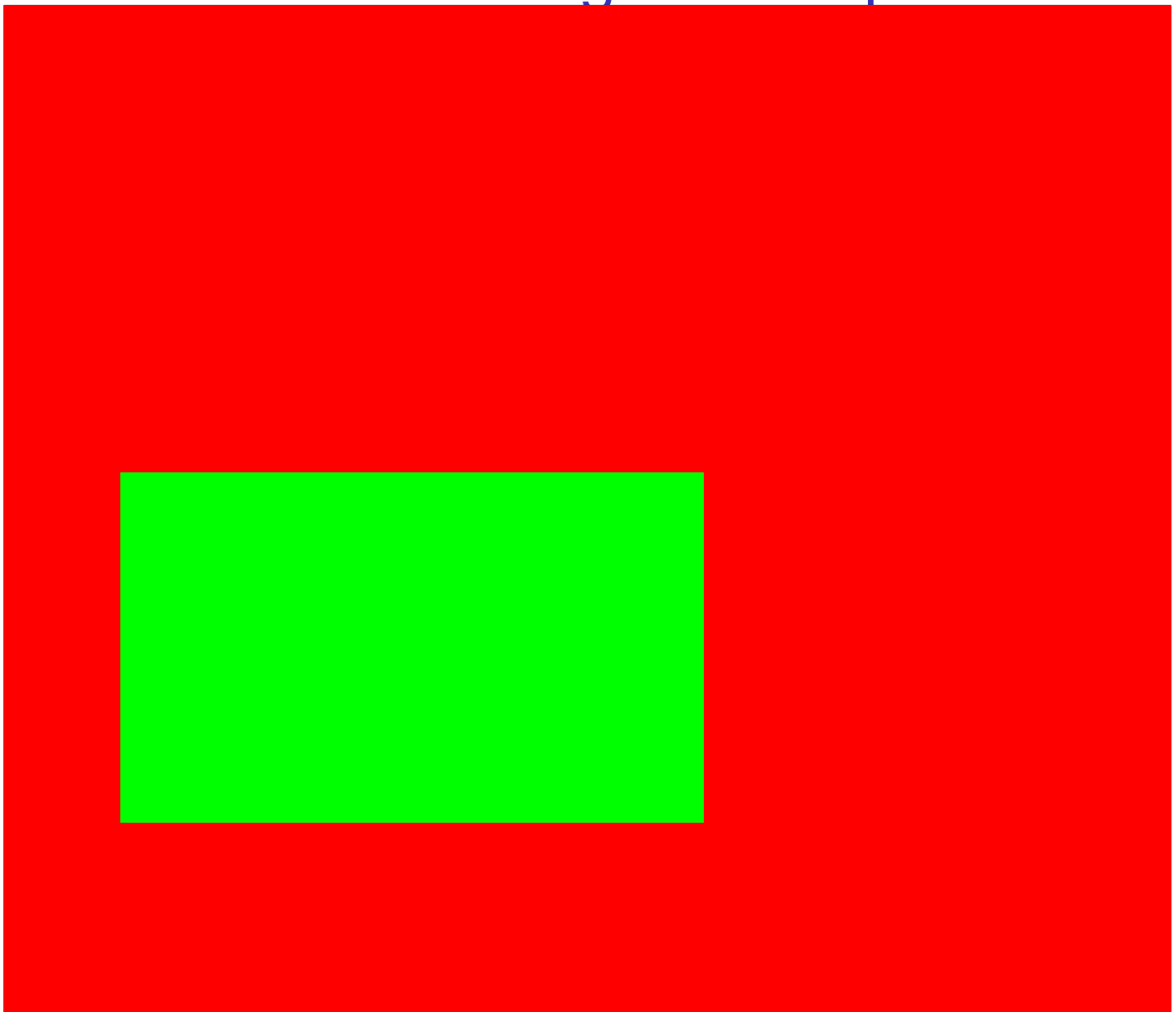
Theoretical analysis

- Computational learning theory
- Algorithmic learning theory
- Inductive inference

Diagrams

- A point on the plane is a string of words.
- The plane consists of every possible string of words.
- A language is an infinite set of sentences: a region in the plane. e.g. a rectangle.
- Points inside the rectangle represent grammatical sentences. Points outside the rectangle represent ungrammatical sentences.

Target concept



Terminology

- The instance space X: the plane
- Instances x are points in the plane
- A concept C is a rectangle which divides the points into positive and negative examples
- The concept class is the set of target concepts
- The concept space $\mathcal{C}$ is in this case the set of all axis aligned rectangles (infinitely many)

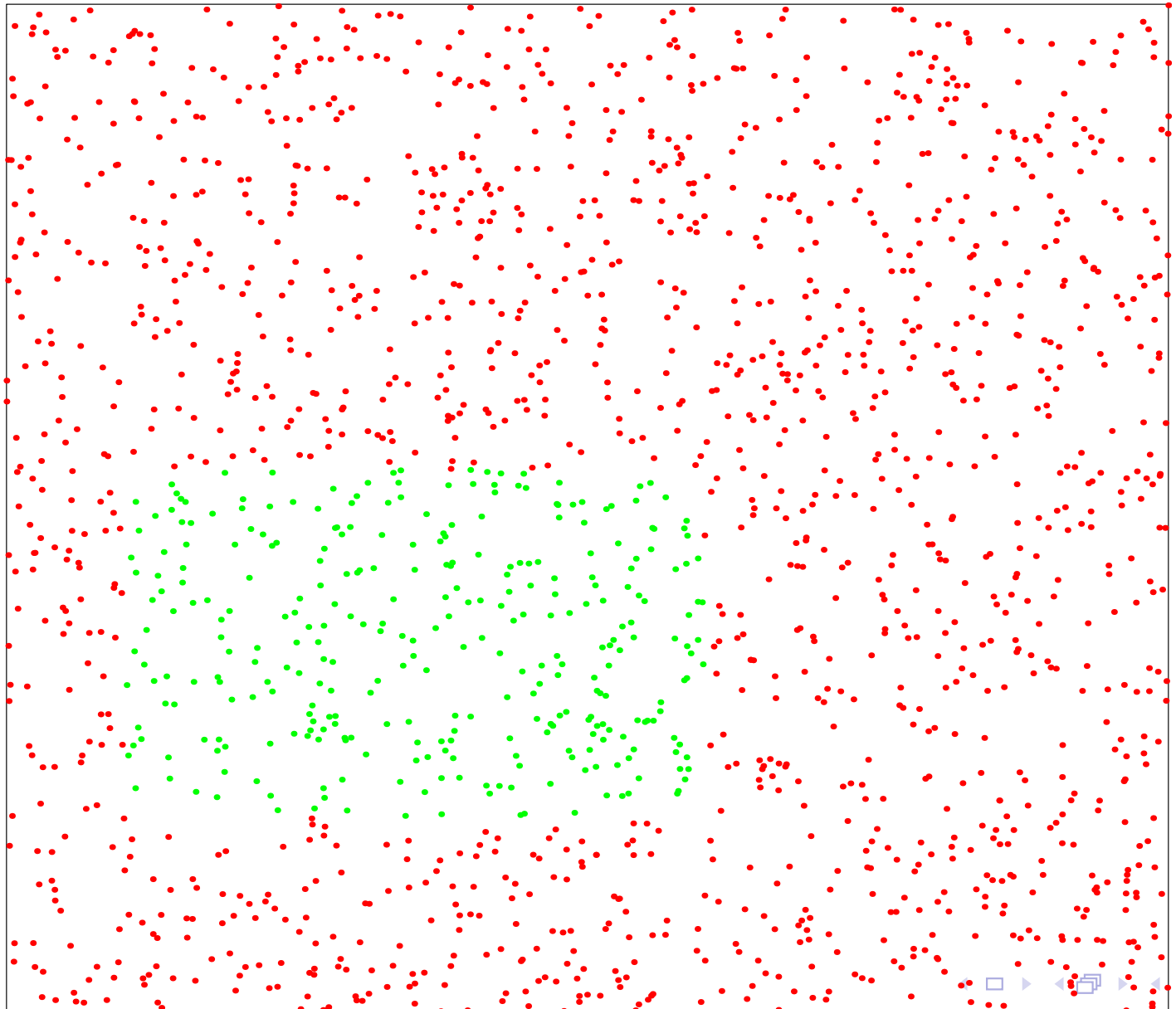
Labeled data

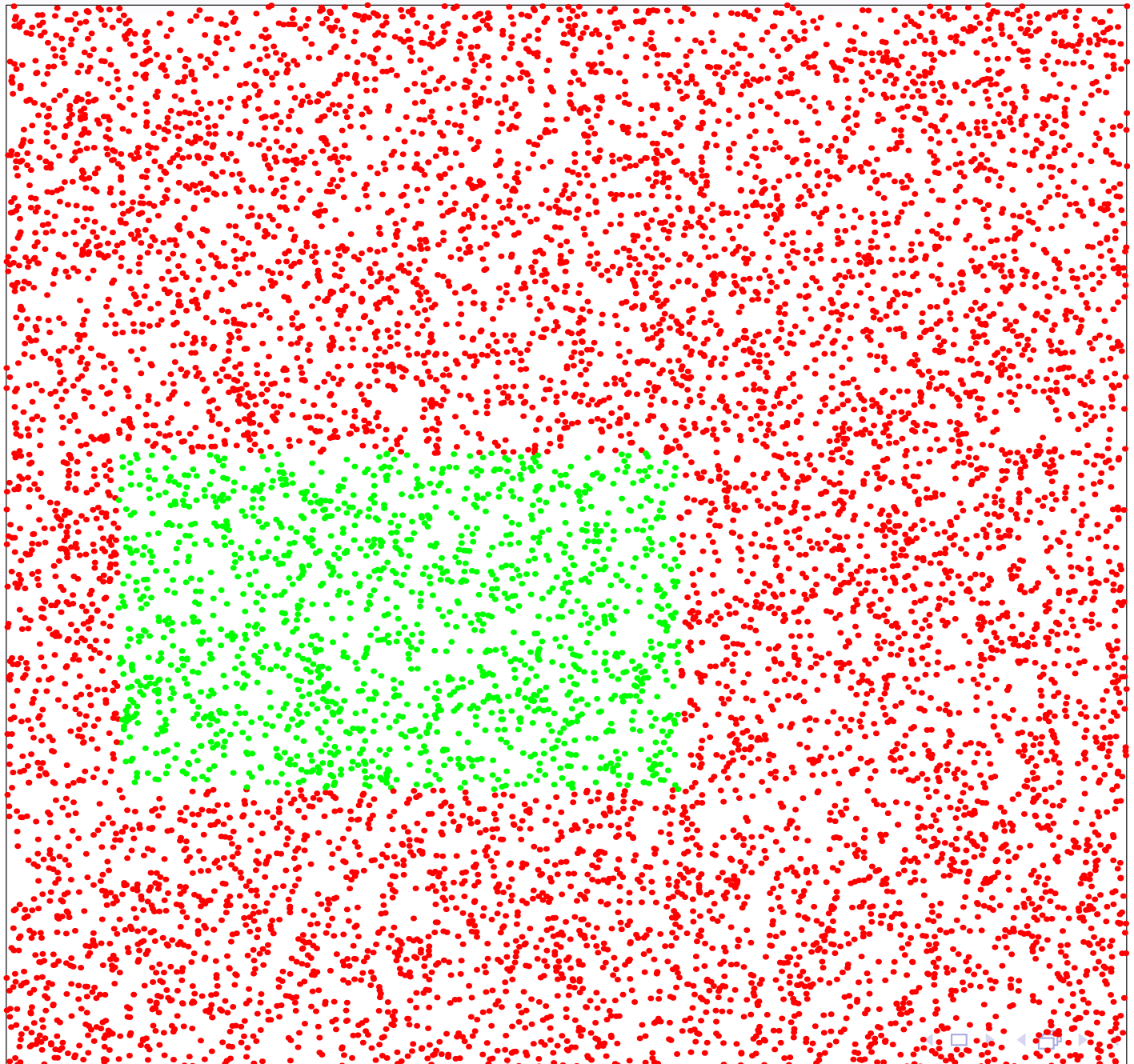
We have to learn from a finite set of examples/instances

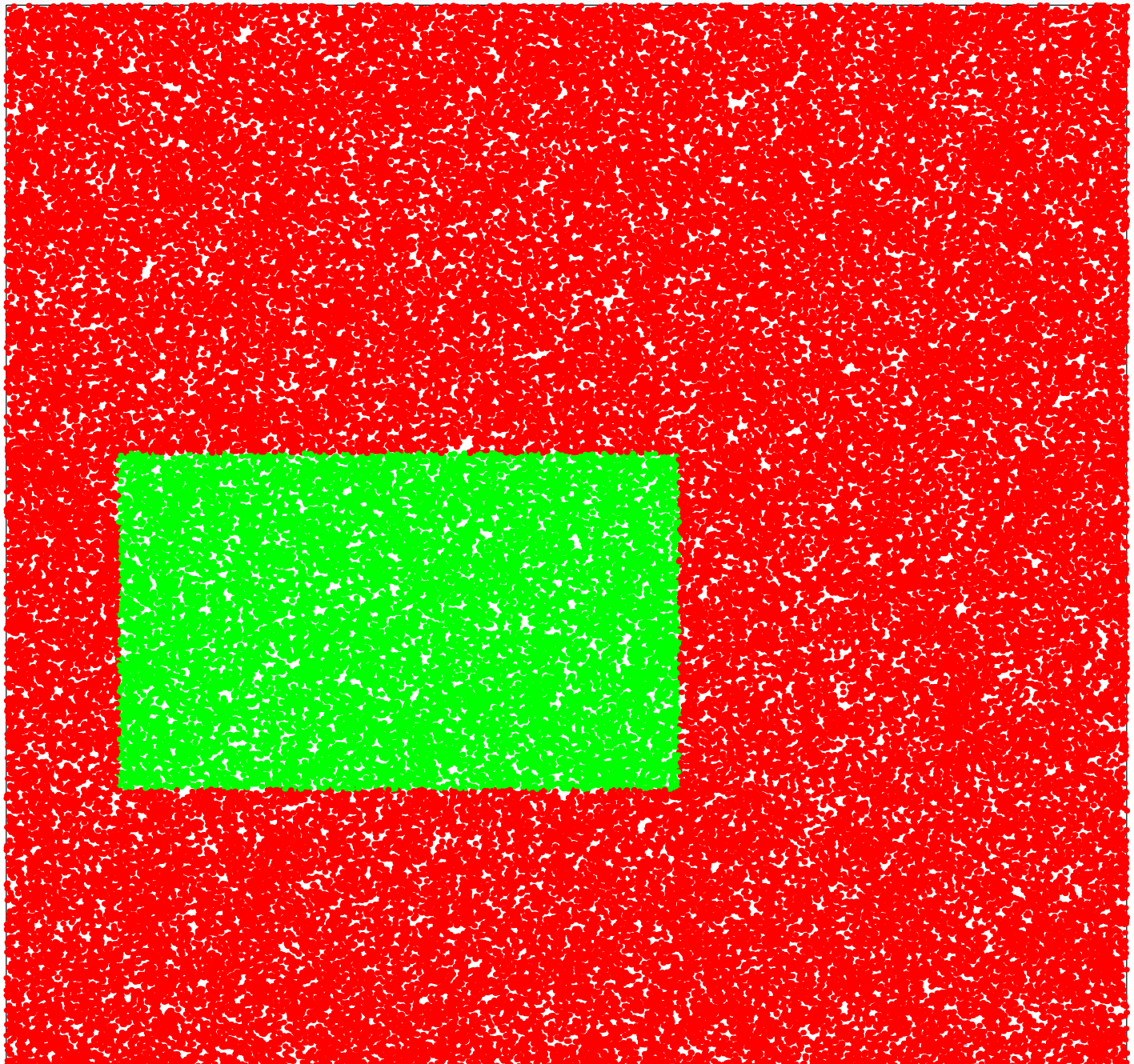
Supervised learning

each example is labeled:
green means it is in the language – grammatical
red means it is not in the language – ungrammatical

Finite data



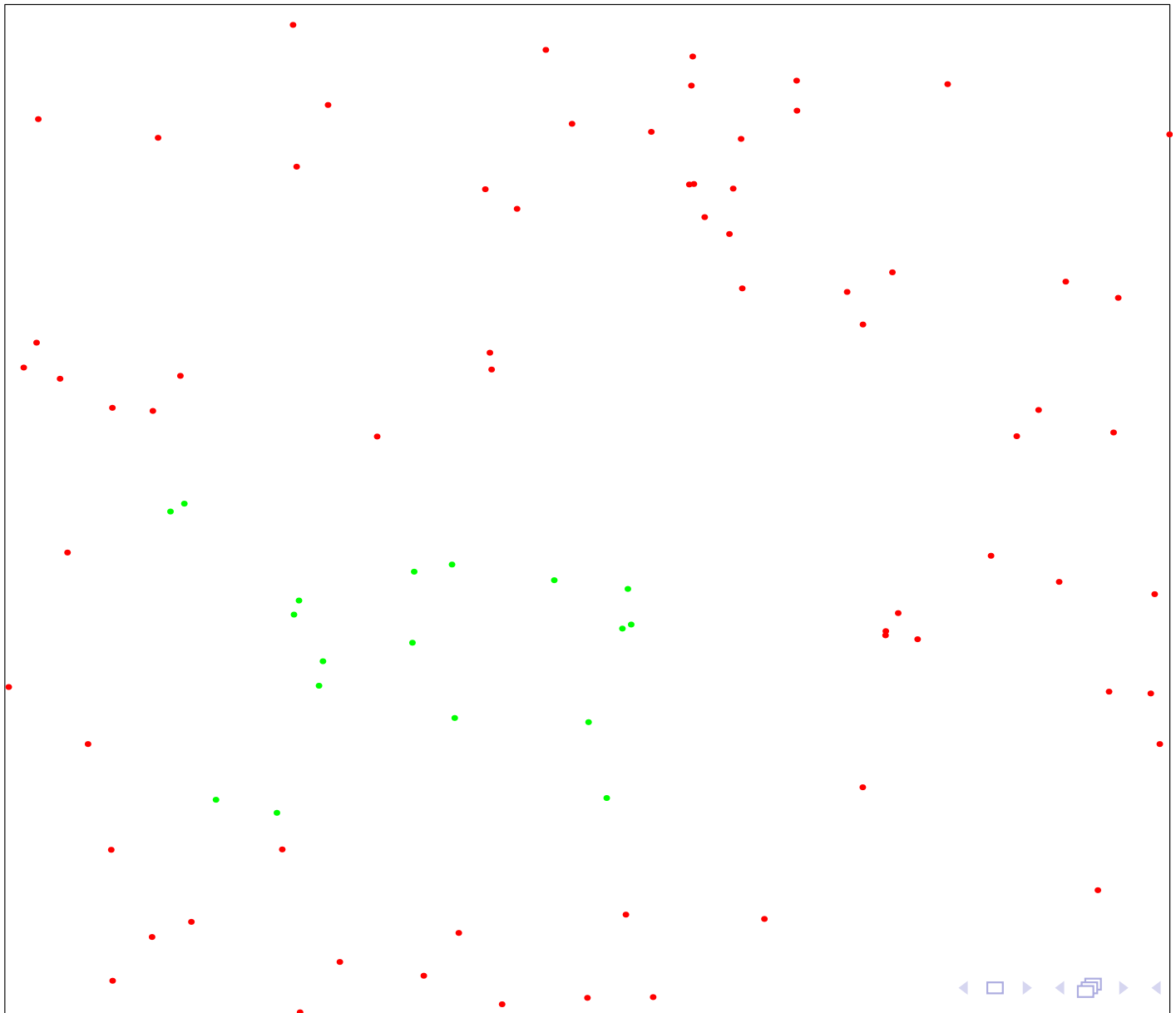




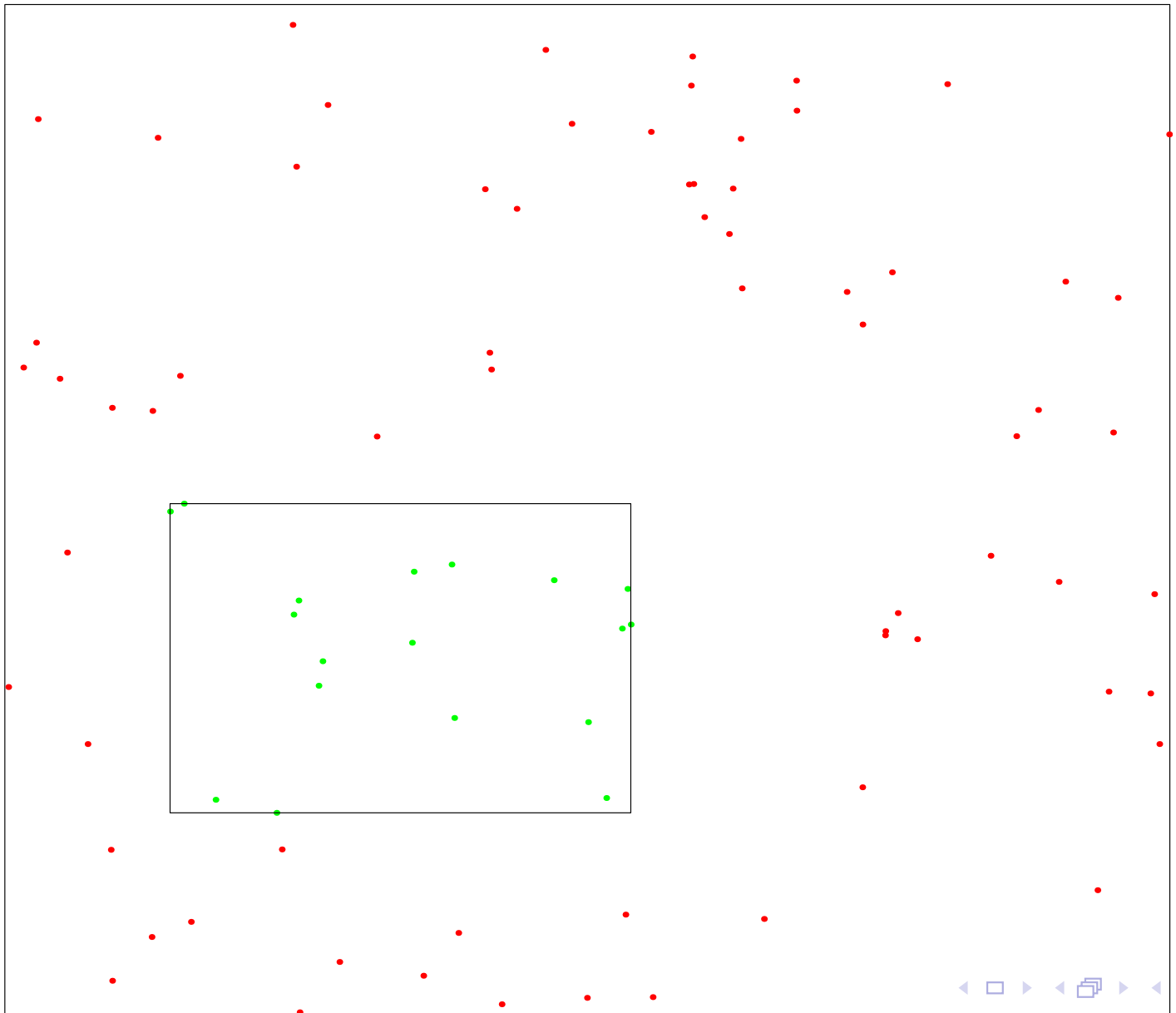
Obvious comment

- More data means a more accurate hypothesis.
- You might never get an exactly correct hypothesis.

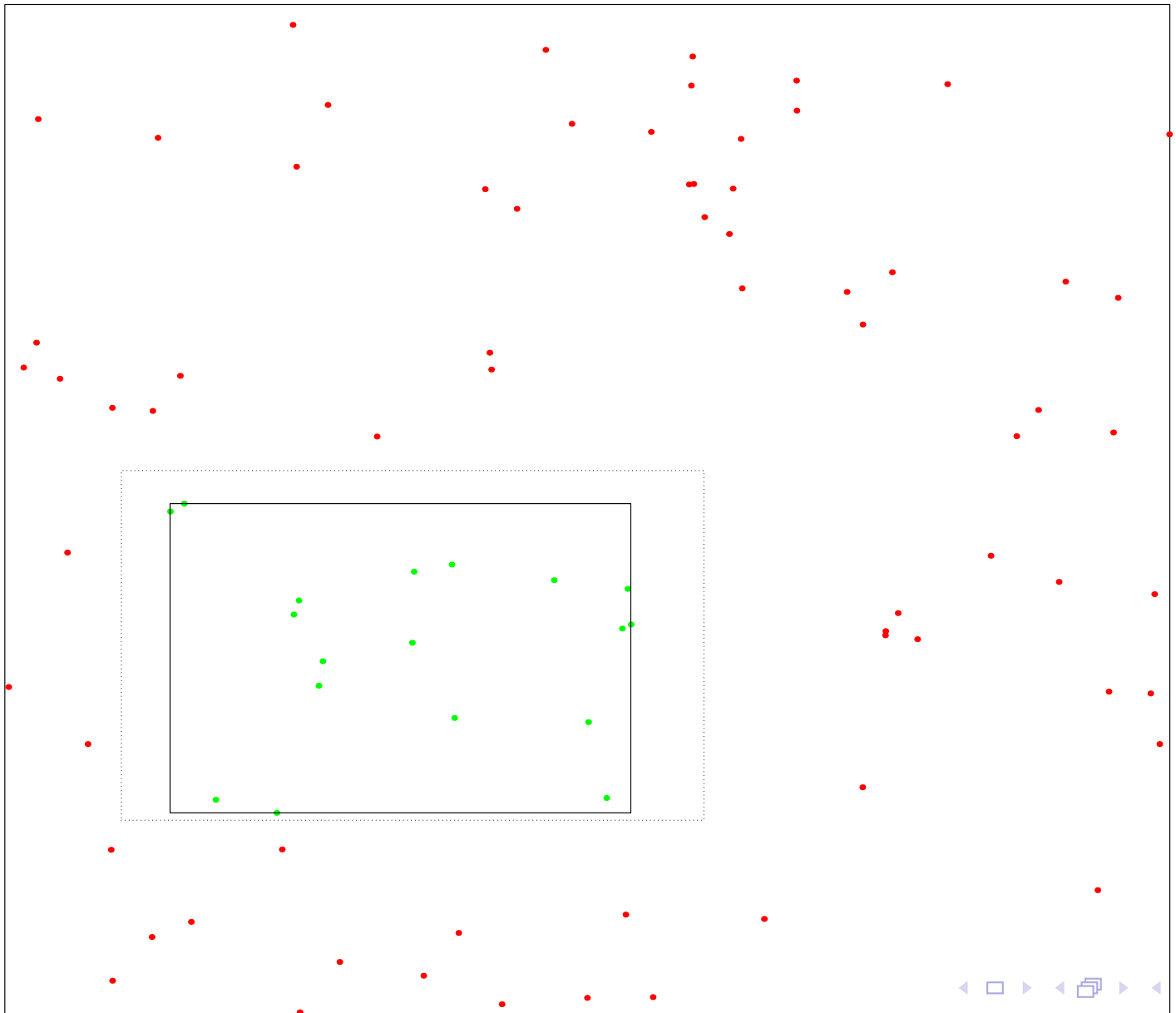
Error



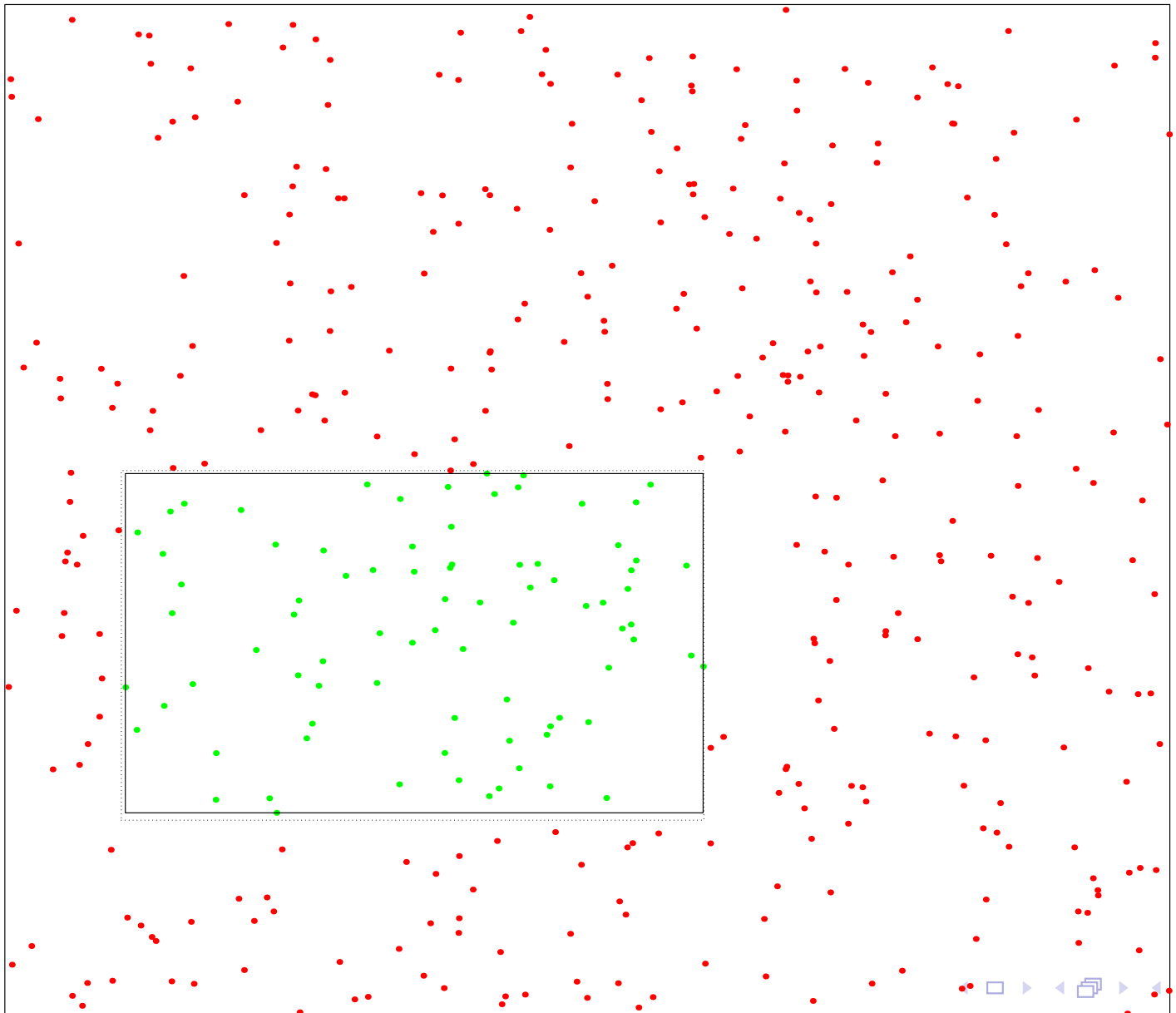
Error



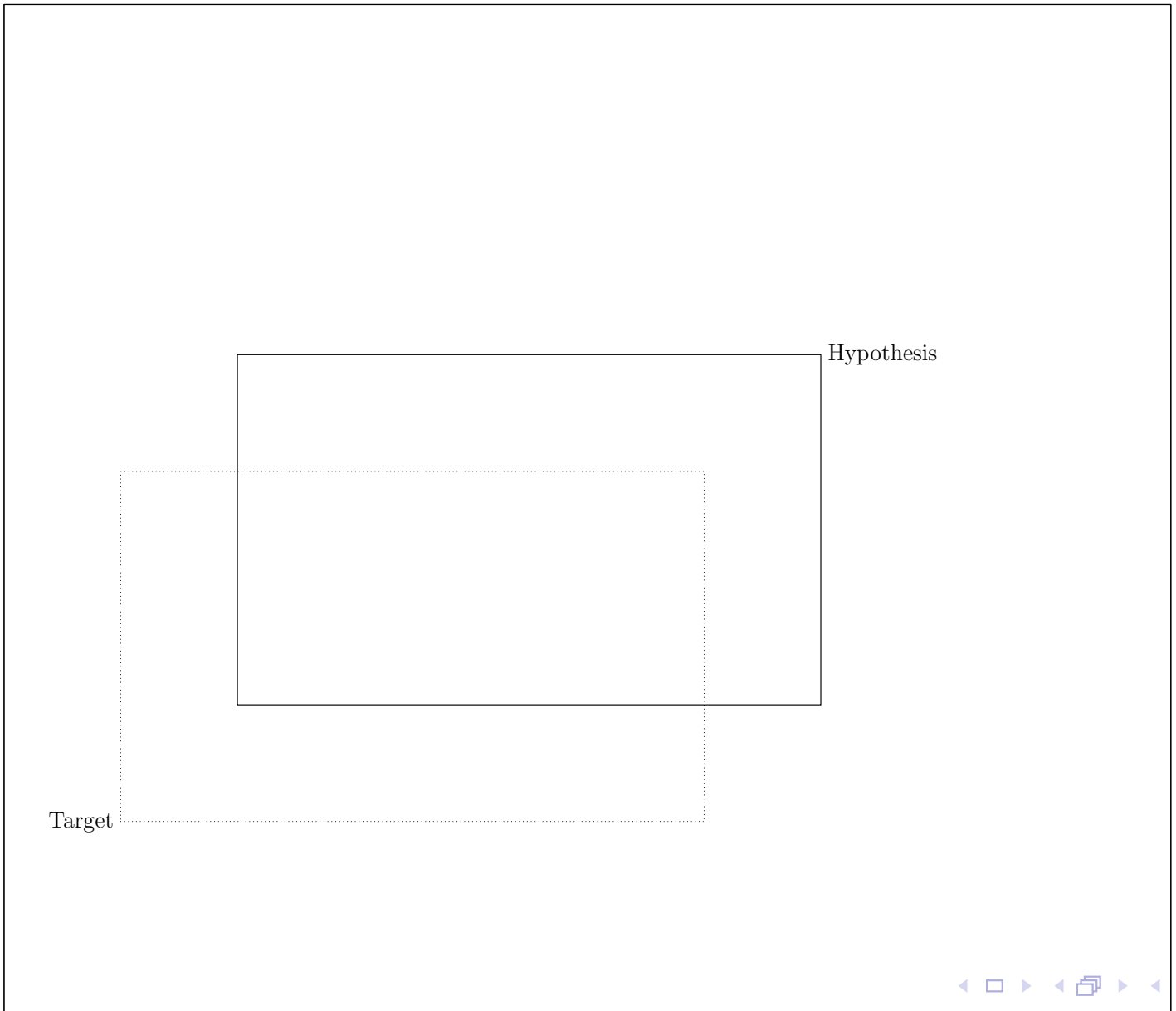
Error



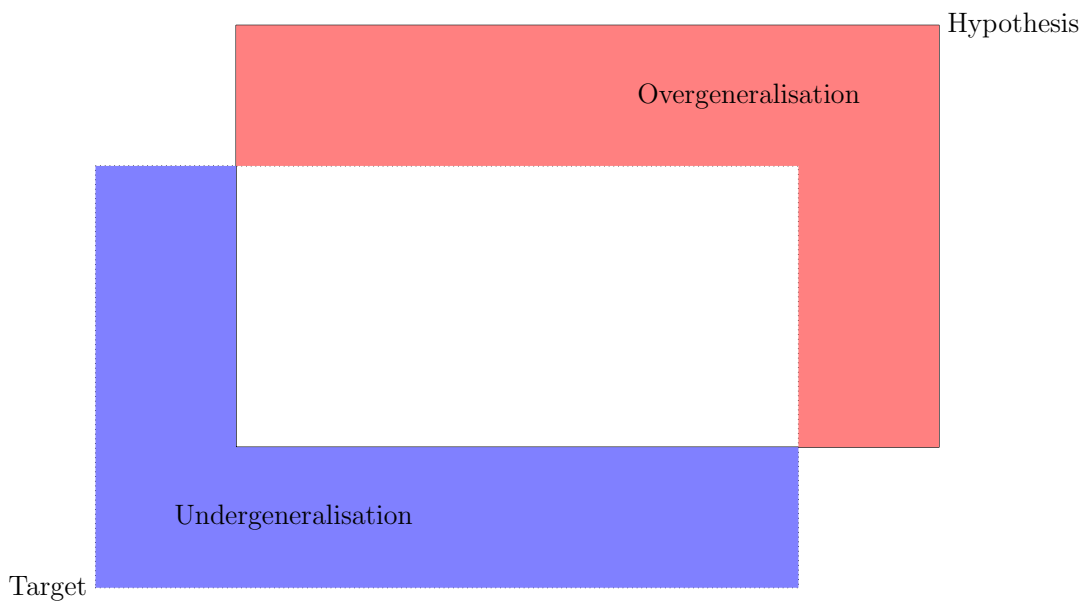
Error



Error



Measuring error



Measuring error

Why is this right? We cannot look in someones head and see if they have the right hypothesis.

Observation

- Do they object to sentences that we think are acceptable?
- Do they produce sentences that we think are unaccetaple?

The most we can say is that these events are rare.

Hypothesis class versus concept class

Hypothesis class

The set of hypotheses that the learner will generate under various inputs.

Concept class

The set of target concepts that the learner may encounter.

Hypothesis class versus concept class

Hypothesis class

The set of hypotheses that the learner will generate under various inputs.

Concept class

The set of target concepts that the learner may encounter.

These can be different:

- The learner might encounter some triangles even though it always generate rectangles
- or the learner might not know that it is only going to receive rectangles and conjecture pentagons sometimes.

Hypothesis class versus concept class

"Proper" learning

In the example, the learner has prior knowledge of the concept class.
Hypothesis class equals concept class.

The hypothesis class might be too small

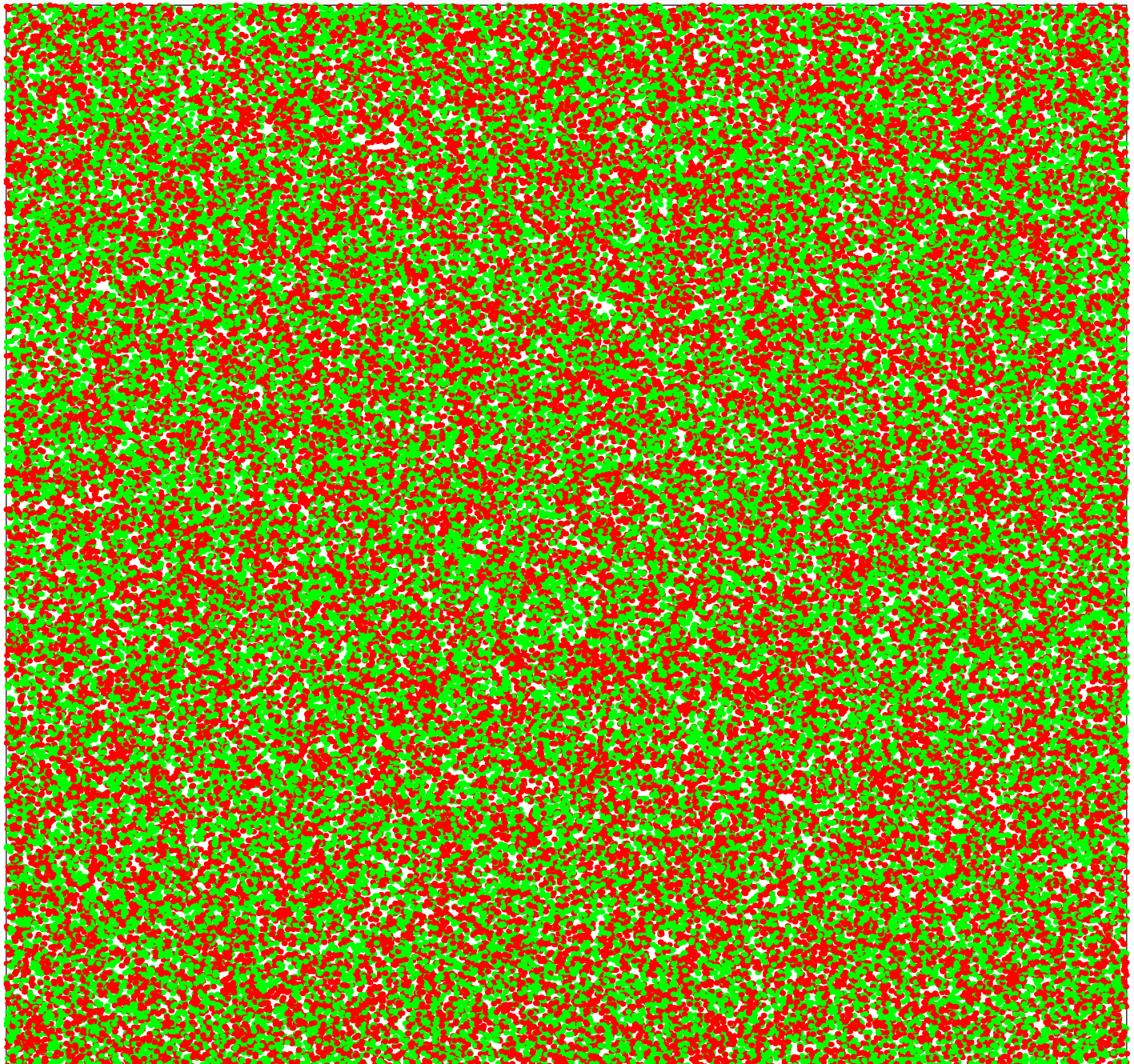
In this case the learner will sometimes fail

The concept class might be much smaller than the hypothesis class

For example, the hypothesis class might be all convex polygons.

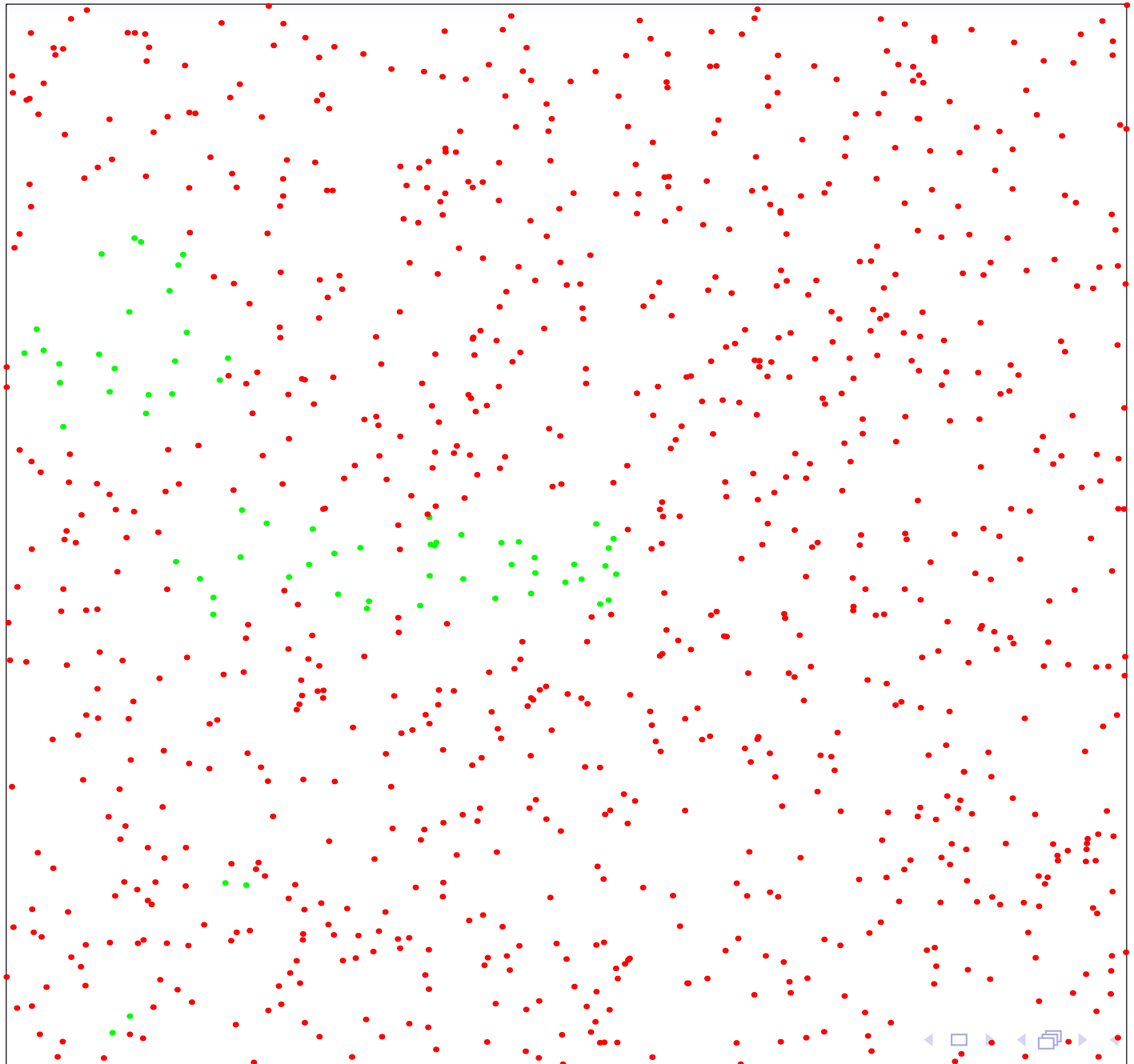
Concept class must be bounded

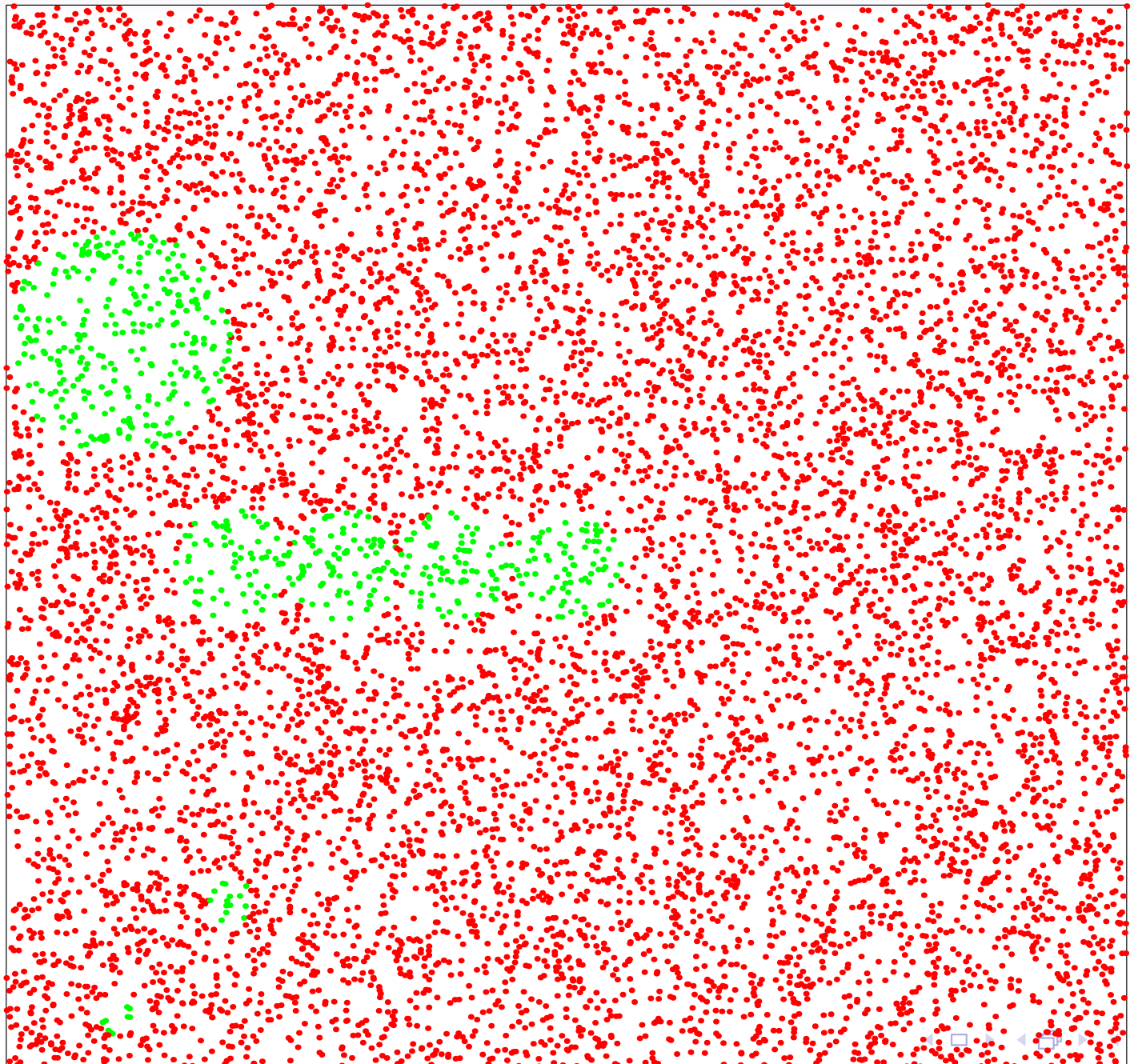
- We can't learn arbitrary sets.
- There must be some restrictions on the concepts: e.g. connected regions.

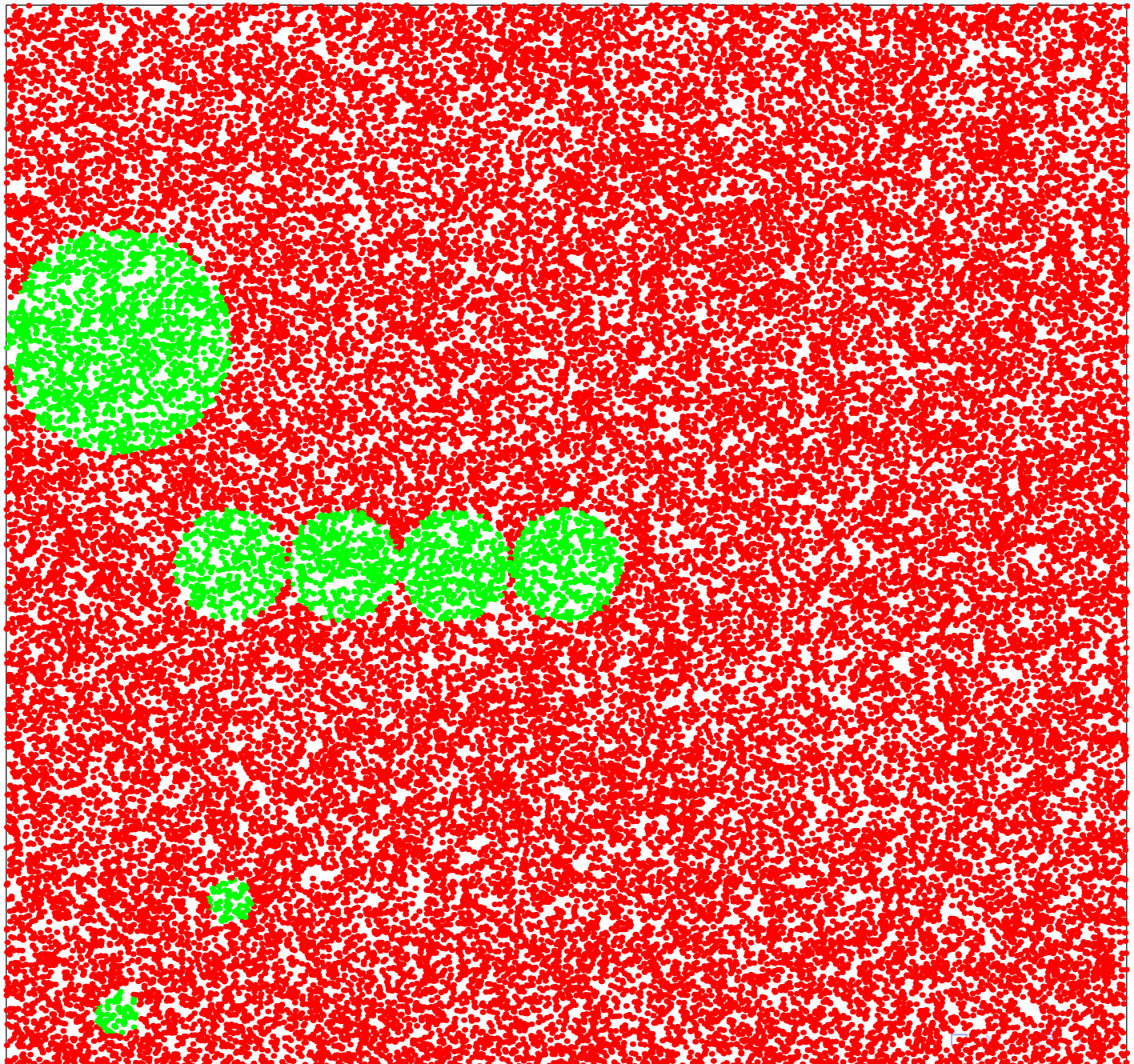


Complex hypotheses

If the shape is more complex then we will need more data to learn.





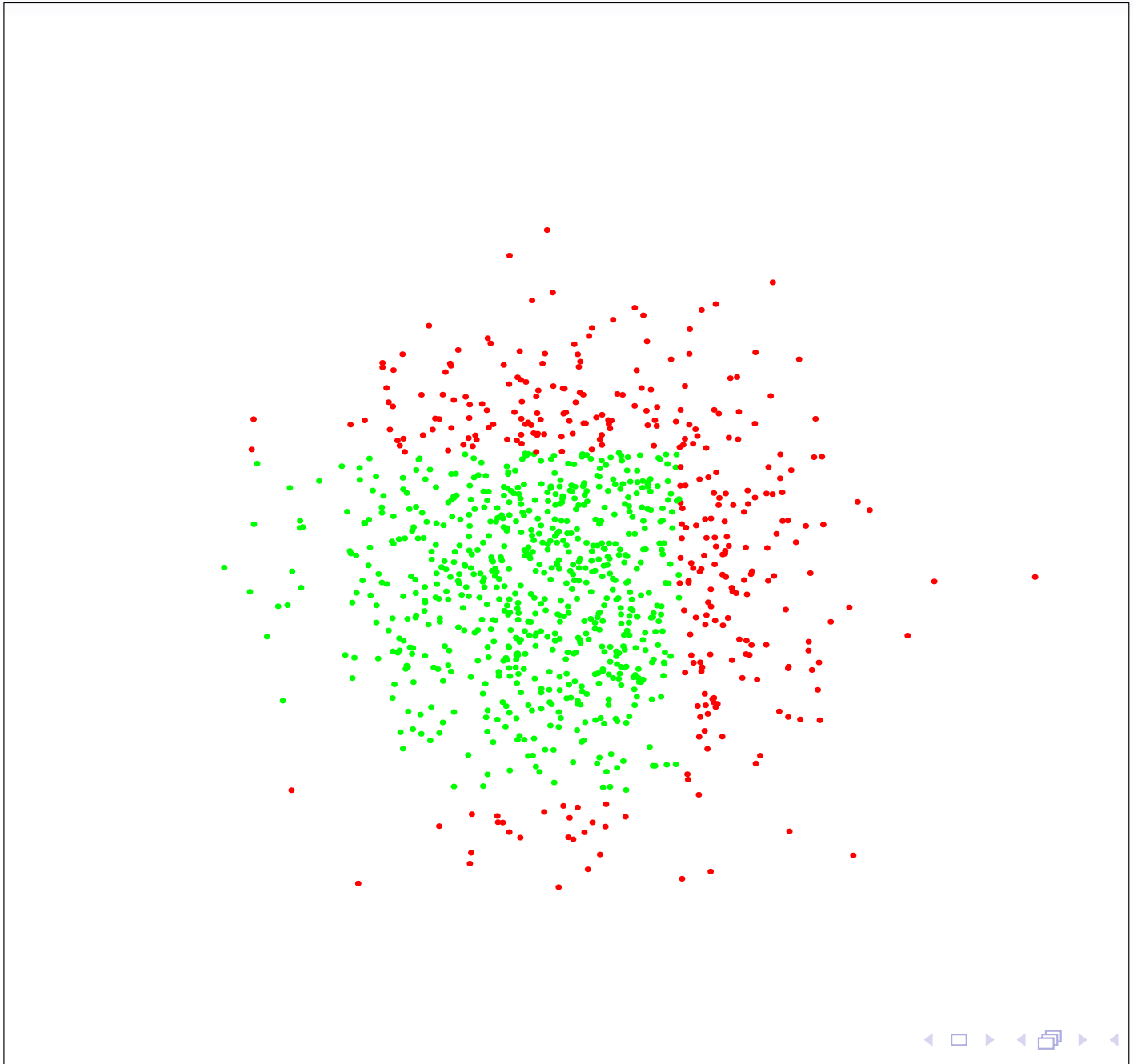


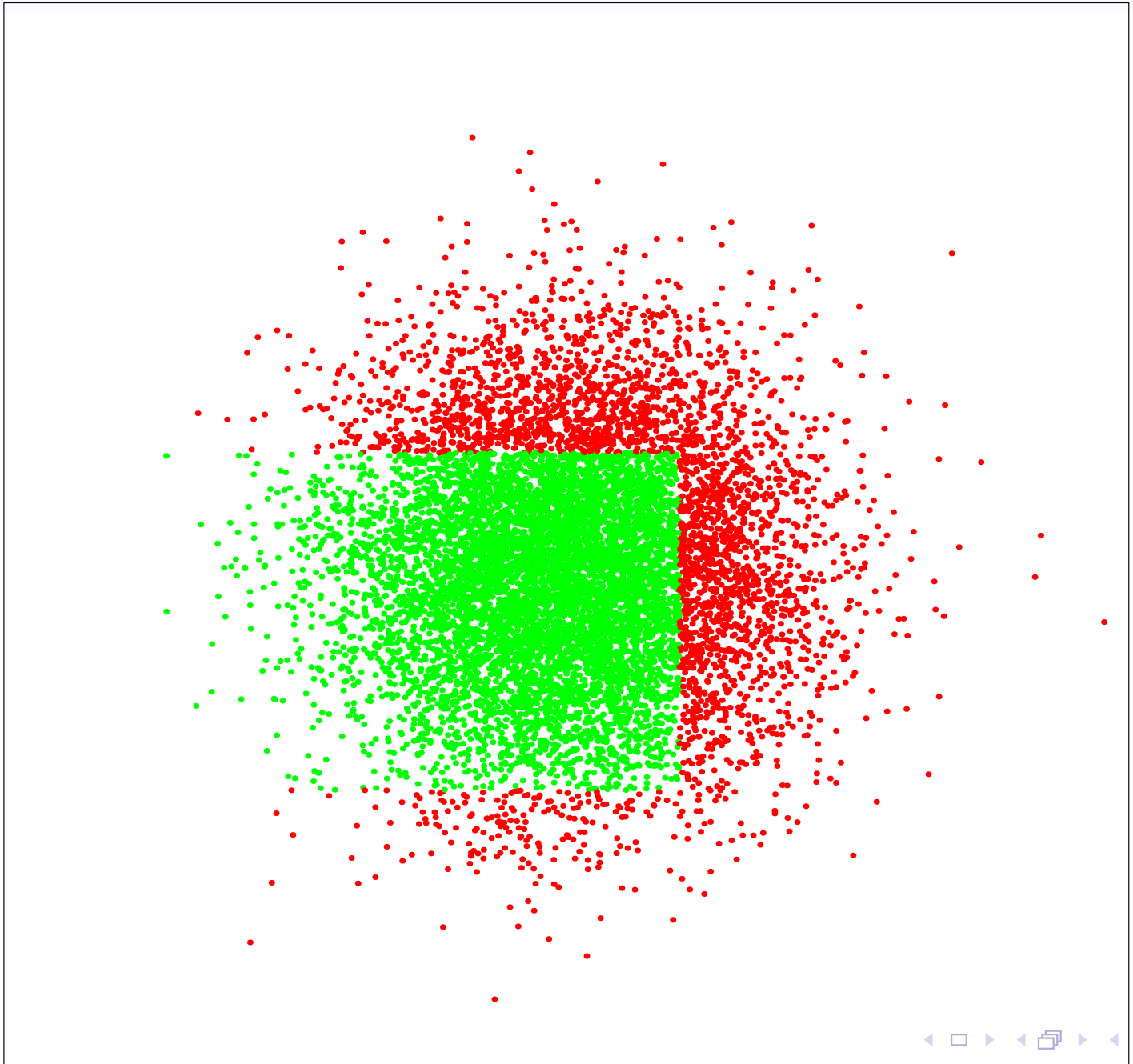
Complex hypotheses

- If we only have 1000 data points, we can't hope to learn a complex hypothesis class of 10000 tiny balls.
- But if we had a lot of data (much more than 10000) then we could still learn it
- So amount of data we need to learn depends on the accuracy and the complexity of the target.

Distribution of examples

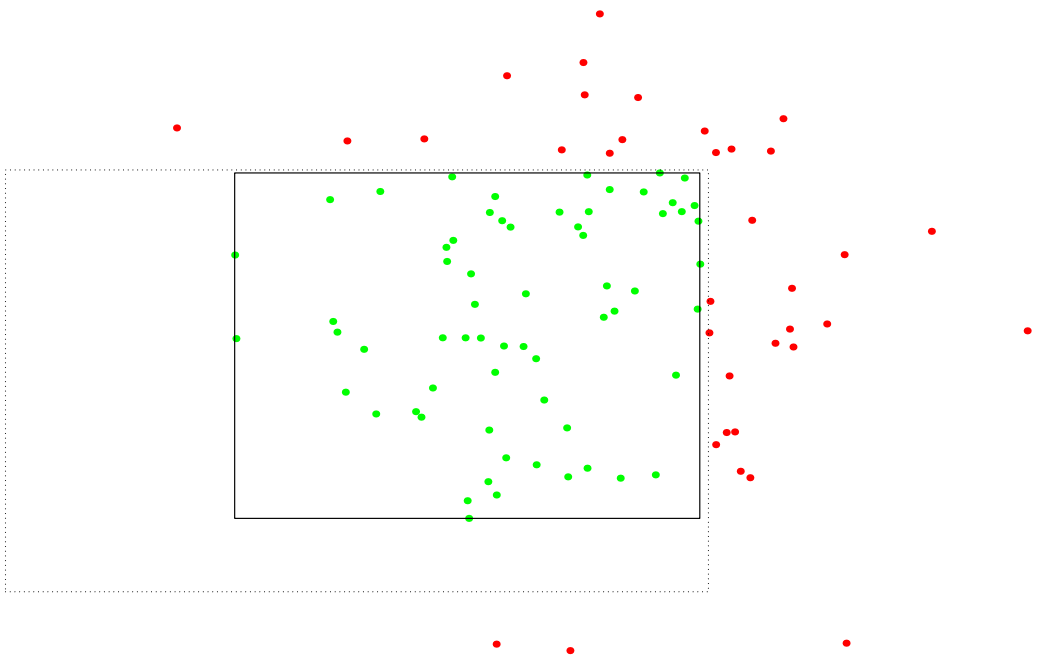
- So far all of these points are distributed evenly, uniformly over the plane.
- The area of a region is proportional to the probability
- But it might be that points are more likely to fall in one area of the plane than another.

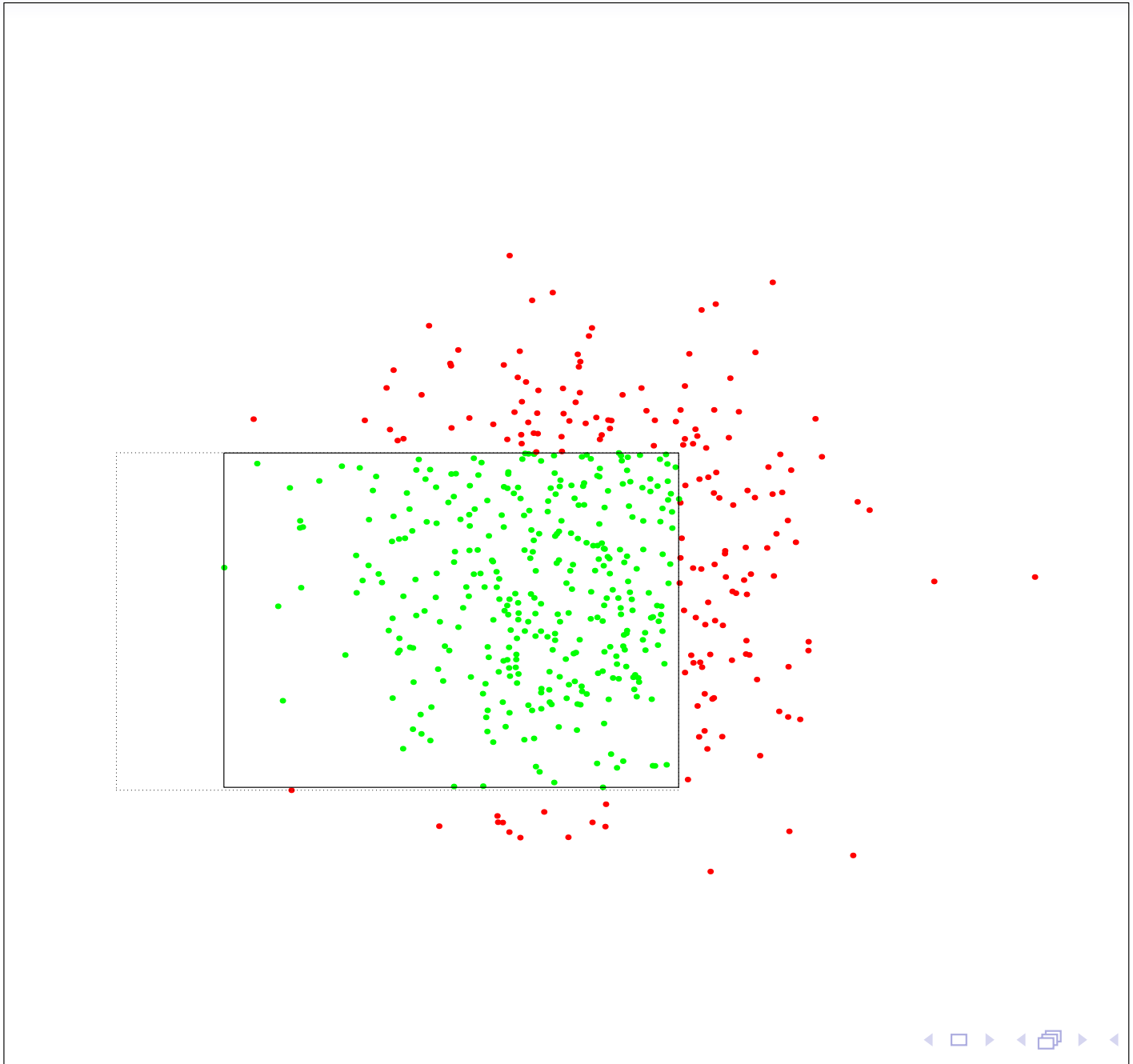




Error

- In this case the error must be “fair”.
- We can’t expect the learner to get the left part right, but it doesn’t matter as they are unimportant.
- We will measure the error by the probability of the area not the area





Rectangles

Rectangles can be learned by the simple algorithm: pick the smallest rectangle that includes the target concept.

Prior knowledge

That the concepts are rectangles.

Is this necessary?

Can we conclude that the learner must know that the concepts are rectangles?

Rectangles

Rectangles can be learned by the simple algorithm: pick the smallest rectangle that includes the target concept.

Prior knowledge

That the concepts are rectangles.

Is this necessary?

Can we conclude that the learner must know that the concepts are rectangles?

No

There are other algorithms that **don't** make this assumption that can also learn rectangles and many other shapes as well.

Difference in goals

Engineering versus Cognitive modeling

Machine learning

Solving problems

Proving that classes can be learned

Language acquisition

Understanding properties of the learner

What can we infer about the learner from its performance?

Unlabeled examples

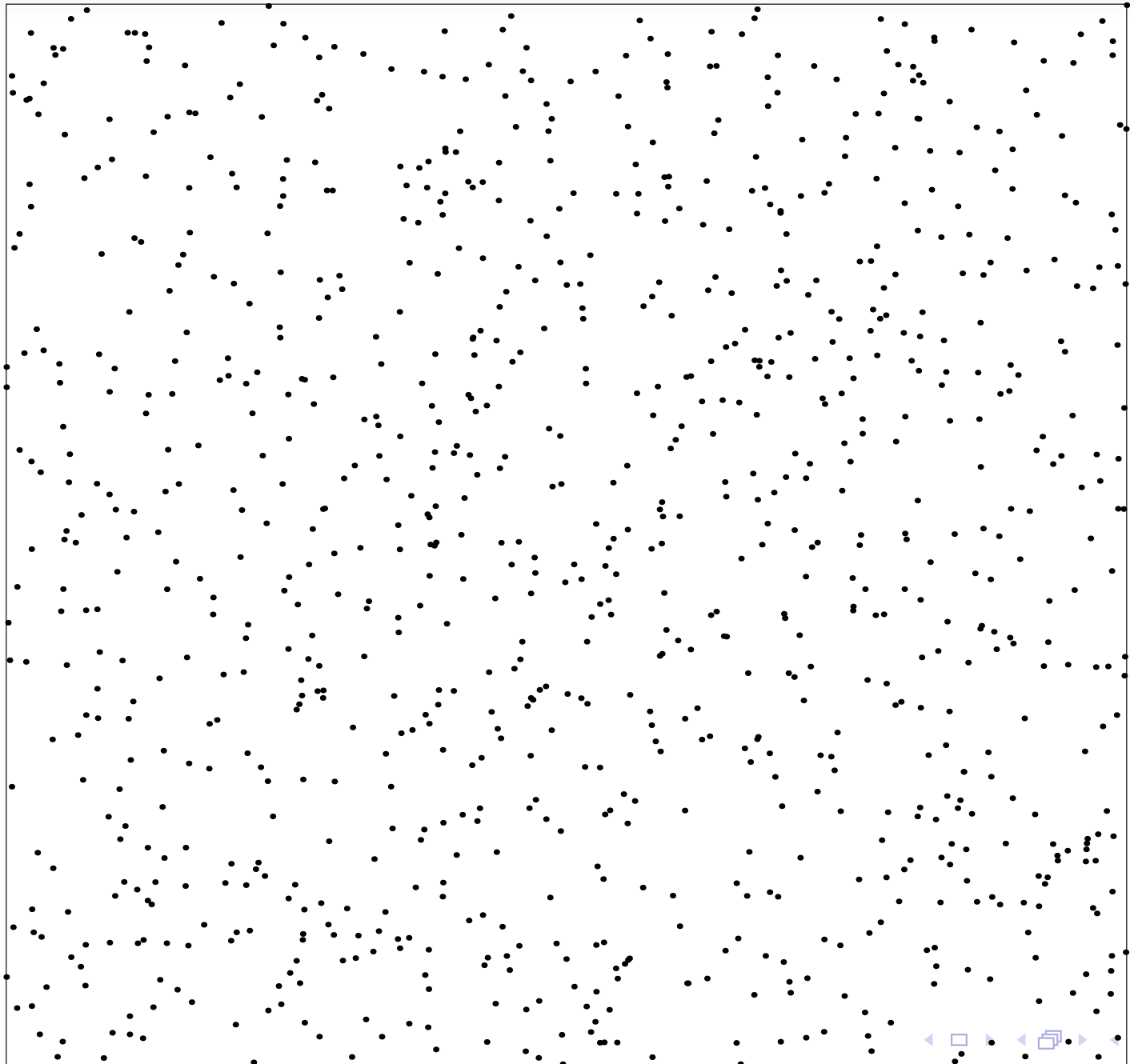
Labeled examples

Each point comes with a color that tells you whether it is in the concept or not.

Unlabeled examples

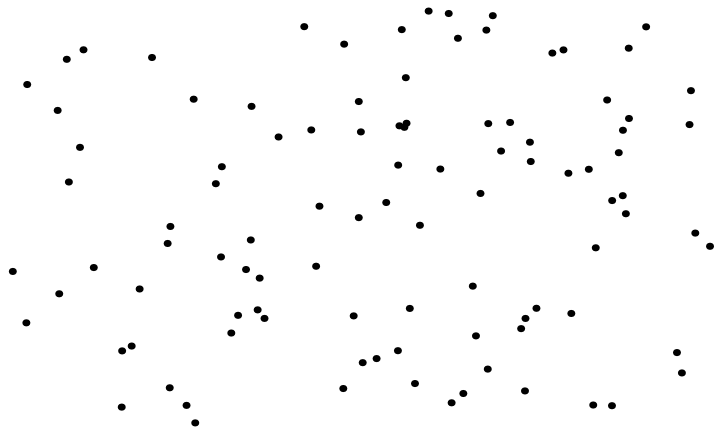
Each point is black

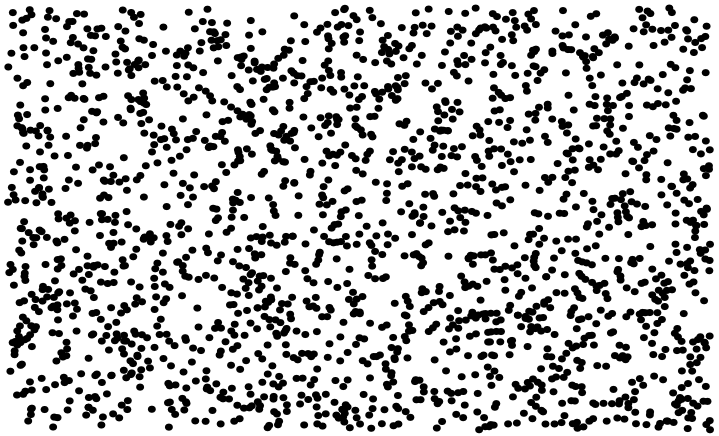
If we just have unlabeled examples then the distribution of the examples must be restricted in some way or we can't learn.



Positive only examples

- If they are unlabeled but restricted to positive examples then we can still learn.
- No negative data





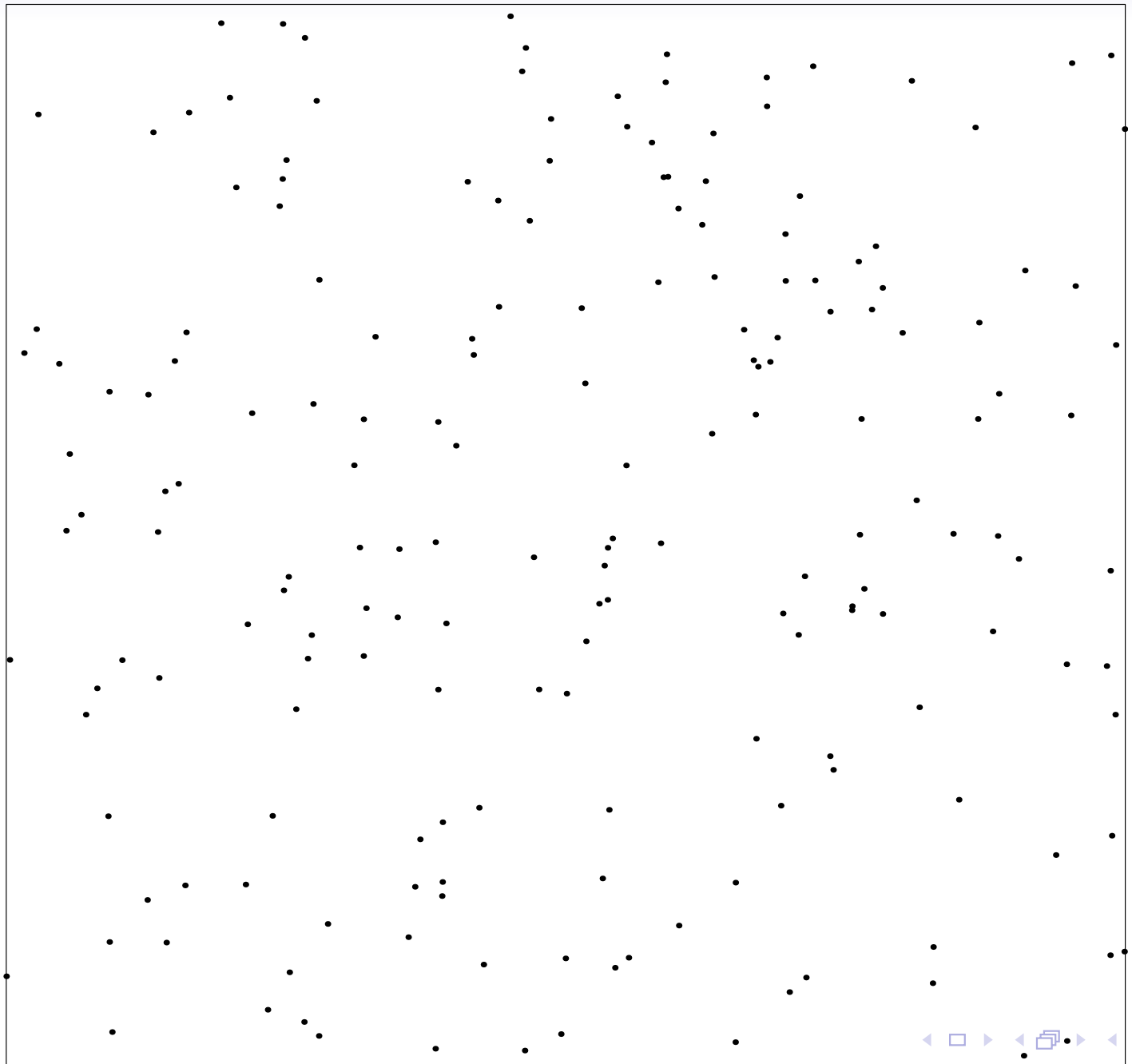
Asymmetry

Small concept

If the concept is very small, then positive examples are much more useful than negative examples.

Large concept

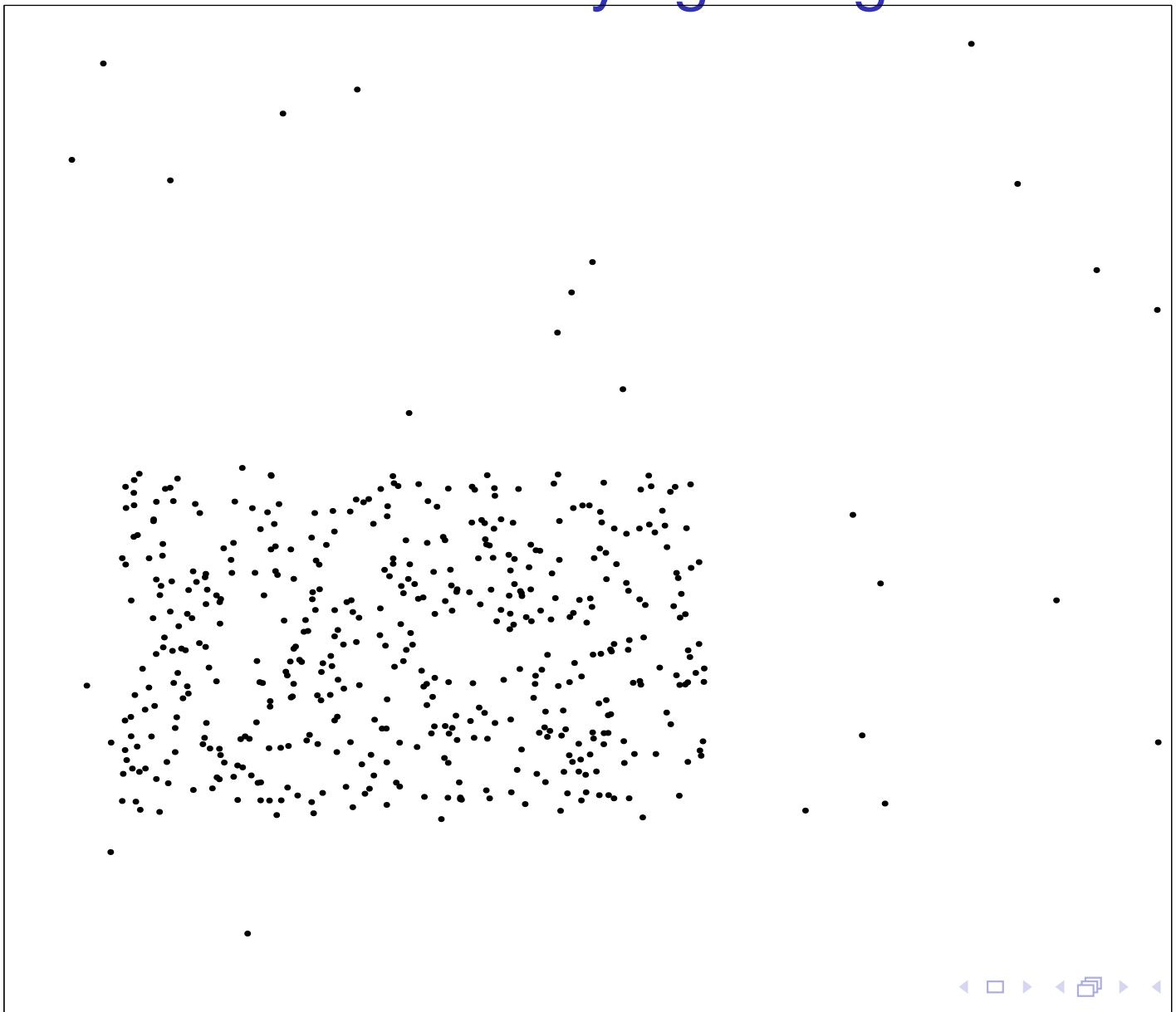
If the concept is very large, then negative examples are much more useful than positive ones.



Noise

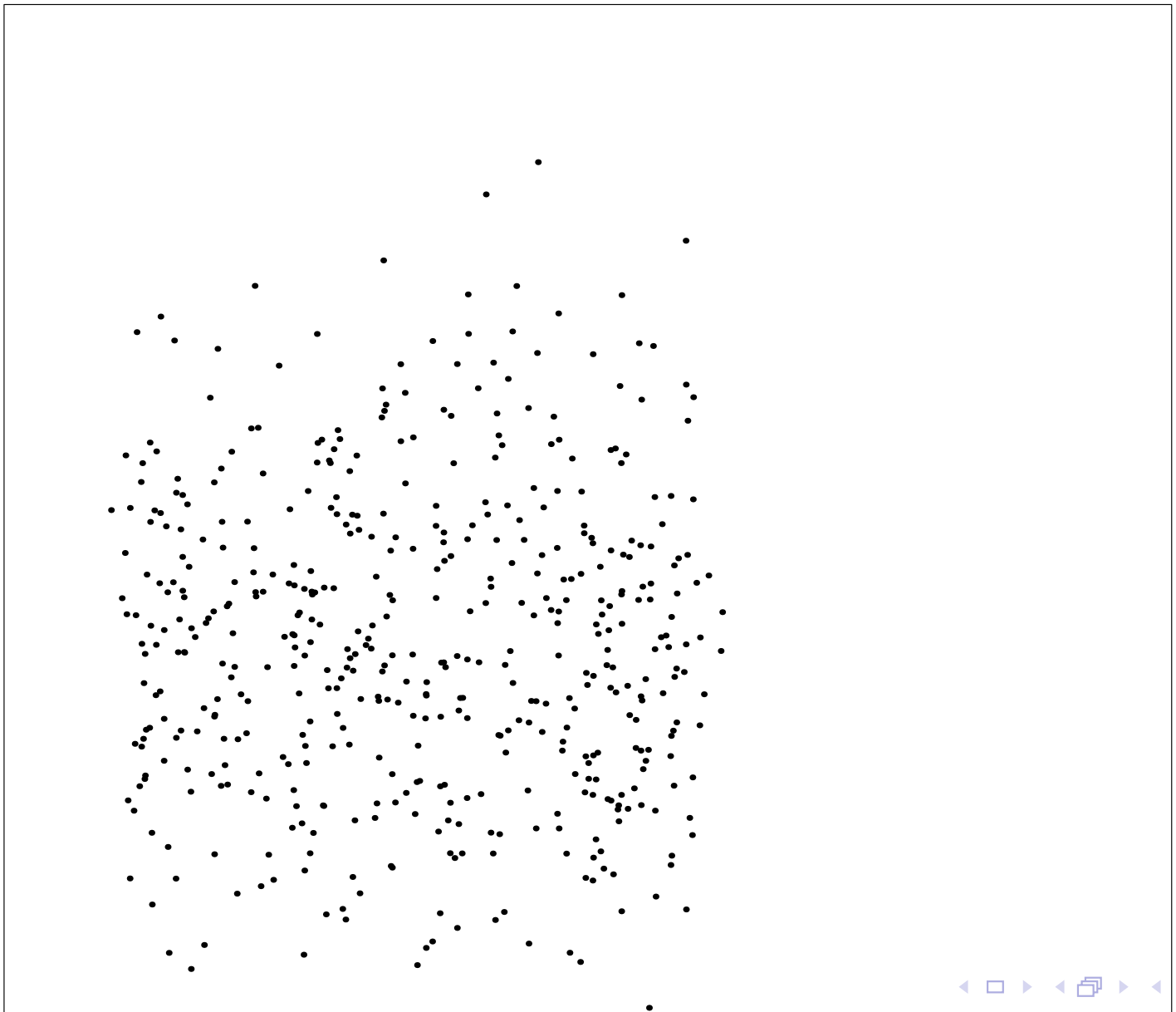
- In the real world, there is “noise”.
- A general problem in learning and perception for all domains.
- We are assuming that all of the examples are in the concept – i.e. are grammatical
- In reality, some will be misheard or corrupted in some way.

Noise with underlying categorial distinction



Noise with blurred boundary

Mixture

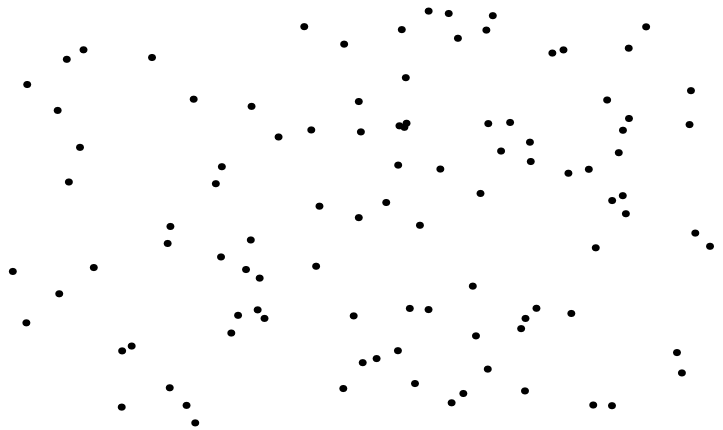


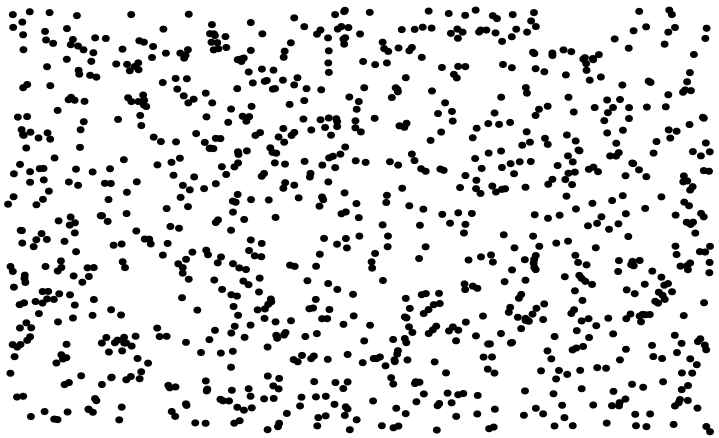
Negative evidence

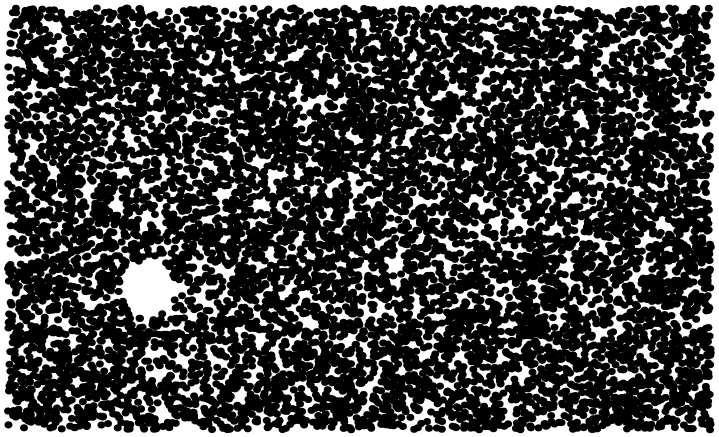
- Negative evidence is not considered a problem in NLP or in unsupervised learning: either in theory or in practice.
- Direct negative evidence is completely useless in practice: almost all long strings of English words are ungrammatical.
  - Jim address tasting array umpet tag ever zoo minibeasts dodo
  - party Victoria claps wrecking weakness spanked grips apricots lunchbox bell
  - surgery gymnast taxi washable ropes cleaner measurer Scotsman rummage gracious

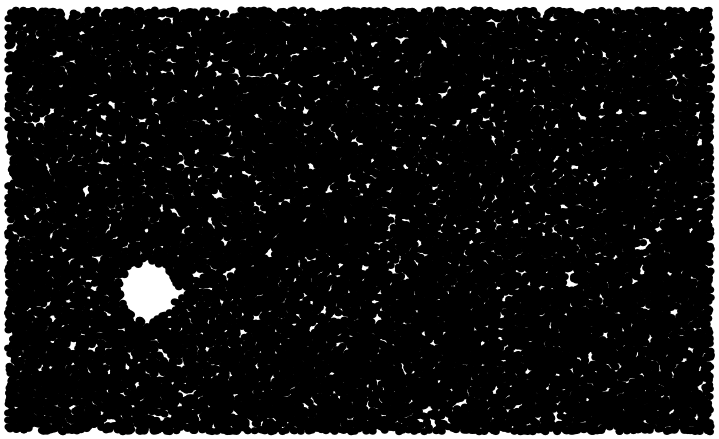
Overgeneralisation

- the claim is sometimes made that recovering from overgeneralisation is impossible with only positive data
- this example is too simple to exhibit this.









Shrinking the hypothesis

Why do we switch from the larger incorrect hypothesis that is too general to a smaller one, even though we have not seen any labeled negative examples?

Shrinking the hypothesis

Why do we switch from the larger incorrect hypothesis that is too general to a smaller one, even though we have not seen any labeled negative examples?

Indirect negative evidence

Because there is a space with no examples in, where we would expect there to be examples.

Outline

E Mark Gold

Language Identification in the Limit, 1967

Seminal paper and worth reading in the original.

- Almost the first formal model of learning
- Very influential but different from mainstream machine learning
- Still useful as it often leads to simple proofs and analysis
- Often forms the basis for APS arguments

Some typical quotes

Names removed to protect the innocent.

Quotes from some recent papers that allude to Gold's paper:

- The input does not include reliable negative evidence, . . . logical arguments suggest that in the absence of such evidence there must be strong innate constraints on the possible forms of grammars (and then a citation to Gold)
- (Gold) provided a logical proof which concluded that, without explicit error correction, the rules of a logical system with the structural complexity of a natural language grammar could not be inductively discovered, even in theory.

Some typical quotes (II)

Names removed to protect the innocent.

- Gold (1967) . . . obtained results that implied that natural languages could not be learned only on the basis of positive evidence
- Gold showed that, for even simple classes of languages, no procedure (statistical or other) exists that could learn a language without non-trivial a priori assumptions.
- The problem is presented even more strikingly by Gold (1967) who, simulating language acquisition on a computer, argues that an unbiased learner who had to induce the rules of grammar from strings of input would require more than a human lifetime.

Some typical quotes (III)

Names removed to protect the innocent.

- Gold asked the question: under what conditions is it possible to learn the correct context free grammar of a language given a set of training instances? His most significant result was that it is impossible to learn the correct language from positive examples alone. If a blind inductive program is given an infinite sequence of positive examples the program cannot determine a grammar for the correct context free language in any finite time.
- Gold (1967) proved that a general learner who has no *a priori* knowledge of the language to be learned cannot learn any language that has an infinite number of sentences from text presentation.

The Gold Paradigm

- In Gold's (1967) Identification in the Limit (IIL) paradigm a language consists of a set of strings
- There is a target language which is unknown to the learner T.
- The learner is presented with an infinite sequence of strings, possibly labeled as to whether they are grammatical or not
- At each step, the learner must produce a hypothesis $H_1, H_2, \ldots$
- As the learner gets more information, the hypotheses H_i should converge to the target T.

Alternative IIL Models

Two types of "presentation" of a language:

Positive data only

The input is a sequence of unlabelled examples from the languages: $s_1, s_2, \ldots$

- It never sees any sentences that are not in L
- Every sentence in L must appear at least once in the sequence.
- No other constraints at all on repetition, order etc.

Alternative IIL Models

Two types of "presentation" of a language:

Positive data only

The input is a sequence of unlabelled examples from the languages: $s_1, s_2, \ldots$

- It never sees any sentences that are not in L
- Every sentence in L must appear at least once in the sequence.
- No other constraints at all on repetition, order etc.

Positive and negative data

the examples are labeled as to whether they are in L or not in L
every possible string occurs in the sequence
no other constraints
Nothing like language acquisition

Convergence in the Gold Paradigm

- For a language L and a presentation of L the learner identifies in the limit the language L, if there is some N such that for all $n > N$, $G_n = G_N$, and G_N is a correct representation of L.
- IIL requires that a learner converge on the correct representation G_L of a language L in a finite but unbounded period of time.
- Alternatively, the learner only changes his hypothesis finitely many times, and ends on a correct hypothesis
- It only makes a finite number of mistakes.

IIL

IIL of a language

We say an algorithm A identifies in the limit a language L, if for every presentation of L, A converges in this sense.

IIL of a class of language

We say an algorithm A identifies in the limit a class of languages $\mathcal{L}$, if for every L in $\mathcal{L}$, A identifies in the limit L.

General properties

- Learnability is a property of classes of languages not languages

General properties

- Learnability is a property of classes of languages not languages
- If a class C of languages is not learnable, then any class that contains C is also not learnable.
- If a class C of languages is learnable then any smaller class is also learnable

Positive Evidence Only: The Class of Finite Languages

- The class of finite languages includes all and only languages with a finite number of strings.
- This class is itself infinite, as there are an infinite number of finite languages.

- **Gold Result 1:**
  The class of finite languages is identifiable in the limit on the basis of positive evidence.

The Rote learner

Prior knowledge/hypothesis class

Representation as a finite list or set
$G = \{w_1, w_2, \ldots, w_k\}$

Algorithm

Simply memorise the examples seen so far.

Proof of Gold Result 1

- If L has k elements, then for any presentation of L, there is a finite point N at which every element of L has appeared at least once.
- At this point G_N will be correct, and it will not change.

Positive Evidence Only: Finite Classes of Languages

- A finite class of languages $\mathcal{L}$ contains only a finite number of languages in the class.
- $\mathcal{L}$ may contain infinite languages, which are languages with an infinite number of strings.

- **Gold Result 2:**
  Any finite class of languages is identifiable in the limit on the basis of positive evidence.

Algorithm

A general algorithm for any finite class of languages.

Prior knowledge

Algorithm has an ordered list of the languages in the class.
$L_1, L_2, \ldots, L_n$

Learner

Pick the first element in the list that includes the data seen so far.

Proof of Gold Result 2

- Assume that the list is ordered so that if $L_i \subset L_j$, then L_i occurs before L_j (i.e. $i < j$)
- Suppose the target is L_k.
- A will never return a hypothesis after L_k.

Proof of Gold Result 2

- Assume that the list is ordered so that if $L_i \subset L_j$, then L_i occurs before L_j (i.e. $i < j$)
- Suppose the target is L_k.
- A will never return a hypothesis after L_k.
- Every hypothesis L_i before L_k does not contain L_k, and so there is a string e_i in $L_k \setminus L_i$

Proof of Gold Result 2

- Assume that the list is ordered so that if $L_i \subset L_j$, then L_i occurs before L_j (i.e. $i < j$)
- Suppose the target is L_k.
- A will never return a hypothesis after L_k.
- Every hypothesis L_i before L_k does not contain L_k, and so there is a string e_i in $L_k \setminus L_i$
- Once it sees e_i it will no longer return L_i
- So once it sees $\{e_1, \ldots, e_{k-1}\}$ it will have converged.

Note that it may be difficult to find such an ordering of the list.

Positive Evidence Only: Supra-Finite Languages

Negative result

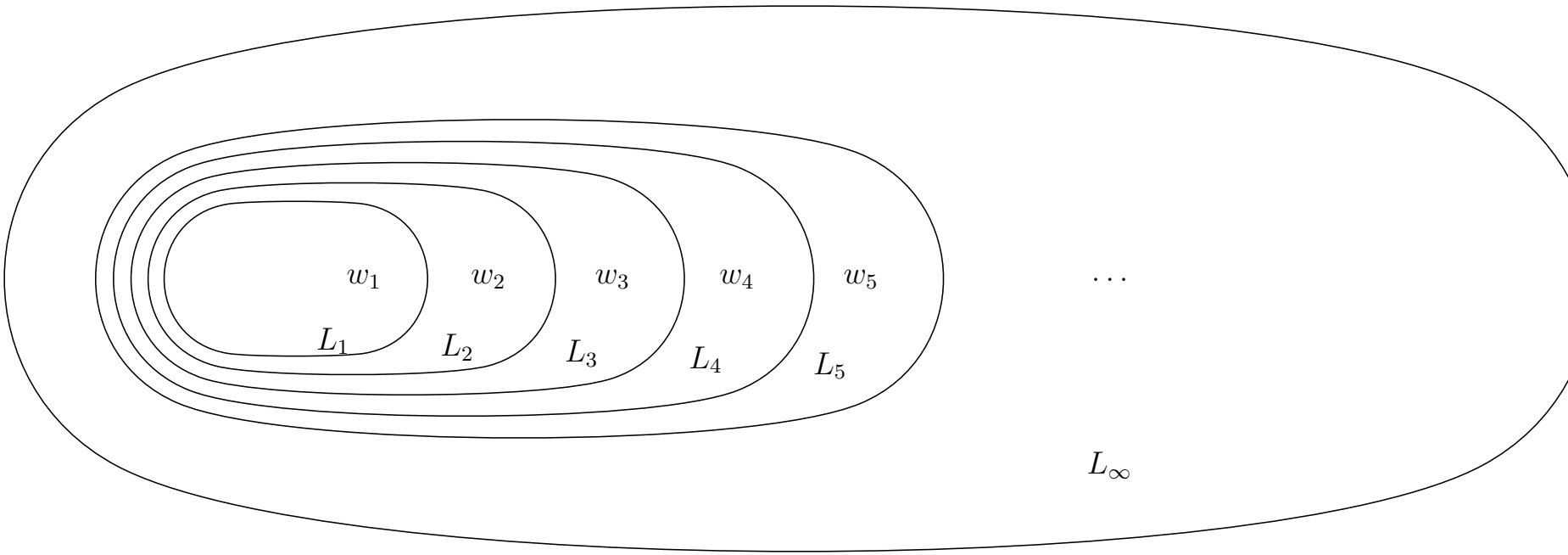
- A supra-finite class of languages is any class that contains all finite languages and at least one infinite language.
- The most influential Gold result proves the non-learnability in the limit of any such class of languages in the positive evidence only IIL model.

- **Gold Result 3:**
  Any class that is supra-finite is not identifiable in the limit on the basis of positive evidence.

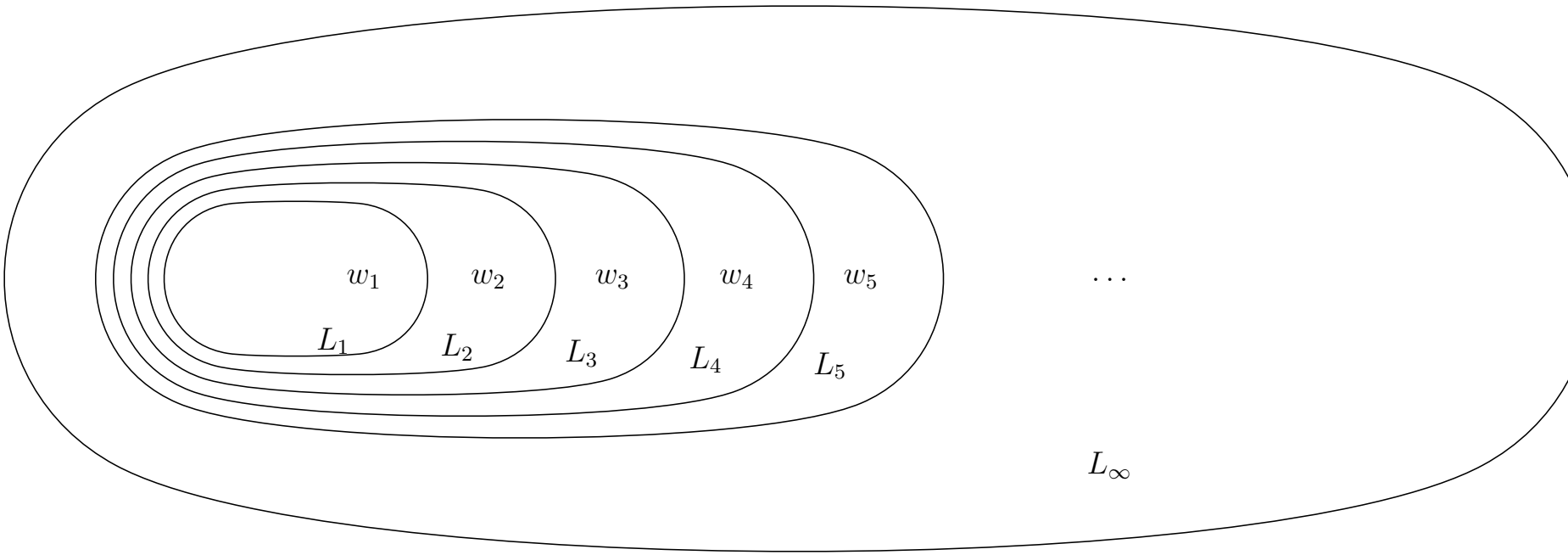
Corollary: the classes of regular/context-free/context-sensitive languages are not IIL from positive data.

Proof of Gold Result 3

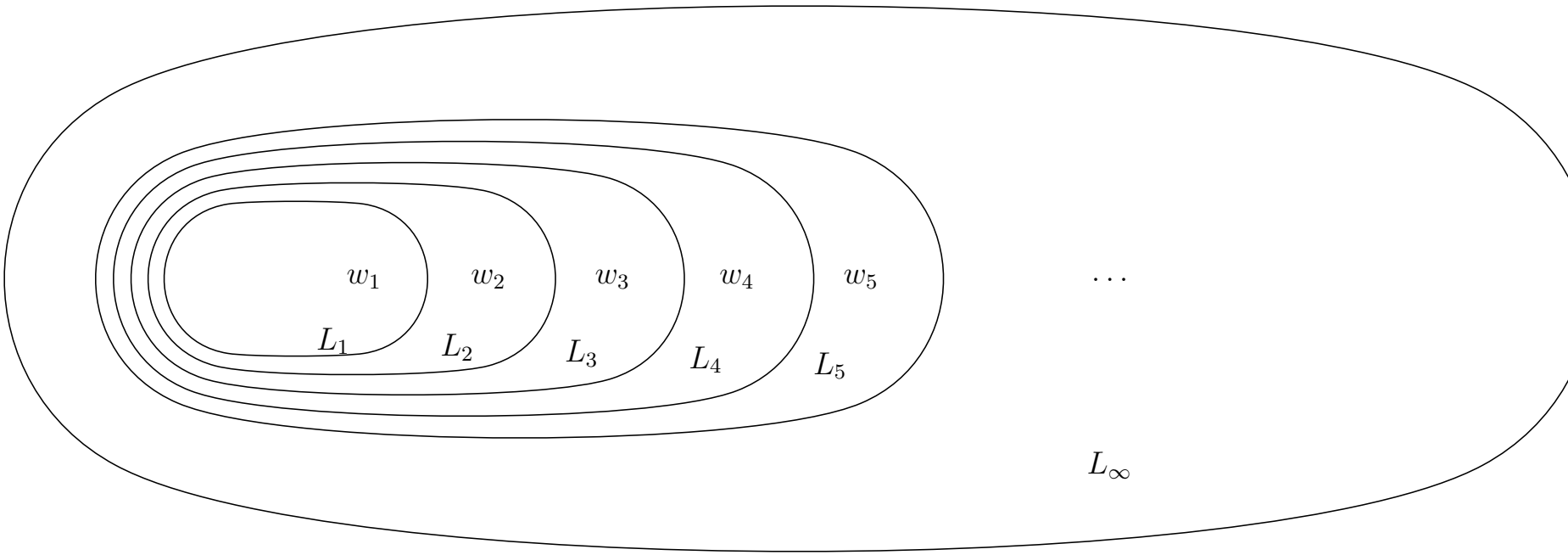
- Take $\mathcal{L}$ to be a supra-finite class of languages, and let $L_\infty \in \mathcal{L}$ be an infinite language.
- Arrange the elements of L_∞ in an infinite sequence $w_1, w_2 \ldots$.
- $L_1 = \{w_1\}$, $L_2 = \{w_1, w_2\}$ $\ldots$
- Suppose that there is an algorithm $\mathcal{A}$ that can identify $\mathcal{L}$ in the limit.
- We construct a presentation on which $\mathcal{A}$ fails to converge, which demonstrates that there can be no such $\mathcal{A}$.



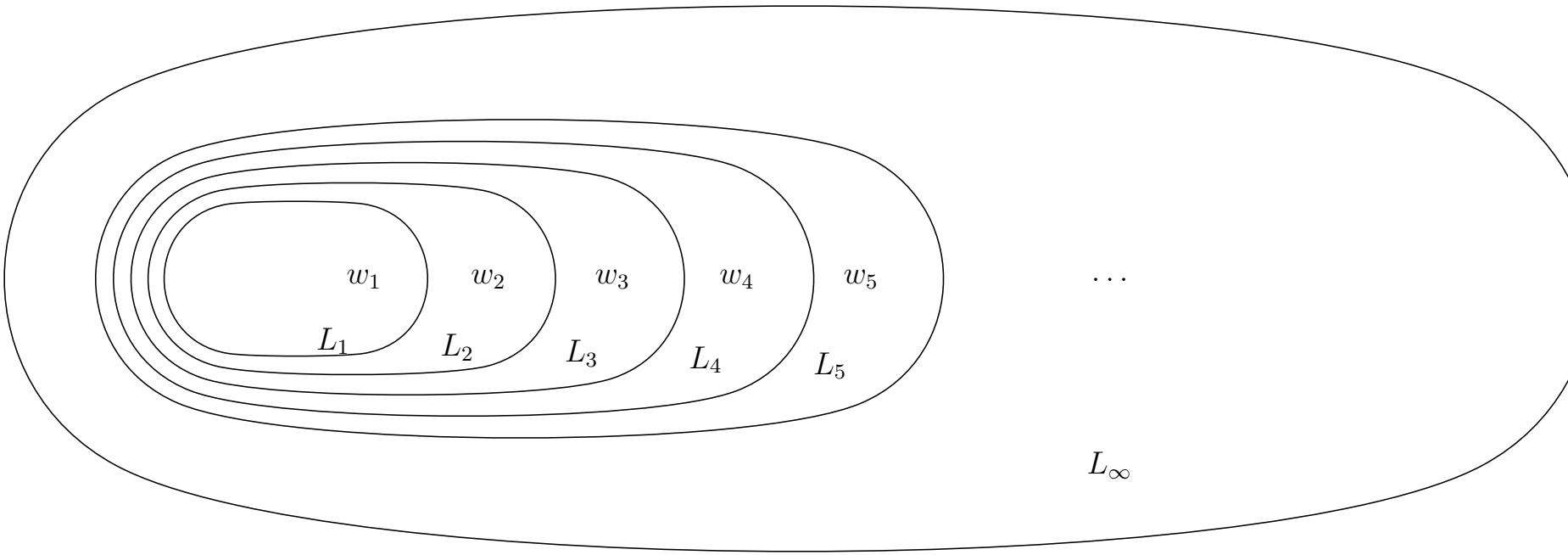
$w_1, w_1, w_1,$



$w_1, w_1, w_1, w_2, w_2, \ldots, w_2,$



$w_1, w_1, w_1, w_2, w_2, \ldots, w_2, w_3, \ldots$



$w_1, w_1, w_1, w_2, w_2, \ldots, w_2, w_3, \ldots w_4, \ldots, w_n \ldots w_{n+1} \ldots$

Comments on the proof

- The learner must learn for every possible presentation
- Even when the presentation is by an adversary
- Adversary can look inside the head of the learner
- Very pessimistic result because the learning environment is unrealistically strict

Positive and Negative Evidence: The Class of Recursive Languages

- A recursive language L is a language for which there is a primitive recursive function that enumerates L's string set S, and L's complement set S'.
- The class of recursive languages includes the class of context-sensitive languages as a proper subclass.

- **Gold Result 4:**
  The class of recursive languages is identifiable in the limit in the model in which the learner has access to both positive and negative evidence for each string in a presentation.

Compatibility

At any given time we will have seen:

- Some positive examples: $p_1, \ldots, p_i$
- Some negative examples: $n_1, \ldots, n_j$

A hypothesis H is compatible with the data if

- All of the positive examples are in $L(H)$
- None of the negative examples are in $L(H)$

Compatibility

At any given time we will have seen:

- Some positive examples: $p_1, \ldots, p_i$
- Some negative examples: $n_1, \ldots, n_j$

A hypothesis H is compatible with the data if

- All of the positive examples are in $L(H)$
- None of the negative examples are in $L(H)$

If our hypothesis is wrong, then eventually we will know it.

Algorithm

Weak prior knowledge

An infinite list of all of the representations in no particular order.

Algorithm

Return the first in the list that is compatible with the data so far.

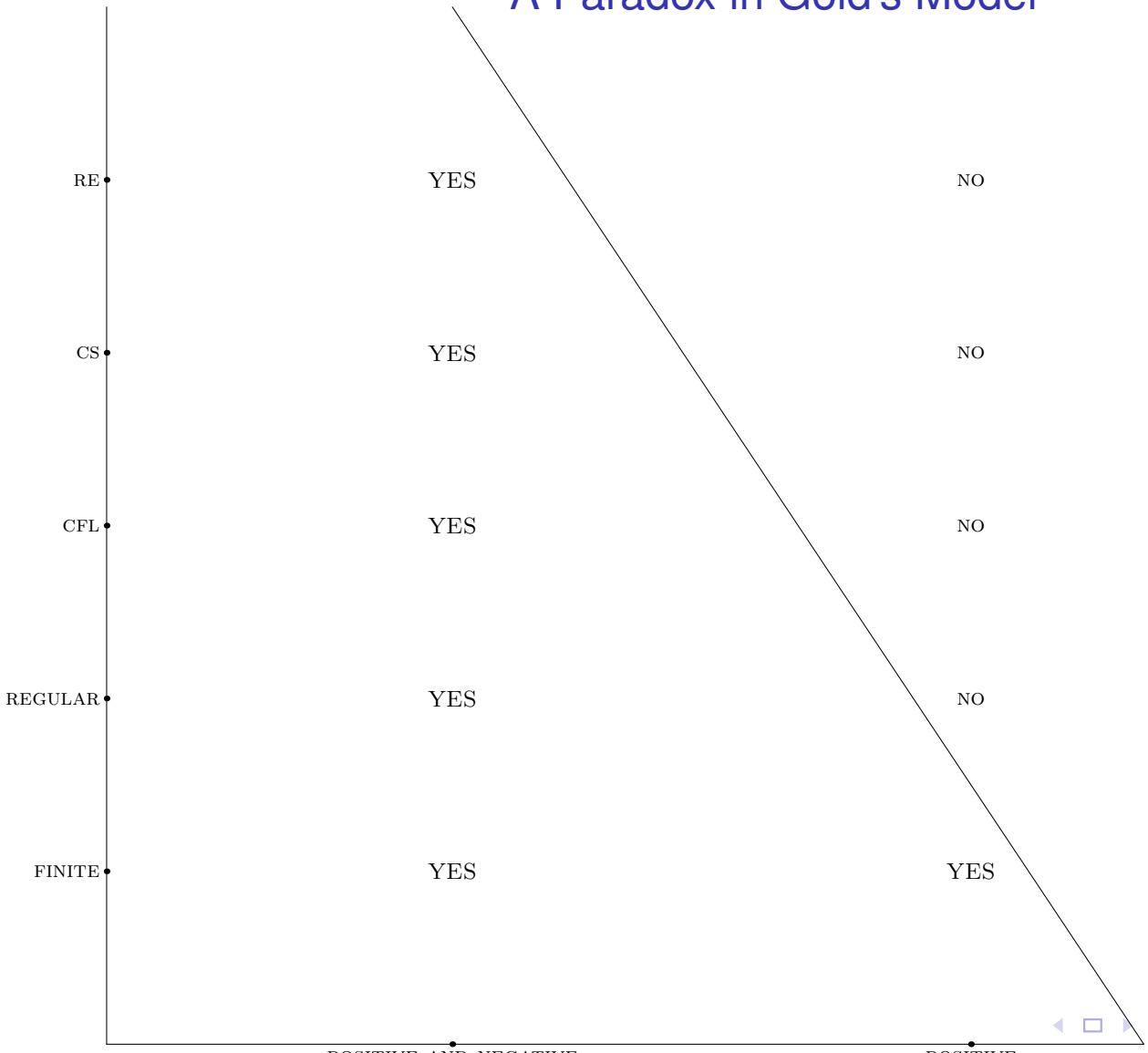
Proof of Gold Result 4

Suppose the right answer is at position i in the list:

- We will never hypothesise an answer after i.
- For any j before i, there will be some data that is incompatible with j.
- Therefore we will at some point reject all hypotheses before i.

Summary

A Paradox in Gold's Model



Discussion

- This model motivates parametric models of acquisition.
- Gold positive only model is way too pessimistic.

Outline

Key point

Probabilistic learning radically changes the learnability results.

Angluin (1988)

> *These positive results on learning large classes of languages from stochastically generated positive examples suggest that the assumption of stochastically generated samples is able to compensate for the lack of explicit negative information in the samples.*

Indirect Negative Evidence

Chomsky (1981):

> *A not unreasonable acquisition system can be devised with the operative principle that if certain structures or rules fail to be exemplified in relatively simple expressions, where they would expect to be found, then a (possibly marked) option is selected excluding them in the grammar, so that a kind of "negative evidence" can be available even without corrections, adverse reactions etc.*

Independence Assumptions for Statistical Learning

Basic assumptions:

- The events are independent of each other
- The distribution is constant and doesn't change.
- This assumption is specified in the principle that events are *Independently and Identically Distributed* (IID).
- The IID is an idealization, and it is open to obvious challenges in the case of sentences uttered in a discourse. (i.e. it's false!)
- Local dependencies clearly do exist among sentences in particular discourse contexts.
- Probabilities of sentences change with time of day, weather, from year to year etc.

Examples of dependencies

Dependencies

- Intersentential dependencies: polar interrogatives followed by yes/no.
- Some more linguistically interesting: “Where did who go?”

Non stationarity

- Good morning versus good evening
- 2011 vs 2010 vs 2009 . . .

Idealisations

If we are learning syntax, then we ignore intersentential factors.

Law of large numbers

Law of large numbers (LLN)

The average number of times we see something converges to the probability.
the average number of heads when flipping coins converges to 50%

Law of large numbers

Law of large numbers (LLN)

The average number of times we see something converges to the probability.
the average number of heads when flipping coins converges to 50%

- IID assumptions imply the LLN.
- What we need for learning is the LLN not IID
- Weaker more complicated assumptions lead to the LLN as well (ergodic, rapidly mixing etc.)

We will use the IID as a place holder for more realistic assumptions.

Chomsky on Statistical Modeling of Grammar

A general antipathy to probabilistic methods.

- Chomsky (1957) rejects the use of statistical methods to represent the distinction between grammatical and ungrammatical strings.
  1. Colourless green ideas sleep furiously.
  2. Furiously sleep ideas green colourless.
- (1) and (2) both have a probability approaching nil (in 1957) of appearing in a corpus or actual speech.
- (1) is syntactically well formed, even if semantically anomalous, while (2) is not.

Chomsky on Statistical Modeling of Grammar

Chomsky (1957) (p. 17)

If we rank the sequences of a given length in order of statistical approximation to English, we will find both grammatical and ungrammatical sequences scattered throughout the list; there appears to be no particular relation between order of approximation and grammaticalness. Despite the undeniable interest and importance of semantic and statistical studies of language, they appear to have no direct relevance to the problem of determining or characterizing the set of grammatical utterances. I believe that we are forced to conclude that grammar is autonomous and independent of meaning, and that probabilistic models give no particular insight into some of the basic problems of syntactic structure.

Several different problems

Chomsky's argument confuses:

- The limitations of a finite state models
- The limitations of maximum likelihood estimation
- The difference between probabilities in the model and frequency in the training data
- Semantically and syntactically anomalous sentences

Languages and Distributions

Distribution

A distribution D over Σ^* assigns a probability to every sequence of words s

this number represents how likely that sentence is to be uttered

- A language model specifies a probability distribution for the strings of a language.
- It is reasonable to ask whether learning a language L involves acquiring a distinct formal representation of L, or whether we can reduce knowledge of L to knowing its distribution.
- In the former case the target of probabilistic learning is a (possibly non-probabilistic) grammar.
- In the latter the language model is itself the target of learning, and languages are identified directly with their distributions.

Arguments for Reducing Languages to Distributions

- There are strong arguments for both views.
- Identifying languages with their distributions is motivated by the fact there is a substantial amount of psycholinguistic evidence showing that frequency based learning is central to language acquisition.
- Assuming that knowledge of a language consists in mastering a language model provides a natural and direct explanation for our capacity to filter out the noise of ill formed sentences in the PLD.
- Taking the target of acquisition to be a language model eliminates an additional formal object, and so simplifies the account of language learning.

Arguments against Reducing Languages to Distributions

- It is not possible to identify grammaticality directly with high frequency of occurrence (see later)
- It confuses real world knowledge with linguistic knowledge
- “My dog broke its wing” is rare because dogs do not have wings, not because of anything about English.

Learning distributions

Trivial example

We have only two sentences in our language: A and B
The distribution is just $p(A) = \alpha$, $p(B) = (1 - \alpha)$
We observe an infinite sequence of As and Bs
We want to learn the value of α

Learning distributions

Trivial example

We have only two sentences in our language: A and B
The distribution is just $p(A) = \alpha$, $p(B) = (1 - \alpha)$
We observe an infinite sequence of As and Bs
We want to learn the value of α

Learning model

We want the hypothesis distribution to be close to the true target distribution.

Asymptotic results for learning distributions

If we want to learn distributions then we have very powerful positive results.

Horning, 1969

PCFGs can be learned
A limited result but has been greatly extended by subsequent work

Angluin, 1988

Any “computable” distribution can be learned from IID samples
Any “computably approximatable” distribution can be learned from IID samples

Chater and Vitanyi, 2007

Abandons IID assumption and gets a similar result.

Efficiency

- These results ignore computational complexity of the inference process – they allow an unbounded running time.
- They also ignore the amount of data that the learner can use – they allow unbounded amounts of data
- They are impossible to run on non-trivial problems
- They are too optimistic as models of learning — so these positive results are not informative.
- There are efficient algorithms for learning some classes of distributions – e.g. PDFAs.

Appropriate model

Probabilistic model

Distribution of appropriate type over positive and negative examples without labels

Non-Probabilistic model

Gold style positive presentation plus membership queries

Gold's Conclusions on Language Acquisition

"If one accepts identification in the limit as a model of learnability, then this conflict must lead to at least one of the following conclusions:

1. The class of possible natural languages is much smaller than one would expect from our present models of syntax. That is, even if English is context-sensitive, it is not true that any context sensitive language can occur naturally. Equivalently, we may say that the child starts out with more information than that the language it is presented is context-sensitive.
2. The child receives negative instances by being corrected in a way we do not recognise.
3. There is an a priori restriction on the class of texts which can occur such as a restriction on the order of text presentation."

Four possibilities

Options

- Reject IIL as a model
- Restrict the class of languages (but this does not mean that the hypothesis class needs to be restricted)
- Argue for negative evidence of some kind
- Consider a restriction (e.g. stochastic presentation) on the input data

Outline

Efficient learnability

The *Tractable Cognition Thesis* (van Rooij, 2008)

> *Human cognitive capacities are constrained by the fact that humans are finite systems with limited resources for computation.*

Efficient learnability

The *Tractable Cognition Thesis* (van Rooij, 2008)

> *Human cognitive capacities are constrained by the fact that humans are finite systems with limited resources for computation.*

This applies to parsing, generation, learning equally.

Abstraction

Modeling brains rather than computers . . .

Two Kinds of Complexity for Learning

Sample Complexity

- The learner must only use limited amounts of data.

Computational complexity

- It must also be able to process this data within the bounds of time and computational resources available to it to solve the learning task

These are distinct, but interact.

General theory of complexity

Introduction to basic ideas of complexity in theoretical computer science

Standard tool

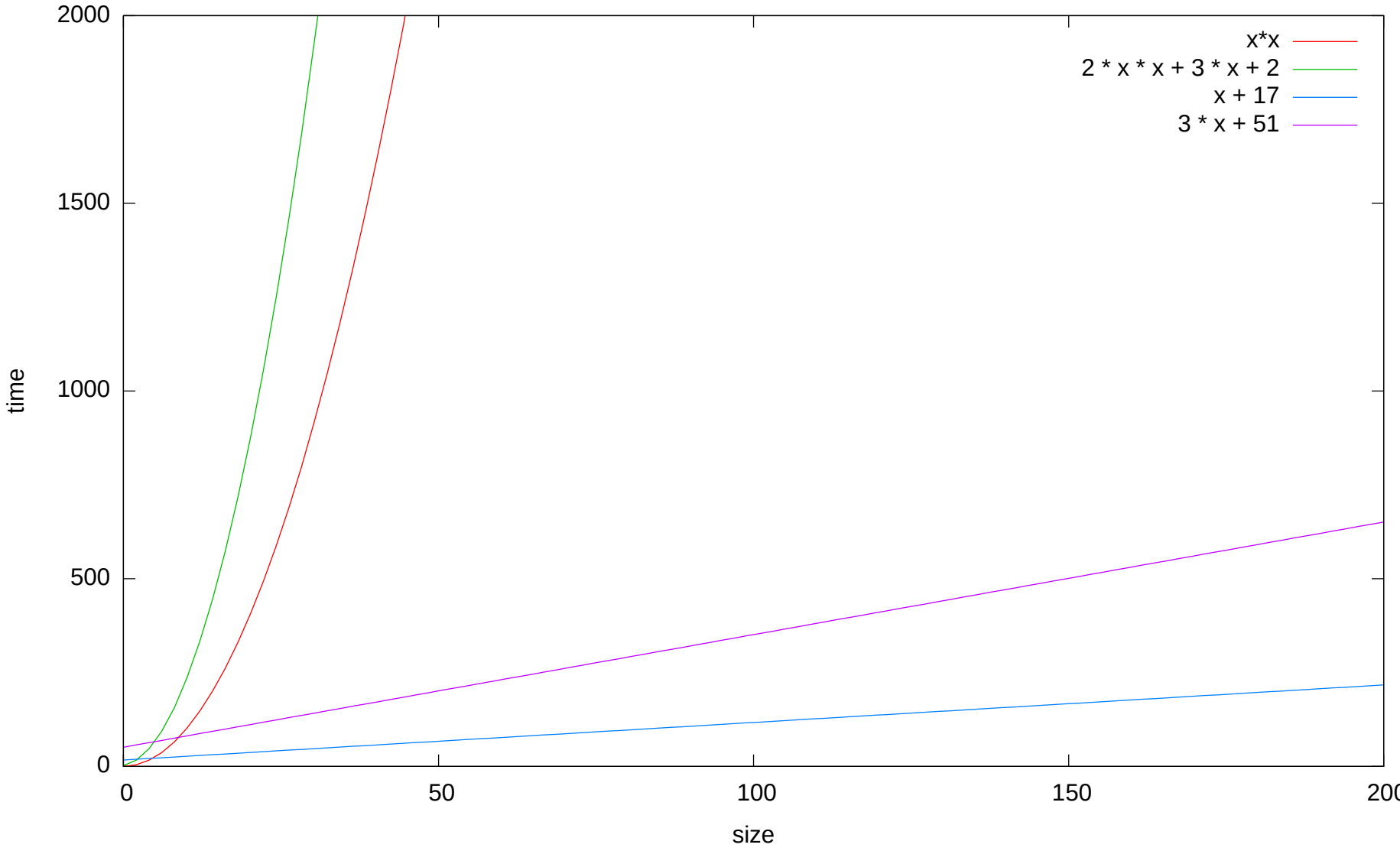
Worst case asymptotic polynomial

"As we solve larger and more complex problems with greater computational power and cleverer algorithms, the problems we cannot tackle begin to stand out. The theory of NP-completeness helps us understand these limitations and the P versus NP problem begins to loom large not just as an interesting theoretical question in computer science, but as a basic principle that permeates all the sciences." (Fortnow, 2009)

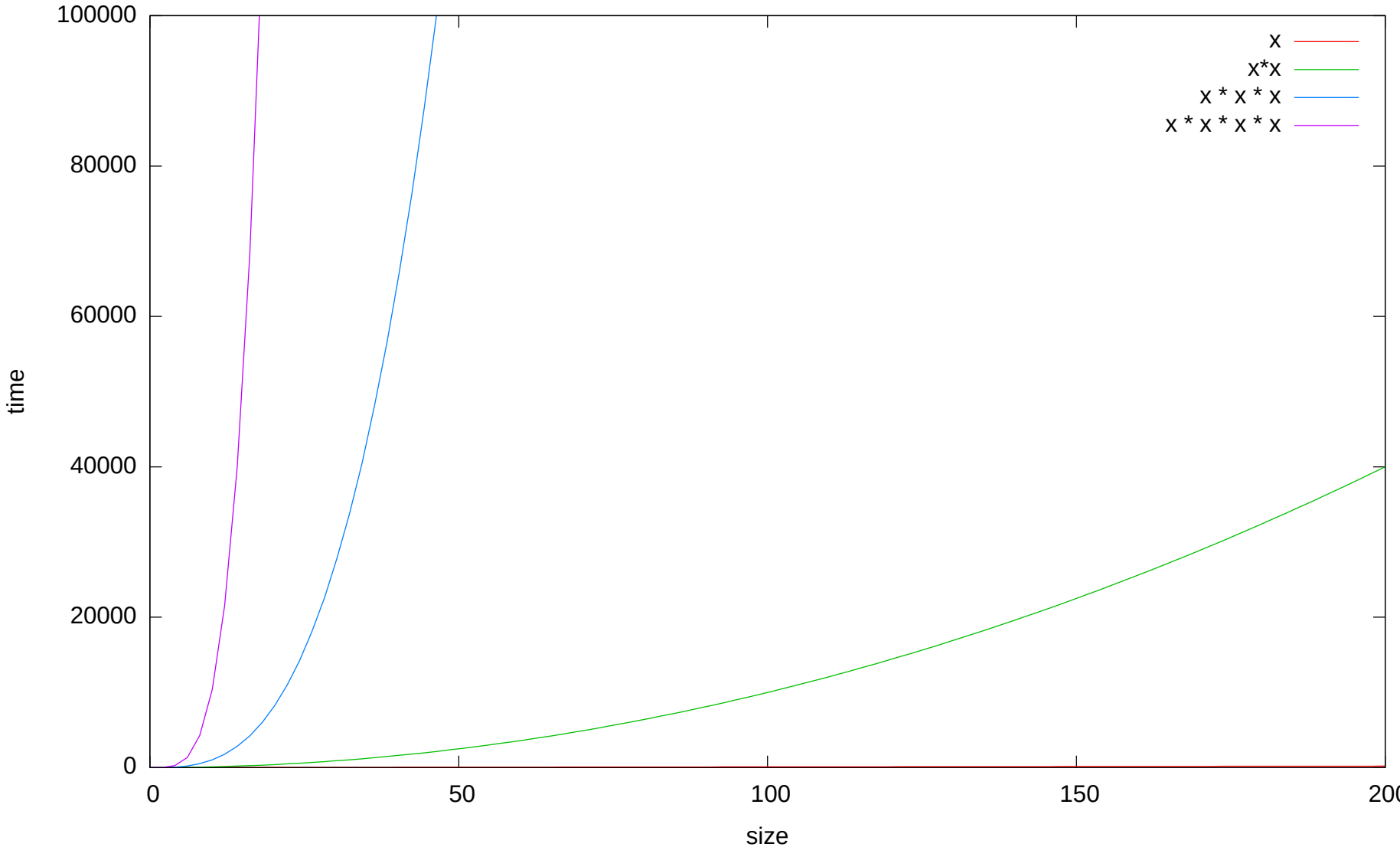
A Complexity Hierarchy

- The complexity of an algorithm is a measure of the resources in space and time that it needs to complete the task for which it is designed.
- This measure is expressed as a function of the size of the input to which it applies.
- The complexity property of an algorithm is specified in terms of the maximum (ie. worst case) quantity of resources that it needs to complete a task.
- Complexity properties form a hierarchy of processing difficulty.

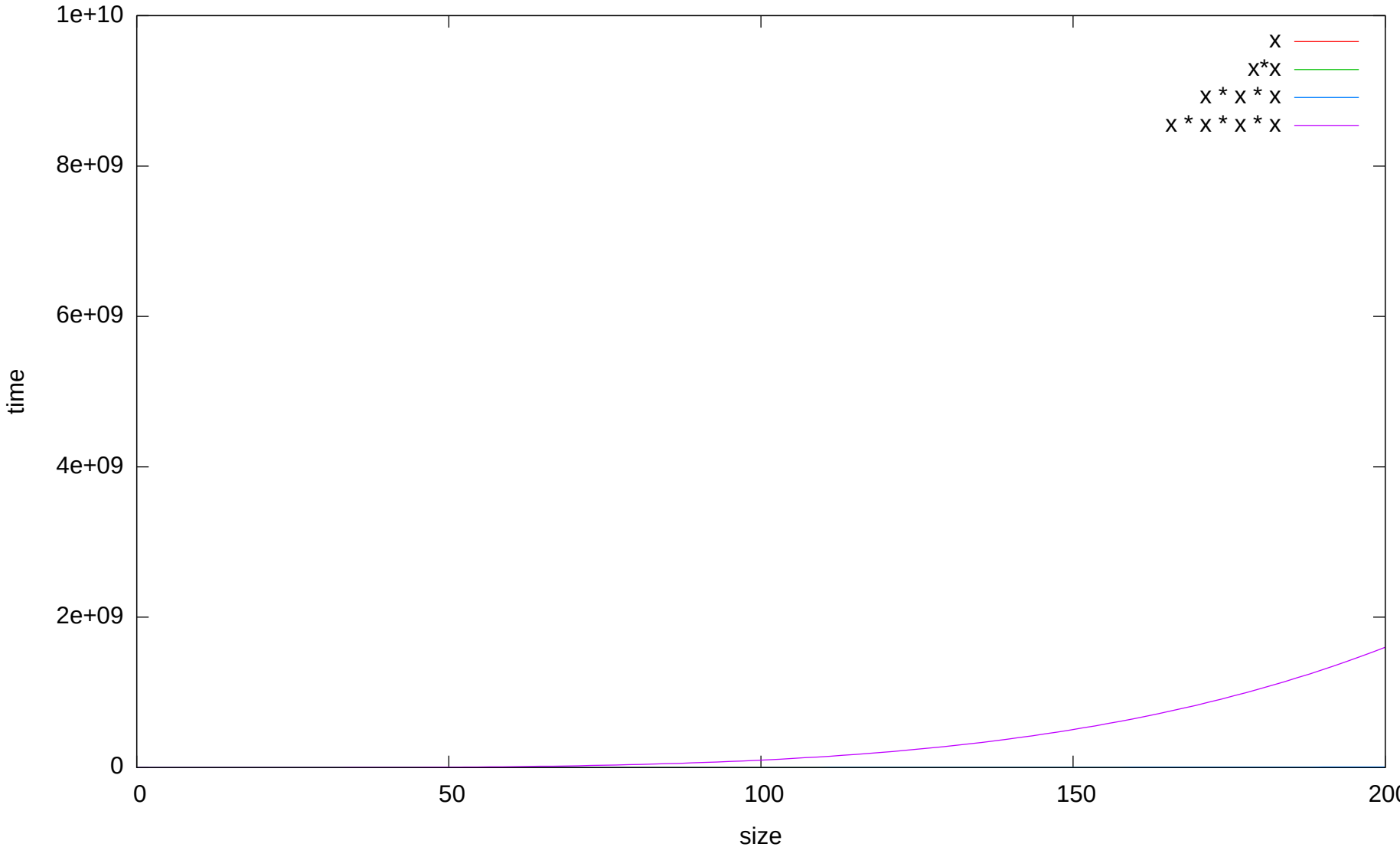
Complexity diagrams



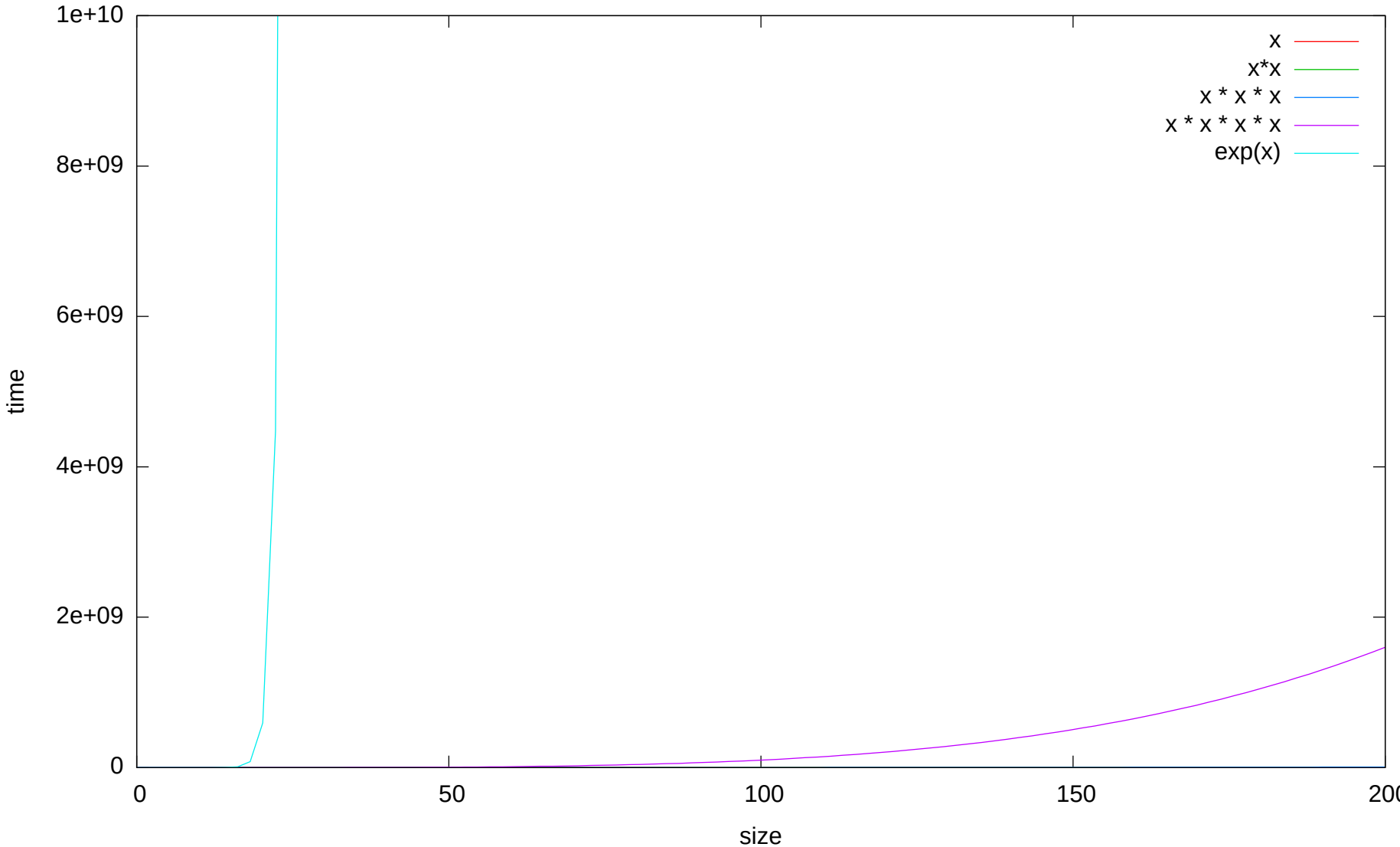
Complexity diagrams



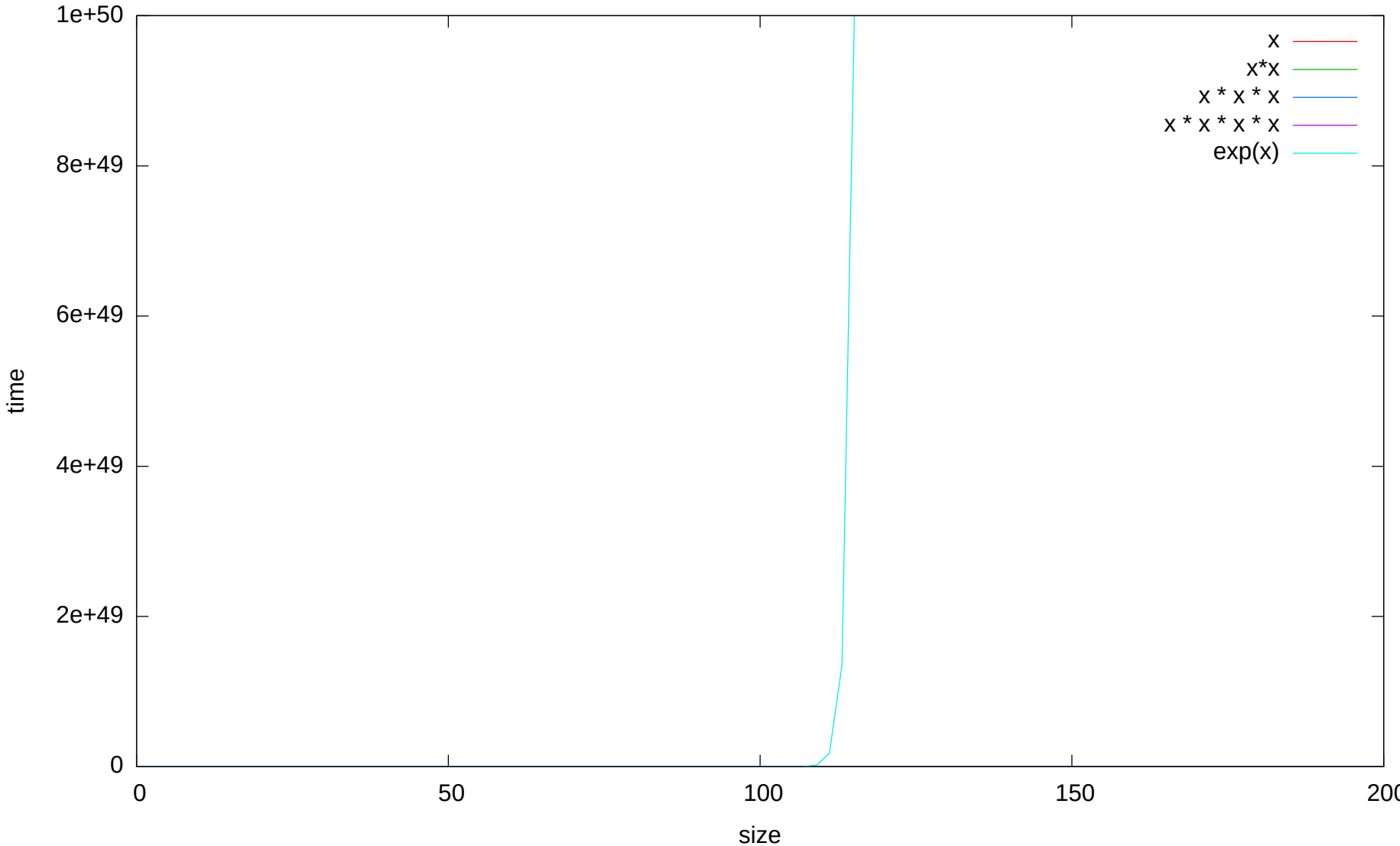
Complexity diagrams



Complexity diagrams



Complexity diagrams



NP-Completeness

- Tasks of polynomial complexity and below are considered tractable.
- Those of exponential complexity are not.
- The set of problems which can be solved by non-deterministic polynomial (NP) procedures is known as *NP-complete*.
- These problems are generally regarded as at least exponential in complexity, but this result has not been proven.
- The question of whether or not P(olynomial) = NP remains an important open problem in theoretical computer science.

Decidability and Complexity

Decidability

Is there an algorithm that can perform a task?

Tractability

Is there an efficient algorithm that can perform a task?

- The decidability of a problem is distinct from the complexity involved in solving it.
- It can be the case that a problem is decidable, but solving it may be an intractable task.

Efficient Learning and the Target Representation

- The size of the representation of the target class is a central factor in determining the learnability of that class.
- If a representation is specified as a grammar G, its size can be measured in terms of the number of rules, the length of these rules, and the number of (non-vocabulary) symbols in G.
- So, for example, the size of a phrase structure grammar is specified in terms of the number and length of its production rules, and the cardinality of its non-terminal symbols.
- For learning to be efficient there must be a tractable (at most polynomial) function that expresses the rate of growth of the data and the computation required for learning, relative to the size of the target representation.

Efficient Learning and the Target Representation

- A learning algorithm $\mathcal{A}$ may be efficient in sample complexity, but it will still not be able to learn very large representations.
- Assume that the size of the data set that $\mathcal{A}$ requires is the square of the size of the target representation G.
- If G is of cardinality 100, then $\mathcal{A}$ needs only $100^2 = 10,000$ samples to learn it, which is not particularly demanding.
- However, if G is of cardinality 10^{100}, then $\mathcal{A}$ must have 10^{200} data samples to acquire G, which is considerably larger than the size of the PLD accessible to human language learners.

Compact vs. Large Representations

- Two distinct types of representation may be weakly equivalent in that they generate the same language class.
- But they may differ significantly in their size and their computational properties.
- It is frequently the case that a more expressive grammar can be more compact than a less expressive one.
- However, the rules of the larger grammar may be easier to learn from the observable data.

Regular and Context Free Grammars

Perfors et al. 2008

CFG rules for NP and N

$NP \rightarrow NP\ PP \mid NP\ CP \mid NP\ C \mid N \mid det\ N \mid adj\ N\ pro \mid prop$
$N \rightarrow n \mid adj\ N$

RG rules for NP and N

$NP \rightarrow pro \mid prop \mid n \mid det\ N \mid adj\ N\ pro\ PP \mid prop\ PP \mid n\ PP \mid det\ N_{PP} \mid adj\ N_{PP}\ pro\ CP \mid prop\ CP \mid n\ CP \mid det\ N_{CP} \mid adj\ N_{CP}\ pro\ C \mid prop\ C \mid n\ C \mid det\ N_C \mid adj\ N_C\ N \rightarrow n \mid adj\ N$

$N_{PP} \rightarrow n\ PP \mid adj\ N_{PP}$
$N_{CP} \rightarrow n\ CP \mid adj\ N_{CP}$
$N_C \rightarrow n\ C \mid adj\ N_C$

Finite variation

- Morphological variation for nouns and verbs is finite.
- The verbs and nouns in a language may admit of a large set of alternative morphological forms.
- For example, Hebrew verbs are expressed through seven binyanim, and Arabic through fourteen.
- When these binyanim are combined with other features of verb morphology (person, number, tense, and gender), they generate a large matrix of verb forms.
- These forms are more compactly represented by rules than by tables.
- Given the limited amount of data available in the PLD, learning requires the compact rule representation of the target, rather the large finite variation table.

VC Dimension and Shattering

- The Vapnik-Chervonenkis (VC) dimension of a concept space $\mathcal{C}$ is a measure of $\mathcal{C}$'s complexity for PAC learning.
- It expresses a relation between $\mathcal{C}$ and the samples of a data set in terms of the maximal number of data points in a sample that the elements of $\mathcal{C}$ can cover or *shatter*.
- The VC dimension of $\mathcal{C}$ is a crucial factor in determining the learnability of the class.

Learning in Finite and Infinite Concept Spaces

- Finiteness of $\mathcal{C}$ does not insure computational efficiency of PAC learning.
- Conversely, tractable convergence is possible in certain cases of an infinite $\mathcal{C}$.
- Hence, in the PAC framework the finiteness assumptions of the P&P view of UG do not, in themselves, solve the learning theoretic problems posed by language acquisition.

Efficiency of Learning in a Finite Concept Space

- The number of training examples required scales with $\log |\mathcal{C}|$.
- Within the P&P framework, assuming n binary parameters, the size of the concept space is $|\mathcal{C}| = 2^n$.
- The number of examples scales then with $\log 2^n$ which is just n.
- However, the size of $|\mathcal{C}|$ grows exponentially with the number of parameters, and so finding the best hypothesis becomes increasingly difficult.
- If the parameters are interdependent and difficult to estimate from observed data, then a finite class may still not be efficiently learnable.

PAC Learning in an Infinite Concept Space

- The VC-dimension of the space is critical in determining the rate of convergence on a target for an infinite concept space.
- The VC-dimension of $\mathcal{C}$ is the largest value of m such that there is a training sample of size m that is shattered by $\mathcal{C}$.
- A training sample is shattered by $\mathcal{C}$ if, for each of the 2^m possible labelings of a sample (assignments from {0,1} to its elements), there is a concept in $\mathcal{C}$ that assigns that labeling.

Shattering and VC Dimensions: An Example

- Assume that each member of $\mathcal{C}$ is associated with n *real-valued* parameters, and so the concept space is uncountably infinite.
- Suppose, for example, that the function to be learned maps points in one-dimensional space onto 0 and 1.
- A concept space $\mathcal{C}$ is a subset of all possible functions of this kind.

Shattering and VC Dimensions: An Example

- $\mathcal{C}$ might, for instance, contain just those functions that assign 1s to all points within an interval *int* of a line and 0s to all points outside of *int*.
- The VC dimension of $\mathcal{C}$ is the cardinality of the largest set of points for which all possible labelings of the points are expressed by elements of $\mathcal{C}$ (they are shattered by $\mathcal{C}$).
- The VC dimension of a $\mathcal{C}$ consisting only of *int* functions is 2, as the hypotheses in $\mathcal{C}$ shatter any set of 2 points in a line, but not all sets of 3 points.

Intervals in a Real Number Line

Kearns and Vazirani (1994)

```
    <-----(X)----(Y)------>

a. <-----(1)----(1)------>
b. <-----(1)----(0)------>
c. <-----(0)----(1)------>
d. <-----(0)----(0)------>
```

```
a. <---[-(1)----(1)-]---->
b. <---[-(1)-]--(0)------>
c. <-----(0)--[-(1)-]---->
d. <-----(0)----(0)-[-]-->
```

Intervals in a Real Number Line

VC dimension is 2

The pair of points in a-d can be covered by all possible labelings that the interval hypotheses in $\mathcal{C}$ specify.

e. <---(1)---(0)---(1)--->

The labeling of the triple in e cannot be expressed by any single bracketing.

Tractable Learning in an Infinite Concept Space

- A concept space $\mathcal{C}$ has infinite VC-dimension if for any value of m, there is a training sample of size m that is shattered by $\mathcal{C}$.
- PAC learning is possible if and only if the VC-dimension of the concept space is finite.
- The number of training examples required is roughly linear in the VC-dimension of the concept space, and so efficient PAC learning is possible in an infinite $\mathcal{C}$ if the VC-dimension of $\mathcal{C}$ is relatively small.
- It is possible to improve the convergence rate for PAC learning by adding a *prior bias* over the elements of $\mathcal{C}$.

Efficient learning

Sample complexity polynomial

For any ϵ, δ there is a polynomial $p(1/\epsilon, 1/\delta)$ that is an upper bound on the number of samples the learner needs.

- Pick $\epsilon = 0.01, \delta = 0.001$
- We then get a number N, say, 1,000,000.
- For any concept, if we get N samples we have to produce a hypothesis which is 99% accurate with a probability of at least 0.999.
- This is independent of how complex the target concept is.

PAC Learning and the Class of Finite Languages

- We can easily see that any finite set of strings is shattered by the class of finite languages.
- Therefore the class of finite languages has infinite VC dimension.
- Therefore, this class is unlearnable in the PAC framework.
- By contrast, the class of finite languages is identifiable in the limit in Gold's positive evidence only model.

PAC Learning and the Class of Finite Languages

- We can easily see that any finite set of strings is shattered by the class of finite languages.
- Therefore the class of finite languages has infinite VC dimension.
- Therefore, this class is unlearnable in the PAC framework.
- By contrast, the class of finite languages is identifiable in the limit in Gold's positive evidence only model.
- This is very strange! A rote learner can learn finite languages easily.

Finite languages

- Pick $\epsilon = 0.01, \delta = 0.001$
- Suppose N is 1,000,000.

But what if the finite language had 2,000,000 unrelated sentences?

Imposing an Upper Bound on the Size of the Language Class

- Suppose that $\mathcal{C}$ has infinite VC-dimension.
- Let $\mathcal{C}^k$ be all of the elements of size at most k.
- For any k, $\mathcal{C}^k$ has finite VC-dimension.
- Therefore, the class of languages in $\mathcal{C}^k$ is uniformly PAC learnable.
- The full class is the union of $C^1, C^2, \ldots$.
- Similarly, bounding the size of regular grammars and CFGs results in finite VC dimension and renders these classes uniformly learnable.

APS Arguments

A bad argument (Nowak et al. 2002):

- The fact that uniform PAC learning requires a finite VC dimension for the target class might appear to support a version of the APS.
- The concept class for language acquisition must be restricted to a set of grammars that has finite VC-dimension to insure acquisition.
- Learners need to have prior knowledge of these bounds on the target language class.
- This learning prior is clearly domain specific, and so it entails a form of linguistic nativism.

PAC Learnability and Bounded Language Classes

- In fact, this argument does not go through when we distinguish the target class from the hypothesis space.
- As in the case of IIL, learners can formulate hypotheses that fall outside a PAC learnable class.
- Any class of finite, regular, or context free languages can be learned up to an arbitrary cardinality bound k.
- This cardinality bound need not be specified as part of the learning algorithm.

PAC Learnability and Bounded Language Classes

- The algorithm can test progressively larger representations of a language against the data until it arrives at the target hypothesis.
- So for any k, the class of finite/regular/context free languages of k cardinality can be uniformly PAC learned.
- The union of these bounded classes gives the full unbounded class.
- Haussler et al. (1991) prove a theorem stating that a learner, without prior knowledge of a bound on the size of the concept class $\mathcal{C}$, can uniformly learn any subset $\mathcal{C}^k$ of C, and so can non-uniformly learn the entire class $\mathcal{C}$.

PAC Learnability and Bounded Language Classes

- The algorithm can test progressively larger representations of a language against the data until it arrives at the target hypothesis.
- So for any k, the class of finite/regular/context free languages of k cardinality can be uniformly PAC learned.
- The union of these bounded classes gives the full unbounded class.
- Haussler et al. (1991) prove a theorem stating that a learner, without prior knowledge of a bound on the size of the concept class $\mathcal{C}$, can uniformly learn any subset $\mathcal{C}^k$ of C, and so can non-uniformly learn the entire class $\mathcal{C}$.
- The concept class $\mathcal{C}^k$ is bounded, but the hypothesis class $\mathcal{C}$ is not.

Finite languages: The Trivial Learner

- The rote learner simply memorises all examples that it sees.
- This learner does not have any prior bound on the size (number of strings in the language).
- So even if we accept the PAC model as a learning paradigm, it doesn't support domain specific nativist conclusions.

Uniform Learning

- The PAC framework requires that learning be at a uniform rate for the elements of a class, in relation to the available data, as specified by a constant upper bound on the size of the data set.
- As we have seen in the case of infinite VC dimension, this condition is problematic if the class contains representations of unbounded complexity.
- By allowing for non-uniform learning in which different elements of such a class can be acquired at rates expressed by distinct polynomial functions on data sets, it is possible to expand the set of learnable classes.

Uniformity in Language Acquisition

- Language acquisition proceeds uniformly.
- At the ages of 5 – 7 children in different language communities have achieved comparable levels of mature linguistic ability.
- It is not the case that children only learn Japanese at age 15, while they learn English at age 5

Uniform Acquisition of Natural Languages

- It is generally agreed that natural languages exhibit roughly the same degree of complexity in their grammars.
- Christiansen and Chater (2008), Kirby (2001), Kirby (2007), and Kirby and Hurford (2002) explain this property on the basis of information theoretic conditions on transmission and learning, which shape the evolution of language.
- If this account is correct, then the common complexity of languages is not due to UG, but largely to domain general constraints on human learning and information processing.
- On this view, even if learning is, in general, non-uniform for target classes, language acquisition will be uniform across languages because of their shared complexity properties.

Computational complexity in learning

- Suppose that we have enough data. Can we find the best hypothesis efficiently?
- Typically we have exhaustive algorithms that search through all possibilities.
- But the space may be too large or infinite.

Some Negative Complexity Results on Learnable Classes

- *Maximum Likelihood Estimation* (MLE) is a form of probabilistic inference in which we select the model (or set of parameters for a model) that makes the data most likely.
- When a fixed set of data D is given, the learner chooses an element, from a restricted set of models, that maximises the probability of D, given that model.
- Estimating the parameters of stochastic DFSAs to maximize the likelihood of a data set is tractable and relatively easy.
- Abe and Warmuth (1992) show that identifying an arbitrary stochastic NFSA that maximizes the likelihood of a data set is NP hard.

Cryptography and Learning Complexity

- Work on complexity in learning has established a connection between cryptographic decoding and learning.
- Any formalism that is rich enough to encode some cryptographic primitives, may be unlearnable.
- In the case of language acquisition this relation is problematic.
- Encryption is designed to be difficult to decode, while natural languages are supposed to be easily learned and to facilitate communication.
- Using cryptographic assumptions Kearns and Valiant (1994) prove that in the classical PAC learning framework even Acyclic Deterministic Finite Automata, which generate only finite languages, are hard to learn.

Learning Complexity and the Chomsky Hierarchy

- These negative results indicate that learning even the class of regular languages, the lowest member of the Chomsky hierarchy of formal languages, is NP-hard (or similar).
- This learning complexity property is inherited by all other classes in the hierarchy.
- Therefore, the class of Context Free and the class of Context Sensitive languages are also unlearnable.
- As natural languages exhibit at least context free, and, in some cases, context sensitive weak generative capacity, then the negative complexity results might be taken to suggest that they are unlearnable without strong learning biases.

Impagliazzo, 1995

“There is a large gap between a problem not being easy and the same problem being difficult. A problem could have no efficient worst-case algorithm but still be solvable for "most" instances, or on instances that arise in practice. Thus a conventional completeness result can be relatively meaningless in terms of the "real life" difficulty of the problem, since two problems can both be NP-complete, but one can be solvable quickly on most instances that arise in practice and the other not.”

Stratifying Hard Classes

- The fact that a class of problems is NP-complete does not entail that all of its elements are intractable.
- It is frequently possible to identify subsets of the class that allow for polynomial or even linear time solutions.
- Stratifying a hard class in this way offers a strategy for separating out the tractable problems within it.
- It *may* be the case that the tasks one is interested in fall within the tractable part of an NP-hard class.

Regular languages

Clark and Thollard, 2004

PAC-learning probabilistic deterministic finite state automata.

- Efficient in data and computation time
- Distinguishability parameter μ separates out harder instances from easier ones.

Clark, 2006

Similar result for a class of CFGs:
Unambiguous NTS languages:

- Several extra parameters to make learning tractable.

Outline

P&P Models of UG and a Finite Class of Grammars

- P&P models of UG encode strong biases with a finite number of parameters, which are often binary valued ({0,1}).
- Assuming that a P&P model of UG has k binary parameters, it specifies a finite set of possible grammars with a cardinality of at most 2^k.
- As the class of grammars is finite, then it can be identified in the limit from positive evidence only in the Gold paradigm.
- As this class has VC-dimension of at most k, it exhibits manageable sample complexity, and so it is PAC learnable ignoring complexity.

P&P Models of UG and Learning Complexity

- However, the finiteness properties of a P&P model do not, in themselves, solve the complexity of learning problem.
- Learning can be computationally intractable in a finite hypothesis space, if the space is large and no efficient algorithm exists for finding the best hypothesis.
- The hardness results in probabilistic learning (e.g. Kearns and Valiant) all use a finite class of languages, with a set of binary parameters.
- P&P models might be learnable: this can only be determined when the class of parameters and grammars has been precisely defined.
- Conversely, learning in an infinite hypothesis space can be efficient, if a learning algorithm exists for exploring it.

Parameter setting algorithms

Fodor and Sakas (2004)

"Parameter setting as a concept is still paramount, but parameter setting as a process has become a source of problems rather than solutions"

Strategies:

- Original idea of 'Triggering'
- Gibson & Wexler (1994) – Triggering Learning Algorithm
- Sakas & Fodor (2000) – Structural Triggers Learners
- Yang (2002) – General probabilistic learning

Hardness of Estimating Parameters

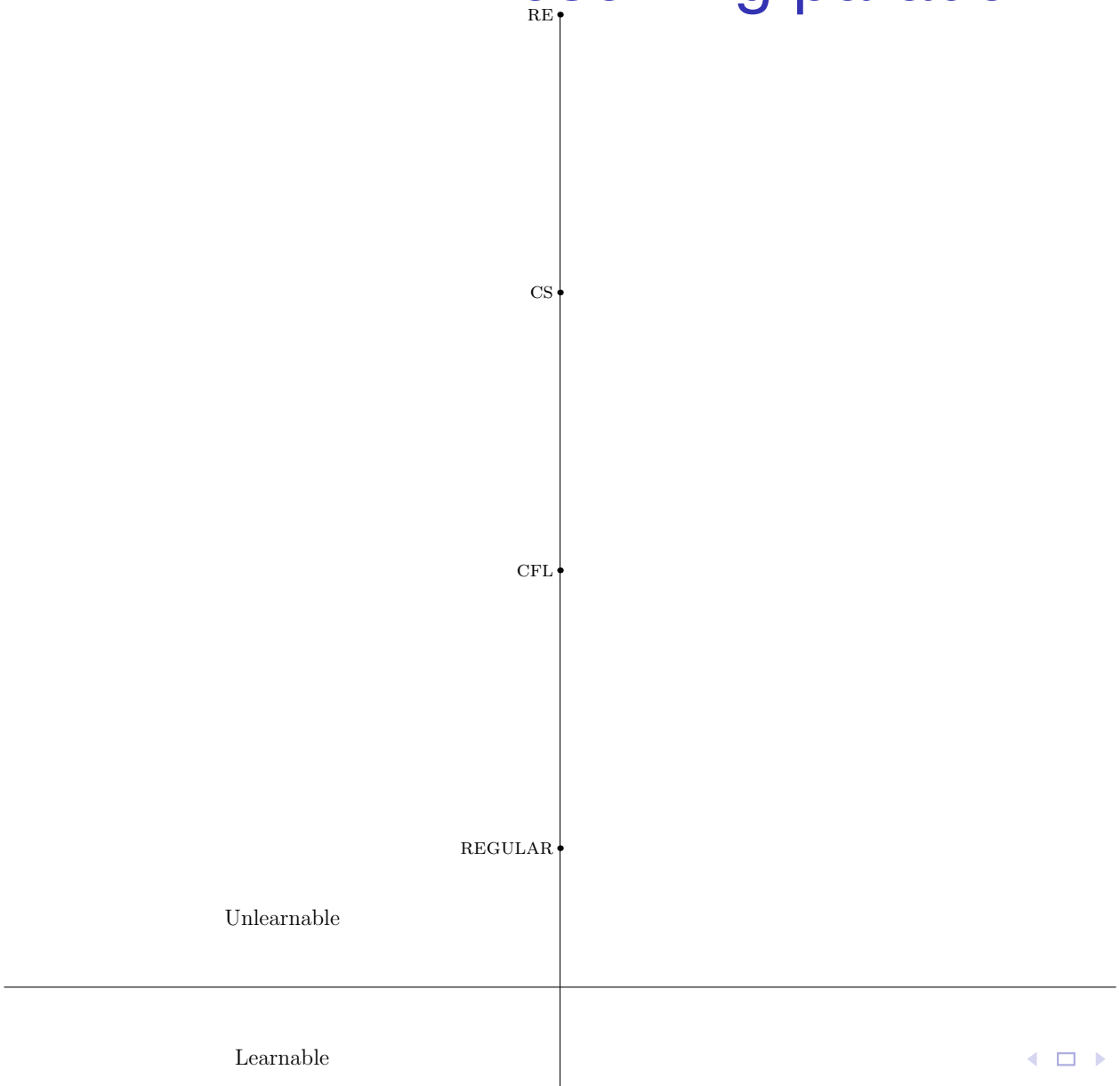
- In general the ease with which the values of parameters can be estimated depends on their relative independence, and the difficulty of identifying them from the data.
- If parameters in a model are heavily interdependent, then it is hard to distinguish their individual values on the basis of the data.
- Moreover, for parameter values to be learnable, it seems that the properties that they specify must be easy to discern in the data.

P&P Models and Complexity

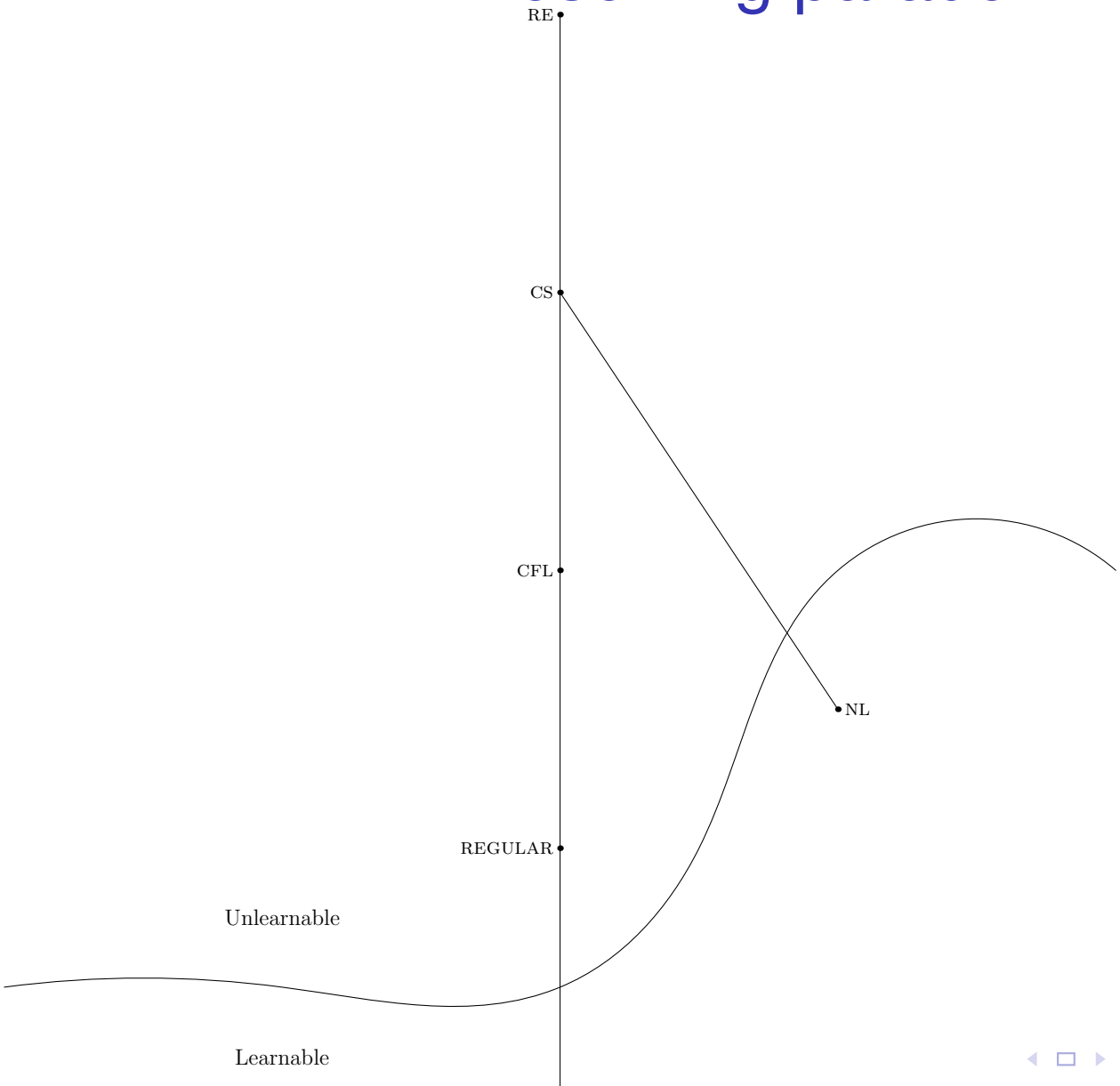
- The parameters that P&P models assume tend to be highly interdependent through entailment relations.
- They generally correspond to abstract syntactic properties which are far removed from observed data.
- They are rarely (if ever) specified as a complete set, or with the precision required for demonstrating learning results.
- Therefore, these models do not (at least in their current form) offer a solution to the complexity of learning problem.

Outline

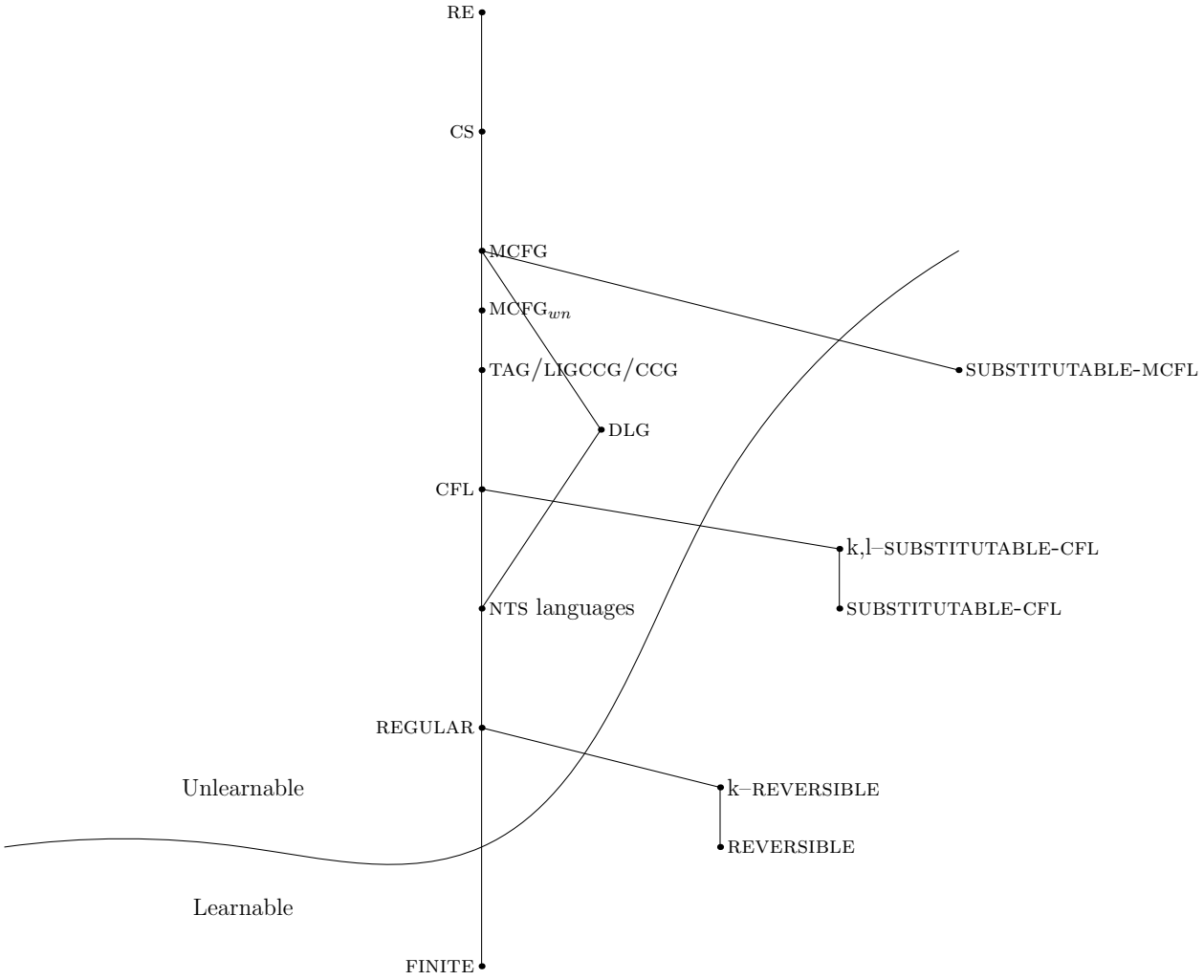
Resolving paradox



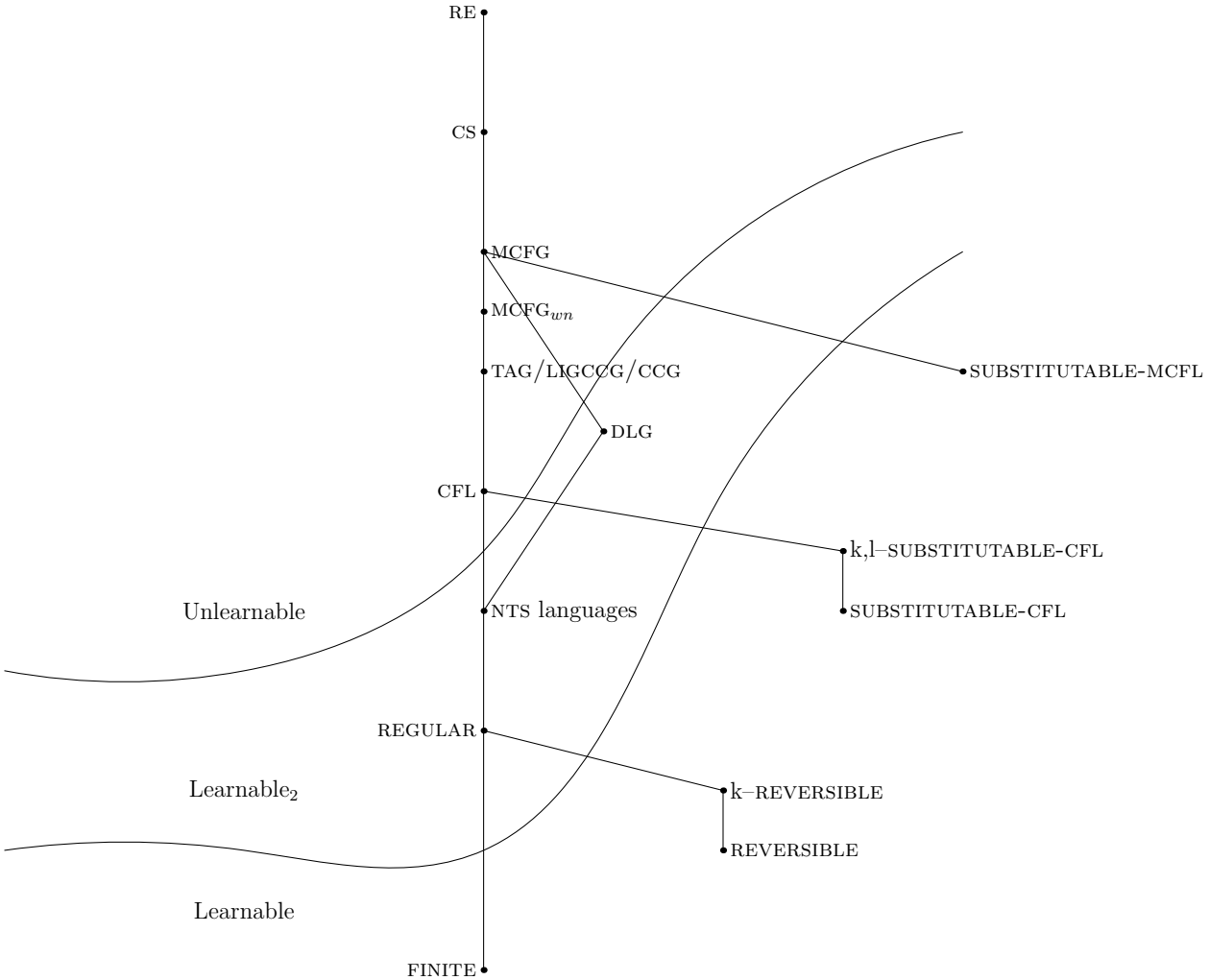
Resolving paradox



Resolving paradox



Resolving paradox



Conclusions

- Learning involves both sample and computational complexity.
- Computational complexity is a major challenge for efficient learning of natural languages.
- Even learning the class of regular languages is computationally hard.
- The finiteness assumptions of P&P models do not solve the computational complexity problem for learning.