Learning Noisy Perceptrons by a Perceptron in Polynomial Time

Edith Cohen

UC Berkeley *AT&T Labs – Research*

http://www.research.att.com/~edith

http://www.cs.berkeley.edu/~edith

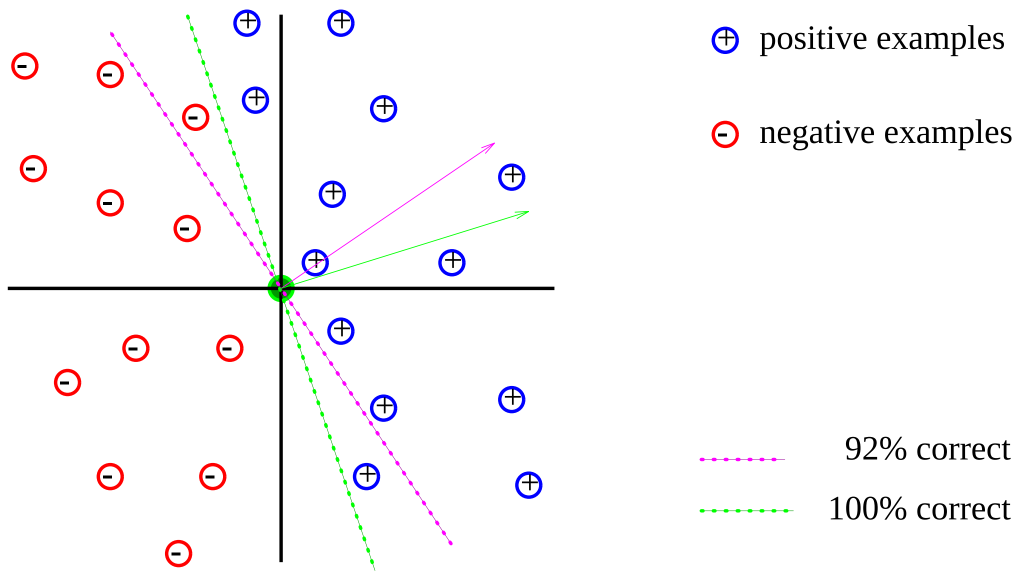
edith@research.att.com

Outline

- background: The perceptron (LTF) problem
  - ◇ equivalent formulation
  - ◇ algorithms
- Perceptrons (LTFs) with random classification noise
  - ◇ background
  - ◇ Our results
- outline of our algorithm
  - ◇ The Ellipsoid algorithm
  - ◇ our separation oracle
  - ◇ witness identification theorem
- our witness identification algorithm
- summary and open problems

The Perceptron Problem

- Input: Labeled examples (points in R^n labeled $+$ or $-$)
- Output: A hyperplane that separates positive and negative examples



We assume examples are drawn according to some (arbitrary) distribution.
Goal: Find a vector $\vec{w} \in R^n$ that classifies correctly $(1-\epsilon)$ of the examples, where $\vec{x} \in R^n$ is classified $+$ if $\vec{x} \cdot \vec{w} > 0$ and $-$ otherwise.

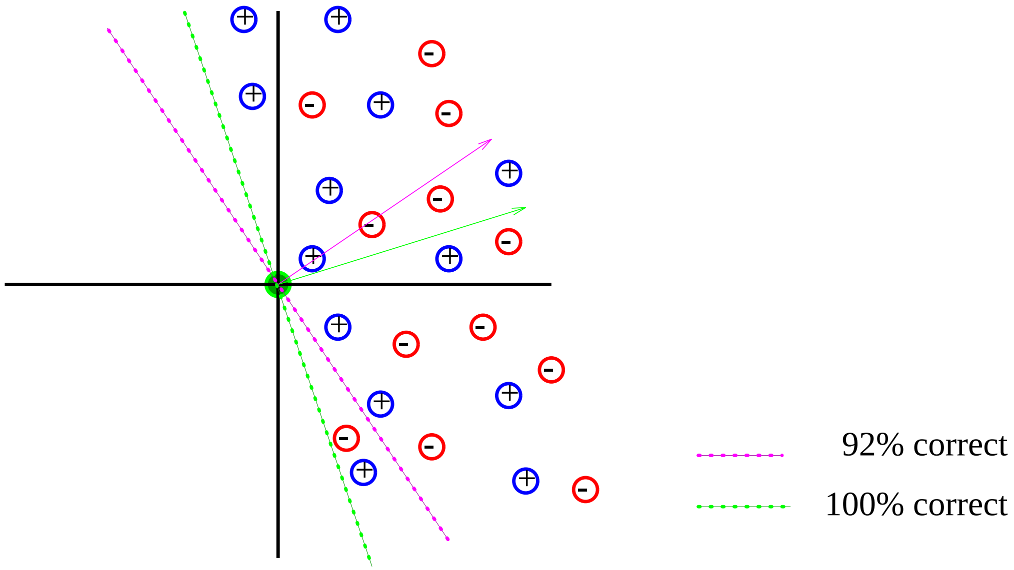
Equivalent Formulation

- Negate all samples with a negative label.
- We obtain a distribution χ over R^n concentrated in a halfspace.

That is,

$$\exists \vec{h} \in R^n \text{ such that } \forall \vec{x} \in \chi, \vec{h} \cdot \vec{x} > 0 \ .$$

Goal: Find a vector $\vec{v} \in R^n$ that classifies $(1-\epsilon)$ of the distribution. That is, $\mathrm{Prob}_{\vec{x} \in \chi}\{\vec{v} \cdot \vec{x} < 0\} \leq \epsilon$.



Algorithms for The Perceptron Problem

◇ This problem is poly-time solvable using LP:

- Draw $k = O(\frac{n}{-\epsilon \log \epsilon} + \frac{-\log \delta}{\epsilon})$ samples $\vec{x}_1, \ldots, \vec{x}_k$ from χ.
- Solve the linear system:
  find $\vec{v}$ subject to $\vec{v} \cdot \vec{x}_i > 0$ (for $i = 1, \ldots, k$).
- VC-dimension arguments imply that the solution is good with confidence $\geq 1 - \delta$.

◇ The greedy perceptron algorithm [Ros62] [Agm54] solves the problem but is not polynomial.

- Running time depends on wiggle room: the largest σ for which there exists a $\vec{h}$ for which $\forall \vec{x} \in \chi$, $\vec{h} \cdot \vec{x} > \sigma$
- Behaves well in practice.

Perceptron Problem with Random Classification Noise

Samples are subject to Random Classification Noise of intensity
$\Delta < 0.5$:
The label of every example is corrupted (flipped) with probability Δ.

In equivalent formulation: We see samples from noisy distribution $\tilde{\chi}$.
A sample from $\tilde{\chi}$ is the same as drawing $\vec{x} \in \chi$ and output

$$\tilde{\vec{x}} = \begin{cases} \vec{x} & \text{with probability } 1 - \Delta \\ -\vec{x} & \text{with probability } \Delta \end{cases}$$

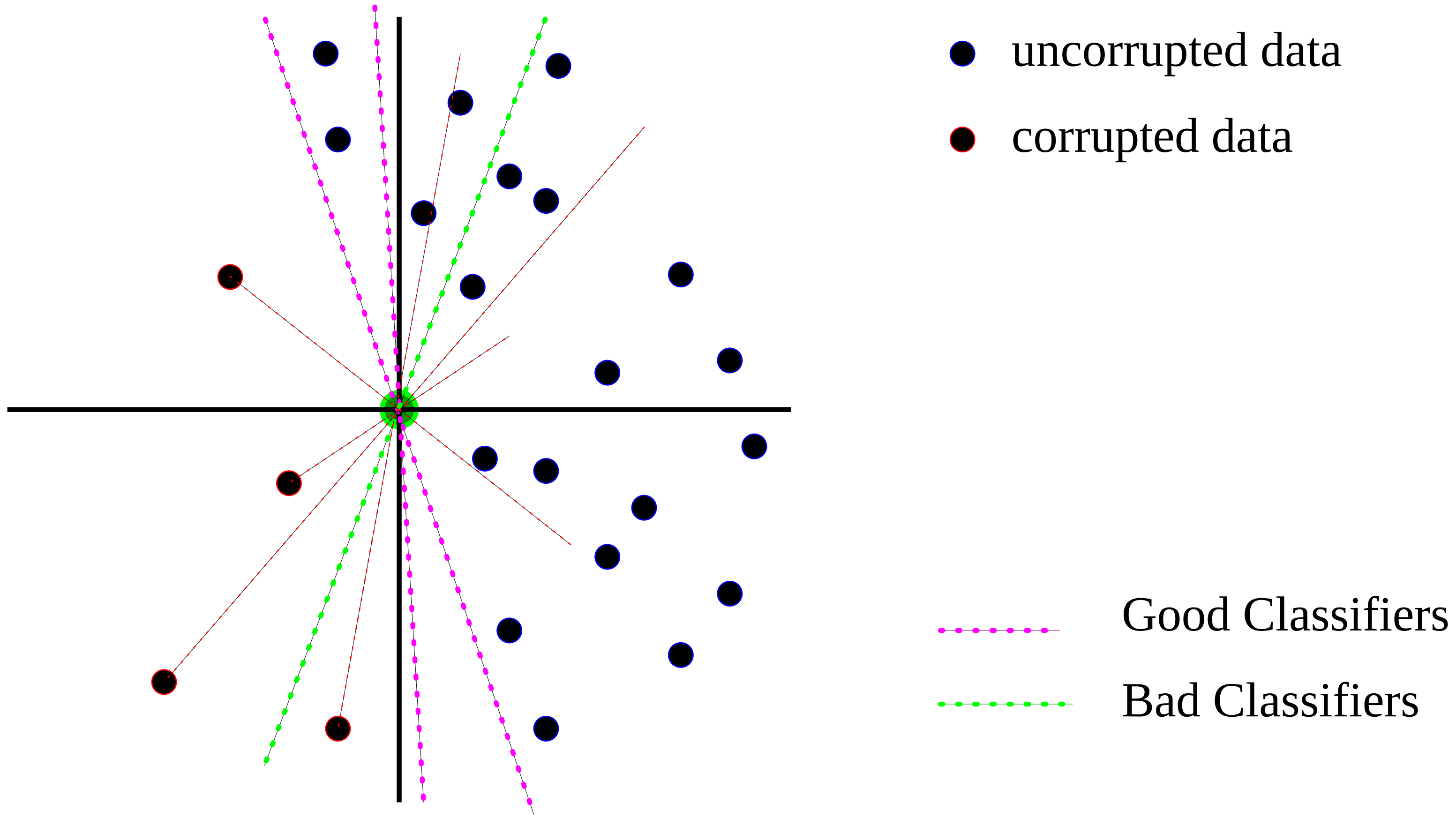
(flip the sign of each sample with probability Δ.)

Goal: Find $\vec{v} \in R^n$ that classifies well the original distribution χ.

Question: Can we find a good linear classifier in polynomial time?
Equivalently: Are noisy perceptrons PAC learnable with a hypothesis which is itself a linear threshold?

Difficulty: Simple LP approach does not apply any more.

...Perceptron Problem with Random Classification Noise



Some previous work on noisy perceptrons

[Kearns 93] established PAC-learnability for radially symmetric distributions (using his statistical query model).

[Bylander 94] modified the perceptron algorithm to work with noisy data, when the data is well separated with respect to any hyperplane.

[BFKV 96] established PAC learnability using a hypothesis which is a (polynomially) long decision rule of linear thresholds.

- They used a variant of the perceptron algorithm to obtain a weak hypotheses: a linear threshold that classifies correctly $\geq 1/\mathrm{poly}$ of distribution and outputs "don't know" for others.
- The above was applied recursively to obtain a polynomially long decision rule that classifies correctly $(1-\epsilon)$ of χ.

Our results

The existence of a poly-time algorithm to learn noisy perceptrons with a single linear threshold remained an open problem.

Our main result: Noisy perceptrons (with random classification noise) are PAC learnable with a hypothesis which is itself a single linear threshold.

Equivalently: We find $\vec{v} \in R^n$ such that $\mathrm{Prob}_{\vec{x} \in \chi}\{\vec{v} \cdot \vec{x} < 0\} < \epsilon$ with high confidence $(1 - \exp(-\mathrm{poly}))$ in time polynomial in $n, \epsilon^{-1}, (1 - 2\Delta)^{-1}, b$.

Our results extend, with limits, to more general noise models.

Outline of Our Approach

- Main loop: The Ellipsoid algorithm: solves, via a separation oracle, the large and non-explicit system of linear inequalities: find $\vec{h} \in R^n$ such that $\forall \vec{x} \in \chi, \vec{h} \cdot \vec{x} > 0$.
  ($\vec{x}$ can be any nonnegative linear combination of vectors in χ)
- Poly time separation oracle: Given a vector $\vec{u} \in R^n$. If $\vec{u}$ is a good classifier say so (easy). Else, find a violated constraint (w.v.h.p). Ingredients:
  - "outliers removal" [BFKV]
  - averaging many random samples to reduce effects of noise, estimating "centers of mass", applying Chernoff bounds
  - restricting the distribution to "slices" of the space taken with respect to current $\vec{u}$.

◇ The main challenge was the separation oracle. Given a bad classifier $\vec{u}$, identifying a violated $\vec{w} \in \chi$ with very high confidence is hard. We see many violated $\vec{x} \in \tilde{\chi}$ but we can not just pinpoint the uncorrupted ones.

Using the Ellipsoid algorithm

The Ellipsoid algorithm (Khachyian) is applicable to an inaccessible large and non-explicit system of linear inequalities. It can find a solution in polynomial time (and poly many oracle calls) when:

- We can "access" the system via a Separation Oracle:
  - input: a vector $\vec{u} \in R^n$.
  - output: oracle tells us if $\vec{u}$ is feasible. If not, oracle gives us one violated inequality.
- We can bound (part of) the feasible region F by some ball B_0.
- $\frac{\mathrm{vol}(B_0)}{\mathrm{vol}(B_0 \cap F)} \leq \exp(\mathrm{poly})$

Our system of inequalities:
find $\vec{v}$ such that $\vec{v} \cdot \vec{x} > 0$, for all $\vec{x}$ which are nonnegative (nonzero) linear combinations of χ vectors.

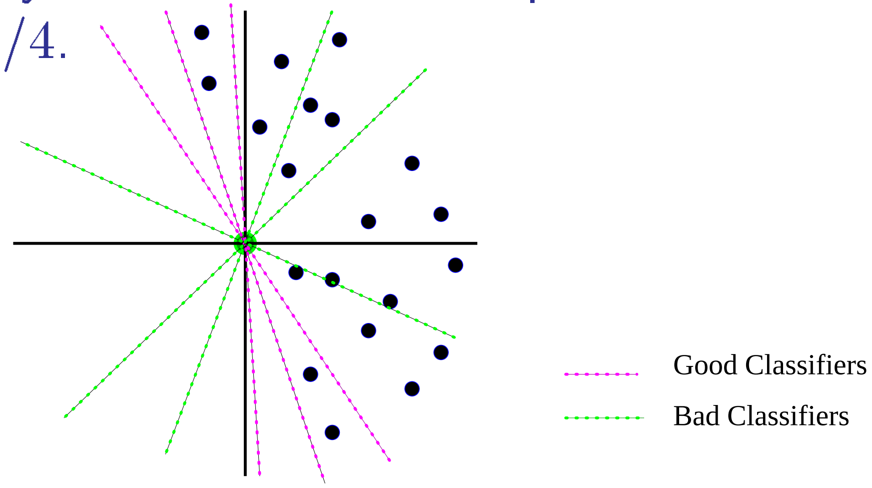
Feasible vectors $\vec{h} \in F \Rightarrow$ good classifiers

Identifying Good and Bad Classifiers

- A vector $\vec{u} \in R^n$ is a good classifier if $\mathrm{Prob}_{\vec{x}\in\chi}\{\vec{u}\cdot\vec{x} < 0\} < \epsilon$. Equivalently, if $\mathrm{Prob}_{\vec{x}\in\tilde{\chi}}\{\vec{u}\cdot\vec{x} < 0\} < \Delta + \epsilon(1-2\Delta)$.
- A vector $\vec{u} \in R^n$ is a bad classifier if $\mathrm{Prob}_{\vec{x}\in\chi}\{\vec{u}\cdot\vec{x} < 0\} > \epsilon/2$ Equivalently, if $\mathrm{Prob}_{\vec{x}\in\tilde{\chi}}\{\vec{u}\cdot\vec{x} < 0\} > \Delta + \epsilon(1-2\Delta)/2$.

Lemma: Goodness/badness of vectors can be identified with very high probability in polynomial time.

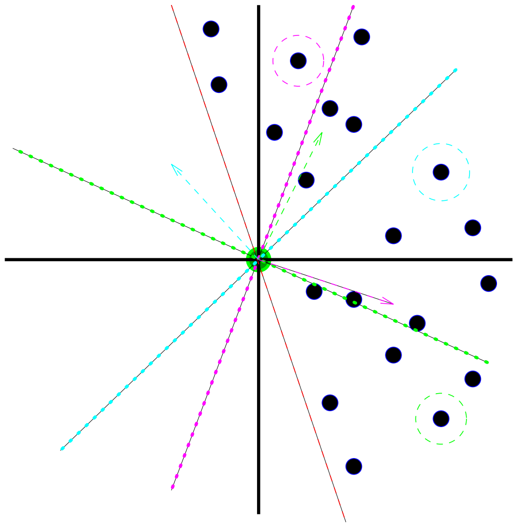
Algorithm: Draw a set S of samples from $\tilde{\chi}$. Let $S^- \subset S$ be samples missed by $\vec{u}$ ($\vec{u}\cdot\vec{x} < 0$). Estimate miss-rate by $\hat{m} = \frac{|S^-|}{|S|}$. Chernoff bounds establish that using a polynomial size S, w.v.h.p the estimate $\hat{m}$ is off by less than $(\epsilon(1-2\Delta))/4$.



Witnesses of badness

Let $\vec{h} \in F$ be a "perfect classifier."
Consider a bad classifier $\vec{b} \in R^n$. A witness for the badness of $\vec{b}$ is a vector $\vec{w} \in R^n$ such that:

- $\vec{w} \cdot \vec{h} > 0$ (perfect classifiers classify $\vec{w}$ positively and correctly)
- $\vec{w} \cdot \vec{b} < 0$ ($\vec{b}$ does not classify $\vec{w}$ correctly)



bad classifiers and corresponding witnesses

Witness identification theorem: If $\vec{b}$ is a bad classifier, a witness $\vec{w}$ can be found in polynomial time w.v.h.p.

Identifying witnesses w.h.p. in polynomial time

For any bad $\vec{b}$, we obtain a witness w.v.h.p. Outline:

- center of mass $\vec{m}^-(\chi)$ of all misclassified vectors on χ can be estimated (unbiasdly) from $\tilde{\chi}$ and constitutes a witness.
  Caveat: estimate not accurate enough (with poly many samples) for witness properties to hold. Variance of $\vec{b} \cdot \vec{x}$ and $\vec{h} \cdot \vec{x}$ too high.
- Fix: Outliers removal [BFKV] removes in polytime all outliers: for all $\vec{u} \in R^n$. we have $\max_{\vec{x} \in \chi} |\vec{u} \cdot \vec{x}| < \beta \, \mathrm{E}_{\vec{x} \in \chi} |\vec{u} \cdot \vec{x}|$ where β is poly.
  Caveat: Helps, but still estimate not good enough
- Fix: Instead of estimating $\vec{m}^-(\chi)$, estimate a linear combination with small $\nu > 1/\mathrm{poly}$: $\vec{m}^\nu = (1 - \nu)\vec{m}^-(\chi) + \nu \vec{m}^+$. Now $\hat{\vec{m}}^\nu \cdot \vec{h} > 0$ but caveat: Sometimes $\vec{m}^+(\chi) \cdot \vec{b} \geq 2^{\mathrm{poly}} |\vec{m}^-(\chi) \cdot \vec{b}|$. Hence, we may have $\vec{m}^\nu \cdot \vec{b} > 0$ and lose witness property.
- Fix: Before doing all the above, restrict χ (and $\tilde{\chi}$) to a symmetric slice of R^n containing vectors with similar dot-product value with $\vec{b}$: $\{\vec{x} \in R^n | a < |\vec{b} \cdot \vec{x}| < 1.1a\}$

Finally works!

Estimating centers of mass

Let $H^- = \{\vec{x} | \vec{x} \cdot \vec{b} < 0\}$ (halfspace of vectors missed by classifier $\vec{b}$)
Let $H^+ = \{\vec{x} | \vec{x} \cdot \vec{b} > 0\}$ (halfspace of vectors captured by classifier $\vec{b}$)
We assume WLOG that Δ is given.

Observation 1: The center of mass $\vec{m}^-(\chi)$ of χ on H^- is a witness.
Observation2: $\vec{m}^-(\chi)$ can be estimated from the noisy distribution $\tilde{\chi}$.

We have

$$\begin{aligned} \vec{m}^+(\tilde{\chi}) &= (1-\Delta)\vec{m}^+(\chi) - \Delta\vec{m}^-(\chi) \\ \vec{m}^-(\tilde{\chi}) &= (1-\Delta)\vec{m}^-(\chi) - \Delta\vec{m}^+(\chi) \end{aligned}$$

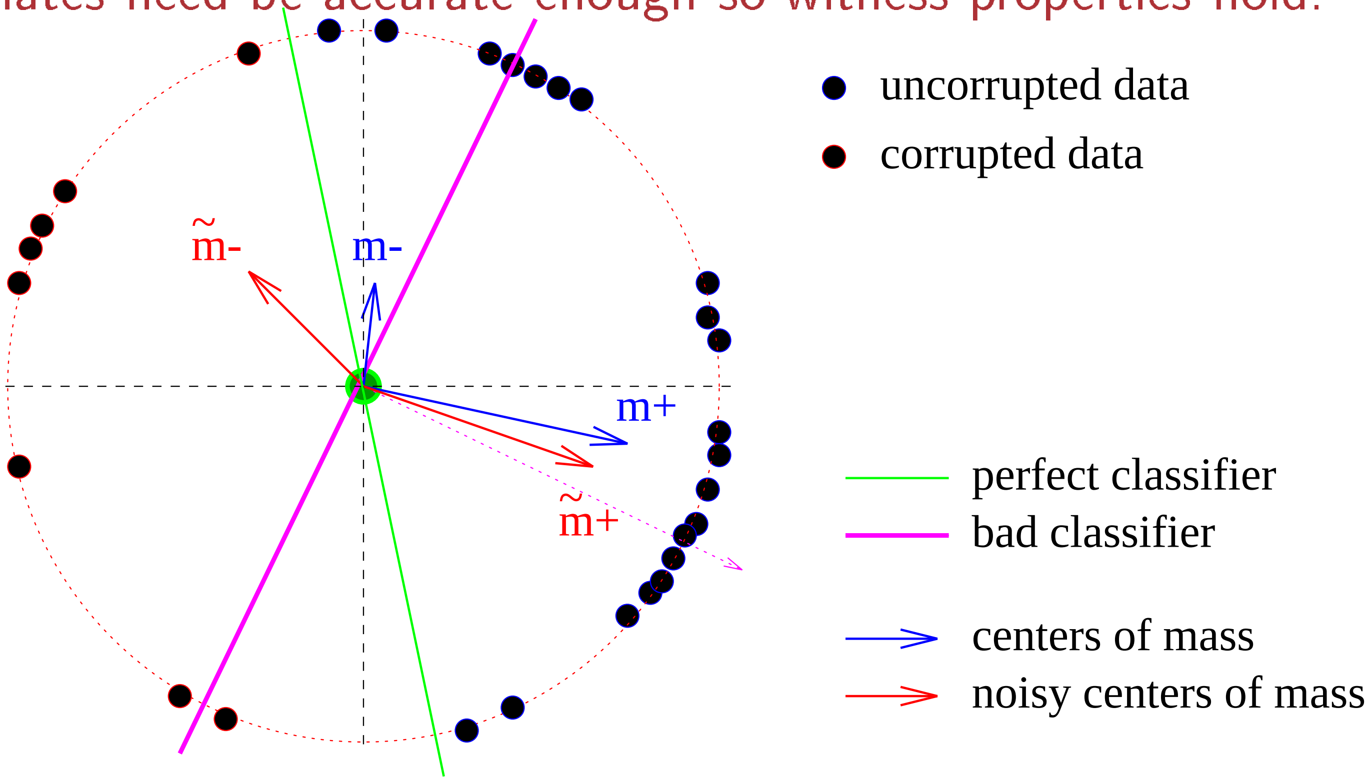
solving yields:

$$\begin{aligned} \vec{m}^+(\chi) &= \frac{(1-\Delta)\vec{m}^+(\tilde{\chi}) + \Delta\vec{m}^-(\tilde{\chi})}{1-2\Delta} \\ \vec{m}^-(\chi) &= \frac{(1-\Delta)\vec{m}^-(\tilde{\chi}) + \Delta\vec{m}^+(\tilde{\chi})}{1-2\Delta} \end{aligned}$$

...Estimating centers of mass

S is a set of noisy samles from $\tilde{\chi}$.
$S^- = \{\vec{x} \in S | \vec{x} \cdot \vec{b} < 0\}$, $S^+ = \{\vec{x} \in S | \vec{x} \cdot \vec{b} > 0\}$.

Estimates: $\hat{\vec{m}}^-(\tilde{\chi}) = \Sigma_{\vec{x} \in S^-} \vec{x}/|S|$, $\hat{\vec{m}}^+(\tilde{\chi}) = \Sigma_{\vec{x} \in S^+} \vec{x}/|S|$.
Estimates need be accurate enough so witness properties hold.



Estimate accuracy and use of linear combinations

After removal of outliers, a polynomial number of samples guarantee (wvhp) for (exponentially many) $\vec{u} \in R^n$:

$|(\vec{m}^-(\tilde{\chi}) - \hat{\vec{m}}^-(\tilde{\chi})) \cdot \vec{u}| \leq \mathrm{E}_{\vec{x}\in\chi}\{|\vec{u} \cdot \vec{x}|\}/\mathrm{poly}$ and

$|(\vec{m}^+(\chi) - \hat{\vec{m}}^+(\tilde{\chi})) \cdot \vec{u}| \leq \mathrm{E}_{\vec{x}\in\chi}\{|\vec{u} \cdot \vec{x}|\}/\mathrm{poly}$

Proof: Chernoff bounds.

Caveat: not good enough as is. Since sometimes $|\vec{m}^-(\chi) \cdot \vec{u}| < 2^{-\mathrm{poly}}\, \mathrm{E}_{\vec{x}\in\chi}\{|\vec{u} \cdot \vec{x}|\}$ (also for $\vec{u} \in \{\vec{b}, \vec{h}\}$)

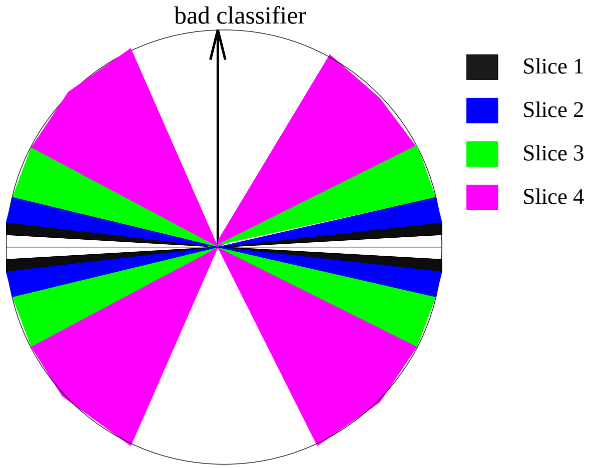
Idea: For all $\vec{x} \in \chi$, $\vec{x} \cdot \vec{h} > 0$. Hence,
$\mathrm{E}_{\vec{x}\in\chi}\{|\vec{h} \cdot \vec{x}|\} = \mathrm{E}_{\vec{x}\in\chi}\{\vec{h} \cdot \vec{x}\} = \vec{m}^-(\chi) \cdot \vec{h} + \vec{m}^+(\chi) \cdot \vec{h}$.
$\Rightarrow$ for $\nu > 1/\mathrm{poly}$, the linear combination
$\vec{m}^\nu = (1-\nu)\vec{m}^-(\chi) + \nu\vec{m}^+(\chi)$ can be estimated in polynomial time so that $\hat{\vec{m}}^\nu \cdot \vec{h} > 0$.

Problem: Note that $\vec{m}^-(\chi) \cdot \vec{b} < 0$ and $\vec{m}^+(\chi) \cdot \vec{b} > 0$. Sometimes we have $|\vec{m}^-(\chi) \cdot \vec{b}| < 2^{-\mathrm{poly}}|\vec{m}^+(\chi) \cdot \vec{b}|$, hence, $\vec{m}^\nu \cdot \vec{b} > 0$ (not a witness !)

Restricting χ to a slice

Solution: When considering a bad $\vec{b} \in R^n$, we first restrict the distributions χ and $\tilde{\chi}$ to a slice of the space where all inner products with $\vec{b}$ have about the same value. Hence, on a slice, we always have $|\vec{m}^-(\chi) \cdot \vec{b}| > |\vec{m}^+(\chi) \cdot \vec{b}|/\text{poly}$.

A slice looks like that: $\{\vec{x} \in R^n | a < |\vec{b} \cdot \vec{x}| < 1.1a\}$



We partition the space to polynomially many slices.
We can always find one slice such that:

- $\vec{b}$ is bad classifier on χ restricted to the slice.
- The slice is dense (the measure of χ on the slice is $\geq 1/\text{poly}$).

open problems (further work?)

- Implementation: the polynomial version of the algorithm is not practically efficient. However, a non-polynomial heuristic version (replace Ellipsoid algorithm, skip outliers removal...) may be.
- Consider other noise models:
  - Malicious noise is NP-hard, but exactly how hard?
  - Approach extends to variable (unknown) noise rate $\Delta(\vec{x})$ that is nondecreasing with proximity to threshold.
  - Approach extends to noise distribution $\Delta(\vec{x})$ which is arbitrary but provided.
  - What happens if noise level is $\Delta(\vec{x}) \leq \Delta_{Max} < 1/2$ arbitrary and unknown?