Lower Bounds for Reductions

Matti Kääriäinen

(joint work with John Langford)

Talk at the Atomic Learning workshop, TTI Chicago
March 25, 2006.

Outline

- Goals
- Case studies:
  - A lower bound for reducing structural sequence learning to binary classification
  - A lower bound for reducing probability prediction to binary classification
- Back to general concerns

Why lower bounds for reductions?

Utilitarian:

- Prove optimality of existing reductions
- Highlight difficult cases (so that they can be circumvented)
- Compare different reduction strategies
- Compare difficulty of different learning tasks

Other:

- Gain understanding on upper bounds
- Study inherent limitations of the reductions approach

Case study I: A sequence prediction problem

Basic setup:

- Given $\vec{X} = (X_1, \ldots, X_T)$, predict $\vec{Y} = (Y_1, \ldots, Y_T) \in \{0, 1\}^T$
- $(\vec{X}, \vec{Y}) \sim D$ iid from some distribution
- Loss of classifier $\vec{f} = (f_1, \ldots, f_T)$ measured by expected Hamming distance

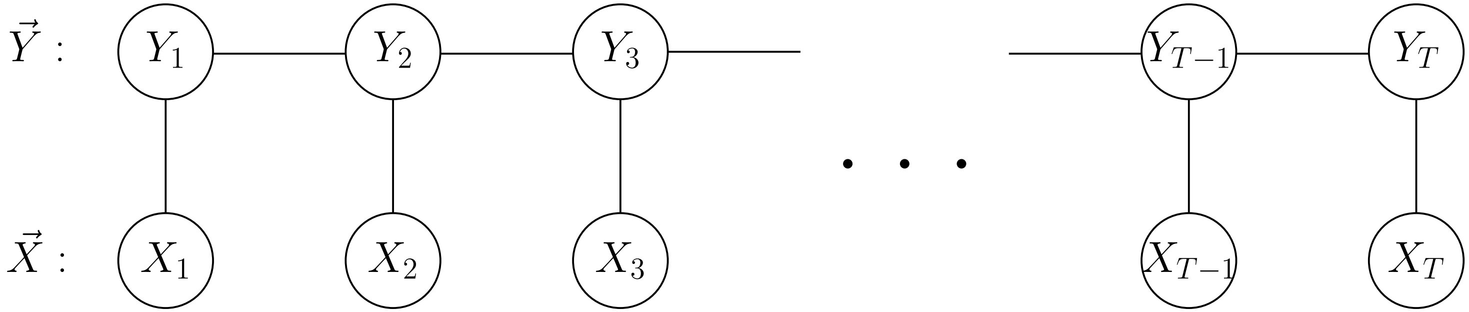
$$\text{Ham}(\vec{f}) = \mathbb{E}_{(\vec{X},\vec{Y})\sim D}\left[\sum_{i=1}^{T} I_{\{Y_i \neq \hat{Y}_i\}}\right],$$

where $\hat{Y}_i = f_i(\vec{X})$.

To make the task (at least look) easier, we assume more structure (on D and $\vec{f}$).

Comb structure

To facilitate learning, the learner assumes that D has the following comb structure:

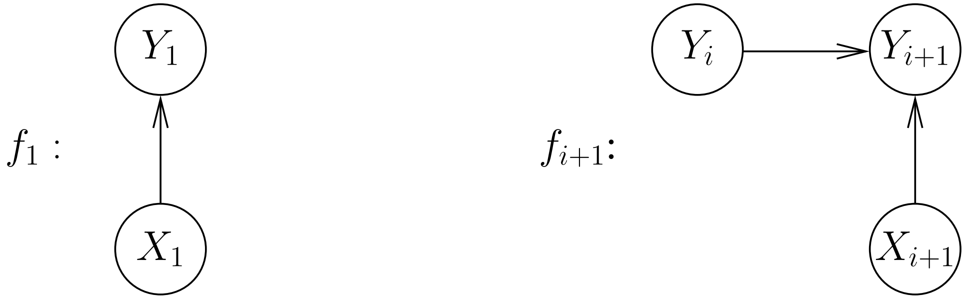


Reduction to binary prediction

Reduction to T binary tasks can be done in many ways, e.g.

- $f_i \colon \prod_{j=1}^{T} \mathcal{X}_i \rightarrow \{0,1\}$
- $f_i \colon \{0,1\} \times \mathcal{X}_i \times \{0,1\} \rightarrow \{0,1\}$
- $f_i \colon \{0,1\} \times \mathcal{X}_i \rightarrow \{0,1\}$
- $f_i \colon \mathcal{X}_i \times \{0,1\} \rightarrow \{0,1\}$

We choose $f_i \colon \{0,1\} \times \mathcal{X}_i \rightarrow \{0,1\}$ for now. As a picture:



Protocol for learning

Learner:

1. Obtain set of training examples $\{(\vec{X}^j, \vec{Y}^j) \mid j = 1, \ldots, n\}$ sampled from D
2. Learn:
   - $\hat{f}_1 : \mathcal{X}_1 \to \{0, 1\}$
   - $\hat{f}_{i+1} : \{0, 1\} \times \mathcal{X}_{i+1} \to \{0, 1\}$
3. Given a test example $\vec{X} = (X_1, \ldots, X_T)$, predict
   - $\hat{Y}_1 = \hat{f}_1(X_1)$
   - $\hat{Y}_{i+1} = \hat{f}_{i+1}(\hat{Y}_i, X_{i+1})$

Ultimate goal: Minimize Ham$(\vec{f})$.

Measuring error of components of $\vec{f}$

- The two natural ways to measure error of $\hat{f}_{i+1}$:

  1. $P_{(\vec{X},\vec{Y})\sim D}[\hat{f}_{i+1}(Y_i, X_{i+1}) \neq Y_{i+1}]$
  2. $P_{(\vec{X},\vec{Y})\sim D}[\hat{f}_{i+1}(\hat{Y}_i, X_{i+1}) \neq Y_{i+1}]$

  These are very very different!

  We choose number 1, end denote it by $\varepsilon(\hat{f}_{i+1})$.

- For technical reasons, we assume in the analysis that

$$P_{(\vec{X},\vec{Y})\sim D}[\hat{f}_{i+1}(Y_i, X_{i+1}) \neq Y_{i+1} \mid Y_i = 0/1] = \varepsilon(\hat{f}_{i+1})$$

Result 1

Theorem: There exists a problem D with the comb structure such that even if $\varepsilon(\hat{f}_i) = \epsilon$ for all i, we have

$$\text{Ham}(\vec{f}) = \frac{T}{2} - \frac{1-(1-2\epsilon)^{T+1}}{4\epsilon} + \frac{1}{2} \approx \frac{T}{2}.$$

Proof idea: Let $Y_1 = f(X_1)$ for some f, let $Y_{i+1} = Y_i$, and let the X_i, $i > 1$, be independent from everything.

Show that the stochastic process $Z_i = I_{\{Y_i \neq \hat{Y}_i\}}$ is a 2-state Markov chain with transition matrix

$$A = \begin{bmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{bmatrix}.$$

Rest follows by known properties of this Markov chain and algebra.

Comments

Result shows that:

- Errors can sometimes accumulate fast

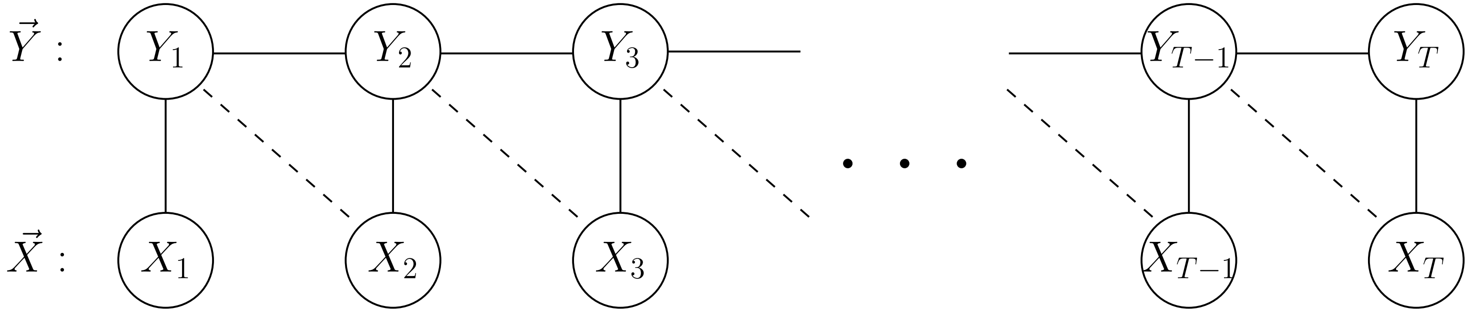
But:

- Very uninteresting problem
- Easy to solve by using the decomposition $f_i : \prod_{j=1}^{T} \mathcal{X}_i \to \{0, 1\}$
- Hard to believe that $\varepsilon(\hat{f}_i) = \epsilon$ for $i > 1$

To make making errors more believable, introduce more dependencies. . .

Another comb structure

Suppose the dependencies are covered by:



Result 2

Theorem: There exists a problem D with the extended comb structure such that if $\varepsilon(\hat{f}_{i+1}) = \epsilon$ for all i, then

$$\text{Ham}(\vec{f}) = \frac{T}{2} - \frac{1 - (1 - 2\epsilon)^{T+1}}{4\epsilon} + \frac{1}{2} \approx \frac{T}{2}.$$

Proof idea: One can construct a D such that

- D has the extended comb structure
- Learning the $\hat{f}_i$s is non-trivial
- The process $Z_i = I_{\{Y_i \neq \hat{Y}_i\}}$ is still the same 2-state Markov chain as before.

The difficult task D

Abstract version of the following parity problem:

- X_is independently sampled raster images of zeros and ones (possibly, say, different fonts for different is)
- Y_i parity of digits represented by $X_1, \ldots, X_i$

More formally:

- $\tilde{f}_i(X_i)$ the digit represented by image X_i
- $Y_i = \bigotimes_{j=1}^{i} \tilde{f}_j(X_j) = Y_{i-1} \otimes \tilde{f}_i(X_i)$

One can furthermore assume that

$P_{(\vec{X},\vec{Y})\sim D}[Y_i = 0] = P_{(\vec{X},\vec{Y})\sim D}[Y_i = 1] = 0.5$.

Better solutions?

- The theorems show that the decomposition $f_i\colon \{0,1\} \times \mathcal{X}_i \to \{0,1\}$ does not work
- Is the decomposition $f_i\colon \prod_{j=1}^{T} \mathcal{X}_j \to \{0,1\}$ any better?
  - Shouldn't be, as Y_{i+1} really depends only on Y_i and X_{i+1}.
- Similar reasoning applies to other decompositions.

Conclusion: The *task* is difficult.

Easy parity problem

- Idea: Replace X_1 by the vector $(X_1, (Y_1, \ldots, Y_T))$
- Effects:
  - Breaks comb structure
  - Learning using the decomposition $f_i \colon \{0,1\} \times \mathcal{X}_i \rightarrow \{0,1\}$ still hard
  - Learning using the decomposition $f_i \colon \prod_{j=1}^{T} \mathcal{X}_j \rightarrow \{0,1\}$ very easy
- Thus, using the decomposition $f_i \colon \{0,1\} \times \mathcal{X}_i \rightarrow \{0,1\}$ can be a bad idea if assumption on comb structure wrong

So which way to decompose?

Answer depends on D, any one of

- $f_i \colon \prod_{j=1}^{T} \mathcal{X}_i \to \{0,1\}$
- $f_i \colon \{0,1\} \times \mathcal{X}_i \times \{0,1\} \to \{0,1\}$
- $f_i \colon \{0,1\} \times \mathcal{X}_i \to \{0,1\}$
- $f_i \colon \mathcal{X}_i \times \{0,1\} \to \{0,1\}$

can be superior (or very very bad).

Lessons learned?

Possible conclusion:

- The simplifying comb assumption doesn't make things simple — maybe it is the wrong assumption?
- The comb assumption is an upper bound on dependencies in D, but reality needs not be worst-case — should one add an assumption that there are no strong long distance dependencies?
- Some completely different assumptions that better capture "locality"?

End of sequence prediction

Case study II: Lower bound for reducing probability prediction to binary classification

Task: Learn a probability predictor $p\colon \mathcal{X} \to [0,1]$ with small mean squared error

$$\mathbb{E}_{(X,Y)\sim D}[(D(1|X) - p(X))^2] = \mathbb{E}_{X\sim D_X}[(D(1|X) - p(X))^2]$$

Here, D is a distribution on $\mathcal{X} \times \{0,1\}$ generating the training data.

Question: Informally, if we assume the capability to solve binary classification to some accuracy, how well can we hope to solve probability prediction?

The reduction approach

- General strategy:
  1. Map the probability prediction problem D into a binary prediction problem $\tilde{D}$ with some domain $\tilde{X}$
  2. Learn a binary predictor $c \colon \tilde{\mathcal{X}} \to \{0,1\}$ with small *generalization error* $\text{Pr}_{(x,y)\sim\tilde{D}}[c(x) \neq y]$
  3. Construct p from c
- The probing reduction: Instance of the above, transforms

$$\text{Pr}_{(x,y)\sim\tilde{D}}[c(x) \neq y] = \epsilon$$

to

$$\mathbb{E}_{(X,Y)\sim D}[(D(1|X) - p(X))^2] = 2\epsilon.$$

- Is probing optimal?

List of assumptions

A1: For $x, x' \in \mathcal{X}$, $x \neq x'$, we have $\tilde{\mathcal{X}}_x \cap \tilde{\mathcal{X}}_{x'} = \emptyset$, where $\mathcal{X}_x$ is the subset of $\tilde{\mathcal{X}}$ that contains all points $\tilde{x}$ that may affect $p(x)$.

A2: For each $x \in \mathcal{X}$, there is a way to choose the predictions for c in the set $\mathcal{X}_x$ so that $|p(x) - D(1|x)| \geq \alpha$, where α is a constant, say $\alpha = 0.5$.

A3: The set $\mathcal{X}$ can be partitioned into disjoint pieces of probability ϵ each.

A4: There exists a classifier c whose generalization error on $\tilde{D}$ is zero.

Result

Theorem: Suppose

- The transformation from c to p satisfies A1&A2
- D satisfies A3
- $\tilde{D}$ satisfies A4

Then there exists a c with generalization error ϵ that transforms to p with mean squared error at least $\alpha^2\epsilon$.

Justification of assumptions

A1: Without some control on how p depends on c, one can make c be an error correcting encoding of a good $p \Rightarrow$ no lower bounds possible

A2: If p does not depend on c, then p can be arbitrarily good independently of $c \Rightarrow$ no lower bounds possible

A3&A4: Probably not that serious, and can be relaxed.

Thus, the assumptions cannot be dropped altogether, but they may be unnecessarily strict.

Lessons learned?

The lower bound may be useful in the following ways:

- Shows that probing is close to optimal in the class of reductions satisfying the assumptions ⇒ fair to market probing as “optimal”
- Improvements upon probing have to violate some of the assumptions ⇒ lower bound narrows down the search space for potentially better reductions
- Shows why probing cannot be easily improved:
  - With binary error rate ϵ, up to an ϵ-fraction of the input space for probability prediction may remain totally unknown

End of probability prediction

Back to generalities on reductions and what can be done with them.

Constructive reductions (upper bounds)

- Given:
  - Problem classes A and B
  - A method M for solving instances of B
- Reduction: a (not too complex) mapping $f : A \rightarrow B$ s.t.
  - A solution to $f(a) \in B$ can be (sufficiently easily) transformed to a solution to $a \in A$.
- If a reduction f from A to B exists, then A can be solved by combining f and M.

Destructive reductions (lower bounds)

- Given:
  - Problem classes A and B
  - Task A known to be hard
- Reductions: Mappings $f\colon A \longrightarrow B$ with same properties as before.
- If a reduction from A to B exists, then also B is hard (i.e., B has hard instances).

Reductions in learning

Constructive:

- Statements of the form
  - If $f(a) \in B$ can be solved (to some accuracy), then $a \in A$ can be solved (to some related accuracy)
- Unrealistic to assume that all instances of B are sufficiently easy
- How to characterize or analyze the difficulty of $f(a)$?

Destructive:

- Perhaps not that meaningful, as all tasks B are known to have hard instances anyway (?)

Instead of lower bounds *by* reductions, we look for lower bounds *for* reductions.

Lower bounds for learning reductions in general

Assume:

- A “black box” method for solving instances of B
- Something on the properties/structure of f
- Something on the reconstruction strategy that transforms solutions to $f(a) \in B$ to solutions to $a \in A$.

Prove:

- There is a limit to the accuracy to which $a \in A$ can be solved, given that
  - The reduction/reconstruction strategy satisfies the assumptions
  - The accuracy in solving $f(a) \in B$ is independent of f

Additional details to consider

- How to measure accuracy (on A and B)? Possibilities:
  - Training set error
  - Mistake bounds
  - Test set error
  - Something else?
- Often, reductions are mappings $f\colon A \to B^k$ for some large k.
  - How to measure the joint performance of solutions to $f_1(a), \ldots, f_k(a)$?

Inherent limitations

- Without extra assumptions, any statement of the form
  - If $f(a) \in B$ can be solved (to some accuracy), then $a \in A$ can be solved (to some related accuracy)

  can be made true by ensuring $f(a) \in B$ cannot be solved/is hard enough to solve.

- Thus, one can argue that
  - Lower bounds are impossible
  - Upper bounds are meaningless

- To ensure that reductions makes sense, one needs
  - Deeper insight to the difficulty of $f(a) \in B$
  - Extra assumptions and/or extra care

Future work

- Main open problem:
  - **What makes a reduction natural?**