Rademacher Complexity and Grammar Induction Algorithms: What It May (Not) Tell Us.

Sophia Katrenko[1] and Menno van Zaanen[2]

[1] Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands
S.Katrenko@uva.nl
[2] TiCC, Tilburg University, Tilburg, The Netherlands
M.M.vanZaanen@uvt.nl

Abstract. This paper revisits a problem of the evaluation of computational grammatical inference (GI) systems and discusses what role complexity measures can play for the assessment of GI. We provide a motivation for using the Rademacher complexity and give an example showing how this complexity measure can be used in practice.

1 Introduction

Various aspects of grammatical inference (GI) have been studied extensively from both theoretical and practical points of view [3]. These include formal learnability results in the frameworks of the identification in the limit and PAC learning, as well as empirical methods. In the latter case, given a finite amount of sequential data, the aim is to find the underlying structure that was used to generate the data. Empirical approaches usually fall into the unsupervised learning paradigm and explore vast volumes of unlabeled sequences. One of the widely discussed questions in the literature concerns the performance of GI methods and their means of assessment. Van Zaanen and Geertzen [5] identify four evaluation strategies: the *looks-good-to-me*, *rebuilding an apriori known grammars*, *language membership detection* and *comparison against a treebank* approaches. All have weaknesses, some of which can be attributed to subjectivity, low scalability, and a bias towards specific grammars.

In practice, the comparison against a gold standard remains the most popular evaluation strategy. For instance the empirical comparison of ABL and EMILE [4] was based on unlabeled precision and recall. In this paper, we do not focus on accuracy of the GI methods but on their overfitting. In particular, it is known from statistical learning theory that classifiers prone to overfitting do not provide high generalization. In what follows, we give a definition of Rademacher complexity and discuss how to use it in the context of GI.

2 Rademacher Complexity

A goal of a learning system is to be able to analyze new, unseen examples and predict them correctly. In other words, given a set of n examples $\{(x_i, y_i)\}_{i=1}^{n}$

drawn i.i.d. from the joint distribution P_{XY}, it is supposed to produce a classifier $h : X \to Y$ such that it is able to categorize a new example $x \in X$. Any incorrect predictions that a classifier makes on a training set are counted as its empirical error $\hat{e}(h) = \sum_{i=1}^{n} I(h(x_i) \neq y_i)$, where I is an indicator function which returns 1 in the case $h(x_i) = y_i$ and 0, otherwise. Even though a classifier has access only to the limited number of examples (training set), one would ideally wish the empirical error on training examples $\hat{e}(h)$ to be close to the true error $e(h)$.

In statistical learning theory, it is common to describe the difference between true and empirical errors in terms of generalization bounds. These bounds typically depend on the number of training examples and capacity of a hypothesis space H. If a hypothesis space is very large and there are only few training examples, the difference between true and empirical errors can be large. Capacity closely relates to the notion of overfitting and emphasizes the fact that even if a classifier performs very well on the training set, it may yield poor results on a new data set. It is measured either by Vapnik-Chervonenkis dimension or Rademacher complexity and here we focus on the latter.

Definition 1 *For n training examples from a domain X, a set of real-valued functions H (where $h \in H$, $h : X \to \mathbb{R}$), a distribution P_X on X, the Rademacher complexity $R(H, X, P_X, n)$ is defined as follows:*

$$R(H, X, P_X, n) = E_{\mathbf{x}\sigma}\big(\sup_{h \in H} \big|\frac{2}{n}\sum_{i=1}^{n} \sigma_i h(x_i)\big|\big) \tag{1}$$

where $\sigma = \sigma_1, \ldots, \sigma_n$ are random numbers distributed identically and independently according to the Bernoulli distribution with values ± 1 (with equal probability), and the expectation is taken over σ and $\mathbf{x} = x_1, \ldots, x_n$.

Equation 1 shows that Rademacher complexity depends on the number of training examples n. In particular, larger number of examples will lead to lower complexity and, consequently, overfitting will also be low. In the binary case, where $h : X \to \{-1, 1\}$, Rademacher complexity ranges from 0 to 2. In a nutshell, Rademacher complexity shows how well a classifier can match random noise.

The use of Rademacher complexity to bound generalization error is discussed in [1] and is illustrated below.

Theorem 1 *(Bartlett and Mendelson) Let P_{XY} be a probability distribution on $X \times \{-1, 1\}$ with marginal distribution P_X on X, H be a set of functions such that each $h \in H, h : X \to \{-1, 1\}$. Let $\{(x_i, y_i)\}_{i=1}^{n}$ be a training set sampled i.i.d. from P_{XY}. For any $\delta > 0$, with probability at least $1 - \delta$, every function $h \in H$ satisfies*

$$e(h) - \hat{e}(h) \leq \frac{R(H, X, P_X, n)}{2} + \sqrt{\frac{\ln(1/\delta)}{2n}} \tag{2}$$

Equation 2 shows that if Rademacher complexity is high and a number of training examples is small, the generalization bound will be loose. Ideally, one would like to keep Rademacher complexity as low as possible, and a number of training examples sufficiently large.

3 Grammar Induction: Some Considerations

Tailoring Rademacher complexity to GI is not trivial because even though it is evaluated against existing *annotated* resources, it does not always fall in a typical supervised learning scenario. We assume that a grammar induction algorithm maintains several hypotheses and chooses the best one available, h_g. Depending on the input data, there are three possible strategies.

Supervised GI When a GI method is supervised, i.e. it is trained on sentences with their corresponding constituency structures, Rademacher complexity can be used to measure overfitting. This is the case of probabilistic context-free grammars (PCFGs). To measure Rademacher complexity, we need to specify what is an input space X and an output space Y. Usually, GI methods take a text corpus as input and generate constituents as output, which may suggest that X is a set of sequences (sentences) and Y is a set of subsequences (constituents). When comparing the output of an algorithm against a structured version of the sentences (i.e. a treebank), one considers how many constituents where found by a GI method and whether they match annotations. Consequently, we assume a hypothesis to be a mapping from constituents to binary labels, $h_g : X \to \{-1, 1\}$. Labels indicate whether a constituent from the gold standard was found by a GI algorithm (1) or not (-1).

To summarize, in the supervised case one may use the following evaluation scheme. For each constituent $x_i, i = 1, \ldots, n$ from the gold standard corpus, we generate a random label σ_i. In addition, we have a binary prediction from the GI method which indicate whether this constituent is generated by this particular method, $h_g(x_i)$. Finally, Rademacher complexity is computed as described in Equation 1.

Semi-supervised GI The second scenario is applicable when a GI method uses both labeled and unlabeled data. In such a case, transductive Rademacher complexity may be used, which is a counterpart of a standard Rademacher complexity.

Unsupervised GI In a fully unsupervised scenario, a GI method does not make use of labeled data for training and in this case we need another measure of overfitting instead of Rademacher complexity. However, in order to see what would happen in the case we simulate an evaluation proposed for supervised scenario, we have applied Alignment-Based Learning (ABL) [4] on the 578 sentence the Air Traffic Information System (ATIS3) subset of the Penn treebank (the edit distance-based alignment algorithm and the term probability selection learning method). As baselines, we also consider left and right branching binary tree structures. The generated structures have been compared against the ATIS3 gold standard, not taking empty constituents (traces) and the constituents spanning the entire sentence into account.

Table 1 shows that complexity rates for all three algorithms are low, which suggests that overfitting is low. Figure 1 illustrates that increasing the size of the training data lowers Rademacher complexity, although the differences are small here as well.

Table 1: Rademacher complexity and standard deviation on the ATIS3 corpus (100 runs).

Settings	Rademacher complexity
ABL	0.0267 (± 0.0227)
left branching	0.0295 (± 0.0215)
right branching	0.0304 (± 0.0219)

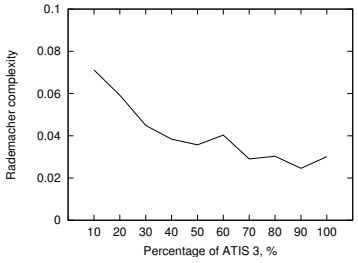

Fig. 1: Learning curve of Rademacher complexity of ABL on ATIS3 corpus.

4 Conclusions

In this paper, we discuss how to use Rademacher complexity to analyze existing grammar induction algorithms. In addition to commonly used measures, such as unlabeled precision or recall, the use of Rademacher complexity allows to measure overfitting of a method at hand. Since complexity is computed for a data sample, it makes it possible to study overfitting for the entire text collection, as well as on some subsets defined based on the sentence length or certain linguistic phenomena.

Rademacher complexity is well suited for supervised and semi-supervised settings. However, it remains an open question how overfitting should be measured in a completely unsupervised scenario. Recent work on clustering [2] suggests that, similarly to supervised learning, it is possible to restrict a function space in order to avoid overfitting. In future, we plan to investigate whether these findings can be used for unsupervised grammar induction.

References

1. Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:2002.
2. S. Bubeck and U. von Luxburg. Nearest neighbor clustering: A baseline method for consistent clustering with arbitrary objective functions. *JMLR*, 10:657 – 698, 2009.
3. Alexander Clark. *Unsupervised Language Acquisition: Theory and Practice*. PhD thesis, COGS, University of Sussex, 2001.
4. Menno van Zaanen and Pieter Adriaans. Alignment-Based Learning versus EMILE: A Comparison. In *Proceedings of the Belgian-Dutch Conference on Artificial Intelligence (BNAIC)*, pages 315–322, 2001.
5. Menno van Zaanen and Jeroen Geertzen. Problems with evaluation of unsupervised empirical grammatical inference systems. In *Proceedings of the ICGI 2008*, pages 301–303, 2008.