Model Selection of Sequence Prediction Algorithms by Compression

Du Xi DUCIS_CN@126.COM
reniku.cn / Shanghai Shenlou Mechatronics,
1018 Miyun Road,
Shanghai, China

Dai Zhuang DAIZ0420@GMAIL.COM
School of Transportation Science and Engineering, Beihang University,
XueYuan Road No.37, HaiDian District,
Beijing, China

Abstract

This paper describes estimating performance of sequence prediction algorithms and hyperparameters by compressing the training dataset itself with the probablities predicted by the trained model. With such estimation we can automate the selection and tuning process of learning algorithms. Spectral learning algorithm are experimented with.

Keywords: Minimum Description Length, Information-theoretic Approaches, Spectral Learning, Unsupervised Learning

1. Introduction

Using Kolmogorov complexity, minimum description length or the output of pragmatic compression algorithms to evaluate quality of predictions made by machine learning algorithms is a recurring theme (Cilibrasi and Vitnyi, 2005a) (Cilibrasi and Vitnyi, 2005b) (Mahoney, 1999) in ML/AI-related studies to the extent equating compression with general intelligence (Chaitin, 2002) (Chaitin, 2006) (Hutter, 2001).In general, we believe that the better a model predicts the probability distribution of future inputs, the better compression of historical inputs can be achieved using such predicted probabilities, and vice versa. In this short paper we apply such ideas on deciding the hyperparameters of prediction algorithms.

2. Compression by Prediction

We can compute a probability for any given sequence with a trained spectral learning (or any other prediction algorithm) model. In principle, a model that predicts the most accurately should also compress a non-biased (uniformly sampled) dataset to the smallest size. Note that the trained model itself should be included in the compressed data, but this is currently ignored for ease of implementation.

Noticing that only the length of the compressed data is needed, we choose to compute the theoretically optimal entropy-coded length for every sequence without actually computing the compressed form of the sequence. Let estimated probability of the sequence to be

compressed be p. The optimal entropy-coded length of the sequence is simply

$$c_{compressed} = -log_2 p \ .$$

The real size of a compressed sequence will always be larger than this theoretical minimum if we actually implement the compression, and converge to the minimum as the encoding scheme gets optimized.

The spectral learning model generates an unignorable amount of invalid estimates less than 0 or greater than 1, which means only some of the sequences can be compressed. The number of invalid estimates also vary significantly with the hyperparameters, so we cannot simply remove the sequences with invalid estimates. We are not sure about the cause of the negative/greater-than-one estimates, whether it is inherent to the model, specific to the implementation, or due to the runtime environment.

Anyway, any lossless compression algorithm can make some 'compressed' sequence longer than the original, and it is a trade-off whether to leave the sequence as is when the 'compressed' sequence is longer, which in turn requires an extra bit for every sequence. Because of the invalid estimates, the only viable option is to use the extra bit, and to compress the sequence only when the estimate is valid and the size of the compressed form is actually smaller.

For the uncompressed sequences, the code length is estimated as

$$c_{uncompressed} = -l \log_2 \frac{1}{m} = l \log_2 m$$

where l is the length of the sequence to be compressed and m is the size of the alphabet. At first glance, it may look attractive to utilize the frequency of the individual symbols when encoding the uncompressed sequences, but it is a compression scheme by itself with its own merits and pitfalls, and leads to some kind of ensemble method (Now that we are using unigram probabilites, why not n-gram or something more advanced?). So to keep things tractable we just stick to the completely uncompressed form.

The total compressed size is

$$t = \sum_{i=1}^{n_{compressed}} (-\log_2 p_i) + \sum_{i=1}^{n_{uncompressed}} l_i \log_2 m + n_{compressed} + n_{uncompressed} \ .$$

Note that ($n_{compressed}$+$n_{uncompressed}$) representing the space occupied by the 1-bit marks is just the total number of sequences and is constant for a given dataset.

Other than the total compressed size t, the number of sequences compressed $n_{compressed}$ and the average compression rate r of the sequences that are compressed are also considered, with

$$r = \frac{\sum_{i=1}^{n_{compressed}} -\log_2 p_i}{\sum_{i=1}^{n_{compressed}} l_i \log_2 m} \ .$$

3. Experiment

The algorithm has been applied to the 16 problems of the SPiCe contest. For each problem, we arbitrarily select a number of triples of hyperparameters, train the model with them,

run the compress test with the trained models, and finally run some of the models against the public test set. The models were manually picked to be run against the public test set if they show any signs of 'features', mostly local extrema of any of the three indicators, although the final submissions to the private test were still chosen among the public-tested models based on their public test score. Finally for each (problem,hyperparmeters) pair we obtain a score s and three compression-related indicators $n_{compressed}$, t and r.

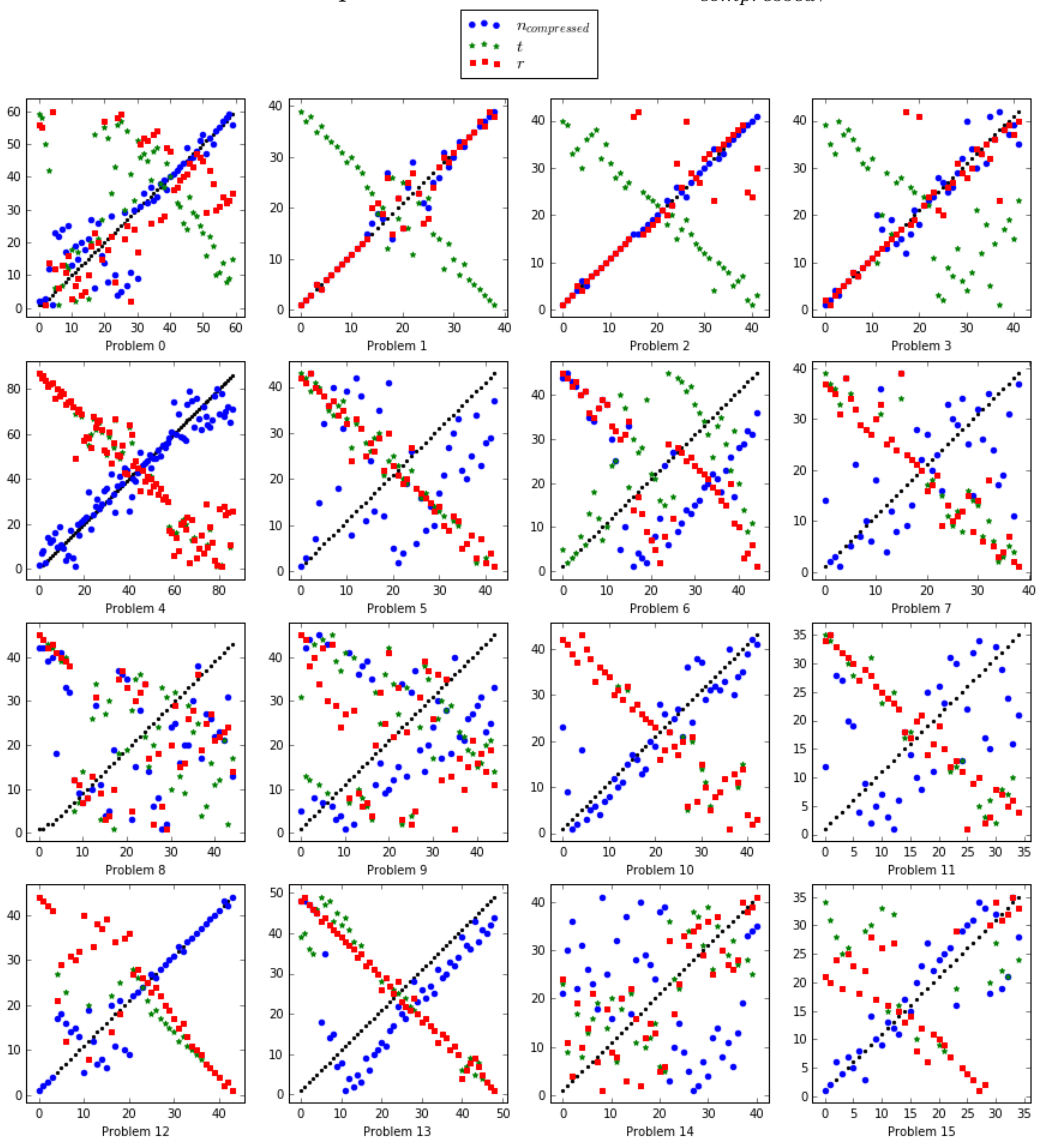

Figure 1: experiment results

As only whether better compression means better prediction is relevant, we rank the score s and all indicators ($n_{compressed}$,t,r) within each problem, and investigate the relationships among the rankings instead.

The final result can be exhibited as Fig. 1 showing the relationship between the ranking of public test score s and the rankings of the three indicators. As the figure shows, at least one of the rankings of the three indicators (the total compressed size s, the number of sequences compressed $n_{compressed}$, and the average compression rate r) have obvious correlation with the score ranking, except for problem 8, 9 and 14. In problem 0, 4, 6, 10, 12, 13, 15, $n_{compressed}$ ranking has a strong positive relationship with the score s ranking. In problem 0, 1, 2, 3, 4, 5, 7, 10, 11, 12, 13, total compressed size t ranking has a strong negative relationship with the score s ranking. In problem 1, 2, 3, the average compression rate r ranking has a strong positive relationship with the score ranking, and in problem 4, 5, 7, 10, 11, 12, 13, an obvious negative relationship exists between the average compression rate r ranking and the score s ranking. Note that $n_{compressed}$ always positively correlate with the score s if the correlation is visible, and t always negatively correlates with the score s if the correlation is visible, while r can correlate both ways, which requires further investigation.

So, to sum all circumstances up, it can be generally said that the ranking of indicators can be used to predict the ranking of public test scores, which in turn proves the effectiveness of the compression test.

4. Conclusion

From the experiments it can be concluded that compressing the training data can be used to assess prediction performance of a model, even without correctly including the predicting model in the compressed representation. Further work can be done on testing with more hyperparameters and datasets, compressing the predicting model or ensembling more learning algorithms.

References

Gregory Chaitin. On the intelligibility of the universe and the notions of simplicity, complexity and irreducibility, 2002.

Gregory Chaitin. The limits of reason. *Scientific American*, 294(3):74–81, 2006.

Rudi Cilibrasi and Paul M. B. Vitnyi. Clustering by compression. *IEEE Transactions on Information Theory*, 51:1523–1545, 2005a.

Rudi Cilibrasi and Paul M. B. Vitnyi. The google similarity distance, 2005b.

Marcus Hutter. Towards a universal theory of artificial intelligence based on algorithmic probability and sequential decisions. In *Proc. 12th European Conf. on Machine Learning (ECML-2001)*, volume 2167 of *LNAI*, pages 226–238, Freiburg, Germany, 2001. Springer. ISBN 3-540-42536-5. URL http://arxiv.org/abs/cs.AI/0012011.

Matthew Mahoney. Text compression as a test for artificial intelligence. In *In AAAI/IAAI*, pages 486–502, 1999.