Journal of Machine Learning Gossip 1 (2003) 11-19

Data Set Selection

Doudou LaLoudouana DOUDOULA3@HOTMAIL.COM
Lupano Tecallonou Center
Selacie, GUANA

Mambobo Bonouliqui Tarare FUZZYBEAR@YAHOO.COM
Lupano Tecallonou Center
Selacie, GUANA

Abstract

We introduce the community to a new construction principle whose practical implications are very broad. Central to this research is the idea of improving the presentation of algorithms in the literature and making them more appealing. We define a new notion of capacity for data sets and derive a methodology for selecting from them. Our experiments demonstrate that even not-so-good algorithms can be shown significantly better than competitors. We present some experimental results, which are very promising.

1. Introduction

Learning is a marvelous subject. A year spent in artificial intelligence is enough to make one believe in God. Unfortunately, so far it has been handled only from a particular one-sided point of view. The VC-theory known only by some people does not offer what we would think would be right to ask from such a theory—we want good bound for our algorithms. With this article, we offer a brand-new approach that allows you to present your algorithm in a much more principled and rigorous way than was possible before. Many researchers, especially those publishing in prestigious conferences such as NIPS or ICML, have tried to show when their algorithms are better (in some sense) than some other given set of algorithms. To do this, they have employed techniques of *data set selection*. It is strange, then, that learning theorists, as they call themselves, over the last 50 years have concentrated on the *model selection problem* and not the data selection problem, which is what people actually do. The two problems can, in some sense, be seen as duals of one another, but not because by solving one you can solve the other. Or vice-versa. In this article, we lay the foundations and introduce to the community of machine learning peers and other engineers a new induction principle: structural data set minimization. Essentially, we begin to formalize the sometimes ad hoc engineering approach of the selection procedure that everyone already practices. In doing so, we find concrete bounds for when the data selected really is better than other data sets and implement less ad hoc algorithms for finding such data sets. We show our approach outperforms the classical approach.

In contrast to its conents, the structure of the paper follows a classical trend: Section 1 presents some nice bounds you can use in lots of situations, Section 2 shows how to use these bounds by designing new algorithms. Section 3 describes some experiments which, of course, are good.[1] Section 4 concludes the article with smart thoughts and future work we will do.[2]

2. Bounds

Let us introduce some notation. Assume a researcher has invented an algorithm A^* and he wishes to show that his pride and joy is superior with respect to some loss function ℓ to a given fixed set of algorithms[3] $A_1, .., A_n$ that other researchers have proposed. For this purpose, the researcher selects some data sets using what is called an *empirical data set minimization* method. The latter consists of taking some fixed set of data sets $D_1, .., D_d$ and finding a data set D^* in the set such that:

$$\ell(A^*, D^*) < \min_{i=1,..,n} \ell(A_i, D^*).$$

Note that this problem is ill-posed. A natural generalization would be to find more than one data set in which your algorithm performs well, but this is a difficult problem that has not been solved so far by the community. Current efforts in solving this problem have focussed on producing more artificial data sets rather than algorithms to achieve this goal.

We have the following theorem:

Theorem 1 *Let $\mathcal{D}$ be a set of training sets, then assume that the space of algorithms is endowed with a fixed distribution $\mathbb{P}$ (which could be anything a priori), then with probability $1-\eta$ over a sampling on the algorithm, and for all $\gamma > 0$, we have:*

$$\forall D \in \mathcal{D}, \quad {R_{gen}}^A[D] \le {R_{emp_\gamma}}^A(D) + O\left(\sqrt{\frac{\Phi(\mathcal{D})}{m} \log(1/\eta)}\right),$$

where $\Phi(\mathcal{D})$ is the capacity *of the set of training sets defined as:*

$$\Phi(\mathcal{D}) = \max\{m \textit{ s.t. } \exists \textit{ algorithms } A_1, .., A_m \textit{ s.t. } \forall (r_{11}, .., r_{ij}, .., r_{mm}) \in [0,1]^{m(m-1)/2}, \\ \exists D \in \mathcal{D}, \quad \forall i \neq j |\ell(D, A_i) - \ell(D, A_j)| \le r_{ij}\}. \quad (1)$$

[1] Contrary to other authors, we include all our experiments in this paper, even the bad ones. But, well, we did not get any bad ones.

[2] If (and only if) we can get funding for it.

[3] Normally, a small set so he does not have to do too many experiments.

We are proud now to supply the following elegant proof.

Proof Let us denote by m the number of points in the training set. We see that introducing a ghost algorithm A':

$$\mathbb{P}_A\left(\sup_{D\in\mathfrak{D}} \left|{R_{\text{emp}}}^{A}[D] - {R_{\text{gen}}}^{A}[D]\right| > \epsilon\right) \le \mathbb{P}_{A,A'}\left(\sup_{D\in\mathfrak{D}} \left|{R_{\text{emp}}}^{A}[D] - {R_{\text{emp}}}^{A'}[D]\right| > \epsilon\right),$$

which is trivially insensitive to permutations, so we can condition over the algorithm A and A'. We then also have the right to play with the swapping permutation, as has been done in the theoretical-but-not-used-in-practice VC framework, which means that we work only with the values of (σ_1, σ_2). After some more steps (which we admit for brevity), this leads to the removal of the supremum. We are then left with a sum of two random variables, and this sum can be controlled using the Bennett-Bernstein inequality. It is here where the tricky part begins. It is known that averaging over two random variables does not give you accurate control of their expectation. But, this can be overcome if we consider many exact replica of the first two variables. At that point, we can choose from as many as we want! And, we can control the expectation, because now the value of m is large. We call this technique the *replica trick*. Note that the replica trick is used many times over during the invention of an algorithm: When exploring the space $\mathcal{A}$ of all possible algorithms, the same algorithm has been visited many times but with negligible variations so that if you use an ϵ-insensitive loss functions, these algorithms appear to be equivalent.[4] ■

The theorem we just proved should be considered as the dual of the theorem of Vapnik and Chervonenkis. And this should be the case because it is just the dual of it. We believe our result is the more natural setting for everyday algorithm design. Maybe our approach is complementary to that of Vladimir and Alexei, but we are one step forward because we can infer/compute the probability over the data sets just by looking at the UCI repository database. Just try to do the same with your set of functions and we'll talk. Anyway, our approach has a lot of properties in common with the classical VC framework. One of them is that, unfortunately, we cannot say much even though we try to. Or, to say it differently, we have our own "no free lunch" theorem, although we try to forget it. Here is our no free brunch!! theorem:

Theorem 2 (No Free Brunch!!) *The generalization error of two data sets over all algorithms is the same:*

$$E_A[{R_{gen}}^{A}[D]] = E_A[{R_{gen}}^{A}[D']],$$

meaning that there is no better data set than the one you already have.

[4] In fact, embedding machine-learning papers into a vector space, we found a large cluster with all points at a distance of less than 0.05 apart—the classical significance threshold used in statistics. We ran k-means clustering 50 times, but kept coming up with the same single big cluster.

The consequences of the theorem can be tough to take: it means that if you don't do well, you must not be very skilled since you can't blame the data. We still have not worked out what the researchers had for breakfast, by the way.

The preceding discussion leads to the natural consequence that to say anything one should restrict the set of algorithms to some natural set. In a companion paper we prove that the set of data sets restricted to all Bayesian algorithms has infinite dimension. The same is true for the set of all kernel algorithms if you leave the following parameters free: loss function (hinge loss, pound loss, epsilon-insensitive loss, ℓ_1, ℓ_2, ℓ_2+, Huber, hyperbolic tangent, Bregman, Breiman, etc.), regularizer (RKHS, $||w||^2$, $\sum_i \alpha_i^2$, $\sum_i \alpha_i$, KL divergence, relative entropy, $\log(\max_i \alpha_i)$ etc.) and the set of functions (we do not list here all possible kernels, but in future work we plan to examine the kernel-set-selection phenomena) and the optimization procedure (perceptron, gradient descent, chunking, quadratic, linear, stochastic, simulated annealing, analytic, random)[5].

The second major contribution of this paper is that our bound has exposed one of the major problems of empirical data set minimization—the de facto method over ten years of NIPS publications.[6] In particular, it is clear that the second term in the bound (the confidence interval over the set of data sets) is not taken into account. This leads us to propose a new construction principle called the Structural Data Sets Minimization (SDSM), which we describe next.

3. Structural Data Sets Minimization

In order to appreciate the following section, we ask the reader for a little patience and to concentrate a little. Assume you have an increasing set of data sets $\mathcal{D}_1, .., \mathcal{D}_k$ (*e.g.*, USPS, NIST, REUTERS 1, REUTERS 2, ...) , which are roughly included in each other:

$$\mathcal{D}_1 \subset \mathcal{D}_2 \subset \ldots \subset \mathcal{D}_k.$$

Then, we would like to apply the theorem of the previous section and choose the best data sets for our algorithm (*i.e.*, the one that will be used in the paper presenting these potentially new algorithms). Using the union bound, we found that with probability $1 - \sum_{i=1}^k \eta_i$, for all $D \in \cup \mathcal{D}_i$:

$$R_{\text{gen}}{}^A[D] \leq \underbrace{\min_{D \in \mathcal{D}_i} R_{\text{emp}}{}_\gamma^A(D) + O\left(\sqrt{\frac{\Phi(\mathcal{D}_i)}{m} \log(1/\eta_i)}\right)}_{\Psi(i)}. \tag{2}$$

So, we can pick the i^* such that $\Psi(i)$ is minimized over i, and then choose the best data set on $\mathcal{D}_{i^*}$. The consistency of this procedure can be ensured by a theorem.

[5] We do not cite the individual papers, we refer the interested reader to the NIPS volumes.

[6] See www.nips.com.

This section suggests that to find the best data set for your algorithm, you can consider many data sets and compute their capacity and then pick the best one not only in terms of empirical errors (which is strangely often called test error in many papers), but also in terms of this capacity.

Here is a nice picture (Figure 1).

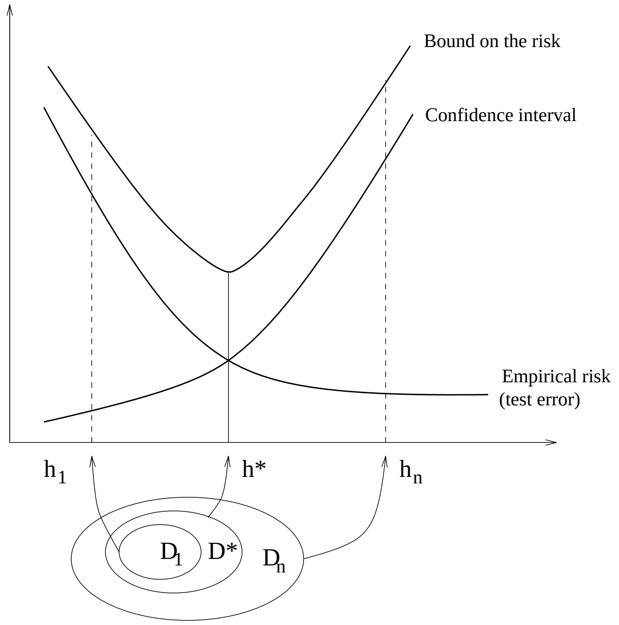

Figure 1: Graphical depiction of the structural data set minimization construction principle.

4. Algorithm Ordering Machine

We have now prescribed a formal method for choosing between real data sets. However, we don't really know the capacity of these real-world data sets, and only approximations can be calculated.[7] For this reason, we suggest using toy problems. It turns out that we can prescribe an efficient algorithm for searching for the best toy problem for a given algorithm A.

The highly successful and efficient algorithm prescribed is the following. Once again, assume a researcher has invented[8] an algorithm A^* and she wishes to show that it is superior with (dis)respect to some loss function ℓ to a given fixed set of

[7] Although, empirical observations show that most published algorithms do well, so we suspect the capacity must be quite high.

[8] And by "invented" an algorithm, we mean "slightly altered someone else's" algorithm.

algorithms $A_1, .., A_n$ that other researchers have proffered. Let us try to choose a fixed class of artificial data sets $\mathcal{D}$, for example, artificial data that is generated by a mixture of k Gaussians with covariance matrixes $c_{1,..k}$ in a d-dimensional space with a noise model e and f extra noisy features. Let us define $w = (c, d, e, f)$, we have the following theorem:

Theorem 3 *The capacity $\Phi(\mathcal{D})$ is bounded by:*

$$\Phi(\mathcal{D}) \leq \min\left(R^2\|w\|^2, n\right),$$

where n is the number of parameters (sum of dimensions of the vectors c, d, e and f), and R is the largest possible value of any coordinate of any vector in the data sets.

Some people would argue that this theorem is valid only for data sets parameterized by Gaussian distributions, but actually it can model a lot of real life problems. The interested reader can read the literature about Gaussian processes for examples.

Now, the task is to optimize the parameters such that algorithm A appears much better than the others. Let us choose A=Bruto, A_1=SVM, A_2=MARS, A_3=k-NN, A_4=C4.5, and we embed this set of algorithms with a uniform distribution to ensure no bias. This can be done with the following minimization:

$$\min_{c,d,e,f} \Phi(\mathcal{D}_{c,d,e,f})$$

$$\text{subject to:} \qquad {R_{\text{emp}}}^{A}[D] < {R_{\text{emp}}}^{A_i}[D] - \epsilon, \quad i = 1, .., 4$$

,

which is equivalent to:

$$\min_{w=(c,d,e,f)} \|w\|^2$$

$$\text{subject to:} \qquad {R_{\text{emp}}}^{A}[D] < {R_{\text{emp}}}^{A_i}[D] - \epsilon, \quad i = 1, .., 4$$

If you wish to find a data set where your algorithm does not achieve 0% test error, you can easily generalize this algorithm to the linearly inseparable case by introducing slack variables ξ_i.

The closeness to SVM is striking. Note that we are maximizing the margin between the algorithms to ensure a paper is accepted. The relation with statistical tests is open and has not been analyzed rigorously yet, but we have an upcoming paper on statistical test selection, so that even a small margin can guarantee a strong result.

5. Experiments

We proceed to prove our methodology by showing we can always find a good solution even for bad algorithms. So, we proceed with the example given above. We must admit that it has already been done in the literature, but we provide a deeper analysis

Algorithms	Without noise	With noise
Linear SVM	0.5 (0.005)	0.5 (0.01)
Poly-2 SVM	0.5 (0.1)	0.5 (0.2)
Poly-5 SVM	0.7 (0.2)	0.7 (0.1)
Poly-10 SVM	0.9 (0.8)	0.8 (0.6)
Mars	0.2 (0.2)	0.4 (0.1)
Bruto	**0.001 (0.00009)**	**0.002 (0.0001)**

Figure 2: Results for $w = (0.00001, 50000154839, 34, 3.14159, 2.755, 1, 2, 3, 4, 5, 6, 7, 8, 9, -1, -2, -3 \dots)$. Here, w corresponds to the parameters of the generated data set. Unfortunately, we have difficulties interpreting w.

(and also better margin!). In Table 2, we present the results we obtained with the best value for w.

Note that it is very difficult to discriminate between a linear SVM and a poly-2 SVM. The algorithms exhibit similar behavior, and we were not able to worsen the results of the poly-2 SVM, although it would have been nice. Poly-2 SVM performs well on a large number of data sets so the optimization was difficult—this may explain why we got strange value for w. On the other hand, it is quite clear in the table that Bruto is much better than all the other algorithms even MARS, although MARS has much in common with Bruto. Thus, our algorithm was able to discriminate between ϵ distances in the space of $\mathcal{A}$. We omit our other experiments for brevity but the results were good (trust us).

6. Conclusion

We will now reiterate what we said before, we repeat the abstract and introduction. The problem with any unwritten law is that you don't know where to go to erase it. This is the same for other matters, by the way. Consider, for instance, notation, such as that it is assumed that m or ℓ always refer to the number of training examples. Sometimes, it is n, but this occurs mainly when the authors are new to the field. Note that we also are new in the field but we did not use n. On the other hand, we have handled quite handily the use of greek letters. Anyway, the question we are discussing right now is to know when and how to stop an unwritten law. We believe this could be the place and the time, and, as a mark of courage, we choose not reiterate the introduction. This may then sound weird that we, as outsiders, put a stone in the sea of machine-learning papers. Indeed, We do not know the hydrodynamic laws of such a sea, nor do we know who discovered water, but we're pretty sure that it wasn't a fish. Not even a big fish with a fancy latin name.

Let us stop being polemic for a moment and come back to our contribution. Central to our new research is the idea of improving the presentation of algorithms in literature and to make them more appealing. We defined a new notion of capacity for data sets and derived a methodology for manipulating it. The experiments showed that even for not-so-good algorithms, you can show that they are significantly better than all other algorithms. The message is powerful and may not be understood at a first reading, so we say it plainly to avoid any confusion: we emplore all researchers to dig out their old failed algorithms and turn them into successful papers.

To be complete, we present in this last paragraph the future work we plan to do some day. We will make the link between data set selection and human neural information processing, which, so far, researchers have shown happens, in females, in the human brain[9]. We will consider whether data set selection is implemented via chemical stimulation of the neurons, possibly in the hippycampus. In humans, it could consist of, when failing to learn a task, buggering off and learning something else instead[10].

At last, we would like to say a mild word of caution. We hope that the community learns from this breakthrough and applies our methodology in their future research so they will not be left behind: our algorithms will far outperform theirs.[11]

References

P. Bartlett. The sample complexity of pattern classification with neural networks: The number of citations is more important than the number of readers. *Biowulf Transactions*, 1998.

C.M. Bishop. *Neural Not works for Pattern Recognition.* Oxford University Press, 1995.

N. Cristianini and J. Shawe-Taylor. *Data Set Selection for Dummies.* Egham University Press, 2005.

J. Friedman, T. Hastie, and R. Tibshirani. Data sets for additive models: a statistical view of bragging and boasting. Technical report, Stanford University, 1998.

G. Fung and O. L. Mangasarian. Data selection for support vector machine classifiers. In *Proceedings of KDD'2000*, 2000.

D. MacKay. I did it my way. Self published, 2002.

[9] Note that for males, some people conjecture it should be in some unknown other place, while yet others conjecture it doesn't exist at all. We will refer to this as the D-spot and note that, at the least, it is very hard to find, especially if the guy is overweight.

[10] Notice how many people learn to juggle around exam time.

[11] Finding the occasional germ of truth awash in a great ocean of confusion and bamboozlement requires intelligence, vigilance, dedication and courage. But, if we don't practice these tough habits of thought, we cannot hope to solve the truly serious problems that face us—and we risk becoming a nation of suckers, up for grabs by the next charlatan who comes along.

R. Shapire, Y. Freund, P. Bartlett, and W.S. Lee. Bushing the margin: A new explanation for the effectiveness of u.s. voting methods. *The Anals of Statistics*, 1998.

V.N. Vapnik, A.Y Chervonenkis, and S. Barnhill. *The VC Way: Investment Secrets from the Wizards of Venture Capital*. Biowulf Publishing (now out of print), 2001.

D.H. Wolpert. The lack of a priori distinctions between learning algorithm research. *Neutral Computation*, 1996.