Statistical Learning Machines: Notes from Theory and Practice

James D. Malley, Ph.D.

Center for Information Technology
National Institutes of Health
Bethesda MD   USA

jmalley@mail.nih.gov

The ultimate user-friendly learning machine.

As the machine studies the data, it teaches us and we are the learners.

Outline

Background

Machine Mythology: Learning and Unlearning Machines

Strategies for Learning: Open Books and Black Boxes

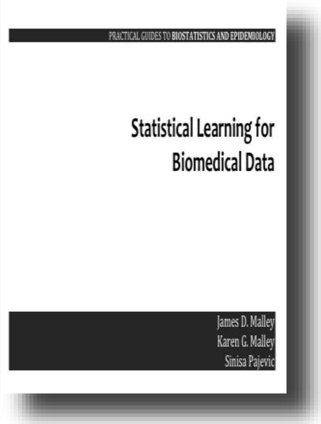
Research Questions

Background Resources

A Probabilistic Theory of Pattern Recognition
Luc Devroye, László Györfi, Gábor Lugosi
(Springer, 1996) = DGL

Applications of Mathematics
Stochastic Modelling and Applied Probability
31
Luc Devroye
László Györfi
Gábor Lugosi
A Probabilistic Theory of Pattern Recognition
Springer

Statistical Learning for Biomedical Data
JD Malley, KG Malley, S Pajevic
(Cambridge University Press, 2010)


PRACTICAL GUIDES TO BIOSTATISTICS AND EPIDEMIOLOGY
Statistical Learning for Biomedical Data
James D. Malley
Karen G. Malley
Sinisa Pajevic

The Bayes machine

The minimum Bayes loss rule is defined using the conditional probability function $\eta(x)$

$$\eta(x) = \Pr[Y = 1 | X = x]$$

And the Bayes machine (rule) $g*$ is

$$\begin{aligned} g^*(x) &= 1 \quad \text{if } \eta(x) \geq ½ \\ &= 0 \quad \text{if } \eta(x) < ½ \end{aligned}$$

The Bayes rule is best

No other rule (machine) can in principle have a lower Bayes error loss than $g^*(x)$

But we never know the true probability function $\eta(x)$
So we can't define $g^*(x)$

And it is very hard to estimate $\eta(x)$ not knowing any details about the distribution of the data

However, it's *not* necessary to accurately estimate $\eta(x)$ to get good binary decision rules

Background Details

Given training data D_n with sample size n
consisting of n training vectors

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

For $X = x_i$ a d-dimensional vector of features (attributes)
$Y = y_i$ a binary outcome $Y = 0, 1$

Want to predict new Y given new X
With machine (rule) $g(X)$: $g(X) = 0, 1$
generated using training data D_n : $g_n(X) = g(X; D_n)$

Evaluate machine using Bayes error (loss) L_n

$$L_n(g) = \Pr [\ g_n(X) \neq \mathrm{Y},\ \text{given training data } D_n\ ]$$

Unlearning Machines

Myth #1 There exists a super machine that will classify all data really well

FACT: Given machine *A* there always exists a universally good machine *B* and a distribution *D* such that:

1. For the data *D* the true Bayes error probability is exactly zero
2. Machine *A* has error probability > Machine *B* error probability (and for all sample sizes from *D*)

Some Cautions

1. Given a machine *A* there is not necessarily a machine *B* that is better for *every* data set.

2. Given a machine *A* and a machine *B* there is not necessarily a data set such that *B* is better than *A* on *that* data.

3. Both of these related—but not equivalent—assertions are open research questions.

4. Need basic probability and advanced combinatorial methods to make progress. See DGL, Chapter 1

    converges in probability = Bayes consistent

    converges a.e. = strongly Bayes consistent

Unlearning Machines

Myth #2 A machine must be complex and really quite clever in order to be very good

FACT: There are many simple, practical machines that are very good.

A key example: The *1*-nearest neighbor machine (*1*-NN) requires no training, no tuning, no hard-won optimization and yet for data with true Bayes error L the Bayes loss L_{nn} is

$$L < L_{nn} < 2L$$

So if the true Bayes error is small then *without any further work*, the Bayes loss for *1*-NN is also small

More details about Myth #2

Note that the apparent (observed) estimated error on the training data D_n for the *1*-NN machine is always exactly zero.

It is not magical or mysterious for a machine to have this property. The logitboost machine can have this property.

So, it does not *necessarily* mean we are overfitting the data and can't make good predictions on new data.

Hence *how* we estimate error on the training data D_n is important.

Unlearning Machines

Myth #3 (version 1)
A good machine needs very little data

FACT: For any good machine there is a data set D_n such that it is far above its large sample Bayes error

Given any small constant c there is a data set such that any sample of size n implies that the Bayes error is in the interval

$$½ > L_n > c$$

That is, the machine gets stuck arbitrarily close to coin-tossing. See DGL, Chapter 6

Unlearning Machines

Myth #3 (version 2)
A very good machine needs very little data

FACT: For a given good machine there exists data such that the estimated Bayes error converges arbitrarily slowly to its large sample lower limit.

1. This is true even for universally strongly Bayes consistent machines.

2. The Bayes error can be held above any decreasing sequence, for every *n*.

Unlearning Machines

Myth #4 A weak machine must be abandoned

FACT: Many weak machines (all with high error) can be combined to generate a provably very good machine (low error).

1. Basic idea goes back to Condorcet, 1785 (!)
1. Weak, provably inconsistent machines may be very strong when decisions are pooled (Gérard Biau et al., 2008).
1. These ensemble, or committee methods include *boosting*
1. Committee decisions are part of the *Random Forest* strategy
1. The method of Mojirsheibani (1999) goes further. . .

A superior committee method for these uncertain times

Mojirsheibani (1999, 2002) showed that any collection of machines could be pooled to get:

1. A group machine that is as least as good as the strongest machine in the set
2. A group machine that is Bayes optimal if any machine in the group is Bayes optimal
3. And we don't need to know which machine is which in (1) or (2).
4. Method is large sample optimal—see *all* the cautions above about small or fixed sample sizes
5. Method is a kind of single decision tree

Unlearning Machines

Myth #5 Finding optimal parameter estimates is the same as finding good machines

FACT: For the binary classification problem the real issue is getting good {0, 1} predictions and low Bayes error.

1. This is *not* the same as finding, say, the minimum squared error for any parameters in the machine code.

2. DGL has key examples showing that MSE can be minimized and yet the resulting machine has large error probability, far from minimum.

Unlearning Machines

Myth #6 A good binary machine works *because* it is a good estimator of the group probability

FACT: Simple examples show otherwise.

1. So, a good probability machine is certainly a good binary decision machine, but

1. An excellent binary rule can be not very good as a probability estimator.

1. Basic idea: a good machine only has to get a probability estimate that is on the same side of the decision boundary as the Bayes probability $\eta(x)$.
   It can be *very* different from the Bayes probability.

Unlearning Machines

More on Myth #6 . . .

4. Good machines like logitboost and Random Forest do not instinctively estimate the group probability.

5. Random Forest seems to be better than logitboost about estimating the group probability, but clearly isn’t that good.

6. It *might* be possible to re-engineer logitboost (LB) or Random Forest (RF) but this is not certain.

7. And both LB and RF can be seen as forms of neural nets, so some second hidden layer—transforming output from the initial hidden layer—might work.

Unlearning Machines

Myth #7 A good machine requires careful tuning to work well

FACT: Good machines need only some tuning, not much.
Most of the tuning rules just track the sample size.

Neural nets: the number of nodes, k, for the first hidden layer and with the sigmoid threshold, should grow but not too fast:

$k \approx \surd n$

Nearest neighbors: number of neighbors, k, should have

k/n going to zero, as n goes to infinity

Single decision trees: sample size, k_n, of the smallest cell (terminal node) should have

$k_n / \log n$ going to infinity, as n goes to infinity

Unlearning Machines

More on Myth #7 . . .

1. Sharper results and improved tuning requires more technique, especially

   Vapnik-Chervonenkis (VC) dimension

2. VC upper and lower bounds provide optimal probability statements about Bayes error and are known exactly for many machines.

3 The VC dimension is a measure of the flexibility of a machine, and needs to be high but not too high.

   Otherwise the machine will overfit: do well on sample, but not do well on test data.

Still more on Myth #7. . .

We ignore the VC dimension at our own peril.

It is *not* a theoretical swamp to be avoided by statisticians.

It is as important for practical reasons as is the bootstrap.

Unlearning Machines

Myth #8 A machine must see all the data and act as a global device in order to be good

FACT: A Bayes consistent machine must be local, and need be only weakly global.

1. This is recent work by Zakai and Ritov, 2008
2. This seems obvious for a nearest machine or decision tree, but it also holds for Support Vector Machines (known to be consistent) and boosting (also consistent)
3. The technical definition of *local* must be made precise, but it basically means that the machine doesn't need to see data far from a test point.

Unlearning Machines

Myth # 8 There must exist some unique small set of most predictive features

FACT: Numerous simple examples show the nonuniqueness of important features.

1. Good models need not be unique. Or, it might take infinite amounts of data to detect any difference between them.
2. Biological processes are typically not unique or singular
3. Relentless use of univariate methods over large feature sets is provably mistaken; See DGL for examples.

Unlearning Machines

Myth # 9 Competing sets of important features should be ranked and combined

FACT: Distinct lists of important features cannot always be combined and maintain logical self-consistency

1. Known to Condorcet, 1785 (again!)
2. Relates to the voting paradox of Arrow, 1951
3. Carefully studied by Saari and Haunsperger, 1991, 1992, 2003
4. It might be possible to use a kind of probabilistic ranking instead; this is a research question

Summarizing what we have unlearned

1. Benchmarking over a set of machines on several data sets
   —to find a grand super machine for *all* data—
   is provably mistaken

2. It *is* informative to look at a set of machines on a single data set to see how they behave on that data

3. Choosing a winner is often just unnecessary, since a committee will usually do better even with very weak machines:

   Random Forest and Random Jungle
   are excellent committee machines

Summarizing what we have unlearned

4. A good binary rule is not the same as a good group probability estimator—nor does it have to be
5. Machines need some tuning but not much if there is any signal in the data
6. Predictive feature sets can be identified but not uniquely so, and usually not with univariate screening

Our Academy of Higher Learning

1. *The Rosetta Stone Principle*
   If Nature has anything to tell us She will say the same thing in at least three different languages
2. All information is local and collective,
   only rarely global or singular
3. Validating a machine should occupy 85% of your brain power; actually running or tuning the machine should be minor.
4. Nature may be mysterious or random but She is not malicious.
5. Committee methods give us hope in a dark universe

Practical strategies for learning

1. Use a small set of machines to detect signal in the data
2. Pre-balance the data using over (under) sampling
3. Extensively validate the results (out of bag, permutation)
4. Have the machines declare some small set of features as most predictive
5. Send the small feature set to a familiar method, such as logistic regression: *move from black box to open book*
6. Check for error drift from the large machine to small machine: use Agresti-Tango methods for paired outcomes
7. Freely acknowledge that many different machines can do equally well

Research Questions

1. Find a solution for merging multiple lists of features
2. Develop methods for network and clique detection among entangled, weakly predictive features
3. Find good probability estimating machines;
   Re-engineer, or transform Random Forest? logitboost? nearest neighbor?
4. Find good machine that handles missing data *without* imputation; See Mojirsheibani and Montazeri, *JRSS-B*, 2007

Acknowledgements

Deanna Greenstein (National Institutes of Health, Bethesda MD)
Gérard Biau (Université Pierre et Marie Curie, Paris)
Luc Devroye (McGill University, Montreal)
Steven Stanhope (University of Delaware)
Joan Bailey-Wilson (National Institutes of Health, Bethesda MD)
Majid Mojirsheibani (Carleton University, Ottawa)
Andreas Ziegler (University of Lübeck, Germany)
Inke König (University of Lübeck, Germany)
Lawrence Brody (National Institutes of Health, Bethesda MD)

jmalley@mail.nih.gov