

A Generic Approach for Statistical Stability in Model Distillation

Yunzhe Zhou^{1*}, Peiru Xu² and Giles Hooker³

^{1*}Department of Biostatistics, University of California, Berkeley.

²Department of Mathematics, University of California, Berkeley.

³Department of Statistics, University of California, Berkeley.

*Corresponding author(s). E-mail(s): ztzyz615@berkeley.edu;

Contributing authors: xpr2019@berkeley.edu;

ghooker@berkeley.edu;

Abstract

Model distillation has been a popular method for producing interpretable machine learning. It uses an interpretable “student” model to mimic the predictions made by the black box “teacher” model. However, when the student model is sensitive to the variability of the data sets used for training, the corresponded interpretation is not reliable. Existing strategies stabilize model distillation by checking whether a large enough corpus of pseudo-data is generated to reliably reproduce student models, but methods to do so have so far been developed for a specific student model. In this paper, we develop a generic approach for stable model distillation based on central limit theorem for the average loss. We start with a collection of candidate student models and search for candidates that reasonably agree with the teacher. Then we construct a multiple testing framework to select a corpus size such that the consistent student model would be selected under different pseudo sample. We demonstrate the application of our proposed approach on three commonly used intelligible models: decision trees, falling rule lists and symbolic regression. Finally, we conduct simulation experiments on Mammographic Mass and Breast Cancer datasets and illustrate the testing procedure throughout a theoretical analysis with Markov process.

1 Introduction

Despite widespread adoption and powerful predictive performance, the models that result from machine learning are algebraically complex black boxes, which are not straightforwardly interpretable to humans. Providing explanations for the predictions of these models can be important in confirming the desiderata of ML systems including fairness, privacy, reliability and causality (Doshi-Velez and Kim, 2017) and can also help assess trust when one plans to deploy a new model (Ribeiro et al., 2016). Such explanations can be approached by two kinds of methods: intrinsic explanations and post hoc explanations, depending on the point at which the interpretability is obtained (Du et al., 2019). Intrinsic explanations are achieved by using models with intrinsically explainable structures such as linear models, decision trees and rule-based models. In contrast, post-hoc explanations require another model to provide insights after the existing model is trained. These explanations can be either model-specific, which derive explanations by examining specific internal model structures, such as measuring feature importance of tree-based ensemble models (Breiman, 2002); or model agnostic which treats the model as a black-box and doesn’t inspect its internal model structure. Model agnostic methods can be approached by a broad class of methods which develop values to measure feature importance, such as permutation feature importance (Breiman, 2001), SHAP values (Lundberg and Lee, 2017), to name a few. It can be also achieved by *model distillation*, which produces an understandable “student” model to mimic the predictions of the original black-box “teacher” model.

Model distillation has been widely used in interpretable machine learning (Tan et al., 2018; Zhou et al., 2018, 2021). The black box teacher model is first trained to perform well and then a new corpus of pseudo-data set of example features is generated with corresponding predictions from the teacher used as response. The student is then trained using this new data. The idea behind this is to use student as an explanation of the teacher’s predictions. Commonly used student models include decision trees (Johansson et al., 2011), generalized additive models (GAMS, Tan et al. (2018)), and LIME (Ribeiro et al., 2016). However, when a student model is sensitive to the variability of the data used to train it, a small perturbations of the data can result in different explanations, raising questions about their utility. We label this source of variance Monte Carlo variability; in this paper we develop means to control it by using a large enough pseudo-sample. Another source of variability comes from the uncertainty of teacher model; this variability is irreducible without collecting more training data. It can be important to quantify uncertainty in explanations due to uncertainty in the teacher; in this paper we focus on controlling Monte Carlo variability as a first step.

Monte Carlo variability can be controlled by developing tests to ensure enough samples are generated to train the student. An existing strategy for tree models makes use of the greedy training strategy to examine the stability of the split chosen at each node during the tree growing process (Zhou et al., 2018). Specifically, at each node the stability of the selected split is assessed

by examining the stability of the Gini split criterion and estimating the probability that an independent pseudo-sample would result in the same split being chosen. Then a multiple testing procedure is employed to account for selecting features between multiple potential splits. This framework is repeated at each split to make sure enough pseudo-data is generated to obtain a stabilized tree. Similar approaches were used by (Zhou et al., 2021) to stabilize the explanations produced by LIME (Ribeiro et al., 2016). There the Least Angle Regression algorithm (LARS, Efron et al. (2004)) for generating the LASSO path is considered and the procedure focuses on the asymptotic behavior of the statistics for selecting the next variable to enter the model. Similarly, multiple testing is conducted to incorporate other alternative features. However, both strategies are limited to specific student models.

In this paper, we propose a *generic method to control the Monte Carlo variability in model distillation*. We are still interested in how many pseudo samples are needed so that repeated runs of the explanation algorithm under same conditions yield the consistent results. We develop a testing procedure available for general student models without a readily analyzable training algorithm that we can exploit. We start with a collection of candidate student models. This can be generated by Monte Carlo simulation, Bayesian posterior sampling (van de Schoot et al., 2021) or any other algorithm which searches for candidates that reasonably agree with the teacher. Then we define a measure of agreement between candidate models and teacher model based on average loss. We then use a central limit theorem for the concentration of the average loss to estimate the probability that an independent pseudo-sample would choose a different candidate, and how many more samples are needed to reduce that probability below an acceptable tolerance. These techniques bear considerable similarity to sample size calculations in statistical hypothesis testing. We consider three commonly used intelligible models to demonstrate our proposed approach: decision trees (DT), falling rule lists (FRL) and symbolic regression (SR).

We observe several important factors in our framework. First, we require the number of candidate models to be sufficiently large to guarantee adequate coverage. Second, a larger corpus size can result in more complex student models and a great variety of structures to choose from, making stabilization much more difficult. The complexity of the students also hinders their interpretability and we therefore truncate the complexity of student models. Finally, when we have two competing candidate models that are difficult to distinguish, the testing procedure can demand an extremely large corpus size. Therefore, we set an upper bound for the corpus size such that when this maximum size is reached, the test stops and the best candidate model is chosen. This can prevent the testing procedure from running too long when two candidate models are not easily distinguishable. We note, however, that this can compromise the goal of stability. We conduct sensitivity analysis to study the effect of the number of candidate models, the model complexity constraint and the maximum sample size.

Mathematically, we treat the testing procedure as a Markov process with a stopping rule and states given by the corpus size where the transition probability is explicitly calculated. We further derive an upper bound for the new corpus size required by the test and calculate the probability that the test will stop given the current corpus size.

The paper is organized as follows. We first introduce our proposed hypothesis testing framework along with three interpretable student models as examples in Section 2. Then we develop some theoretical analysis by regarding the testing procedure as a Markov process and derive some useful bounds for explanation of stabilization difficulty in Section 3. In Section 4, we conduct experiments on two real datasets, as well as a sensitivity analysis for several important factors. Conclusions and future work are discussed in Section 5. All technical theorems and proofs are provided in the supplementary Appendix.

2 Methodology

In this section we present the details of our proposed generic approach for stable model distillation. We first derive the asymptotic properties for the average loss based on a central limit theorem, which can be used to calculate the probability that a fixed alternative model structure would be selected under a different pseudo sample. We can then perform a sample size calculation to choose a large enough corpus size to control this probability and further employ a multiple testing procedure to control for other alternative candidate models. We summarize our proposed algorithm and discuss several important hyperparameters in the testing framework. Finally, we discuss application of our testing framework to specific cases of candidate student models.

2.1 Asymptotic properties for the Average of Losses

In the context of model distillation, given a black box teacher model F , we produce a pseudo sample $\{(X_i, Y_i)\}_{i=1}^n$ of size n , where the response $Y_i = F(X_i)$ is the prediction of the teacher model. Consider a collection of candidate student models S_1, \dots, S_N with different structures. We define a measure of agreement between potential models and the teacher based on the average loss:

$$L(F, S_j) = \frac{1}{n} \sum_{i=1}^n l(F(X_i), S_j(X_i)),$$

for $j = 1, \dots, N$. $l(\cdot, \cdot)$ is an appropriate loss function for the problem at hand. Then we define the “best” student model S_k as the one which minimizes this average of losses:

$$S_k = \arg \min_{S_j} L(F, S_j)$$

Denote $d_{jk} = L(F, S_j) - L(F, S_k)$ as the loss-gap between the best student model S_k and a competing student model S_j . According to central limit theorem, as $n \rightarrow \infty$,

$$\sqrt{n}(d_{jk} - \mu_{jk})/\sigma_{jk} \implies N(0, 1) \quad (1)$$

satisfies

$$\begin{aligned} \mu_{jk} &= \mathbb{E}[l(F(X), S_j(X)) - l(F(X), S_k(X))], \\ \sigma_{jk}^2 &= \text{Var}[l(F(X), S_j(X)) - l(F(X), S_k(X))]. \end{aligned}$$

Or asymptotically,

$$d_{jk} \sim N\left(\mu_{jk}, \frac{\hat{\sigma}_{jk}^2}{n}\right), \quad (2)$$

where the variance estimate $\hat{\sigma}_{jk}^2$ is estimated from the empirical variance of the values $l(F(X_i), S_j(X_i)) - l(F(X_i), S_k(X_i))$ for $i = 1, \dots, n$.

We are interested in whether enough pseudo samples are generated to distinguish between candidate models with different structures. Suppose we have already observed that $d_{jk} > 0$, we want to calculate the probability that this will still hold in a repeated experiment. Assume that we have another independently generated pseudo sample denoted by $\{(X_i^*, Y_i^*)\}_{i=1}^n$. Similarly, we denote

$$L^*(F, S_j) = \frac{1}{n} \sum_{i=1}^n l(F(X_i^*), S_j(X_i^*)).$$

Then we can check whether the same decision is obtained by assessing $d_{jk}^* = L^*(F, S_j) - L^*(F, S_k) > 0$. Equation 2 implies

$$d_{jk}^* - d_{jk} \sim N\left(0, \frac{2\hat{\sigma}_{jk}^2}{n}\right),$$

which leads to the approximation that

$$d_{jk}^* | d_{jk} \sim N\left(d_{jk}, \frac{2\hat{\sigma}_{jk}^2}{n}\right).$$

In order to control $P(d_{jk}^* > 0)$ at confidence level $1 - \alpha$, we then need

$$d_{jk} > Z_\alpha \sqrt{\frac{2\hat{\sigma}_{jk}^2}{n}},$$

where Z_α is the $(1 - \alpha)$ -quantile of a standard normal distribution.

Define the p-value $p_{n,j}$ for a fixed confidence level α and n by

$$p_{n,j} = P(d_{jk}^* < 0) := P\left(Z > \frac{\sqrt{n}d_{jk}}{\sqrt{2\hat{\sigma}_{jk}^2}}\right), \quad (3)$$

where Z follows the standard normal distribution. We want to find n' such that

$$p_{n',j} = P\left(Z > \frac{\sqrt{n'}d_{jk}}{\sqrt{2\hat{\sigma}_{jk}^2}}\right) < \alpha.$$

This requires that

$$n' > \frac{2Z_\alpha^2 \hat{\sigma}_{jk}^2}{d_{jk}^2}. \quad (4)$$

In order to account for multiple competing student models, we employ a multiple testing correction to control the probability that the same decision would be obtained again under an independent pseudo-sample. Consider the $N - 1$ competing student models for model S_k . We test the hypothesis $\mathcal{H}_{j,0} : d_{jk} > 0$ and obtain the p-values $p_{n,j}$ for $j \in \mathcal{J}_k = \{1, 2, \dots, n\} \setminus \{k\}$. We can control the probability of selecting any other candidate by Bonferroni's inequality

$$P(d_{jk} > 0, \forall k \neq j) \leq \sum_{j \in \mathcal{J}_k} p_{n,j}$$

and we can stop if this sum is less than the predetermined tolerance α . Otherwise, we require more samples for distinguishing the candidate models. In practice, in order to find the sample size n' necessary for stabilization, we can employ a linear search by setting

$$n' = (1 + tL)n \quad (5)$$

and increasing t with the rate L (e.g. 0.1) until $\sum_{j \in \mathcal{J}_k} p_{n',j} \leq \alpha$.

2.2 Generic distillation for statistical stability

Based on the asymptotic properties derived above, we come up with a generic distillation approach for statistical stability. We begin with a real dataset \mathcal{D} and train a black box teacher model F on it. We do not specify the form of F but assume it is not intelligible by itself. And we are interested in explaining its predictions

By using the pretrained teacher model F , we can generate a synthetic data $\{(X_i, F(X_i))\}_{i=1}^{n_{\text{init}}}$ of initial size n_{init} . This can be generated according to

either a fixed or estimated distribution. We use a kernel smoother throughout the paper (Wand and Jones, 1994). There are also many state-of-the-art approaches with deep generative modeling, such as mixture density network (Bishop, 1994), generative adversarial networks (GAN) (Creswell et al., 2018) and variational autoencoder (VAE) (Kingma and Welling, 2013). Alternatively, we can ignore dependence between different covariates and use a simple Gaussian distribution or binomial distribution to model each covariate according to data types.

Candidate student models can be collected by broad class approaches including Monte Carlo simulation, Bayesian posterior sampling or any other algorithm which searches for candidates that reasonably agree with the teacher. We provide more concrete examples in Section 2.3 below. Instead of focusing on different values of model parameters, we are more concerned with the interpretation implied by different model structures (e.g. the tree structures of decision tree). Thus, we divide the candidate models into equivalence classes according to different structures. Specifically, we create equivalence classes $\Omega_1, \Omega_2, \dots, \Omega_M$ for the collections of candidate models S_1, \dots, S_N , such that if S_i and S_j belong to the same equivalence class Ω_k , they should share the same structure; otherwise, their structures are different and they are separated into different equivalence classes. We constrain the complexity of candidate student model with the cutoff C (e.g. maximum depth of decision tree and maximum length of the rule list).

Next we generate a synthetic sample of size n and define the loss function $L(F, S)$. We choose the student model $S_{(i)}$ with the smallest loss inside each equivalence class Ω_i :

$$S_{(i)} = \arg \min_{S_j \in \Omega_i} L(F, S_j),$$

for $i = 1, 2, \dots, M$ and we call $S_{(i)}$ as the representative of the class Ω_i . In the same spirits of Section 2.1, we define the best candidate student among these representatives by

$$S_{(k)} = \arg \min_{S_{(j)}} L(F, S_{(j)}).$$

Then we can derive the p-value $p_{n,(j)}$ for each competing model $S_{(j)}$ with Equation (3). The null hypothesis is rejected if $\sum_{(j)} p_{n,(j)} < \alpha$. Otherwise, we perform linear search in (5) for the updated size n' . Then we iterate previous procedures by going back to generate a new collection of student models and implement the hypothesis test based on a new synthetic data of size n' . We repeat this process until the testing is passed or a maximum sample size n_{\max} is reached. We summarize the whole procedures in Algorithm 1.

There are several important factors in our algorithm framework which we list as below.

- **Number of candidate student models (N):**

N is defined by the number of candidate student models from which we select the best student. Since the total number of equivalence classes M increases with N , a larger N implies a greater variety of model structures for the candidate students. We require N to be sufficiently large for adequate

Algorithm 1 Generic Distillation

Input: Real dataset \mathcal{D} , number of candidate student models N , significant level α , maximum sample size n_{\max} , initial sample size n_{init} , complexity constraints C .

Initialize: Fit a teacher model F on \mathcal{D} and set $n = n_{\text{init}}$.

Step1: Produce a collection of candidate student models $\{S_1, \dots, S_N\}$ under the complexity constraints C and divide them into equivalence classes $\{\Omega_1, \dots, \Omega_M\}$ according to different model structures.

Step2: Generate a synthetic data $\{(X_i, Y_i)\}_{i=1}^n$ of size n and define the loss function $L(F, S)$ based on this. Pick up the student with the smallest loss inside each equivalence class $S_{(i)} = \arg \min_{S_j \in \Omega_i} L(F, S_j)$ for $i = 1, 2, \dots, M$ as the representative.

Step3: Let $S_{(k)} = \arg \min_j L(F, S_{(j)})$ to be the best candidate student among the representatives and derive the p-value $p_{n,(j)}$ for each competing model $S_{(j)}$ with Equation (3).

Step4: If $\sum_{(j)} p_{n,(j)} > \alpha$ and $n < n_{\max}$, perform linear search in (5) to get the updated size n' . Otherwise, return the output.

Step5: Set $n = \min\{n', n_{\max}\}$ and go back to **Step 1**.

Output: $S_{(k)}$.

coverage. As shown in Section 4, the algorithm is relatively insensitive to the choice of N as long as it is large enough. Nevertheless, a relatively small N would lead to poor performance of the stabilization because some important model structures might not be available and therefore not chosen as the best student model. In addition, as the sample size n increases, the structures missed at initial stages may be chosen later.

• **Complexity constraint of models (C):**

A large sample size n would make the distillation process more difficult to stabilize because it results in more complex student models and greater variety of structures. Besides, a broad class of model structures would lead to many local minima of the loss function and make it intractable to get a stable optimization solution. Therefore, we need to truncate the complexity of student models. We introduce the notation C for the constraint of complexity. In Section 2.3, we provide more specific examples of C under different candidate models.

• **Maximum sample size (n_{\max}):**

It is highly possible to have two competing candidate models $S_{(j)}$ and $S_{(k)}$ such that μ_j and μ_k are very close to each other. As a result, it is difficult to distinguish between these two models and our testing procedure would suggest an extreme large n' , which is computationally infeasible. So in practice, we set the cutoff n_{\max} as the maximum sample size for n to reach. If the hypothesis test suggests a n' larger than n_{\max} , we stop our testing procedure and use the best model chosen under synthetic data of size n_{\max} . By setting this upper bound for n , we can prevent the testing procedure

from running too long when two candidate models make very similar predictions. However, a relatively small choice of n_{\max} would lead to decrease in testing power and the resulting distillation structures are less stable. Thus, there is a tradeoff between the computational complexity and the stability of distilled model in terms of n_{\max} .

2.3 Candidate Student Models

In this paper, we consider three specific cases of intelligible student models: decision trees, falling rule lists and symbolic regression; we incorporate them into our testing framework respectively.

Decision Tree (DT)

DT uses a flowchart-like tree structure where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node holds a class label (Breiman et al., 2017; Quinlan, 1987). The tree-like structure of DT makes it easy to interpret since the terminal nodes of the tree provides explanations for the model predictions (Ribeiro et al., 2016). It is among the most popular machine learning algorithms which can visually represent the decision making process (Wu et al., 2008). There are various DT algorithms including ID3 (Wu et al., 2008), C4.5 (Quinlan, 2014), C5.0 (Kuhn et al., 2013), and CART (Loh, 2011) (update this citation?). We focus on CART throughout this paper.

Specifically, for regression task, given the data $\{(X_i, Y_i)\}_{i=1}^n$, CART splits the whole feature space into Q units R_1, R_2, \dots, R_Q and defines an output value c_q at each unit R_q , which can be estimated by taking the average of Y_i with its corresponded $X_i \in R_q$. Then the prediction model can be formalized as

$$f(x) = \sum_{q=1}^Q c_q I(x \in R_q).$$

This model is trained by greedily selecting splits which most reduce the current loss, usually given by squared error for regressions and gini index or entropy for classification. Figure 3 shows an example of DT.

For the purpose of model distillation, we constrain the maximum depth of the tree structure at C . In order to collect candidate DT models, we use Monte Carlo simulations by generating a set of synthetic datasets $\mathcal{D}_1, \dots, \mathcal{D}_N$ and training a DT model separately on each of them. This produces a group of candidate student models S_1, \dots, S_N . In addition, we divide them into equivalence classes according to different tree structures; S_i and S_j fall into the same equivalence class if they use the same feature at each split. Alternatively, in Zhou et al. (2018), a greedy structure of DT is used for producing stable model distillation.

Falling Rule List (FRL)

FRL is an interpretable probabilistic decision list for binary classification consisting of an ordered list of if-then rules (Wang and Rudin, 2015). It is very useful in clinical practice which requires a natural decision-making process, where the observations with highest risks (e.g. symptoms of high severity diseases) are classified first and then the second most at-risk observations are considered, and so on. To be more specific, FRL is modeled by the following parameters:

- Size of list: $H \in \mathbb{Z}^+$.
- IF clauses: $c_h(\cdot) \in B_X(\cdot)$ for $h = 0, \dots, H - 1$.
- Risk scores: $r_h \in \mathbb{R}$ for $h = 0, \dots, H$ and $r_{h+1} \leq r_h$ for $h = 0, \dots, H - 1$.

where $B_X(\cdot)$ is the space of all possible IF clauses and $c_l(\cdot)$ is an identity function on feature space X , which is 1 if the set of conditions are satisfied and 0 otherwise. The value of r_h is transformed by a logistic function to a risk probability between 0 and 1. r_h is also assumed to be monotone with $c_0(\cdot)$ as the rule at the top of the list. To estimate the parameters and make inference with FRL, a Bayesian approach is used to characterize the posterior over falling rule lists given training data and a combination of Gibbs sampling and Metropolis-Hastings is employed for generating a trajectory of candidate models to perform posterior sampling. See more details in Wang and Rudin (2015). We provide an example of FRL in Table 1.

In the context of model distillation, we define C to be the maximum length of the rule list and constrain $H \leq C$. We take the advantage of Bayesian framework by using posterior sampling for collecting candidate models. Then the generated models are further divided into equivalence classes with regard to the rule list structures. According to the Bernstein–von Mises theorem (Doob, 1949), as the sample size n goes towards infinity, such candidate models generated from posterior sampling are asymptotically equivalent to those produced by Monte Carlo simulations with multiple synthetic datasets. However, in practice, the posterior sampling process of FRL frequently falls into a local region and the resulting produced candidate students may fail to have an adequate coverage of possible list structures. Therefore in practice, we first generate a few synthetic datasets $\mathcal{D}_1, \dots, \mathcal{D}_P$ and produce the trajectory T_j of candidate models for each \mathcal{D}_j . Then we aggregate T_1, \dots, T_P together as the final collection of candidate models. This is equivalent to using several starting points to resolve the local optima in intractable optimization problems.

Conditions			Probability
IF	IllDefinedMargin AND Age ≥ 60	THEN malignancy risk is	90.20%
ELSE IF	IrregularShape	THEN malignancy risk is	82.46%
ELSE IF	SpiculatedMargin	THEN malignancy risk is	44.83%
ELSE		THEN malignancy risk is	5.74%

Table 1: Example of Falling Rule List.

Symbolic Regression (SR)

SR is a supervised learning model which searches the space of mathematical expressions that best fit an observed data set. Its model structure is represented by an expression tree, which consists of mathematical operators, analytic functions, constants, and state variables (Augusto and Barbosa, 2000). Figure 1 shows a specific example that uses expression tree to represent $a * y - b + z/x$. The search process is carried out via genetic programming, in which a population of candidates is evolved to mimic the biological evolutionary process (Koza, 1992). Specifically, we first start with a population which consists of randomly generated equation models. Then a set of transformations is employed on this population to generate the “children”, including mutation, crossover, and traversing. We choose the top units from the children in terms of their prediction performance as our second generation of the population and apply transformations to them again to produce next generations. The whole procedure is repeated until the maximum number of generations is reached.

In the interpretability context, SR returns an analytical model with interpretable symbolic equations (Affenzeller et al., 2014; Ferreira et al., 2020; Aldeia and de França, 2022). By representing the symbolic formulas with an expression tree, we define C by the maximum depth of the expression tree in context of SR. We use the population at the last generation as the collection of candidate student models. In this case equivalence classes cannot be defined merely in terms of the expression tree structure because two different tree structures can sometimes represent the same symbolic formulas. For example, the equations $x + 1$ and $x + 2 - 1$ have different expression tree structures. Another example is concerned with the symmetry of expression tree: $x + y$ and $y + x$ would result in two different tree structures which are reverse to each other. Thus, when creating the equivalence class, we would take these scenarios into account and also we ignore the difference of the numerical numbers in the formulas. The SymPy package in Python can achieve this by changing formulas with the same meaning into identifiable objects (Meurer et al., 2017).

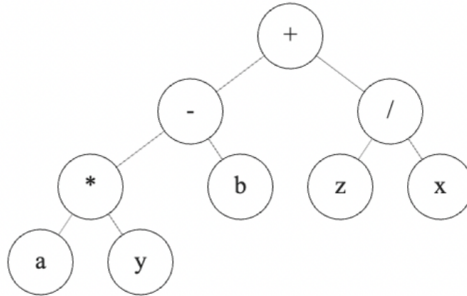


Fig. 1: Expression tree for formula $a * y - b + z/x$.

3 Theory

In this section, we illustrate the proposed procedure through a theoretical analysis by treating it as a Markov process with a stopping rule and demonstrate how large n would be required or how difficult the stabilization would become to distinguish between a group of candidate student models.

Consider a collection of candidate student models with different structures S_1, \dots, S_N and the teacher $F(X)$. We start with the following normality assumption to establish the theoretical properties of our testing framework:

Assumption 1. *Suppose for any $j \neq k$, the average loss difference between any two candidate models follows a normal distribution:*

$$d_{jk} = \frac{1}{n} \sum_{i=1}^n [l(S_j(X_i), F(X_i)) - l(S_k(X_i), F(X_i))] \sim \mathcal{N}\left(\mu_{jk}, \frac{\sigma_{jk}^2}{n}\right).$$

Assumption 1 imposes the normality condition for the loss function l . It directly implies the normality of the average loss difference, on which our testing framework relies. We define the standardized difference by

$$S_{jk} = \frac{\mu_{jk}}{\sigma_{jk}},$$

and then define the smallest absolute value of S_{jk} between any two candidate models by

$$S^* = \min_{j \neq k} |S_{jk}|,$$

which measures the distance between two candidate students with the most similar prediction performance. Empirically, the distribution of S^* depends on the complexity of the model space for the student candidates. Mathematically, if we denote the complexity constraint as C , we assume that,

Assumption 2. *Both $N = N(C)$ and $S^* = S^*(C)$ are the functions of the complexity C .*

Assumption 2 demonstrates that the complexity constraint would affect the number of candidate student models required for the testing framework and the difficulty to distinguish between the two closest student candidates. For example, when we consider DT for distillation, there are $N(C) = O(2^{d^C})$ possible candidate tree structures in total under the maximum depth C , where d is the total number of features. $S^*(C)$ should also decrease with C , since deeper tree structures would make it more difficult to distinguish between them.

The whole testing procedure can be treated as a Markov process with the stopping rule. In this framework, the correct sample size gives the state of the

markov chain. We then have two possible transition types. Then the transition goes two possible directions for next state:

- (1) the test rejects the null and the procedure stops.
- (2) the test shows n is not large enough to distinguish between the candidate students and another larger sample size n' is suggested. Figure 2 describes this Markov process given the current sample size n , where ω_1 is the probability that the procedure stops so n is kept since there is enough power to distinguish between different candidate students; and ω_2 is the probability that a new sample size n' is suggested.

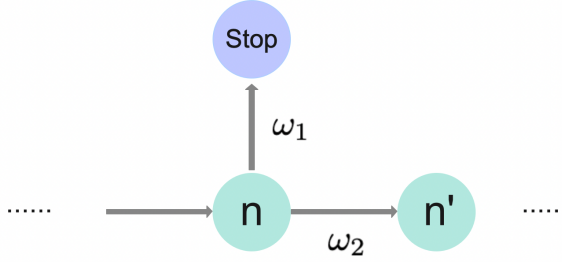


Fig. 2: Markov Process for Testing Framework in Algorithm 1

Using the notation above, we represent the transition probability by

$$p(n'|n) = \omega_1 p_1(n'|n) + \omega_2 p_2(n'|n),$$

where p_1 denotes the transition probability if the procedure stops while p_2 is the case when a new sample is suggested.

The following theorem demonstrates how the difficulty of distinguishing between different candidate models is related to the complexity constraint for our testing framework.

Theorem 1 *Suppose Assumptions 1 and 2 hold, for relatively large n , we have*

- (i) *The probability that the testing procedure stops is bounded below by*

$$\omega_1 \geq 1 - N(C) \sqrt{\frac{8}{\pi} \log \left(\frac{N(C)}{2\alpha} \right)} e^{-\frac{1}{2} \left(\sqrt{n} S^*(C) - 2 \sqrt{\log \frac{N(C)}{2\alpha}} \right)^2},$$

which is equivalent to

$$\omega_2 < N(C) \sqrt{\frac{8}{\pi} \log \left(\frac{N(C)}{2\alpha} \right)} e^{-\frac{1}{2} \left(\sqrt{n} S^*(C) - 2 \sqrt{\log \frac{N(C)}{2\alpha}} \right)^2}.$$

- (ii) *With high probability,*

$$n' < (4n \log n) N^2(C) \log \left(\frac{N(C)}{2\alpha} \right).$$

Theorem 1 illustrates how challenging the stabilization problem can be if the model is quite complex (i.e. relatively high value of C). Specifically, as C increases, $S^*(C)$ becomes small and the term $\sqrt{n}S^*(C)$ shrinks to 0. In this case the lower bound for ω_1 is close to the order of $1 - O\left(\sqrt{\log[N(C)]}\right)$, indicating a high probability that the transition to a new sample size will happen. On the other hand, since $N(C)$ also increases with C , it leads to a larger upper bound for the new sample size n' , implying that more pseudo samples are required to distinguish between these candidate models with more complex structures.

This result of Theorem 1 is quite intuitive. It shows the importance of truncating the model complexity so that the term $\sqrt{n}S^*(C)$ is of large magnitude and the upper bound for ω_1 is exponentially decaying to 0. Otherwise, without further truncating the model complexity, we can expect the Markov process will result in n' that exceeds computational capacity.

4 Experiments

In this section we present experiments on two real-world datasets using our proposed generic model distillation algorithm, along with a sensitivity analysis for several important factors in our testing algorithm.

4.1 Model Settings

We begin with the general setup for our experiments. We focus on the binary classification task in our experiments and use the cross entropy as the loss function L , that is,

$$L(F, S) = \frac{1}{n} \sum_{i=1}^n l(F(X_i), S(X_i)),$$

$$l(F(x), S(x)) = F(x) \log(S(x)) + (1 - F(x)) \log(1 - S(x)),$$

where $F(x)$ is the binary label predicted by teacher F at feature point x while $S(x)$ is the student’s predicted probability that the label is 1.

We first train a random forest F on the real data and keep it fixed as the teacher throughout the whole experiment. If the covariates are continuous, we use the kernel smoother with bandwidth equal to 2 for generating the synthetic data. For the binary covariates, we flip the value (e.g. 1 to 0 or 0 to 1) of each covariate with the probability $p = 0.1$. There are several binary covariates representing the one hot encoding of a specific feature, instead of flipping each covariate separately, we choose a different label at random with probability 0.1. We repeat our experiments 100 times with the fixed pretrained teacher but different random seeds to generate each synthetic dataset.

We use the same notation for hyperparameters from Algorithm 1 and fix them across the experiments in Section 4.2 and 4.3. And we only vary one or

two of them in the sensitivity analysis of Section 4.4 and 4.5. Specifically, we set

- (1) Significant level $\alpha = 0.05$.
- (2) The number of repetitions is 100.
- (3) Initial sample size $n_{\text{init}} = 1000$.
- (4) Maximum sample size $n_{\text{max}} = 100000$.
- (5) Complexity constraint $C = 3$.
- (6) Number of candidate students $N = 100$ for DT, 10000 population units for SR, and $P = 10$ trajectories with length 1000 produced by posterior sampling for FRL.

All the experiments are conducted in Python 3 (Van Rossum and Drake, 2009). We use `scikit-learn` package for the implementation of random forests and DT (Pedregosa et al., 2011). We code from the supplementary open resource materials in Wang and Rudin (2015) for FRL and `gplearn` for SR (Stephens, 2018).

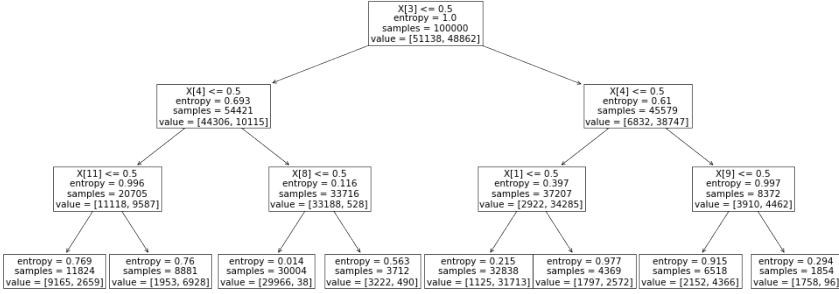


Fig. 3: Tree structure encoded by “[3, 4, 11, L, L, 8, L, L, 4, 1, L, L, 9, L, L]”.

4.2 Mammographic Mass Data

Mammography is a widely used technique for breast cancer screening and the Mammographic Mass data were first employed in (Elter et al., 2007) to predict the severity (malignant or benign) of a mammographic mass lesion. This data set contains 961 samples and 6 features including the BI-RADS assessment, patient’s age, shape, margin and density mass of the tumor, and severity of the breast cancer, collected at the Institute of Radiology of the University Erlangen-Nuremberg between 2003 and 2006. The data is further processed with one-hot encoding for the nominal features. The density mass is converted into a binary variable using the cutoff of 2. The patient’s age is partitioned into four intervals with cutoffs 30, 45 and 60 and then also converted into one-hot encoding.

We assess the performance of our proposed method on this data. Tables 2-4 display the proportion of each model structure among the 100 repetitions

Without Stabilization		With Stabilization	
Structure	Proportion	Structure	Proportion
[3, 4, 11, L, L, 8, L, L, 4, 1, L, L, 9, L, L]	44%	[3, 4, 11, L, L, 8, L, L, 4, 1, L, L, 9, L, L]	100%
[3, 4, 11, L, L, 8, L, L, 4, 1, L, L, 10, L, L]	20%		
[4, 3, 11, L, L, 1, L, L, 3, 8, L, L, 9, L, L]	18 %		
[4, 3, 11, L, L, 1, L, L, 3, 8, L, L, 10, L, L]	9 %		
...	...		

Table 2: Proportion of each model structure out of 100 repetitions for DT on Mammographic Mass Data with or without stabilization for distillation. We set $\alpha = 0.05$, $n_{\text{init}} = 1000$, the number of candidate models $N = 100$, the maximum depth of tree $C = 3$, and $n_{\text{max}} = 100000$. We encode tree structure with preorder traversals (root, left, right) of the nodes, where “L” represents the node is terminated and the integers denote the following features: “1” is “OvalShape”, “3” is “IrregularShape”, “4” is “CircumscribedMargin”, “8” is “SpiculatedMargin”, “9” is “ $30 < \text{Age} < 45$ ”, “10” is “ $45 \leq \text{Age} < 60$ ” and “11” is “ $\text{Age} \geq 60$ ”.

Without Stabilization		With Stabilization	
Structure	Proportion	Structure	Proportion
[IllDefinedMargin, Age ≥ 60], [IrregularShape], [SpiculatedMargin]	21%	[IrregularShape], [IllDefinedMargin, Age ≥ 60], [SpiculatedMargin]	94%
[SpiculatedMargin, Age ≥ 60], [IllDefinedMargin, Age ≥ 60], [IrregularShape]	20%	[SpiculatedMargin, Age ≥ 60], [IllDefinedMargin, Age ≥ 60], [IrregularShape]	5%
[IrregularShape], [IllDefinedMargin, Age ≥ 60], [SpiculatedMargin]	19%	[IllDefinedMargin, Age ≥ 60], [IrregularShape], [SpiculatedMargin]	1%
...	...		

Table 3: Proportion of each model structure out of 100 repetitions for FRL on Mammographic Mass Data with or without stabilization for distillation. We set $\alpha = 0.05$, $n_{\text{init}} = 1000$, the number of trajectories $P = 10$, the maximum length of rule list $C = 3$, and $n_{\text{max}} = 100000$.

for DT, FRL, and SR. We consider both cases where the stabilization (Algorithm 1) is applied or not. We propose to use preorder traversals to represent the structure of the tree in DT. It traverses the nodes of the tree from the root, to left and then to right. We use positive integers to represent different features and “L” to indicate a terminal leaf node. For example, we can encode the tree structure in Figure 3 with “[3, 4, 11, L, L, 8, L, L, 4, 1, L, L, 9, L, L]”. The structure of FRL is an ordered list of if-then rules, with each rule of the list constructed by using one or more features. In Table 3, “[IllDefinedMargin, Age ≥ 60], [IrregularShape], [SpiculatedMargin]” represents the decision process of first checking whether the tumor has irregular margin and

Without Stabilization		With Stabilization	
Structure	Proportion	Structure	Proportion
X3 + X4 + X11 + 1	66%	X3 + X4 + X11 + 1	83%
X1 + X3 + X4	16%	X1 + X3 + X4	10%
X1 + X3 + X4 + X11 + 1	7%	X1 + X3 + X4 + X8 + X11	4%
X3 + X4 + X8 + 1	5%	X1 + X3 + X4 + X8	1%
...

Table 4: Proportion of each model structure out of 100 repetitions for SR on Mammographic Mass Data with or without stabilization for distillation. We set $\alpha = 0.05$, $n_{\text{init}} = 1000$, the number of populations at last generations $N = 10000$, the maximum depth of expression tree $C = 3$, and $n_{\text{max}} = 100000$. Here, “1” is “OvalShape”, “3” is “IrregularShape”, “4” is “CircumscribedMargin”, “8” is “SpiculatedMargin”, and “11” is “Age ≥ 60 ”.

Without Stabilization		With Stabilization	
Structure	Proportion	Structure	Proportion
[20, 22, 27, L, L, 23, L, L, 22, 27, L, L, 23, L, L]	11%	[20, 22, 27, L, L, 23, L, L, 22, 27, L, L, 23, L, L]	94%
[22, 27, 20, L, L, 20, L, L, 20, 23, L, L, 23, L, L]	5%	[22, 27, 20, L, L, 20, L, L, 20, 23, L, L, 23, L, L]	5%
[20, 27, 22, L, L, 7, L, L, 22, 27, L, L, 23, L, L]	5%	[22, 20, 27, L, L, 27, L, L, 20, 23, L, L, 23, L, L]	1%
[20, 22, 27, L, L, 23, L, L, 22, 23, L, L, 23, L, L]	3%		
...	...		

Table 5: Proportion of each model structure out of 100 repetitions for DT on Breast Cancer Data with or without stabilization for distillation. We set $\alpha = 0.05$, $n_{\text{init}} = 1000$, the number of candidate models $N = 100$, the maximum depth of tree $C = 3$, and $n_{\text{max}} = 100000$. Here “L” represents the node is terminated and the integers denote the following features: “20” is “Worst Radius”, “22” is “Worst Perimeter”, “23” is “Worst Area”, “27” is “Worst Concave Points”.

the age is no smaller than 60, then evaluating whether it has irregular shape, and finally verifying whether it has spiculated margin. For SR, we consider both “plus” and “multiplication” operators and ignore the difference between the coefficients of mathematical formulas.

We observe that our proposed method achieves much better stability and picks up the more consistent structures in all three types of candidate models. Without stabilization, the distilled model structures are quite diverse which lead to different interpretation. We also see that all three types of candidate models select very similar features (e.g. “IrregularShape”, “SpiculatedMargin”, and “Age ≥ 60 ”) under the stabilization.

Without Stabilization		With Stabilization	
Structure	Proportion	Structure	Proportion
[Worst Area 1, Worst Radius 1], [Worst Concave Points 1], [Worst Concavity 1]	33%	[Worst Area 1, Worst Radius 1], [Worst Concave Points 1], [Worst Concavity 1]	100%
[Worst Area 1], [Worst Concave Points 1], [Worst Concavity 1]	21%		
[Worst Area 1, Worst Perimeter 1], [Worst Concave Points 1], [Worst Concavity 1]	9%		
...	...		

Table 6: Proportion of each model structure out of 100 repetitions for FRL on Breast Cancer Data with or without stabilization for distillation. We set $\alpha = 0.05$, $n_{\text{init}} = 1000$, the number of trajectories $P = 10$, the maximum length of rule list $C = 3$, and $n_{\text{max}} = 100000$.

Without Stabilization		With Stabilization	
Structure	Proportion	Structure	Proportion
X20 + X22 + X23 + X27	48%	X20 + X22 + X23 + X27	95%
X20 + X21 + X22 + X27	16%	X20 + X21 + X22 + X27	2%
X7 + X20 + X22 + X27	6%	X7 + X20 + X22 + X27	1%
X7 + X20 + X22 + X23	4%	X20 + X22 + X27	1%
...	...	X1 + X20 + X22 + X27	1%

Table 7: Proportion of each model structure out of 100 repetitions for SR on Breast Cancer Data with or without stabilization for distillation. We set $\alpha = 0.05$, $n_{\text{init}} = 1000$, the number of populations at last generations $N = 10000$, the maximum depth of expression tree $C = 3$, and $n_{\text{max}} = 100000$. Here, “7” is “Mean Concave Points”, “20” is “Worst Radius”, “21” is “Worst Texture”, “22” is “Worst Perimeter”, “23” is “Worst Area”, and “27” is “Worst Concave Points”.

4.3 Breast Cancer Data

The Breast Cancer Data is a widely adopted binary classification dataset in machine learning community (Mangasarian et al., 1995), which describes characteristics of the cell nuclei present in digitized images of fine needle aspirate (FNA) of breast mass. The dataset contains 569 samples and 30 continuous features, including the mean, standard deviation and the worst value of radius, texture, area, smoothness, etc. for each cell nucleus. The label for whether an instance is benign or malignant is also provided for classification task. This dataset can be directly accessed through `scikit-learn` package in Python.

We implement the distillation for a black box random forest model on this data with DT, FRL and SR. Since FRL can only handle binary features, we use `KBinsDiscretizer` in `scikit-learn` to partition continuous feature into intervals. Specifically, it uses the quantile values to generate equally populated bins in each feature and employs one-hot encoding for each bin and we drop one of the encoded columns in common with Wang and Rudin (2015). To reduce

computation complexity, we only use the last ten features of the original data for discretization and one-hot encoding in the experiments of FRL.

We present the results in Table 5-7. As in our results above, we observe that the stabilization selects much more consistent structures and “Worst Area”, “Worst Concave Points” and “Worst Radius” are regarded as quite significant features among all these three tables. For FRL, since each feature is partitioned into different bins, we label each bin with a number after the feature names (e.g. “Worst Area 1”, “Worst Area 2” and so on).

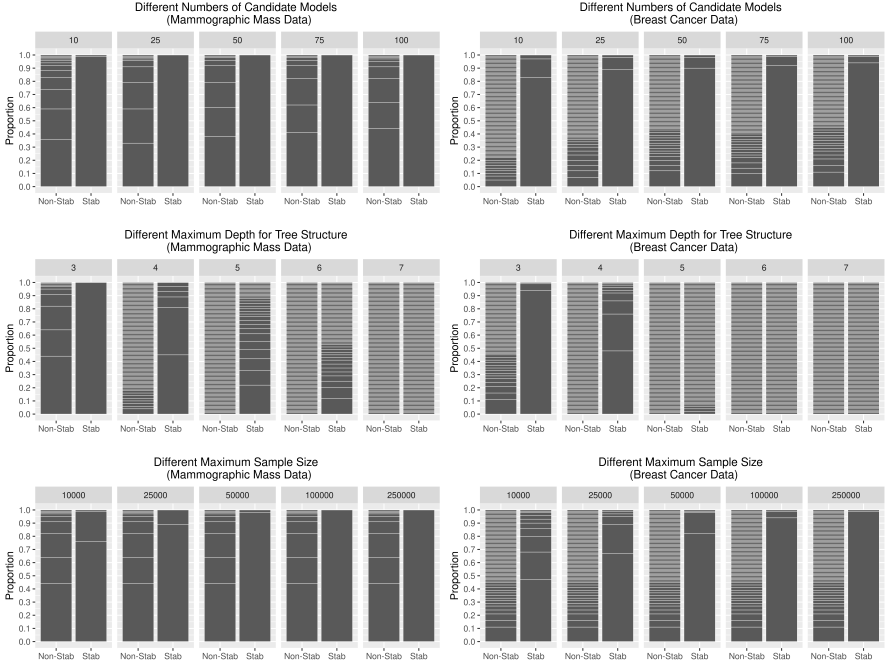


Fig. 4: Sensitivity analysis for DT on both Mammographic Mass Data and Breast Cancer Data. We fix the default settings in Section 4.2 and 4.3 but vary one of the hyperparameters among the different numbers of candidate models (N), maximum depth for tree structure (C) and maximum sample size (n_{\max}). In each column, a single black bar represents a unique structure of the tree, while the height of the bar represents the proportion of that structure out of 100 repetitions.

4.4 Sensitivity Analysis

In Section 2.2, we discuss three important factors in our testing framework: (1) number of candidate student models (2) complexity constraint of models and (3) maximum sample size. We also conduct the sensitivity analysis to explore

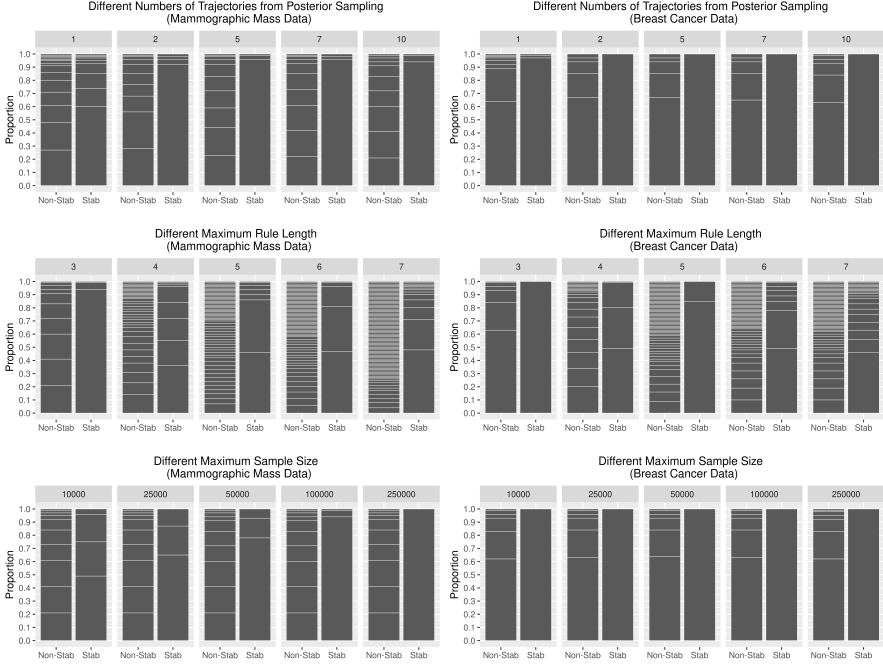


Fig. 5: Sensitivity analysis for FRL on both Mammographic Mass Data and Breast Cancer Data. We fix the default settings in Section 4.2 and 4.3 but vary one of the hyperparameters among the different numbers of trajectories (P) from posterior sampling, maximum length for rule list (C) and maximum sample size (n_{\max}). In each column, a single black bar represents a unique structure of the rule list, while the height of the bar represents the proportion of that structure out of 100 repetitions.

how these three factors affect the performance of our proposed method. In particular, for FRL and SR, a larger number of trajectories from posterior sampling and populations at the last generation is equivalent to an increasing number of candidate student models. We fix the default settings in Section 4.2 and 4.3 but vary one of these three factors.

The resulting plots are shown in Figure 4-6. In each column of the barplot, a single black bar represents a unique structure of the model, while the height of the bar represents the proportion of that structure out of 100 repetitions. All these three types of candidate models share very similar overall patterns. First of all, we need enough initial candidate student models to ensure adequate coverage. Otherwise, the stabilization fails to work well. We also observe that DT and SR have very different requirements for the number of candidate models. For DT, only 10 candidate models already achieve very promising performance, but at least 10000 is required for SR. This can be explained by

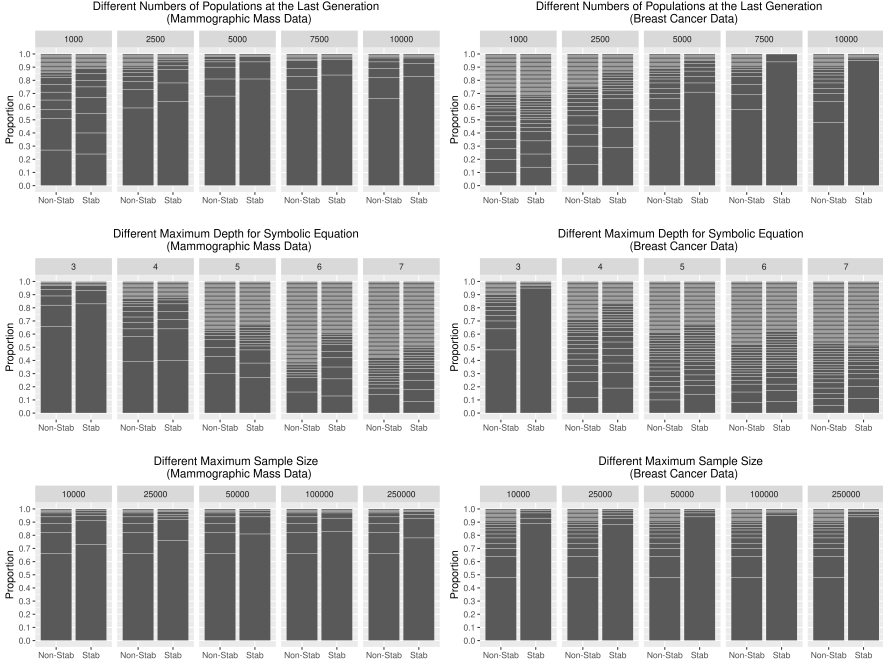
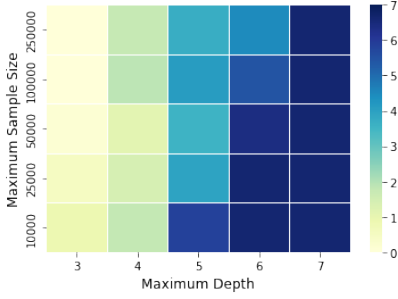


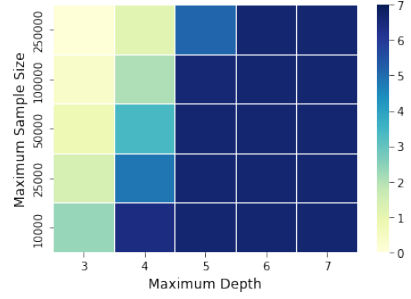
Fig. 6: Sensitivity analysis for SR on both Mammographic Mass Data and Breast Cancer Data. We fix the default settings in Section 4.2 and 4.3 but vary one of the hyperparameters among the different numbers of populations at the last generation (N), maximum depth for expression tree (C) and maximum sample size (n_{\max}). In each column, a single black bar represents a unique structure of the expression tree, while the height of the bar represents the proportion of that structure out of 100 repetitions.

different generating process of candidate models. SR relies on genetic programming to search for candidate structures which can easily fall into local minima or fail to achieve the convergence when given only a small number of populations of symbolic equations in genetic programming. In comparison, DT uses multiple synthetic datasets through Monte Carlo simulations, which selects structures with better coverage and the structures missed at the initial stages may be picked up in later stages with larger sample size.

In addition, when the model structure becomes more complex, it is extremely challenging to produce stable distilled structures without further increasing sample size and additional computational cost. For instance, in Figure 4 for Breast Cancer Data, when the maximum depth of tree is set at 7, the stabilization seems to fail to add any stability for the distilled structures. This is because much more complex model structures result in a collection of candidate models which all have extremely similar performance in the prediction task, and our testing framework suggests a huge number of samples that



(a) Mammographic Mass Data



(b) Breast Cancer Data

Fig. 7: Heatmap for the entropy of the proportions of different structures out of 100 repetitions. We use DT for demonstration and vary both maximum depth for tree structure (C) and maximum sample size (n_{\max}).

is computationally infeasible. Varying the maximum sample size also demonstrates that a choice of relatively small maximum sample would result in failure in our testing framework as a much larger sample size is required to achieve stability.

As a final exercise, we also explore varying both model complexity and maximum sample size. Figure 7 presents the heatmaps for the entropy of the proportions of different structures in 100 repetitions of DT. The colors in each heatmap plot denote the value of entropy. A darker rectangle corresponds to more uncertainty of the produced structure under the maximum sample size and maximum depth. *As an overall trend, increasing the maximum sample size results in more stable distillation but a huge sample size would be required for more complex model structures: given a set of candidate models, distinguishing between different structures would be challenging without further truncating the model complexity or requiring an infeasible computational cost.*

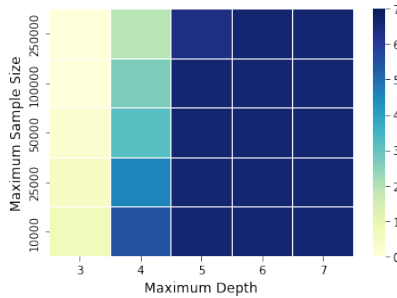


Fig. 8: Sensitivity analysis for DT on Breast Cancer Data when the independent sampling strategy is used. The rest of the setup is the same as Figure 7b.

4.5 Sampling Methods

In Section 2.2, we discussed several sampling methods for generating the synthetic data in model distillation but we focused on the kernel smoother throughout the paper. Alternatively, we also explore the performance of our proposed method when a naive independent sampling is used to generate the synthetic data. We examine the effect of this choice with a specific example of DT on Mammographic Mass Data. To generate the synthetic data, we ignore the dependence between covariates and use an Gaussian distribution to model each covariate separately based on the original data. Based on this sampling strategy, we conduct a sensitivity analysis with the same settings of 4.4. The resulting plot is shown in Figure 8. Comparing it with Figure 7b, We can observe that distillation with the independent sampling method achieves better performance when the maximum depth of DT is 3 but similar or slightly worse performance for higher complexity students. Mathematically, independent sampling leads to more extensive exploration of the feature space and it increases the power of distinguishing between different candidate student models. However, in this empirical experiment, such effect is not very pronounced for DT with more complex structures.

5 Conclusions

In this paper, we proposed a generic approach for stable model distillation by using a hypothesis testing framework. We constructed test statistics based on the central limit theorem to examine whether the candidate models are distinguishable. We showed that our proposed algorithm can ensure sufficient pseudo-data is generated to choose the same structure among a set of candidate student models with high probability. Theoretically, we treat our testing procedure as a Markov process and derived the bounds to measure how difficult the stabilization problem would become when the model complexity is quite large. Empirically, we conducted extensive experiments to demonstrate the efficacy of our proposed method and explore how several important factors of our testing framework affect stabilization performance. Finally, we ended up with the conclusion that if very complex models structures are considered, it will be extremely difficult to distinguish between candidate students without further requiring large pseudo samples, which can render distillation computationally infeasible.

A more rigorous theoretical analysis can be derived by incorporating the error of the central limit theorem through Berry–Esseen theorem. Donsker class assumption might also be imposed (Donsker, 1951) to consider the uncertainty of the candidate student models. There is also another source of variability coming from the uncertainty of teacher model, which we did not consider in this paper. To control this variability, we need to quantify the uncertainty of the teacher based on its internal structure. This will relate to the work on uncertainty quantification so the stabilization method must be developed for each learning algorithm individually (e.g. Ghosal et al. (2022)). Another

brute force way is through bootstrapping but doing so may be computationally infeasible and also lead to inconsistent estimates of the teacher’s uncertainty.

References

- Affenzeller, M., S.M. Winkler, G. Kronberger, M. Kommenda, B. Burlacu, and S. Wagner. 2014. Gaining deeper insights in symbolic regression, Genetic Programming Theory and Practice XI, 175–190. Springer.
- Aldeia, G.S.I. and F.O. de França. 2022. Interpretability in symbolic regression: a benchmark of explanatory methods using the feynman data set. Genetic Programming and Evolvable Machines: 1–41 .
- Augusto, D.A. and H.J. Barbosa 2000. Symbolic regression via genetic programming. In Proceedings. Vol. 1. Sixth Brazilian Symposium on Neural Networks, pp. 173–178. IEEE.
- Bishop, C.M. 1994. Mixture density networks .
- Breiman, L. 2001. Random forests. Machine learning 45(1): 5–32 .
- Breiman, L. 2002. Manual on setting up, using, and understanding random forests v3. 1. Statistics Department University of California Berkeley, CA, USA 1(58): 3–42 .
- Breiman, L., J.H. Friedman, R.A. Olshen, and C.J. Stone. 2017. Classification and regression trees. Routledge.
- Creswell, A., T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A.A. Bharath. 2018. Generative adversarial networks: An overview. IEEE signal processing magazine 35(1): 53–65 .
- Donsker, M.D. 1951. An invariance principle for certain probability limit theorems.
- Doob, J.L. 1949. Application of the theory of martingales. Le calcul des probabilites et ses applications: 23–27 .
- Doshi-Velez, F. and B. Kim. 2017. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 .
- Du, M., N. Liu, and X. Hu. 2019. Techniques for interpretable machine learning. Communications of the ACM 63(1): 68–77 .
- Duembgen, L. 2010. Bounding standard gaussian tail probabilities. arXiv preprint arXiv:1012.2063 .

- Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani. 2004. Least angle regression. The Annals of statistics 32(2): 407–499 .
- Elter, M., R. Schulz-Wendtland, and T. Wittenberg. 2007. The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process. Medical physics 34 11: 4164–72 .
- Ferreira, L.A., F.G. Guimarães, and R. Silva 2020. Applying genetic programming to improve interpretability in machine learning models. In 2020 IEEE congress on evolutionary computation (CEC), pp. 1–8. IEEE.
- Ghosal, I., Y. Zhou, and G. Hooker. 2022. The infinitesimal jackknife and combinations of models. arXiv preprint arXiv:2209.00147 .
- Johansson, U., C. Sönströd, and T. Löfström 2011. One tree to explain them all. In 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 1444–1451. IEEE.
- Kingma, D.P. and M. Welling. 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 .
- Koza, J. 1992. On the programming of computers by means of natural selection. Genetic programming .
- Kuhn, M., K. Johnson, et al. 2013. Applied predictive modeling, Volume 26. Springer.
- Loh, W.Y. 2011. Classification and regression trees. Wiley interdisciplinary reviews: data mining and knowledge discovery 1(1): 14–23 .
- Lundberg, S.M. and S.I. Lee. 2017. A unified approach to interpreting model predictions. Advances in neural information processing systems 30 .
- Mangasarian, O.L., W.N. Street, and W.H. Wolberg. 1995. Breast cancer diagnosis and prognosis via linear programming. Operations Research 43(4): 570–577 .
- Meurer, A., C.P. Smith, M. Paprocki, O. Čertík, S.B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J.K. Moore, S. Singh, et al. 2017. Sympy: symbolic computing in python. PeerJ Computer Science 3: e103 .
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Pasos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12: 2825–2830 .

- Quinlan, J.R. 1987. Generating production rules from decision trees. In ijcai, Volume 87, pp. 304–307. Citeseer.
- Quinlan, J.R. 2014. C4. 5: programs for machine learning. Elsevier.
- Ribeiro, M.T., S. Singh, and C. Guestrin 2016. ” why should i trust you?” explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135–1144.
- Stephens, T. 2018. gplearn: Genetic programming in python. See <https://github.com/trevorstephens/gplearn>.
- Tan, S., R. Caruana, G. Hooker, and Y. Lou 2018. Distill-and-compare: Auditing black-box models using transparent model distillation. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, pp. 303–310.
- van de Schoot, R., S. Depaoli, R. King, B. Kramer, K. Märtens, M.G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen, et al. 2021. Bayesian statistics and modelling. Nature Reviews Methods Primers 1(1): 1–26 .
- Van Rossum, G. and F.L. Drake. 2009. Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.
- Wand, M.P. and M.C. Jones. 1994. Kernel smoothing. CRC press.
- Wang, F. and C. Rudin 2015, 09–12 May. Falling Rule Lists. In G. Lebanon and S. V. N. Vishwanathan (Eds.), Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, Volume 38 of Proceedings of Machine Learning Research, San Diego, California, USA, pp. 1013–1022. PMLR.
- Wu, X., V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, et al. 2008. Top 10 algorithms in data mining. Knowledge and information systems 14(1): 1–37 .
- Zhou, Y., Z. Zhou, and G. Hooker. 2018. Approximation trees: Statistical stability in model distillation. arXiv preprint arXiv:1808.07573 .
- Zhou, Z., G. Hooker, and F. Wang 2021. S-lime: Stabilized-lime for model explanation. In Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, pp. 2429–2438.

Appendix A Proof

A.1 Auxiliary Lemma

Lemma 1. *For very small $\alpha > 0$,*

$$\sqrt{2 \log \left(\frac{1}{10\alpha} \right) - \log \log \left(\frac{1}{10\alpha} \right)} \leq Z_\alpha \leq \sqrt{2 \log \left(\frac{1}{2\alpha} \right)} .$$

Proof

On the one hand, for $x \geq 1$, we have

$$\begin{aligned} 1 - \Phi(x) &= \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt \\ &\leq \frac{1}{\sqrt{2\pi}} \int_x^\infty \frac{t}{x} e^{-t^2/2} dt \\ &\leq \frac{e^{-x^2/2}}{x\sqrt{2\pi}} \\ &\leq \frac{1}{2} e^{-x^2/2}. \end{aligned}$$

Let $x = Z_\alpha$. It implies

$$Z_\alpha \leq \sqrt{2 \log \left(\frac{1}{2\alpha} \right)} .$$

On the other hand, from Duembgen (2010), we know that if $x \geq 1$,

$$\begin{aligned} 1 - \Phi(x) &\geq \frac{1}{3\sqrt{2\pi}x} e^{-x^2/2} \\ &\geq \frac{1}{10x} e^{-x^2/2}. \end{aligned}$$

Let $x = Z_\alpha$. We approximately get

$$Z_\alpha \geq \sqrt{2 \log \left(\frac{1}{10\alpha} \right) - \log \log \left(\frac{1}{10\alpha} \right)}.$$

□

A.2 Proof of Theorem 1

We define the CDF $\Phi(x) = P(Z < x)$ where Z is a standard normal distribution. And we use Z_α to represent the $1 - \alpha$ -quantile of a standard normal distribution. We ignore the randomness of the variance estimate $\hat{\sigma}$ and assume it is known throughout our analysis for simplicity. The whole

proof is divided into two steps, where a simple case of two candidate models is first considered and then it is generalized to multiple candidate models.

Step 1

We start with a simple case when there are only two candidate models j and k . From Assumption 1, we know that

$$d_{jk} = \frac{1}{n} \sum_{i=1}^n [l(S_j(X_i), F(X_i)) - l(S_k(X_i), F(X_i))] \sim \mathcal{N}\left(\mu_{jk}, \frac{\sigma_{jk}^2}{n}\right)$$

Assume $\mu_{jk} > 0$. Because of the randomness of data, we can observe d_{jk} to be either positive or negative, even though the latter case has relatively smaller probability to happen. As a result, our proposed test would reject the null if $d_{jk} > Z_\alpha \sqrt{\frac{2\sigma_{jk}^2}{n}}$ or $d_{jk} < -Z_\alpha \sqrt{\frac{2\sigma_{jk}^2}{n}}$. Hence,

$$\begin{aligned} \omega_2 &= P\left(-Z_\alpha \sqrt{\frac{2\sigma_{jk}^2}{n}} < d_{jk} < Z_\alpha \sqrt{\frac{2\sigma_{jk}^2}{n}}\right) \\ &= \Phi(\sqrt{2}Z_\alpha - \sqrt{n}S_{jk}) - \Phi(-\sqrt{2}Z_\alpha - \sqrt{n}S_{jk}), \end{aligned} \tag{A1}$$

and

$$\omega_1 = 1 - \omega_2 = 1 - \Phi(\sqrt{2}Z_\alpha - \sqrt{n}S_{jk}) + \Phi(-\sqrt{2}Z_\alpha - \sqrt{n}S_{jk}).$$

According to Equation (4), we have

$$n' = \frac{Z_\alpha^2}{Z_{p_n}^2} n,$$

where

$$Z_{p_n} = \frac{\sqrt{n}d_{jk}}{\sqrt{2\sigma_{jk}^2}}.$$

Conditioninal on the case when the test is rejected, we have

$$\sqrt{2}Z_{p_n} \sim \mathcal{N}_{[-\sqrt{2}Z_\alpha, \sqrt{2}Z_\alpha]}(\sqrt{n}S_{jk}, 1),$$

where $\mathcal{N}_{[a,b]}(\mu, \sigma^2)$ represents the truncated normal with mean μ and standard deviation σ between interval $[a, b]$. Thus, the transition probability $p_2(n'|n)$ corresponds to the following random variable:

$$n' = \frac{2Z_\alpha^2}{Z_1^2} n, \tag{A2}$$

where $Z_1 \sim \mathcal{N}_{[-\sqrt{2}Z_\alpha, \sqrt{2}Z_\alpha]}(\sqrt{n}S_{jk}, 1)$. Equation (A2) implies that the magnitude of n' is controlled by the value of Z_1 , which follows a truncated normal distribution. We can use this to derive an upper bound for n' by constraining $|Z_1|$ to be larger than a small number $\delta > 0$. That is if we denote F_{Z_1} as the distribution function of Z_1 ,

$$\begin{aligned} P\left(n' \geq \frac{2Z_\alpha^2}{\delta^2}n\right) &= \omega_2 F_{Z_1}([- \delta, \delta]) \\ &= \Phi(\delta - \sqrt{n}S_{jk}) - \Phi(-\delta - \sqrt{n}S_{jk}). \end{aligned} \quad (\text{A3})$$

Equation (A3) shows that with probability as least $1 - \Phi(\delta - \sqrt{n}S_{jk}) + \Phi(-\delta - \sqrt{n}S_{jk})$,

$$n' < \frac{2Z_\alpha^2}{\delta^2}n.$$

Step 2

Now we can extend the results of Step 1 to the case of multiple candidate models. We relax the bound for multiple testing by regarding it as the Bonferroni correction. Specifically, we assume the model k is picked up as the best model, where k can be regarded as a random variable. Conditional on k , with Equation (A1) and mean value theorem we have

$$\begin{aligned} \omega_2 &= P\left(\sum_{j \in \mathcal{J}} p_{n,j} > \alpha\right) \\ &\leq \sum_{j \in \mathcal{J}} P\left(p_{n,j} > \alpha/(N(C) - 1)\right) \\ &\leq \sum_{j \in \mathcal{J}} \left[\Phi(\sqrt{2}Z_{\alpha/(N(C)-1)} - \sqrt{n}S_{jk}) - \Phi(-\sqrt{2}Z_{\alpha/(N(C)-1)} - \sqrt{n}S_{jk}) \right] \\ &\leq \sum_{j \in \mathcal{J}} \frac{2Z_{\alpha/(N(C)-1)}}{\sqrt{\pi}} e^{-\frac{\left(\sqrt{n}|S_{jk}| - \sqrt{2}Z_{\alpha/(N(C)-1)}\right)^2}{2}} \\ &\leq (N(C) - 1) \max_{j \in \mathcal{J}} \frac{2Z_{\alpha/(N(C)-1)}}{\sqrt{\pi}} e^{-\frac{\left(\sqrt{n}|S_{jk}| - \sqrt{2}Z_{\alpha/(N(C)-1)}\right)^2}{2}}. \end{aligned}$$

When n is quite large such that $\sqrt{n}|S_{jk}| > \sqrt{2}Z_{\alpha/(N(C)-1)}$ for any j, k , we can get

$$\omega_2 \leq (N(C) - 1) \frac{2Z_{\alpha/(N(C)-1)}}{\sqrt{\pi}} e^{-\frac{\left(\sqrt{n}S^*(C) - \sqrt{2}Z_{\alpha/(N(C)-1)}\right)^2}{2}}. \quad (\text{A4})$$

This implies that the probability the test procedure stops should be at least

$$\omega_1 \geq 1 - (N(C) - 1) \frac{2Z_{\alpha/(N(C)-1)}}{\sqrt{\pi}} e^{-\frac{(\sqrt{n}S^*(C) - \sqrt{2}Z_{\alpha/(N(C)-1)})^2}{2}}. \quad (\text{A5})$$

To derive a upper bound for n' , we want to find a n' such that $p_{n',j} \leq \alpha/(N(C) - 1)$ for any $j \in \mathcal{J}$. That is

$$n' = \max_{j \in \mathcal{J}} \frac{(\sqrt{2}Z_{\alpha/(N(C)-1)})^2}{Z_{jk}^2} n,$$

where $Z_{jk} \sim \mathcal{N}_{[-\sqrt{2}Z_{\alpha/(N(C)-1)}, \sqrt{2}Z_{\alpha/(N(C)-1)}]}(\sqrt{n}S_{jk}, 1)$.

Hence, for a small $\delta > 0$ and large n , we have

$$n' < \left[\sqrt{2}Z_{\alpha/(N(C)-1)} \right]^2 \frac{n}{\delta^2}, \quad (\text{A6})$$

with probability at least

$$\begin{aligned} 1 - P\left(\min_{j \in \mathcal{J}} Z_{jk} \in [-\delta, \delta]\right) &\geq 1 - P\left(\min_{j \in \mathcal{J}} Z_{jk} \in [-\delta, \delta]\right) \\ &\geq 1 - \sum_{j \in \mathcal{J}} P\left(Z_{jk} \in [-\delta, \delta]\right) \\ &= 1 - \sum_{j \in \mathcal{J}} \left[\Phi(\delta - \sqrt{n}S_{jk}) - \Phi(-\delta - \sqrt{n}S_{jk}) \right] \\ &\geq 1 - \sqrt{\frac{2}{\pi}} \delta (N(C) - 1) e^{-\frac{(\delta - \sqrt{n}S^*(C))^2}{2}}, \end{aligned} \quad (\text{A7})$$

according to Equation (A3) and mean value theorem.

When $N(C)$ is also quite large, with Lemma 1, we know that $Z_{\alpha/(N(C)-1)} \leq \sqrt{2 \log \left(\frac{N(C)-1}{2\alpha} \right)}$. Substituting it into equations (A4,A5,A8,A7) and let $\delta = \frac{1}{N(C)\sqrt{\log n}}$. We get

$$\begin{aligned} \omega_1 &\geq 1 - N(C) \sqrt{\frac{8}{\pi} \log \left(\frac{N(C)}{2\alpha} \right)} e^{-\frac{1}{2} \left(\sqrt{n}S^*(C) - 2\sqrt{\log \frac{N(C)}{2\alpha}} \right)^2} \\ \omega_2 &< N(C) \sqrt{\frac{8}{\pi} \log \left(\frac{N(C)}{2\alpha} \right)} e^{-\frac{1}{2} \left(\sqrt{n}S^*(C) - 2\sqrt{\log \frac{N(C)}{2\alpha}} \right)^2}. \end{aligned}$$

Similarly, with probability at least $1 - \sqrt{\frac{2}{\pi \log n}} e^{-\frac{\left(\frac{1}{N(C) \log^{1/2} n} - \sqrt{n} S^*(C)\right)^2}{2}}$,

$$n' < (4n \log n) N^2(C) \log \left(\frac{N(C)}{2\alpha} \right). \quad (\text{A8})$$

□