Doctoral Thesis

Boosting and concentration of measure methods in Machine Learning

Candidate: Thiago Rodrigo Ramos
Advisor 1: Roberto Imbuzeiro Oliveira
Advisor 2: Paulo Orenstein

Instituto de Matemática Pura e Aplicada

Rio de Janeiro, October, 2022.

Abstract

In this thesis, we present three applications of boosting and concentration of measure methods in machine learning.

In the first one, we develop a stage-wise boosting algorithm, dubbed ExactBoost. Our method directly optimizes combinatorial and non-decomposable losses, instead of making use of surrogate functions as is often the case in standard boosting methods. We develop an extension of margin theory to the non-decomposable setting and calculate bounds for the generalization error of ExactBoost for many important metrics with different levels of non-decomposability.

Our second application focuses on the Record Linkage problem. We propose a method that uses a variant of AdaBoost to learn a large-margin similarity classifier via a sample of similar/dissimilar items. Then, we construct single-bit hash functions that correlate with the similarity between items. From these, we can build hash codes that significantly speed up searches for similar items in databases.

Finally, in our third application, we work with split conformal prediction, a popular tool to obtain predictive intervals for general statistical algorithms under exchangeable data assumptions. We show how concentration of measure can be used to obtain finite-sample marginal, empirical and conditional guarantees for large classes of non-exchangeable data.

Keywords: boosting, learning to hash, conformal prediction, concentration of measure, time series.

Resumo

Nesta tese, nós apresentamos três aplicações de métodos de "boosting" e concentração de medida em aprendizado de máquinas.

Em nossa primeira aplicação, nós desenvolvemos um algoritmo de "boosting", chamado ExactBoost. Nosso método otimiza diretamente perdas combinatoriais e não-decomponíveis, ao invés de fazer uso de perdas substitutas, como é frequente em algoritmos clássicos. Nós desenvolvemos uma extensão da teoria de margem para o cenário de não-decomponíveis e calculamos cotas para o erro de generalização do ExactBoost aplicado à métricas importantes com diferentes níveis de não-decomposibilidade.

Nossa segunda aplicação foca no problema de "record linkage". Nós propomos um método que usa uma variação do AdaBoost para aprender um classificador de similaridades com grande margem. Então construímos funções "hash" de bit único que correlacionam com a similaridade entre itens. A partir dessas funções, construímos códigos "hash" que aceleram significativamente a busca por itens similares em bancos de dados.

Finalmente, em nossa terceira aplicação, trabalhamos com predição conforme, uma ferramenta popular para obtenção de intervalos preditivos de algoritmos estatísticos quando a hipótese de intercambiabilidade se verifica. Nós mostramos como concentração de medida pode ser usada para obter garantias marginais, empíricas e condicionais para uma grande classe de dados não-intercambiáveis.

Palavras-chave: boosting, aprendizado de funções hash, predição conforme, concentração de medida, séries temporais.

Agradecimentos

Agradeço primeiramente à Célia, Elias e Beatriz por serem minha família, mesmo eu tendo perdido tantas festas ao longo dos anos e aos meus primos (biológico e adotivo) Murilo e Tristãozinho, por aguentarem meus assuntos repetidos nas noites de sábado.

Agradeço também aos amigos que fiz ao longo desses quatro anos no IMPA, em particular, ao Luciano, Júlia, Eduardo, aos Christian(s), Daniel, Adriana, Deborah, João (que também é o baixista da nossa banda), ao Guerra e à Crislaine (que aguentou meu batuque por vários meses).

Um agradecimento especial as meus amigos de Centro Pi: ao meu co-orientando Antônio, ao meu parceiro de interior Lucas R, ao Lucas N por ter me apresentado a melhor pizza do Rio e um desagradecimento ao Lucas S por ter me apresentado ao Jazz. Agradeço também ao Alzira, Rodrigo (nosso Hacker Russo), ao João, ao João, ao João, ao Daniel que me converteu ao Vim, ao Beauclair (vulgo SUDO), à Damiana, ao Jorge e um super agradecimento especial à Carolzinha, que é uma das pessoas mais legais do mundo.

Finalmente, agradeço ao meu pai e mãe acadêmicos (não necessariamente nessa ordem) Roberto e Paulo, que além de grandes profissionais, são pessoas incríveis.

Contents

Chapter 1

Introduction

Concentration of measure [DLLG01, BLM13, Ver18, MRT12] is a subject of intensive research due to its importance in numerous practical and theoretical applications in Statistics, Learning Theory, Discrete Mathematics, Statistical Mechanics, Random Matrix Theory, Information Theory, and High-Dimensional Geometry. Its core idea is to quantify random fluctuations of random variables of interest, usually by bounding the probability that such random variable differs from its expected value (or from its median). For example, using certain concentration of measure inequalities, one can give a finite-sample bound for the difference of the sample average of an i.i.d. sample of random variables and its expectation.

In Machine Learning, for example, we use the theory of concentration of measure to find non-asymptotic worst case scenario bounds for the set of potential outputs of a given model. In that setting, concentration is often combined with measures of complexity of function classes, such as the Rademacher Complexity [BFLS98] and the VC dimension [Vap98].

An important class of Machine Learning algorithms whose theoretical analyses rely on concentration of measure are boosting methods [BFLS98, SF13, MRT12]. The main idea for such methods is to combine different (possibly inaccurate) prediction rules to create a highly accurate resulting prediction rule. One of the most important boosting methods is the AdaBoost algorithm [FS97], whose resulting classification function is chosen in a stage-wise adaptive way. In the AdaBoost setting, concentration of measure and margin maximization theory [BFLS98] are used to prove bounds for the testing error of the output classifier of the algorithm.

This thesis consists of three independent chapters. All involve applications of concentration of measure to Machine Learning, and the first two also involve boosting.

In the first one, we develop a stage-wise boosting algorithm, dubbed ExactBoost. Our method directly

optimizes combinatorial and non-decomposable losses, instead of making use of surrogate functions as is often the case in standard boosting methods. We develop an extension of margin theory to the non-decomposable setting using concentration of measure tools and calculate bounds for the generalization error of ExactBoost. Through extensive examples, we show that such theoretical guarantees translate to competitive empirical performance. In particular, when used as an ensembler, ExactBoost is able to significantly outperform other surrogate-based and exact algorithms available.

Our second application focuses on hashing methods for the Record Linkage problem. We propose a method that uses a variant of AdaBoost to learn a large-margin similarity classifier via a sample of similar/dissimilar items. Then, we construct single-bit hash functions that correlate with the similarity between items. From these, we can build hash codes that significantly speed up searches for similar items in databases. Our theoretical guarantees rely on concentration of measure applied to margin maximization theory and some techniques well known in the field of locallity-sensitive hashing. Additionally, preliminary experiments show our method has competitive performance against other hashing methods for Record Linkage.

Finally, in our third application, we work with split conformal prediction, a popular tool to obtain predictive intervals for general statistical algorithms under exchangeable data assumptions. We show how concentration of measure can be used to obtain finite-sample marginal, empirical and conditional guarantees for large classes of non-exchangeable data. In particular, we show that the empirical coverage bounds for some β-mixing processes match the order of the bounds under exchangeability. The framework introduced also extends to non-stationary processes and to other Conformal Prediction methods, and experiments corroborate our split Conformal Prediction coverage guarantees under dependent data.

In what follows we give more details on each of the three main parts of the thesis.

1.1 Main results of this thesis

1.1.1 ExactBoost: directly boosting the margin in combinatorial and non-decomposable metrics [1]

Several challenging classification tasks involve combinatorial and non-decomposable loss functions [KNJ14, GZPA19]. A combinatorial metric is one that is computed in terms of indicator functions, while non-decomposable metrics are those that cannot be reduced to a sum of loss functions on each sample point. Since such losses are neither differentiable nor parallelizable, common approaches based on convex optimization or stochastic gradient descent are not readily applicable without resorting to

[1] Joint paper [CPR+22] with Daniel Csillag, Carolina Piazza, João Vitor Romano, Roberto I. Oliveira and Paulo Orenstein.

surrogate losses.

Many popular metrics are of this nature. The area under the ROC curve (AUC) is a prime example. Other examples include the Kolmogorov-Smirnov (KS), widely used in the credit industry, and precision at k (P@k), which is usually applied to ranking problems. Generally, the data comes as independent and identically distributed (iid) points $(X_i, y_i)_{i=1}^n$, with features $X_i \in \mathbb{R}^p$ and binary labels $y_i \in \{0, 1\}$, and the goal is to devise algorithms that learn score functions (or classifiers) $S : \mathbb{R}^p \to [-1, 1]$ that correctly distinguish between the two label classes. Let n_0 and n_1 denote the number of labels in each class. These loss functions can be written

$$\widehat{\mathrm{AUC}}(S, y) = 1 - \frac{1}{n_1} \sum_{y_i=1} \frac{1}{n_0} \sum_{y_j=0} \mathbf{1}_{[S(X_i) > S(X_j)]}, \tag{1.1}$$

$$\widehat{\mathrm{KS}}(S, y) = 1 - \max_{t \in \mathbb{R}} \sum_{i=1}^{n} \rho_i \mathbf{1}_{[S(X_i) \le t]}, \tag{1.2}$$

$$\widehat{\mathrm{P@k}}(S, y) = 1 - \frac{1}{n} \sum_{i=1}^{n} y_i \mathbf{1}_{[i \in \mathcal{M}_k]}, \tag{1.3}$$

where $\rho_i = 1/n_0$ if $y_i = 0$ and $\rho_i = -1/n_1$ if $y_i = 1$, and $\mathcal{M}_k$ denotes the set of indices $i = 1, \ldots, n$ achieving the highest k scores. These three examples display different levels of non-decomposability: AUC relies on pairwise interactions, KS has a global threshold chosen optimally, and P@k also has a global threshold but with no optimality structure. Many other popular loss functions belong to the combinatorial or non-decomposable classes, including F-score and partial AUC.

Boosting is a leading technique to deal with classification problems, though it usually requires the development of surrogate losses for combinatorial and non-decomposable metrics. Still, not using the exact metric of interest often incurs in performance degradation, and the development of surrogate losses with optimality guarantees typically require significant work.

We consider, instead, a novel approach that works more generally for losses such as (1.1), (1.2) and (1.3). The procedure, dubbed ExactBoost, is a stagewise optimization algorithm tailored to the exact loss function with a margin condition. While margin theory is readily applicable in the decomposable setting, a novel extension is developed here for non-decomposable losses, yielding provable finite-sample performance guarantees. Given labels $\mathbf{y} = (y_1, \ldots, y_n)$, initial scores $\mathbf{S}_0 = (S_0(X_1), \ldots, S_0(X_n))$, and empirical loss function $\widehat{L} : [-1, 1]^n \times \{0, 1\}^n \to \mathbb{R}$, ExactBoost solves, at iteration $t = 1, \ldots, T$,

$$(\alpha_t, \mathbf{h}_t) = \operatorname*{argmin}_{\alpha, \mathbf{h}} \widehat{L}_\theta(\mathbf{S}_{t-1} + \alpha \mathbf{h}, \mathbf{y}), \tag{1.4}$$

and sets $\mathbf{S}_t = \mathbf{S}_{t-1} + \alpha_t \mathbf{h}_t$, where $h : \mathbb{R}^p \to \mathbb{R}$ is a base learner (e.g., a stump), $\mathbf{h} = (h(X_1), \ldots, h(X_n))$, $\alpha \ge 0$ is its corresponding weight, and, crucially, $\widehat{L}_\theta$ is a margin-adjusted version of the empirical loss

$\widehat{L}$. The combinatorial nature of the losses allows each boosting iteration to be solved relatively quickly. By employing interval arithmetic, ExactBoost is of order $O(pn \log n)$.

While ExactBoost is a competitive standalone estimator, its performance is even better as an ensembler. Using surrogate-based algorithms' predictions as features for ExactBoost allows it to combine them specifically for the chosen loss function, extracting the remaining signal tailored to the loss, akin to transfer learning.

Related work Boosting algorithms for combinatorial and non-decomposable losses [KNJ14, GZPA19] typically employ surrogate metrics, as is the case with Gradient Boosting [Fri01] and AdaBoost [FS97]. Both use approximations of the loss that lead to fast algorithms that are generally sensitive to misclassification error [BJM06]. Still, some loss in performance may follow from not using the exact metric of interest [CM03, FK20]. Recently, there have been efforts to find better surrogates to popular combinatorial losses [FFHO02, Joa05, BCMR12, Aga13, KNJ14, LY18, Tas18, ECPC19, PP20, GSST20, JAN$^+$20, AMŠP20], trading off speed for a more accurate loss function. There has also been interest in developing heavily constrained approaches that use the exact loss function [LJZ14, FC19]. ExactBoost, instead, relies on a novel and general extension of the margin theory for non-decomposable losses [ZXTW13, SF13] to obtain empirical error bounds, such as in [BFLS98, BM02, KP02], not previously available in this setting.

1.1.2 Learning to hash via boosting [2]

Given databases $\mathcal{A}$ and $\mathcal{B}$, we are interested in methods that report pairs $(A, B) \subset \mathcal{A} \times \mathcal{B}$, such that A and B are similar under a certain given measure of similarity. For example, A could be the information of a person in a database $\mathcal{A}$ and B be the information of the same person in a different database $\mathcal{B}$ and we want to conclude that even though A and B are not identical, they refer to the same person.

[2] Submission in preparation. We thank Lucas Nissenbaum, Alex Akira Okuno and Rodrigo Schuller for help with the experiments.

Database $\mathcal{A}$		
Name	Surname	Birthday
Alex	Akira	10-30-1998
Thiago	Ramos	04-16-1944
Rordigo	Schuler	08-30-1993
Paulo	Orenstein	04-01-1963

Database $\mathcal{B}$		
Names	Surnames	Birthday
Rodrigo	Schuller	08-30-1993
Tiago	Ramos	04-16-1994
Roberto	Oliveira	08-06-1977
Alex	Alzira	10-30-1998
Paulo	Orenstein	04-01-19963

Table 1.1: Example of Record Linkage Problem. We would like to match entries in different databases representing the same entity. In this example, equivalent entities are denoted by the same color.

The naive solution for this problem is to comb through whole collections of points in $\mathcal{A}\times\mathcal{B}$ and compute their similarities one-by-one. This solution clearly becomes prohibitively expensive if $\mathcal{A}\times\mathcal{B}$ is large or if checking for similarity is a costly operation.

A way to reduce the number of possible comparisons when searching for similar items is via hashing. A hash function constructs a low-dimension binary representation of the points in $\mathcal{A}$ and $\mathcal{B}$, called hash code. Unlike standard dimensionality-reduction methods, the fact that this representation is binary is important to ensure fast retrieval time [AI06, Cha02, KD09]. Using this binary low-dimensional representation, we can then only compare pairs (A, B) where A and B receive the same hash code. An efficient hash code must map very few neighboring candidates to each hash codes, significantly reducing the associated number of comparisons and ensure that similar items are indeed between these candidates.

We are specifically concerned with the setting where relevant similarities are not given by "obvious" data features but must be learned from the data. Such methods that try to learn hash codes from data are part of a family of methods called Learning to Hash [AB21a, KD09, WZs+18] and usually they provide the best results. An important example where the relevant similarities are not given by "obvious" data features, is the Record Linkage (RL) scenario [Chr12, FS69, EIV07]. In this context, two points A and B are defined as similar if they refer to the same entity. This suggests a classification problem where the goal is to discriminate between similar vs. non-similar pairs of data points. See Table 1.1 for a hypothetical example.

In this work, we propose a method that uses a variant of AdaBoost to learn a large-margin similarity classifier via a sample of similar/dissimilar items [FS97, FS99, MRT18]. The resulting classifier will be a convex combination of "simple functions", such as decision stumps, which has theoretical and

computational advantages.

Given this output, one can find single-bit hash functions that correlate with the similarity between items. Using these, we build hash codes with good computational and theoretical guarantees In particular, we obtain bounds on the Recall and Reduction Ratio metrics, which are standard in the literature.

Related work The RL problem is very important in several data analysis problems, since combining information from multiple databases can provide a more rich and detailed database [Win04]. Often, the records to be matched correspond to entities that refer to people. In this case, there are applications in the health sector to improve health policies [Cla04, KBH02], in statistical agencies to link census data [Win06], in security agencies for fraud in crime detection [JH06]. The RL problem also applies to databases containing information that is not about people, such as records about businesses, movies, consumer products, bibliographic citations, Web search results or genome sequences, bioinformatics and more details can be found in [Chr12]. Usually, hashing methods are applied to the RL problem so that the process of searching for similar items is restricted to blocks of items with matching hash codes [SVSF14, SS18]. For this reason, the RL problem is also known as data or field matching, entity resolution and deduplication.

An important class of methods for creating hash codes is known as Learning to Hash [AB21a, WTF08, KD09, WZs$^+$18]. Such methods use a supervised approach to learn hash codes from the data.

A problem closely related to our work that also makes use of hashing techniques is the nearest neighbor (NN) problem [HPIM12, AI06, BV10]. In the NN scenario, the measure of similarity is given by a predefined distance in a metric space and the goal is to find the nearest point to a query point under this distance. It is important to note that this is not the case we will consider, since there is no clear distance of matches in the RL problem.

In the NN setting, besides Learning to Hash, there is an alternative class of successful methods is called Locality-sensitive hashing (LSH). Locality-sensitive hashing [AI06, HPIM12, KG09, JQK14, OCC13] is an unsupervised method where hash codes are built in such a way that points having small distance, have a higher probability of having the same hash code. Some ideas of the LSH theory are adapted in this work so we use it in contexts where there is no predefined distance between points.

Finally, boosting and margin maximization methods [FS97, FS99, MRT18, FS97] has been shown to be very effective in practice and is based on a rich theoretical analysis based on concentration of measure. To the best of our knowledge, there is no work on boosting for learning to hash for Record Linkage problems. The work of [KYK20] uses boosting techniques to create a LSH method, however its underlying idea is quite different from ours and, as the proposed method is tailored to NN problems, it relies on the existence of a distance in a metric space, which is not our case.

1.1.3 Conformal prediction for dependent data [3]

Conformal prediction (CP), introduced by [VGS05], is a set of techniques for quantifying uncertainty in the predictions of any model, under very general assumptions on the data-generating distribution. CP yields finite-sample coverage guarantees of many kinds, and has generated much recent interest [SV07, LGR$^+$18, RPC19, AB21b, CGD21].

A concrete and popular formulation of CP is split confomal prediction [PPVG02, LGR$^+$18]. Consider a regression setting where the data is a random sample $(X_i, Y_i)_{i=1}^n$ of covariate/response pairs $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$. Split CP proceeds as follows: (i) partition the data indices in three parts: a training set I_{train}, a calibration set I_{cal} and a test set I_{test}, each with sizes n_{train}, n_{cal} and n_{test}; (ii) train a nonconformity score $\widehat{s}_{\text{train}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, for example the residual $\widehat{s}_{\text{train}}(x, y) = |y - \widehat{\mu}|$ of an arbitrary model $\widehat{\mu}$ trained on $(X_i, Y_i)_{i \in I_{\text{train}}}$; (iii) compute the empirical $(1-\alpha)$-quantile $\widehat{q}_{1-\alpha}$ of $\{\widehat{s}_{\text{train}}(X_i, Y_i)\}_{i \in I_{\text{cal}}}$; and (iv) for each $i \in I_{\text{test}}$, define a confidence set

$$C_{1-\alpha}(X_i) \coloneqq \{y \in \mathcal{Y} : \widehat{s}_{\text{train}}(X_i, y) \leq \widehat{q}_{1-\alpha}\}.$$

If the data $(X_i, Y_i)_{i=1}^n$ is exchangeable, then the usual theory of conformal prediction guarantees that the sets $C_{1-\alpha}(X_i)$ have good marginal coverage over the test set; that is, for any $i \in I_{\text{test}}$,

$$\mathbb{P}[Y_i \in C_{1-\alpha}(X_i)] \geq 1 - \alpha - \eta, \tag{1.5}$$

where $\eta = (1-\alpha)/(n+1)$. Equivalently, one can take the lower bound to be $1-\alpha$ by employing $C_{1-\alpha+\eta}$ instead. Additionally, for independent and identically distributed (iid) data and $\eta \gg 1/\min\{n_{\text{cal}}, n_{\text{test}}\}$, [LGR$^+$18] prove empirical coverage over the test set; that is,

$$\mathbb{P}\left[\frac{1}{n_{\text{test}}} \sum_{i \in I_{\text{test}}} \mathbf{1}_{[Y_i \in C_{1-\alpha}(X_i)]} \geq 1 - \alpha - \eta\right] \geq 1 - \delta, \tag{1.6}$$

for $\delta = \exp(-c\,\varepsilon^2 \min\{n_{\text{cal}}, n_{\text{test}}\})$ and some positive $c > 0$. These guarantees can also be shown to hold conditionally under certain conditions [CWZ18, CWZ21, BCRT20].

Unfortunately, the results above are strongly reliant on the data exchangeability. Similarly, most guarantees from the classical theory of CP do not apply to several important data processes, such as time series, spatial models and shifting distributions. Several recent papers have tried to address these issues [CWZ18, XX21, JBA22, GC21, BCRT22], but they generally require the introduction of new CP algorithms specifically tailored to different types of non-exchangeability that are either very computationally intensive or only possess asymptotic guarantees.

The main message of this work is that on many occasions there is no need to introduce specific CP methods for non-exchangeable data. We prove that in such cases split CP possesses the marginal,

[3] Joint paper [OORR22] with João Vitor Romano, Roberto I. Oliveira and Paulo Orenstein.

empirical and conditional guarantees above, up to the addition of a slightly larger penalty term η in (1.5) and (1.6). These guarantees hold in finite samples and make no underlying assumptions on model consistency. While the penalty depends on the nature of the non-exchangeability, we show that in practice the effect is small even for moderately depedent data, and that increasing the calibration set size is a viable corrective. Importantly, split CP is computationally simple, avoiding intensive routines such as bootstrapping, ensembling or blocking. Finally, the method is exactly the same as the one used for the iid data and attests to its robustness, which is essential to ensure its validity in practical settings.

For example, Figure 1.1 shows how split CP's marginal coverage behaves for an AR(1) time series and three different underlying models. The data-generating mechanism is given by $W_t = \lambda W_{t-1} + \varepsilon_t$, $t \in \mathbb{N}$, $\lambda \in \mathbb{R}$ and $\varepsilon_t \sim N(0, 1)$ independently, and models are trained on 11 lags to predict the next element in the sequence. Details are given in Section 4.4. The x-axis is indexed by λ, which can be interpreted as a level of dependence in the data. Note that unless the dependence is very high, split CP still has adequate coverage: autoregressive coefficients up to $\lambda = 0.99$ achieve coverage higher than 89%. Significant losses of coverage only happen when $\lambda \geq 0.999$.

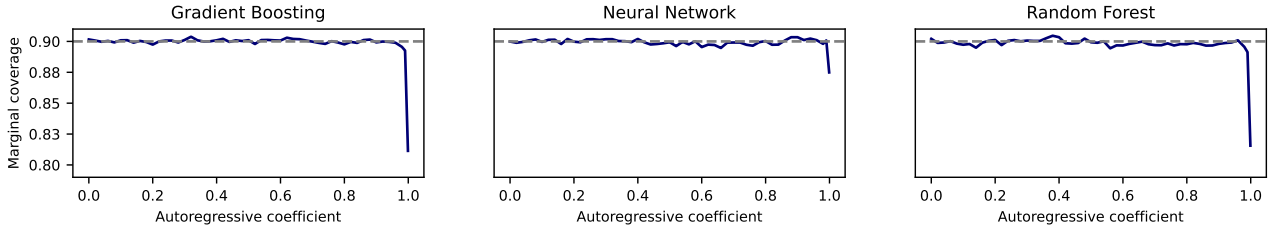


Figure 1.1: Marginal coverage for AR(1) process (solid) and nominally prescribed iid level (dashed) for different values of the autoregressive coefficent and three different models. Split CP holds well even under moderate dependence and undercoverage only happens at very high levels.

To extend split CP's guarantees to non-exchangeable data, we introduce a novel mathematical framework that is based on concentration inequalties and decoupling properties of the data, rather than exchangeability. We consider a concrete application of this framework for the important class of stationary β-mixing distributions [Bra05], and prove that when the β-mixing coefficients are summable the empirical coverage bounds match the order of the iid bounds, corroborating the claim that the non-exchangeability penalties incurred by split CP are small. Further, we show how this framework can be extended beyond to the risk-controlling prediction sets [BAL+21] setting and even to non-split CP methods, such as rank-one-out [LGR+18].

Related work The field of conformal prediction started with the seminal work of Vovk, Gammerman and Shafer [VGS05]; see [SV07] for a survey of early work in the topic. Lei et al. [LGR+18] helped popularize CP in the Statistics community. Since then, there has been an explosion of work on the

topic: see, e.g., [RPC19, CGD21, BAL$^+$21, ABJM21, BCRT20] for significant recent examples, and the survey [AB21b] for an introduction and additional references. We emphasize that the focus of this literature is on exchangeable data.

An important point about guarantees such as (1.5) is that they give marginal coverage. This means that coverage might be better than $1-\alpha$ for certain "easy" values of X_i and much worse for "hard" values. In fact, experiments in [CGD21] confirm this possibility.

The harder goal of pointwise coverage,

$$\mathbb{P}[Y_i \in C_{1-\alpha}(X_i) \mid X_i = x] \geq 1-\alpha-\varepsilon,$$

for small ε, was discussed by Chernozhukov et al. [CWZ21, CWZ18]. They prove that pointwise coverage can be achieved asymptotically when it is possible to learn the conditional distribution of Y_i given X_i.

On the other hand, Barber et al. [BCRT20] show that pointwise coverage is not possible in general, even for iid data. On the positive side, they show that if $\mathcal{A}$ is a family of subsets of $\mathcal{X}$ of finite VC dimension, and the data is iid, then one obtains conditional guarantees

$$\mathbb{P}[Y_i \in C_{1-\alpha}(X_i) \mid X_i \in A] \geq 1-\alpha-\varepsilon,$$

for all $A \in \mathcal{A}$ with $\mathbb{P}[X_i \in A]$ not too small. One of our results is that such conditional guarantees can be extended to dependent data.

CP methodologies for non-exchangeable data have been considered since [VGS05]. In recent years, a few different papers have appeared on this topic.

Gibbs and Candès [GC21] developed a general adaptive approach to conformal prediction that requires no distributional assumptions. Their method is very different in spirit from split CP: it requires the predictive sets to be updated online at each step. While [GC21] achieves empirical coverage under minimal assumptions, they give no guarantee of marginal coverage as in (1.5).

The issue of distributional shift has been considered in some detail. Tibshirani et al. [TBCR19] consider the setting where the distributions of the covariates in the training and calibration sets differ. If their likelihood ratio is known, or can be estimated well, their method achieves good guarantees. By contrast, Barber et al. [BCRT22] consider CP methods that give approximately correct marginal coverage under gradual changes in the data distribution. Their method does not seem applicable to time series, as it requires the distribution of the data to be approximately invariant under permutations between points "close" to the current test value.

We now consider methods that are specific to time series. The first work of this kind seems to be Chernozhukov et al [CWZ18], which employs a slightly convoluted block-based method reminiscent of the block bootstrap. Their results for non-exchangeable data, [CWZ18, Section 3.2], require that an "oracle score function" (a population object) be learned consistently from the data in the training phase. This is contrast to the guarantees of split CP, which are agnostic to the quality of the trained model. On the other hand, they require that the time series be strongly mixing, which is weaker than our β-mixing assumption.

Xu and Xie [XX21] consider another approach to time series, based on ensembling regressors that are trained over bootstrapped subsamples. Like [CWZ18], their theoretical guarantees require the strong assumption that the population regression function is consistently learned from the training data [XX21, Assumption 2]. Also like [CWZ18], they require a weaker mixing assumption than we do. We note in passing that ensembling (which may be desired on its own) can be easily incorporated into the training phase of split CP.

Finally, Jensen et al [JBA22] consider a time-series variant of conformalized quantile regression (see Romano et al. [RPC19]). Like [XX21], the approach of Jensen et al. also involves training an ensemble of methods over bootstrapped samples. No theoretical guarantees are given in [JBA22], but our framework can be used to obtain approximate versions of the bounds in [RPC19].

1.2 Basic tools

Throughout this section, let $\mathcal{S}_{\mathcal{Z},m} := \{Z_1, \ldots, Z_m\}$ be an i.i.d. sample from a probability distribution $\mathcal{D}_Z$ over a feature space $\mathcal{Z}$ (with suitable σ-field) and a family of measurable functions $\mathcal{G}$ from $\mathcal{Z}$ to $\mathbb{R}$. The family $\mathcal{G}$ usually corresponds to a set of candidate functions computed by a Machine Learning method.

1.2.1 Concentration of measure The main concentration of measure result we use in this thesis is McDiarmid's Inequality [McD98]:

Theorem 1.1 (McDiarmid's Inequality)**.** *Let an i.i.d. sample $\mathcal{S}_{\mathcal{Z},m} := \{Z_1, \ldots, Z_m\}$ and assume there exist $c_1, \ldots, c_m > 0$ such that $f : \mathcal{Z}^m \to \mathbb{R}$ satisfies the following condition:*

$$|f(z_1, \ldots, z_i, \cdots, z_m) - f(z_1, \ldots, z_i', \cdots, z_m)| \leq c_i.$$

Define $f(\mathcal{S}_{\mathcal{Z},m}) := f(Z_1, \ldots, Z_m)$, then for all $t > 0$,

$$\mathbb{P}\left[|f(\mathcal{S}_{\mathcal{Z},m}) - \mathbb{E}\left[f(\mathcal{S}_{\mathcal{Z},m})\right]| \geq t\right] \leq 2\exp\left(\frac{-2t^2}{\sum_{i=1}^m c_i^2}\right)$$

Next, we explain how McDiarmid's Inequality will genereally be employed in this thesis.

General usage of McDiarmid's Inequality. Given a family of measurable functions $\mathcal{G}$ from $\mathcal{Z}$ to $[0,1]$, consider

$$f(\mathcal{S}_{\mathcal{Z},m}) = \sup_{g\in\mathcal{G}} \left(\frac{1}{m}\sum_{i=1}^{m} g(Z_i) - \mathbb{E}\left[g(Z_1)\right] \right).$$

Note that we can apply McDiarmid's Inequality for the constants $c_i = 1/m$ and conclude that, for all $t > 0$

$$\mathbb{P}\left[|f(\mathcal{S}_{\mathcal{Z},m}) - \mathbb{E}\left[f(\mathcal{S}_{\mathcal{Z},m})\right]| \geq t \right] \leq 2\exp\left(-2mt^2\right)$$

This implies that, given $\delta > 0$, with probability at least $1-\delta$,

$$\sup_{g\in\mathcal{G}} \left(\frac{1}{m}\sum_{i=1}^{m} g(Z_i) - \mathbb{E}\left[g(Z_1)\right] \right) \leq \sqrt{\frac{\log(2/\delta)}{2m}} + \mathbb{E}\left[\sup_{g\in\mathcal{G}} \left(\frac{1}{m}\sum_{i=1}^{m} g(Z_i) - \mathbb{E}\left[g(Z_1)\right] \right) \right]. \tag{1.7}$$

Equation 1.7 give us a finite-sample estimation of how the empirical mean concentrates around its expectation uniformly over $\mathcal{G}$. The only problem in the previous equation is that we do not know the value of

$$\mathbb{E}\left[\sup_{g\in\mathcal{G}} \left(\frac{1}{m}\sum_{i=1}^{m} g(Z_i) - \mathbb{E}\left[g(Z_1)\right] \right) \right], \tag{1.8}$$

so, in the next subsection, we exhibit classical tools to bound such quantity.

Finally, we introduce another concentration inequality, named Bernstein's Inequality [Ber, MRT12], that will be useful in Chapter 4:

Theorem 1.2 (Bernstein's Inequality)**.** *Let an independent sample $\mathcal{S}_{\mathcal{Z},m} := \{Z_1, \ldots, Z_m\}$. Assume that $\mathbb{E}[Z_i] = 0$ and $Z_i \leq c$ a.s. for all $i = 1, \cdots, m$. Let $\sigma^2 = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Var}(Z_i)$. Then for all $t > 0$,*

$$\mathbb{P}\left[\frac{1}{m}\sum_{i=1}^{m} Z_i \geq t \right] \leq \exp\left(-\frac{mt^2}{2\sigma^2 + 2ct/3} \right).$$

Bernstein's inequality improves McDiarmid's when the random variables Z_i have variances that are much smaller than the bounding constant c.

1.2.2 Rademacher processes and VC dimension Given a family of measurable functions $\mathcal{G}$ from $\mathcal{Z}$ to $\mathbb{R}$, its Rademacher complexity [BM02] is defined as

$$\mathcal{R}_{\mathcal{S}_{\mathcal{Z},m}}(\mathcal{G}) \coloneqq \mathbb{E}_\sigma \sup_{g\in\mathcal{G}} \frac{\sum_{i=1}^m \sigma_i g(Z_i)}{m},$$

where the $\sigma_1, \ldots, \sigma_m$ are i.i.d. uniform over $\{-1, +1\}$ and independent of the Z_i, and $\mathbb{E}_\sigma[\cdot]$ denotes expectation with respect to the σ_i variables only. The averaged Rademacher complexity is defined as

$$\mathcal{R}_m(\mathcal{G}) \coloneqq \mathbb{E}_{Z_1,\ldots,Z_m\sim\mathcal{D}_Z}\mathbb{E}_\sigma \sup_{g\in\mathcal{G}} \frac{\sum_{i=1}^m \sigma_i g(Z_i)}{m}.$$

We assume implicitly throughout this section that that the families $\mathcal{G}$ we consider is nice enough that the supremum is measurable and integrable. A fundamental property of $\mathcal{R}_m(\mathcal{G})$ is the *symmetrization inequality*: if all functions $g \in \mathcal{G}$ are integrable,

$$\mathbb{E}_{Z_1,\ldots,Z_m\sim\mathcal{D}_Z} \sup_{g\in\mathcal{G}} \frac{\sum_{i=1}^m \mathbb{E}_{Z\sim\mathcal{D}_Z} g(Z) - g(Z_i)}{m} \leq 2\mathcal{R}_m(\mathcal{G}). \tag{1.9}$$

A fundamental result used throughout this thesis is what is referred to as the Rademacher Inequality [KP02, Theorem 1].

Theorem 1.3 (Rademacher Inequality)**.** *Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{Z}$ to $[0,1]$. Then for any $\delta > 0$, with probability at least $1-\delta$ over the draw of an i.i.d. sample $\mathcal{S}_{\mathcal{Z},m} \coloneqq \{Z_1, \ldots, Z_m\}$, each of the following* holds for all $g \in \mathcal{G}$*:*

$$\mathbb{E}\left[g(Z)\right] \leq \frac{1}{m}\sum_{i=1}^m g(Z_i) + 2\mathcal{R}_{\mathcal{S}_{\mathcal{Z},m}}(\mathcal{G}) + \sqrt{\frac{\log(1/\delta)}{2m}} \tag{1.10}$$

$$\mathbb{E}\left[g(Z)\right] \leq \frac{1}{m}\sum_{i=1}^m g(Z_i) + 2\mathcal{R}_m(\mathcal{G}) + 2\sqrt{\frac{\log(2/\delta)}{2m}}. \tag{1.11}$$

To prove such result, it suffices to use McDiarmid's and use (1.9) to bound the expectation (1.8).

If the family $\mathcal{G}$ takes values from $\mathcal{Z}$ to $\{-1, 1\}$ we can also define its VC complexity [Vap98] , given by

$$\mathrm{VC}(\mathcal{G}) \coloneqq \max\{m : \Pi_{\mathcal{G}}(m) = 2^m\},$$

where the function $\Pi_{\mathcal{G}}(m)$ is defined as,

$$\Pi_{\mathcal{G}}(m) = \max_{z_1,\cdots,z_m\in\mathcal{Z}} |\{(g(z_1), \cdots, g(z_m)) : g \in \mathcal{G}\}|$$

Supposing that $\mathrm{VC}(\mathcal{G}) = d$, then there is an important property relating the VC dimension of $\mathcal{G}$ and its Rademacher complexity given by,

$$\mathcal{R}_m(\mathcal{G}) \leq \sqrt{\frac{2\log(\Pi_{\mathcal{G}}(m))}{m}} \leq \sqrt{\frac{2\left(\frac{em}{d}\right)^d}{m}}.$$

If our base space is $\mathcal{Z} = \mathbb{R}^d$, then an important example of family of binary functions is the family of Stumps, given by:

$$\text{Stumps} = \left\{\pm\mathbf{1}_{[Z_{(j)} \leq \xi]} \pm \mathbf{1}_{[Z_{(j)} > \xi]} \ : \ \xi \in \mathbb{R}, j \in [p]\right\}, \tag{1.12}$$

with $Z_{(j)}$ denoting the jth coordinate of Z. In this case, if we take $\mathcal{G} = \text{Stumps}$, then we show in Proposition 2.9 that $\mathcal{R}_n(\mathcal{G}) = O(\sqrt{\log p/n})$.

1.2.3 AdaBoost A very important classification method used in this thesis is the AdaBoost algorithm [FS97, FS99, MRT18]. Here we assume a sample $\mathcal{S}'_{\mathcal{Z},m} := \{(Z_1, y_1), \ldots, (Z_m, y_m)\}$ with $Z_i \in \mathcal{Z}$ and $y_i \in \{-1, 1\}$ and we want to learn a classifier for this problem.

The idea behind AdaBoost is to create a convex combination of several "weak" classifiers from a given family $\mathcal{G}$ such that this resulting convex combination has a good generalization error. As we show next, these convex weights are chosen in a clever stage-wise adaptive way, hence the name AdaBoost.

Algorithm 1 AdaBoost algorithm

Require: $\mathcal{S}_{\mathcal{Z},m} = (Z_i, y_i)_{i=1}^m$, number of iterations $T \in \mathbb{N}$, binary family $\mathcal{G}$

for $i \leftarrow 1$ to m **do**
  $Q_1(i) \leftarrow \frac{1}{m}$
end for
for $t \leftarrow 1$ to T **do**
  $g_t^* \leftarrow$ classifier in $\mathcal{G}$ with error $\varepsilon_t = \sum_{i=1}^m Q_t(i)\mathbf{1}_{[y_i g_t^*(Z_i) < 0]} < 1/2$
  $\alpha_t^* \leftarrow \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) > 0$
  $Z_t \leftarrow 2\left[\varepsilon_t(1-\varepsilon_t)\right]^{1/2}$
  for $i \leftarrow 1$ to m **do**
    $Q_{t+1}(i) \leftarrow \frac{Q_t(i)\exp(-\alpha_t^* y_i g_t^*(z_i))}{Z_t}$
  end for
end for
$f^* \leftarrow \frac{1}{\sum_{t=1}^T \alpha_t^*}\sum_{t=1}^T \alpha_t^* g_t^*$
return $\operatorname{sign}(f^*)$

Note that $yf^*(Z) < 0$ only if y and $f^*(Z)$ have different sign, that is,

$$\text{Training Error} := \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{[\text{sign}(f^*(Z_i))\neq y_i]} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{[y_i f^*(Z_i)\leq 0]}.$$

But using the definitions of α_t^*, ε_t and Z_t in Algorithm 1, one can show that

$$\frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{[y_i f^*(Z_i)\leq 0]} \leq 2^T \prod_{t=1}^{T} [\varepsilon_t(1-\varepsilon_t)]^{1/2} \tag{1.13}$$

and under some minor assumptions over $\{\varepsilon_t\}_{t=1}^T$ it is possible to show that the right hand side of (1.13) decays exponentially fast [MRT12, Theorem 7.2]. That is, for a suitable constant $\gamma > 0$,

$$\text{Training Error} \leq \exp\left(-2\gamma^2 T\right), \tag{1.14}$$

justifying the use of the sign function of f^* as the final classifier of AdaBoost Algorithm.

In practice, AdaBoost does not tend to overfit, even after many iterations. A partial theoretical explanation for this phenomenon is given by margin arguments. In Chapter 3 we adapt this analysis to the setting of learning to hash via boosting.

Chapter 2

ExactBoost: directly boosting the margin in combinatorial and non-decomposable metrics

The results in this section were obtained in a joint paper [CPR+22] with Daniel Csillag, Carolina Piazza, João Vitor Romano, Roberto I. Oliveira and Paulo Orenstein. The author of this thesis is responsible for all the theory and mathematical proofs.

2.1 Introduction

Consider data $(X_1, y_1), \ldots, (X_n, y_n) \sim \mathcal{D}$ independently, with $X_i \in \mathbb{R}^p$ features and $y_i \in \{0, 1\}$ labels, and an empirical loss $\widehat{L} : [-1, 1]^n \times \{0, 1\}^n \to [0, 1]$ that is invariant under rescaling and translation in its first argument, such as (1.1), (1.2) and (1.3). The goal is to find a score function $S : \mathbb{R}^p \to [-1, 1]$ where higher scores $S(X_i)$ indicate higher likelihood of $y_i = 1$. It will be assumed that, after t rounds, a score has the form

$$S_t(X_i) = \sum_{r=1}^{t} w_r h_r(X_i), \tag{2.1}$$

with $w_r \geq 0$, $\sum_{r=1}^{t} w_r = 1$ and $h_r \in \mathcal{H}$, where $\mathcal{H}$ is a set of base learners. For stagewise minimization of the empirical loss, one solves $(\alpha_{t+1}, \mathbf{h}_{t+1}) = \operatorname{argmin}_{\alpha \geq 0, h \in \mathcal{H}} \widehat{L}(\mathbf{S}_t + \alpha \mathbf{h}, \mathbf{y})$ and updates the score via $S_{t+1} = (S_t + \alpha_{t+1} h_{t+1})/(1 + \alpha_{t+1})$, where the denominator ensures the weights sum to one, as in (2.1).

This approach produces competitive results on test data in many settings. However, to attenuate overfitting with combinatorial and non-decomposable losses, a margin-adjusted loss $\widehat{L}_\theta$ is justified.

Consider

$$\widehat{L}_\theta(\mathbf{S}, \mathbf{y}) = \widehat{L}(\mathbf{S} - \theta\mathbf{y}, \mathbf{y}), \tag{2.2}$$

where $\theta > 0$ is a margin parameter (though the P@k case is slightly more subtle, see Theorem 2.3). That way, scores for positive labels are artificially decreased, forcing the algorithm to increase the confidence when correctly classifying samples (since losses are translation-invariant, this is equivalent to imposing high confidence on negative cases). This simple adjustment is crucial to provide optimality bounds on the generalization performance of the resulting algorithm (see Section 2.2).

Now, consider the optimization program (1.4). While ExactBoost and its guarantees hold for general sets of base learners $\mathcal{H}$, in practice learners beyond stumps (e.g., trees of higher depths) do not yield significant improvements and can be much more costly computationally. Thus, take $\mathcal{H}$ to be the set of stumps:

$$\mathcal{H} = \left\{ \pm \mathbf{1}_{[X_{(j)} \leq \xi]} \pm \mathbf{1}_{[X_{(j)} > \xi]} \ : \ \xi \in \mathbb{R}, j \in [p] \right\}, \tag{2.3}$$

with $X_{(j)}$ denoting the jth feature of X.

Since the losses are invariant under rescaling and translation of the first argument, ExactBoost must pick

$$\begin{aligned}(\alpha_t, \mathbf{h}_t) &= \operatorname*{argmin}_{\alpha, \mathbf{h}} \widehat{L}_\theta \left(\frac{1}{1+\alpha}\mathbf{S}_{t-1} + \frac{\alpha}{1+\alpha}\mathbf{h}, \mathbf{y} \right) \\ &= \operatorname*{argmin}_{\alpha, \mathbf{h}} \widehat{L} \left(\mathbf{S}_{t-1} - \theta\mathbf{y} + \alpha(\mathbf{h} - \theta\mathbf{y}), \mathbf{y} \right).\end{aligned}$$

Let $\tilde{h}(X) = \tilde{a}\mathbf{1}_{[X_{(j)} \leq \xi]} + \tilde{b}\mathbf{1}_{[X_{(j)} > \xi]} - (|\tilde{b} - \tilde{a}|/2)\theta y$, a function parametrized by $\tilde{a}, \tilde{b}, \xi \in \mathbb{R}$ and $j \in [p]$. Note

$$\begin{aligned}\tilde{h}(X) - \frac{\tilde{a} + \tilde{b}}{2} &= \frac{|\tilde{b} - \tilde{a}|}{2} \left(\pm \mathbf{1}_{[X_{(j)} \leq \xi]} \pm \mathbf{1}_{[X_{(j)} > \xi]} - \theta y \right) \\ &= \alpha(a\mathbf{1}_{[X_{(j)} \leq \xi]} + b\mathbf{1}_{[X_{(j)} > \xi]} - \theta y) \\ &= \alpha(h(X) - \theta y),\end{aligned}$$

where $a, b \in \{-1, 1\}$ and $\alpha \geq 0$. Thus, the program (1.4) is iteratively solved by picking (ξ_t, j_t, a_t, b_t) via

$$\min_{\xi \in \mathbb{R}, j \in [p], \tilde{a}, \tilde{b} \in \mathbb{R}} \widehat{L} \left(\mathbf{S}_{t-1} + \tilde{a}\mathbf{1}_{[X_{(j)} \leq \xi]} + \tilde{b}\mathbf{1}_{[X_{(j)} > \xi]} - \left(1 + (|\tilde{b} - \tilde{a}|)/2 \right) \theta\mathbf{y}, \mathbf{y} \right), \tag{2.4}$$

then setting $\mathbf{S}_t = \mathbf{S}_{t-1} + a_t\mathbf{1}_{[X_{(j_t)} \leq \xi_t]} + b_t\mathbf{1}_{[X_{(j_t)} > \xi_t]}$. Note the discrete nature of combinatorial loss functions allows (2.4) to be solved by only considering a finite set of ξ, $\tilde{a}$ and $\tilde{b}$: for ξ, it suffices to look at the unique values of feature $X_{(j)}$ for $j = 1, \ldots, p$, and for a and b the unique values of $S(X_i)$,

for $i = 1, \ldots, n$. Other values of ξ, a and b do not yield different training losses.

The resulting algorithm is called ExactBoost, as it is based on the exact loss function provided rather than a surrogate loss. To avoid overfitting, subsampling is used (see Section 2.2.4 for theoretical guarantees). Finally, randomized runs of the algorithm are averaged, similar in spirit to random forests, and can be trivially parallelized. Algorithm 2 includes the full pseudocode. It takes as input an initial set of scores, which could for instance be scores trained by other learning models.

Algorithm 2 ExactBoost

function ExactBoost(data $(\mathbf{X}, \mathbf{y})$, initial scores S_0, margin θ, iterations T, estimator runs E)
 for $e \in \{1, \ldots, E\}$ **do**
  $S_e \leftarrow S_0$
  for $t \in \{1, \ldots, T\}$ **do**
   $\mathbf{X}^s, \mathbf{y}^s \leftarrow$ subsample $\mathbf{X}, \mathbf{y}$
   for $j \in \{1, \ldots, p\}$ **do**
    $\widehat{L}(h) \leftarrow \widehat{L}_\theta(S_e(\mathbf{X}^s_{(j)}) + h(\mathbf{X}^s_{(j)}), \mathbf{y}^s)$
    $h_j \leftarrow \operatorname{argmin}_h \widehat{L}(h)$ ▷ Algorithm 3
   end for
   $h \leftarrow \operatorname{argmin}_{h_j} \widehat{L}_\theta(S_e(\mathbf{X}^s) + h_j(\mathbf{X}^s), \mathbf{y}^s)$
   $S'_e \leftarrow S_e + h$
   if $\widehat{L}_\theta(S'_e(\mathbf{X}), \mathbf{y}) \leq \widehat{L}_\theta(S_e(\mathbf{X}), \mathbf{y})$ **then**
    $S_e \leftarrow S'_e$
    $S_e \leftarrow (S_e - \min S_e)/(\max S_e - \min S_e)$
   end if
  end for
 end for
 return $\operatorname{mean}(S_1, \ldots, S_E)$
end function

Algorithm 3 Iterative Minimization

function MINIMIZE(loss $\hat{L}_\theta$, data $\mathbf{X}_{(j)}$, labels $\mathbf{y}$, scores S, margin θ)
 $\Xi \leftarrow [\min \mathbf{X}_{(j)}, \max \mathbf{X}_{(j)}]$
 $A \leftarrow [-1, 1]$; $B \leftarrow [-1, 1]$
 for $k \in \{1, \dots, c\}$ **do**
  $l_\star \leftarrow +\infty$
  for bisections $(\Xi^{(b)}, A^{(b)}, B^{(b)})$ **do**
   for $i \in \{1, \dots, n\}$ **do**
    $s \leftarrow S + A^{(b)}\mathbf{1}_{[\mathbf{X}_{(j)} \leq \Xi^{(b)}]} + B^{(b)}\mathbf{1}_{[\mathbf{X}_{(j)} > \Xi^{(b)}]}$
    $\mathbf{s}_i \leftarrow \underline{s}$ if $y_i = 0$ otherwise $\overline{s}$
   end for
   if $\widehat{L}_\theta(\mathbf{s}, \mathbf{y}) < l_\star$ **then**
    $l_\star \leftarrow \widehat{L}_\theta(\mathbf{s}, \mathbf{y})$
    $\Xi_\star \leftarrow \Xi^{(b)}$; $A_\star \leftarrow A^{(b)}$; $B_\star \leftarrow B^{(b)}$
   end if
  end for
 end for
 $I_\Xi \leftarrow \{\underline{\Xi_\star}, \overline{\Xi_\star}\}$; $I_A \leftarrow \{\underline{A_\star}, \overline{A_\star}\}$; $I_B \leftarrow \{\underline{B_\star}, \overline{B_\star}\}$
 $S(a, b, \xi) \leftarrow S + a\mathbf{1}_{[\mathbf{X}_{(j)} \leq \xi]} + b\mathbf{1}_{[\mathbf{X}_{(j)} > \xi]}$
 $(\xi_\star, a_\star, b_\star) \leftarrow \underset{\xi \in I_\Xi, a \in I_A, b \in I_B}{\operatorname{argmin}} \widehat{L}_\theta(S(a, b, \xi), \mathbf{y})$
 return $S + a_\star\mathbf{1}_{[\mathbf{X}_{(j)} \leq \xi_\star]} + b_\star\mathbf{1}_{[\mathbf{X}_{(j)} > \xi_\star]}$
end function

In order to efficiently solve (2.4), we use an interval arithmetic (IA)-based algorithm: We use the usual IA notations and operations, see [HJvE01]; e.g., $Z = [\underline{Z}, \overline{Z}]$ is an interval, $F(Z) = [\underline{F(Z)}, \overline{F(Z)}]$ means $\underline{F(Z)}$ is the lower bound for the interval $F(Z)$, and $U + V = [\underline{U}, \overline{U}] + [\underline{V}, \overline{V}] = [\underline{U} + \underline{V}, \overline{U} + \overline{V}]$. Let $\odot$ denote elementwise multiplication. The optimization algorithm, whose pseudocode is presented in Algorithm 3, takes the form of a bisection-like iterative procedure: we first assign intervals A, B and Ξ as the search domain, and then, c times, we halve them as follows: for each possible way to halve A, B and Ξ, compute the IA lower bound in the subinterval, $\underline{\widehat{L}(S'(A^{(b)}, B^{(b)}, \Xi^{(b)}), \mathbf{y})} = \widehat{L}(\mathbf{y} \odot \overline{S'(A^{(b)}, B^{(b)}, \Xi^{(b)})} + (1 - \mathbf{y}) \odot \underline{S'(A^{(b)}, B^{(b)}, \Xi^{(b)})}, \mathbf{y})$ (this follows directly from the IA definitions applied to our losses), and pick the one with the lowest IA lower bound as the new search domain. Since each step halves the search domain and gives an extra bit of numerical precision, c is fixed as the precision of the floating-point type.

By using Algorithm 3 to solve (2.4), ExactBoost has a runtime complexity of order $O(pn \log(n))$ and a space complexity of order $O(n)$. Thus, it can scale well even to large datasets, as shown in Section 2.3.

2.2 Theoretical results

This section develops a theory of generalization for ExactBoost under margin-type conditions. It shows, in particular, that the population error of an ExactBoost's score S can be upper bounded by the sum of a margin-adjusted sample error of S plus an error depending on $\mathcal{H}$. Crucially, the latter is controlled uniformly over S and only depends on the class of functions $\mathcal{H}$. Thus, if a method has a margin-adjusted training loss that is sufficiently small relative to θ, then it generalizates well. When $\mathcal{H}$ is the set of stumps (2.3), for example, one can allow for a number of features that is nearly as large as an exponential in the number of positive and negative examples.

The theoretical results are based on the representative losses (1.1), (1.2) and (1.3), which display different levels of non-decomposability. While this affects the guarantees for each loss slightly differently, the proof techniques allow for generalization to other non-decomposable losses, as pointed out below. Importantly, the margin adjustment on each loss is essentially the same.

The results below extend to non-decomposable losses previous work in obtaining empirical bounds for classification tasks [BFLS98, BM02, KP02]. The results presented here differ in spirit from those obtained via surrogate losses [Aga13, KNJ15]. Surrogate metrics can provide upper bounds of the desired loss but often lack a natural quantitative interpretation. The theorems below, on the other hand, show that minimizing a margin-adjusted empirical loss leads, with high probability, to a small population loss.

Notation. Assume $\mathcal{D}$ is a probability distribution over pairs $(X, y) \in \mathbb{R}^p \times \{0, 1\}$, and let $\mathcal{D}_0$ (respectively, $\mathcal{D}_1$) denote the conditional distribution of X when $y = 0$ (respectively, 1). When unambiguous, $\mathcal{D}$ might also denote the marginal distribution of X. The data is $(X_i, y_i)_{i=1}^n \sim \mathcal{D}$ iid, and, conditionally on the number n_1 of indices i with $y_i = 1$ (and also defining $n_0 \coloneqq n - n_1$), the subsamples $\mathbf{X}_1 \coloneqq (X_i : i \in [n], y_i = 1)$ and $\mathbf{X}_0 \coloneqq (X_i : i \in [n], y_i = 0)$ are iid from $\mathcal{D}_1$ and $\mathcal{D}_0$. Score functions $S : \mathbb{R}^p \to [-1, 1]$ are convex combinations of elements in a family of measurable functions $\mathcal{H} : \mathbb{R}^p \to [-1, 1]$. Let $\{\sigma_i\}_{i=1}^n$ be iid uniform over ± 1 and independent from data. Define the Rademacher complexities of $\mathcal{H}$ with respect to $\mathcal{D}$, $\mathcal{D}_0$ and $\mathcal{D}_1$:

$$\mathcal{R}_n(\mathcal{H}) \coloneqq \mathbb{E}_{\mathcal{D}} \left[\mathbb{E}_\sigma \left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \sigma_i h(X_i) \right] \right]$$

$$\mathcal{R}_{n,y}(\mathcal{H}) \coloneqq \mathbb{E}_{\mathcal{D}_y} \left[\mathbb{E}_\sigma \left[\sup_{h \in \mathcal{H}} \frac{1}{n_y} \sum_{i \,:\, y_i = y} \sigma_i h(X_i) \right] \right],$$

for $y \in \{0, 1\}$. Note $\mathcal{R}_{n,y}(\mathcal{H})$ is defined conditionally on n_y, the number of examples with label y. When n_y equals zero, we set $\mathcal{R}_{n,y}(\mathcal{H}) = 1$ by convention. Note $\mathcal{R}_n(\mathcal{H}) = O(\sqrt{\log p / n})$ when $\mathcal{H}$ is as in (2.3).

2.2.1 Margin result for AUC loss The AUC loss, for $(X, X') \sim \mathcal{D}_1 \times \mathcal{D}_0$, and its θ-margin-adjusted version are given by:

$$\begin{aligned}\mathrm{AUC}(S) &\coloneqq 1 - \mathbb{P}\{S(X) > S(X')\},\\ \widehat{\mathrm{AUC}}_\theta(S) &\coloneqq 1 - \frac{1}{n_1}\sum_{i:y_i=1}\frac{1}{n_0}\sum_{j:y_j=0}\mathbf{1}_{[S(X_i)-\theta>S(X_j)]}.\end{aligned}$$

Note $\widehat{\mathrm{AUC}}_\theta(S)$ is one minus the area under the curve when one subtracts θ from the scores of 1-labelled samples. Because AUC relies on pairwise interactions, it is not readily decomposable over each sample point. Still, the U-statistic structure of this loss allows for the following result.

Theorem 2.1. *Given $\theta > 0$, $\delta \in (0,1)$, $n_0, n_1 > 0$, and a class of functions $\mathcal{H}$ from $\mathbb{R}^p$ to $[-1,1]$, the following holds with probability at least $1-\delta$: for all score functions $S : \mathbb{R}^p \to [-1,1]$ obtained as convex combinations of the elements of $\mathcal{H}$,*

$$\mathrm{AUC}(S) \le \widehat{\mathrm{AUC}}_\theta(S) + \frac{4}{\theta}\zeta_{\mathrm{AUC}}(\mathcal{H}) + \sqrt{\frac{2\log(1/\delta)}{\min\{n_0,n_1\}}},$$

where $\zeta_{\mathrm{AUC}}(\mathcal{H}) = \mathcal{R}_{\min\{n_0,n_1\},0}(\mathcal{H}) + \mathcal{R}_{\min\{n_0,n_1\},1}(\mathcal{H})$.

Theorem 2.1 holds conditionally on $n_0, n_1 > 0$, which will hold with very high probability unless $\mathcal{D}$ is too imbalanced towards $y = 0$ or $y = 1$. When $\mathcal{H}$ is given by (2.3), the theorem implies, for constant δ, that the score S produced by the algorithm satisfies $\mathrm{AUC}(S) \le \widehat{\mathrm{AUC}}_\theta(S) + o(1)$ with high probability when $\min\{n_0, n_1\} \gg \theta^{-2}\log p$. Theorem 2.1 can be extended to similar pairwise losses.

2.2.2 Margin result for KS loss For a score S, the KS loss and its margin-adjusted sample version are defined as:

$$\begin{aligned}\mathrm{KS}(S) &= 1 - \sup_{t\in\mathbb{R}}\Big(\mathbb{P}_{X\sim\mathcal{D}_0}\{S(X)\le t\} - \mathbb{P}_{X\sim\mathcal{D}_1}\{S(X)\le t\}\Big),\\ \widehat{\mathrm{KS}}_\theta(S) &= 1 - \max_{t\in\mathbb{R}}\left(\frac{1}{n_0}\sum_{i\,:\,y_i=0}\mathbf{1}_{[S(X_i)\le t]} - \frac{1}{n_1}\sum_{i\,:\,y_i=1}\mathbf{1}_{[S(X_i)-\theta\le t]}\right),\end{aligned}$$

where, by convention, $\widehat{\mathrm{KS}}_\theta(S) = 1$ if $n_1 = 0$ or $n_0 = 0$.

Theorem 2.2. *Given $\theta > 0$, $\delta \in (0,1)$, $n_0, n_1 > 0$, and a class of functions $\mathcal{H}$ from $\mathbb{R}^p$ to $[-1,1]$, the following holds with probability at least $1-\delta$: for all score functions $S : \mathbb{R}^p \to [-1,1]$ obtained as*

convex combinations of the elements of $\mathcal{H}$*,*

$$\mathrm{KS}(S) \le \widehat{\mathrm{KS}}_\theta(S) + \frac{8}{\theta}\zeta_{\mathrm{KS}}(\mathcal{H}) + \sqrt{\frac{\log(2/\delta)}{2}}\left(\frac{1}{\sqrt{n_0}} + \frac{1}{\sqrt{n_1}}\right),$$

where $\zeta_{\mathrm{KS}}(\mathcal{H}) = \mathcal{R}_{n_0,0}(\mathcal{H}) + \mathcal{R}_{n_1,1}(\mathcal{H}) + n_0^{-1/2} + n_1^{-1/2}$.

Thus a score that achieves a small margin-adjusted KS loss will, with high probability, have good performance on the population, if we condition on $n_0, n_1 > 0$. Similarly to Theorem 2.1, when the base learners are stumps, we obtain

$$\mathrm{KS}(S) \le \widehat{\mathrm{KS}}_\theta(S) + C\sqrt{\frac{\theta^{-2}(1+\log p) + \log(2/\delta)}{\min\{n_0, n_1\}}}.$$

Thus for constant δ, good training performance on the margin-adjusted loss leads to good generalization whenever $\theta^{-2}\log p \ll \min\{n_0, n_1\}$.

2.2.3 Margin result for P@k loss For the precision at k loss, given a score $S : \mathbb{R}^p \to [-1, 1]$ and $\alpha \in (0, 1)$, let $t_\alpha(S)$ denote its $(1-\alpha)$-quantile under the population distribution and $\widehat{t}_\alpha(S)$ the sample version,

$$t_\alpha(S) \coloneqq \inf\{t \in \mathbb{R} : \mathbb{P}\{S(X) \le t\} \ge 1 - \alpha\}$$
$$\widehat{t}_\alpha(S) \coloneqq \inf\left\{t \in \mathbb{R} : \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{[S(X_i)\le t]} \ge 1 - \alpha\right\}.$$

The precision at k loss of S (for parameter α) and its margin-adjusted sample version are

$$\mathrm{P@k}_\alpha(S) \coloneqq 1 - \mathbb{P}\{y = 1, S(X) \ge t_\alpha(S)\},$$
$$\widehat{\mathrm{P@k}}_\theta(S) \coloneqq 1 - \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{[y_i=1, S(X_i)-\theta \ge \widehat{t}_\alpha(S)]}.$$

Informally, $\widehat{\mathrm{P@k}}_\theta(S)$ is the sample precision at k when 1-labelled examples have their scores reduced by θ after the threshold $\widehat{t}_\alpha(S)$ has been computed. Similarly to the KS loss, P@k is non-decomposable due to a global threshold $\widehat{t}_\alpha(S)$, but the lack of optimality structure makes proving the next result much more involved.

Theorem 2.3. *Given* $\theta > 0$*,* $\delta \in (0, 1)$*,* $n_0, n_1 > 0$*, and a class of functions* $\mathcal{H}$ *from* $\mathbb{R}^p$ *to* $[-1, 1]$*, define*

$$\bar{\eta}_n(\mathcal{H}) \coloneqq \sqrt{4\mathcal{R}_n(\mathcal{H}) + \frac{4}{\sqrt{n}}} + \sqrt{\frac{\log(3/(\delta - \delta^2))}{n}},$$

Assume $\theta > 2\bar{\eta}_n(\mathcal{H})$ *and* $\mathbb{P}(\min\{n_0, n_1\} > 0) \geq 1-\delta$. *Then the following holds with probability* $\geq 1-\delta$*: if* $\delta' := \delta - \delta^2$*, then for all score functions* $S : \mathbb{R}^p \to [-1, 1]$ *obtained as convex combinations of the elements of* $\mathcal{H}$*, it holds*

$$\begin{aligned}\mathrm{P@k}(S) \leq & \widehat{\mathrm{P@k}}_\theta(S) + \frac{4\mathcal{R}_{n_1,1}(\mathcal{H}) + \frac{4}{\sqrt{n_1}}}{\theta - 2\bar{\eta}_n(\mathcal{H})} \\ & + \bar{\eta}_n(\mathcal{H}) + \sqrt{2\frac{\log(3/\delta')}{n_1}} + \sqrt{\frac{\log(3/\delta')}{2n}}.\end{aligned}$$

The proof techniques of the theorem above can be generalized to other combinatorial losses that use a restricted sample, such as partial AUC.

2.2.4 Subsampling Subsampling can help ExactBoost avoid overfitting. The next proposition is helpful in controlling its impact in the optimization procedure for some losses.

Proposition 2.4. *Let* $\widehat{L}$ *be either the* $\widehat{\mathrm{AUC}}$ *or the* $\widehat{\mathrm{KS}}$ *loss. Consider a subset of indices* $I = I_0 \cup I_1 \subset [n]$ *chosen independently and uniformly at random with equal number of positive and negative cases,* $|I_0| = |I_1| = k$*. Let* h_R *be the optimal stump over the reduced sample* $\{(X_j, y_j)\}_{j\in I}$ *and score* S *and* h_* *the optimal stump over the entire sample* $\{(X_i, y_i)\}_{i\in[n]}$*. Then,*

$$\mathbb{E}[\widehat{L}(S + h_R)] \leq \widehat{L}(S + h_*) + \frac{e}{k},$$

where the expectation is over the choice of I*.*

Hence, using random subsets of observations in ExactBoost with balanced proportions of positive and negative examples leads to an expected error close to the optimal one.

2.2.5 Ensembling Since minimizing the margin-adjusted empirical loss can generalize to the population loss, it is natural to investigate whether ExactBoost can also provide a good ensembling technique for other classifiers. Indeed, for some losses, it is possible to guarantee that the empirical loss of the ensembler is smaller than the empirical loss of each ensembler member.

Denote the vector of scores for the ith data point by $Z_i \coloneqq (S_1(X_i), S_2(X_i), \ldots, S_M(X_i))^T \in \mathbb{R}^M$, M being the number of models, and train ExactBoost over a modified dataset $(Z_i, y_i)_{i=1}^n$. The next proposition shows that the training set performance of ExactBoost over $(Z_i, y_i)_{i=1}^n$ using either the KS or P@k metrics is always at least as good as that of the the best score function over $(X_i, y_i)_{i=1}^n$.

Proposition 2.5. *Let* $\widehat{L}$ *be either the* $\widehat{\mathrm{KS}}$ *or the* $\widehat{\mathrm{P@k}}$ *loss. Consider the score* $S_* : \mathbb{R}^M \to \mathbb{R}$ *obtained*

by ExactBoost over the dataset $(Z_i, y_i)_{i=1}^n$ *with initial score* $S_0 \equiv 0$. *Then:*

$$\widehat{L}_{(Z_i,y_i)_{i=1}^n}(S_*) \leq \min_{1\leq m\leq M} \widehat{L}_{(X_i,y_i)_{i=1}^n}(S_m),$$

where $\widehat{L}_{(Z_i,y_i)_{i=1}^n}(\cdot)$ *and* $\widehat{L}_{(X_i,y_i)_{i=1}^n}(\cdot)$ *denote the loss over the ensemble and the original data.*

Section 2.3 shows that, in practice, ensembling with ExactBoost leads to better results than ensembling with other surrogate-based algorithms. The fact that the inputs for the ensembler can be trained with surrogate-based methods attenuates overfitting, and speeds up ExactBoost by reducing the set of original features p to the number of models M.

2.3 Experiments

To test its performance, ExactBoost is compared against 10 exact and surrogate-based algorithms, on 30 heterogeneous datasets, over three different losses. For ease of presentation, results of 10 representative datasets are shown.

Dataset	Observations	Features	Positives
a1a	1605	119	24.6%
german	1000	20	70.0%
gisette	6000	5000	50.0%
gmsc	150000	10	6.7%
heart	303	21	45.9%
ionosphere	351	34	64.1%
liver-disorders	145	5	37.9%
oil-spill	937	49	4.4%
splice	1000	60	48.3%
svmguide1	3089	4	35.3%

Table 2.1: Dataset properties.

Datasets. Table 2.1 displays the main characteristics of each dataset, which span economic, medical, radar, financial and ecological applications, and range from balanced to imbalanced.

Surrogate benchmarks. ExactBoost is compared to various standard learning algorithms: AdaBoost, k-nearest neighbors, logistic regression and random forest (via their Scikit-Learn implementation in [PVG+11]), gradient boosting (via XGBoost, see [CG16]) and a 4-layer connected neural net (via TensorFlow, see [AAB+15]).

Exact benchmarks. Several algorithms that specifically optimize the performance metric are considered. For KS, the baseline is DMKS [FC19], and, for P@k, the baseline is TopPush [LJZ14]. For AUC, the baseline is RankBoost [FISS03], a boosting algorithm shown to optimize the AUC under

certain conditions in [CM03].

Dataset	RankBoost	DMKS	TopPush
a1a	55.90×	102.78×	0.82×
german	23.98×	1.28×	0.88×
gisette	OOT	55.68×	0.02×
gmsc	OOT	22.89×	0.08×
heart	3.32×	19.00×	5.25×
ionosphere	3.97×	3.48×	2.69×
liver-disorders	1.91×	6.36×	12.53×
oil-spill	5.93×	7.92×	2.18×
splice	49.78×	1.19×	1.20×
svmguide1	220.05×	1.88×	4.27×

Table 2.2: Timings of various exact algorithms vs ExactBoost (above 1× indicates ExactBoost is faster). TopPush is fast but much less precise; see Table 2.3.

Hyperparameters. Hyperparameters were fixed throughout the experiments. Baseline models were trained with the package-provided hyperparameters. Aided by experimental evidence on held-out datasets, ExactBoost uses as default $E = 250$ runs, $T = 50$ rounds, subsampling of 20% and margin of $\theta = 0.05$. See Subsection 2.3.1 for further discussions.

Computational allowance, environment and code. Experiments were run with four Intel Xeon E5-4650 CPUs with 2.60 GHz, 64 threads, and 810 GB of RAM. Code to reproduce figures and tables can be found at `https://github.com/dccsillag/exactboost`. Methods had at most 5 days to run on each dataset.

2.3.1 Effect of Hyperparameters on ExactBoost

ExactBoost has two main hyperparameters that control overfitting: the margin θ and the number of runs averaged E (see Algorithm 2). Figure 2.1 shows how the margin affects the test error for the AUC, KS and P@k losses in three different datasets. Generally, though not always, the loss decreases with small positive margins, but becomes increasing once the margin is too large.

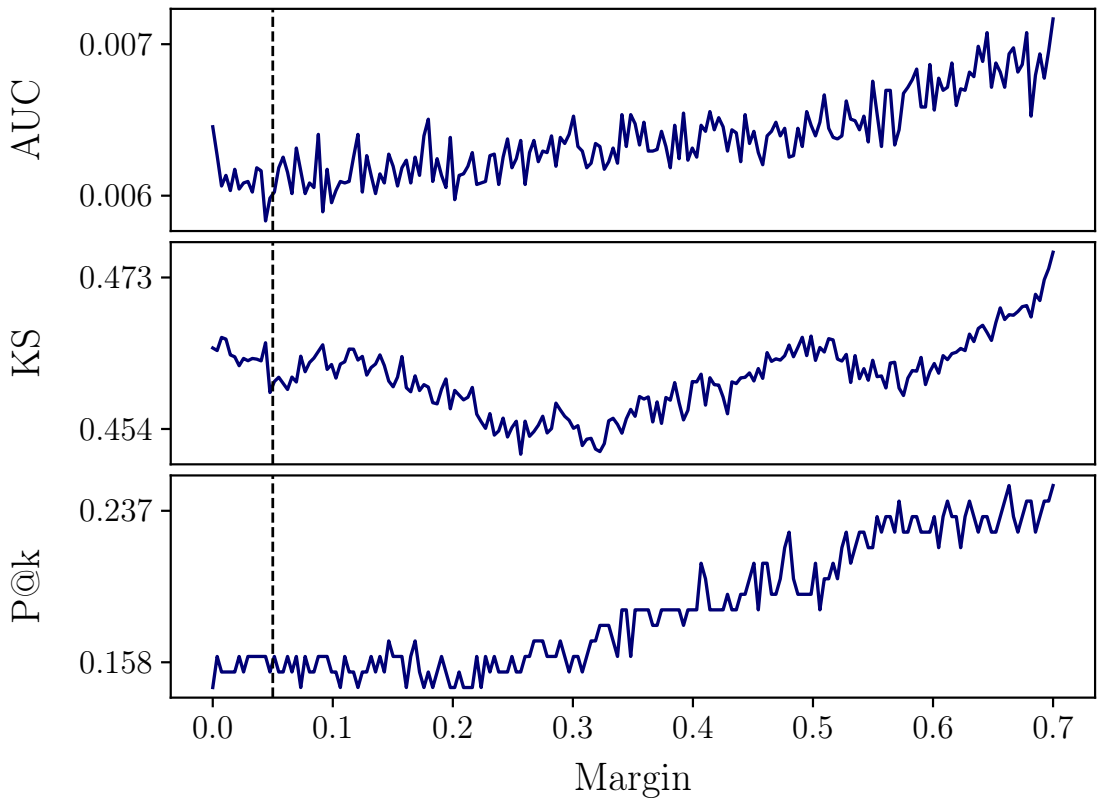


Figure 2.1: Effect of margin on ExactBoost's test performance on `svmguide1`, `gmsc` and `splice`. The vertical line shows the default $\theta = 0.05$. There are gains with small margins; the performance degrades with large θ.

2.3.2 ExactBoost vs exact and surrogate benchmarks The performance of ExactBoost as an estimator is investigated via its 5-fold cross-validated test error. Table 2.3 shows that ExactBoost is generally better than loss-specific alternatives. In particular, the table includes comparisons to additional exact models available in the literature, such as SVMPerf [Joa05], which directly optimizes for multivariate performance metrics such as P@k, and plugin logistic [KNRD14, DKKN17], a fast hybrid method that uses the metric of interest, say AUC, to pick the optimal threshold for logistic regression using a separate data fold. Figure 2.2 shows that ExactBoost also has good performance against surrogate benchmarks.

In terms of timings, Table 2.2 shows that ExactBoost scales well even to large datasets. Note it is faster than other exact alternatives, and while TopPush can be faster, it is generally much less precise (see Table 2.3).

2.3.3 ExactBoost as an ensembler In the experiments below, 5-fold cross-validation is used to compare ExactBoost against other ensemblers. Six base models were used: AdaBoost, k-nearest

Dataset	AUC			KS		P@k		
	ExactBoost	RankBoost	Plugin Logistic	ExactBoost	DMKS	ExactBoost	TopPush	SVMPerf
a1a	$\mathbf{0.11 \pm 0.0}$	0.13 ± 0.0	0.20 ± 0.0	$\mathbf{0.37 \pm 0.0}$	$\mathbf{0.37 \pm 0.0}$	0.26 ± 0.1	0.29 ± 0.1	$\mathbf{0.22 \pm 0.1}$
german	$\mathbf{0.23 \pm 0.0}$	0.24 ± 0.0	0.28 ± 0.0	$\mathbf{0.53 \pm 0.0}$	0.55 ± 0.0	$\mathbf{0.11 \pm 0.0}$	0.26 ± 0.1	0.21 ± 0.0
gisette	$\mathbf{0.01 \pm 0.0}$	OOT	0.03 ± 0.0	0.09 ± 0.0	$\mathbf{0.06 \pm 0.0}$	0.02 ± 0.0	$\mathbf{0.01 \pm 0.0}$	$\mathbf{0.01 \pm 0.0}$
gmsc	$\mathbf{0.21 \pm 0.0}$	OOT	0.38 ± 0.0	$\mathbf{0.44 \pm 0.0}$	0.45 ± 0.0	$\mathbf{0.52 \pm 0.0}$	0.96 ± 0.0	0.85 ± 0.0
heart	$\mathbf{0.09 \pm 0.0}$	0.13 ± 0.0	0.19 ± 0.0	0.30 ± 0.0	$\mathbf{0.28 \pm 0.0}$	$\mathbf{0.04 \pm 0.1}$	0.13 ± 0.1	$\mathbf{0.04 \pm 0.1}$
iono	$\mathbf{0.04 \pm 0.0}$	$\mathbf{0.04 \pm 0.0}$	0.17 ± 0.0	$\mathbf{0.13 \pm 0.0}$	0.28 ± 0.0	$\mathbf{0.03 \pm 0.0}$	0.15 ± 0.1	0.16 ± 0.1
liver	$\mathbf{0.22 \pm 0.1}$	0.32 ± 0.1	0.35 ± 0.1	$\mathbf{0.45 \pm 0.1}$	0.50 ± 0.1	$\mathbf{0.23 \pm 0.1}$	0.47 ± 0.2	0.33 ± 0.2
oil-spill	$\mathbf{0.09 \pm 0.1}$	$\mathbf{0.09 \pm 0.1}$	0.39 ± 0.1	$\mathbf{0.25 \pm 0.1}$	0.45 ± 0.1	$\mathbf{0.52 \pm 0.3}$	0.96 ± 0.1	1.00 ± 0.0
splice	0.04 ± 0.0	$\mathbf{0.02 \pm 0.0}$	0.21 ± 0.0	$\mathbf{0.16 \pm 0.0}$	0.36 ± 0.0	$\mathbf{0.03 \pm 0.0}$	0.12 ± 0.0	0.10 ± 0.0
svmg1	0.01 ± 0.0	$\mathbf{0.00 \pm 0.0}$	0.05 ± 0.0	$\mathbf{0.06 \pm 0.0}$	0.09 ± 0.0	$\mathbf{0.00 \pm 0.0}$	$\mathbf{0.00 \pm 0.0}$	0.03 ± 0.0

Table 2.3: Evaluation of exact benchmarks. OOT indicates the time budget of 5 days was exceeded. ExactBoost has the best performance for all metrics: it is faster and uses less memory than RankBoost and DMKS (see Table 2.2), and much more accurate than Plugin Logistic and TopPush.

neighbors, logistic regression, neural network, random forest and XGBoost. These models were trained on training folds, and their predictions on test folds were used as features for the ensemble models.

Table 2.4 shows the results of using different surrogate and exact models as ensemblers. The surrogate ensemblers were AdaBoost, logistic regression, neural network, random forest and XGBoost, while the exact benchmarks were given by RankBoost (for AUC), DMKS (for KS) and TopPush (for P@k).

ExactBoost is generally the best ensembler available. In fact, it is able to match or overcome the performance of the best base model available and is robust to noisy features coming from poorly performing base models. This is particularly attractive because, given the discrete nature of combinatorial losses, it is often the case that the best performing model changes from dataset to dataset. ExactBoost's success can be interpreted as transfer learning: it is able to better combine high-signal features trained with surrogate losses by considering the exact metric of interest.

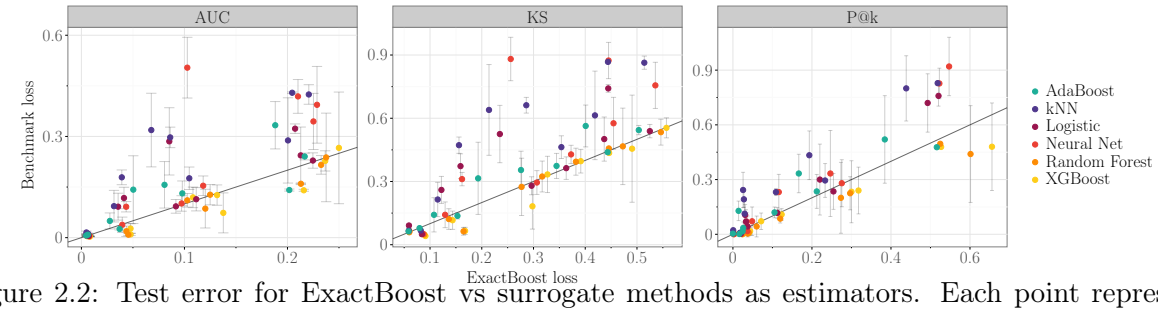


Figure 2.2: Test error for ExactBoost vs surrogate methods as estimators. Each point represents a dataset from Table 2.1. Alternatives are generally worse than ExactBoost or statistically indistinguishable.

2.4 Proofs and technical results

2.4.1 Technical results We present a general theoretical framework that we apply to obtain the margin results in Section 2.4.

Loss	Dataset	ExactBoost	AdaBoost	Logistic	Neural Net	Rand. For.	XGBoost	Exact Bench.
AUC	a1a	$\mathbf{0.13 \pm 0.0}$	0.17 ± 0.0	0.14 ± 0.0	0.15 ± 0.0	0.27 ± 0.1	0.28 ± 0.1	0.16 ± 0.0
	german	$\mathbf{0.23 \pm 0.0}$	0.32 ± 0.0	0.24 ± 0.0	0.50 ± 0.1	0.33 ± 0.0	0.35 ± 0.0	0.30 ± 0.1
	gisette	$\mathbf{0.00 \pm 0.0}$	0.01 ± 0.0	0.01 ± 0.0	0.01 ± 0.0	0.03 ± 0.0	0.02 ± 0.0	0.01 ± 0.0
	gmsc	0.15 ± 0.0	$\mathbf{0.14 \pm 0.0}$	0.31 ± 0.0	0.46 ± 0.0	0.42 ± 0.0	0.41 ± 0.0	0.15 ± 0.0
	heart	$\mathbf{0.12 \pm 0.0}$	0.18 ± 0.1	$\mathbf{0.12 \pm 0.0}$	0.23 ± 0.1	0.19 ± 0.0	0.23 ± 0.1	0.15 ± 0.0
	iono.	$\mathbf{0.04 \pm 0.0}$	0.05 ± 0.0	0.07 ± 0.0	0.07 ± 0.0	0.07 ± 0.0	0.09 ± 0.0	0.05 ± 0.0
	liver	$\mathbf{0.30 \pm 0.1}$	0.34 ± 0.1	0.34 ± 0.1	0.34 ± 0.1	0.38 ± 0.0	0.38 ± 0.0	0.38 ± 0.1
	oil-spill	$\mathbf{0.17 \pm 0.1}$	0.19 ± 0.1	0.29 ± 0.2	0.46 ± 0.1	0.38 ± 0.1	0.35 ± 0.2	0.19 ± 0.1
	splice	$\mathbf{0.01 \pm 0.0}$	$\mathbf{0.01 \pm 0.0}$	0.08 ± 0.0	0.05 ± 0.0	0.04 ± 0.0	0.04 ± 0.0	0.02 ± 0.0
	svmg1	$\mathbf{0.00 \pm 0.0}$	0.01 ± 0.0	0.01 ± 0.0	0.01 ± 0.0	0.03 ± 0.0	0.04 ± 0.0	0.01 ± 0.0
KS	a1a	$\mathbf{0.37 \pm 0.1}$	0.44 ± 0.1	0.40 ± 0.1	0.41 ± 0.1	0.54 ± 0.1	0.57 ± 0.1	0.49 ± 0.1
	german	$\mathbf{0.50 \pm 0.1}$	0.68 ± 0.1	0.53 ± 0.1	0.89 ± 0.1	0.66 ± 0.0	0.69 ± 0.1	0.53 ± 0.1
	gisette	$\mathbf{0.04 \pm 0.0}$	$\mathbf{0.04 \pm 0.0}$	0.07 ± 0.0	0.07 ± 0.0	0.06 ± 0.0	$\mathbf{0.04 \pm 0.0}$	0.10 ± 0.0
	gmsc	$\mathbf{0.43 \pm 0.0}$	0.44 ± 0.0	0.73 ± 0.0	0.95 ± 0.0	0.85 ± 0.0	0.83 ± 0.0	0.46 ± 0.0
	heart	$\mathbf{0.34 \pm 0.1}$	0.38 ± 0.1	0.37 ± 0.1	0.52 ± 0.1	0.38 ± 0.1	0.46 ± 0.1	0.40 ± 0.0
	iono.	$\mathbf{0.13 \pm 0.1}$	0.18 ± 0.1	0.18 ± 0.1	0.17 ± 0.1	0.15 ± 0.1	0.19 ± 0.1	0.27 ± 0.1
	liver	$\mathbf{0.53 \pm 0.1}$	0.60 ± 0.2	0.59 ± 0.2	0.61 ± 0.1	0.76 ± 0.1	0.76 ± 0.0	0.60 ± 0.2
	oil-spill	$\mathbf{0.33 \pm 0.2}$	$\mathbf{0.33 \pm 0.2}$	0.47 ± 0.2	0.89 ± 0.1	0.76 ± 0.2	0.69 ± 0.3	0.63 ± 0.3
	splice	$\mathbf{0.06 \pm 0.0}$	0.09 ± 0.0	0.28 ± 0.0	0.21 ± 0.0	0.09 ± 0.0	0.09 ± 0.0	0.28 ± 0.0
	svmg1	$\mathbf{0.06 \pm 0.0}$	0.08 ± 0.0	$\mathbf{0.06 \pm 0.0}$	$\mathbf{0.06 \pm 0.0}$	0.07 ± 0.0	0.07 ± 0.0	$\mathbf{0.06 \pm 0.0}$
P@k	a1a	$\mathbf{0.22 \pm 0.1}$	0.34 ± 0.1	0.28 ± 0.1	0.32 ± 0.1	0.34 ± 0.2	0.40 ± 0.1	0.29 ± 0.1
	german	$\mathbf{0.13 \pm 0.0}$	0.16 ± 0.1	$\mathbf{0.13 \pm 0.0}$	0.33 ± 0.0	0.20 ± 0.0	0.21 ± 0.1	0.18 ± 0.0
	gisette	0.01 ± 0.0	0.01 ± 0.0	$\mathbf{0.00 \pm 0.0}$	$\mathbf{0.00 \pm 0.0}$	0.02 ± 0.0	0.02 ± 0.0	0.01 ± 0.0
	gmsc	0.51 ± 0.0	$\mathbf{0.48 \pm 0.0}$	0.74 ± 0.1	0.88 ± 0.0	0.65 ± 0.1	0.62 ± 0.0	0.96 ± 0.0
	heart	0.07 ± 0.1	0.19 ± 0.1	$\mathbf{0.06 \pm 0.0}$	0.19 ± 0.1	0.23 ± 0.1	0.29 ± 0.1	0.14 ± 0.2
	iono.	$\mathbf{0.03 \pm 0.0}$	0.04 ± 0.1	0.05 ± 0.0	0.06 ± 0.1	0.09 ± 0.1	0.10 ± 0.1	0.10 ± 0.1
	liver	$\mathbf{0.27 \pm 0.2}$	0.33 ± 0.2	0.33 ± 0.2	0.40 ± 0.3	0.40 ± 0.2	0.33 ± 0.2	0.30 ± 0.2
	oil-spill	$\mathbf{0.44 \pm 0.2}$	0.72 ± 0.2	0.84 ± 0.2	0.92 ± 0.1	0.72 ± 0.2	0.68 ± 0.3	0.68 ± 0.2
	splice	$\mathbf{0.01 \pm 0.0}$	$\mathbf{0.01 \pm 0.0}$	0.04 ± 0.0	0.04 ± 0.0	0.05 ± 0.0	0.05 ± 0.0	0.05 ± 0.0
	svmg1	$\mathbf{0.00 \pm 0.0}$	0.01 ± 0.0	$\mathbf{0.00 \pm 0.0}$	0.01 ± 0.0	0.05 ± 0.0	0.05 ± 0.0	$\mathbf{0.00 \pm 0.0}$

Table 2.4: Evaluation of ensemblers. The exact benchmarks are RankBoost (AUC), DMKS (KS) and TopPush (P@k). ExactBoost is generally the best performer (and top 2 in all cases, for all losses).

Empirical vs. cumulative distribution functions We now note a "margin-type" result relating population and empirical cumulative distribution functions of elements of $\mathcal{G}$. It essentially follows from [KP02, Theorem 1].

Lemma 2.6. *With the above notation, assume further that the functions in $\mathcal{G}$ are bounded by 1 in absolute value. Given $\eta > 0$, the inequality below holds with probability at least $1 - \delta$:*

$$\forall g \in \mathcal{G},\, t \in \mathbb{R} \,:\, \mathbb{P}_{Z \sim \mathcal{D}_Z}\{g(Z) \leq t\} \leq \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{[g(Z_i) \leq t + \eta]} + \frac{4\mathcal{R}_m(\mathcal{G}) + \frac{4}{\sqrt{m}}}{\eta} + \sqrt{\frac{\log(1/\delta)}{2m}}.$$

Similarly, the following holds with probability at least $1 - \delta$:

$$\forall g \in \mathcal{G},\, t \in \mathbb{R} \,:\, \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{[g(Z_i) \leq t]} \leq \mathbb{P}_{Z \sim \mathcal{D}_Z}\{g(Z) \leq t + \eta\} + \frac{4\mathcal{R}_m(\mathcal{G}) + \frac{4}{\sqrt{m}}}{\eta} + \sqrt{\frac{\log(1/\delta)}{2m}}.$$

Proof. We only prove the first of these results, as the second one is similar. Define:

$$\Delta \coloneqq \sup_{g\in\mathcal{G},t\in\mathbb{R}} \left(\mathbb{P}_{Z\sim\mathcal{D}_Z}\{g(Z)\le t\} - \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{[g(Z_i)\le t+\eta]} \right).$$

Since $\|g\|_\infty \le 1$ for all $g \in \mathcal{G}$, the term inside the brackets is equal to 0 for $t \ge 1$ and at most 0 for $t \le -1$. In particular, the supremum defining Δ is nonnegative and achieved for some $t \in [-1, 1]$.

Now consider $\phi_\eta : \mathbb{R} \to [0, 1]$ defined by:

$$\phi_\eta(x) \coloneqq \begin{cases} 1, & x \le 0; \\ 1 - \frac{x}{\eta}, & 0 < x \le \eta; \\ 0, & x > \eta. \end{cases} \quad (x \in \mathbb{R}).$$

Then we see at once that $\mathbf{1}_{[g(Z_i)\le t+\eta]} \ge \phi_\eta(g(Z_i)-t) \ge \mathbf{1}_{[g(Z_i)\le t]}$, so that, for any $g \in \mathcal{G}$ and $t \in [-1, 1]$,

$$\mathbb{P}_{Z\sim\mathcal{D}_Z}\{g(Z)\le t\} - \mathbf{1}_{[g(Z_i)\le t+\eta]} \le \mathbb{E}_{Z\sim\mathcal{D}_Z}\phi_\eta(g(Z)-t) - \phi_\eta(g(Z_i)-t).$$

Therefore,

$$\Delta \le \Delta^* \coloneqq \sup_{g\in\mathcal{G},\, t\in[-1,1]} \left(\mathbb{E}_{Z\sim\mathcal{D}}\phi_\eta(g(Z)-t) - \frac{1}{m}\sum_{i=1}^{m} \phi_\eta(g(Z_i)-t) \right).$$

We now consider Δ^*. The symmetrization inequality (1.9) implies that

$$\mathbb{E}\,\Delta^* \le 2\mathcal{R}_m(\tilde{\mathcal{G}}), \tag{2.5}$$

where $\tilde{\mathcal{G}}$ is the family of all functions of the form $\phi_\eta(g(\cdot)-t)-\phi_\eta(0)$ where $g \in \mathcal{G}$ and $t \in [-1, 1]$. Note also that ϕ_η is $1/\eta$-Lipschitz. Using items 4 and 5 of [BM02, Theorem 12], we see that:

$$\mathcal{R}_m(\tilde{\mathcal{G}}) \le 2\,\frac{\mathcal{R}_m(\mathcal{G}) + \frac{1}{\sqrt{m}}}{\eta}. \tag{2.6}$$

This bounds $\mathbb{E}\,\Delta^*$. To obtain a concentration inequality, notice that the random variable Δ^* is a function of independent random variables $Z_1, \ldots, Z_n$, and that changing the value of one of the Z_i will change the value of Δ^* by at most $1/m$ in absolute value. McDiarmid's inequality implies:

$$\mathbb{P}\left\{ \Delta^* - \mathbb{E}\,\Delta^* \le \sqrt{\frac{\log(1/\delta)}{2m}} \right\} \ge 1-\delta.$$

Combining this with (2.5) and (2.6) finishes the proof. □

The following corollary of Lemma 2.6 will also be useful. It may be viewed as a high-probability

uniform bound for the Levy distance between empirical and population cdf's of $g \in \mathcal{G}$.

Corollary 2.7. *In the setting of Lemma 2.6, let*

$$\bar{\eta}_m(\mathcal{G}) \coloneqq \sqrt{4\mathcal{R}_m(\mathcal{G}) + \frac{4}{\sqrt{m}}} + \sqrt{\frac{\log(1/\delta)}{2m}}.$$

Then either of the following statements holds with probability at least $1 - \delta$:

$$\forall g \in \mathcal{G}\, \forall t \in \mathbb{R} \,:\, \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{[g(Z_i) \le t]} \le \mathbb{P}_{Z \sim \mathcal{D}_Z}\{g(Z) \le t + \bar{\eta}_m(\mathcal{G})\} + \bar{\eta}_m(\mathcal{G});$$

and

$$\forall g \in \mathcal{G}\, \forall t \in \mathbb{R} \,:\, \mathbb{P}_{Z \sim \mathcal{D}_Z}\{g(Z) \le t\} \le \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{[g(Z_i) \le t + \bar{\eta}_m(\mathcal{G})]} + \bar{\eta}_m(\mathcal{G}).$$

Proof. Apply both parts of Lemma 2.6 with $\delta/2$ replacing δ and $\eta = \bar{\eta}_m(\mathcal{G})$. □

Rademacher complexities and U-statistic-type sums of indicators When we consider the AUC metric, we will need a "U-statistic" result for families $\mathcal{G}$. Let $\mathcal{D}'_Z$ be another probability distribution over $\mathcal{Z}$ and $Z'_1, \ldots, Z'_{m'} \sim \mathcal{D}'_Z$ be an i.i.d. sample of size m' from that distribution which is independent from $Z_1, \ldots, Z_m$. We let $\mathcal{R}'_{m'}(\mathcal{G})$ denote the Rademacher complexity of $\mathcal{G}$ with respect to the new sample size m' and the new distribution $\mathcal{D}'_Z$.

Lemma 2.8. *With the above definitions and notation, let $\eta > 0$ and $\delta \in (0, 1)$ be given. Let $m_{\min} \coloneqq \min\{m, m'\} > 0$. Then the following holds with probability at least $1 - \delta$: for all $g \in \mathcal{G}$,*

$$\begin{aligned}\mathbb{P}_{(Z,Z') \sim \mathcal{D}_Z \times \mathcal{D}_{Z'}}\{g(Z) \le g(Z')\} \le \frac{1}{m\,m'}\sum_{i=1}^{m}\sum_{i'=1}^{m'} \mathbf{1}_{[g(Z_i) < g(Z'_{i'}) + \eta]} \\ + 4\,\frac{\mathcal{R}_{m_{\min}}(\mathcal{G}) + \mathcal{R}'_{m_{\min}}(\mathcal{G})}{\eta} + \sqrt{\frac{\log(1/\delta)}{m_{\min}}}.\end{aligned}$$

Proof. The rough outline of this proof is similar to that of Lemma 2.6. We replace indicators by the function ϕ_η; apply symmetrization to bound the expectation of a supremum; and use McDiarmid's inequality to prove concentration. The key difference is at the symmetrization step, where we need to circumvent the fact that we are considering a U-statistic (rather than an i.i.d. sum).

Let ϕ_η be as in the proof of Lemma 2.6, and define

$$\Delta^* \coloneqq \sup_{g\in\mathcal{G}} \left(\mathbb{E}_{(Z,Z')\sim\mathcal{D}_Z\times\mathcal{D}_{Z'}} \phi_\eta(g(Z)-g(Z')) - \frac{1}{m\,m'} \sum_{i=1}^{m}\sum_{i'=1}^{m'} \phi_\eta(g(Z_i)-g(Z'_{i'})) \right).$$

The reasoning in the previous proof shows that:

$$\sup_{g\in\mathcal{G}} \left(\mathbb{P}_{(Z,Z')\sim\mathcal{D}_Z\times\mathcal{D}_{Z'}} \{g(Z)\le g(Z')\} - \frac{1}{m\,m'} \sum_{i=1}^{m}\sum_{i'=1}^{m'} \mathbf{1}_{[g(Z_i)\le g(Z'_{i'})+\eta]} \right) \le \Delta^*. \tag{2.7}$$

Our proof focuses on controlling Δ^*. We first notice a concentration property. Notice that Δ^* is a function of independent variables Z_i and $Z'_{i'}$. Since $\|\phi_\eta\|_\infty = 1$, changing one of the Z_i will change Δ^* by at most $1/m$ in absolute value, and changing a $Z'_{i'}$ will only change Δ^* by at most $1/m'$. Applying McDiarmid's inequality [McD98], we obtain:

$$\mathbb{P}\left\{ \Delta^* - \mathbb{E}\,\Delta^* \le \sqrt{\frac{\log(1/\delta)}{2\left(\frac{1}{\frac{1}{m}+\frac{1}{m'}}\right)}} \right\} \ge 1-\delta,$$

so that in particular

$$\mathbb{P}\left\{ \Delta^* - \mathbb{E}\,\Delta^* \le \sqrt{\frac{\log(1/\delta)}{m_{\min}}} \right\} \ge 1-\delta. \tag{2.8}$$

We now need to bound $\mathbb{E}\,\Delta^*$ in terms of Rademacher complexities. The main difficulty is that Δ^* is not an i.i.d. sum, and the symmetrization inequality (1.9) does not apply directly. However, one can use an averaging argument to obtain an upper bound for the expectation in terms of an i.i.d. sum.

The argument is as follows. Let $\mathcal{I}$ be the set of all pairs (S,f), where $S\subset[m]$ has size $m_{\min}$ and $f: S\to[m']$ is a one-to-one function (note that such (S,f) exist because $m_{\min}=\min\{m,m'\}$). By symmetry, we see that for all $(i,i')\in[m]\times[m']$,

$$\frac{\#\{(S,f)\in\mathcal{I} : i\in S,\, f(i)=i'\}}{\#\mathcal{I}} = \frac{m_{\min}}{m\,m'}.$$

Therefore,

$$\frac{1}{m\,m'}\sum_{i=1}^{m}\sum_{i'=1}^{m'} \phi_\eta(g(Z_i)-g(Z'_{i'})) = \frac{1}{\#\mathcal{I}} \sum_{(S,f)\in\mathcal{I}} \sum_{i\in S} \frac{\phi_\eta(g(Z_i)-g(Z'_{f(i)}))}{m_{\min}}.$$

Now plug the above into the definition of Δ^*, and obtain:

$$\Delta^* = \sup_{g\in\mathcal{G}} \left(\frac{1}{\#\mathcal{I}} \sum_{(S,f)\in\mathcal{I}} \sum_{i\in S} \frac{\mathbb{E}_{(Z,Z')\sim\mathcal{D}_Z\times\mathcal{D}'_Z} \phi_\eta(g(Z')-g(Z)) - \phi_\eta(g(Z_i)-g(Z'_{f(i)}))}{m_{\min}} \right).$$

That is, Δ^* is the supremum of an average over $(S, f) \in \mathcal{I}$. The corresponding "average of suprema" is at least as large, so

$$\Delta^* \leq \frac{1}{\#\mathcal{I}} \sum_{(S,f)\in\mathcal{I}} \sup_{g\in\mathcal{G}} \left(\sum_{i\in S} \frac{\mathbb{E}_{(Z,Z')\sim\mathcal{D}_Z\times\mathcal{D}'_Z} \phi_\eta(g(Z) - g(Z')) - \phi_\eta(g(Z_i) - g(Z'_{f(i)}))}{m_{\min}} \right).$$

Crucially, *all terms in the sum over $(S, f) \in \mathcal{I}$ have the same distribution.* In particular, all terms in the RHS of the preceding display have the same expectation. Considering the case where $S = [m_{\min}]$ and $f(i) = i$ for each $i \in [m_{\min}]$, we conclude:

$$\mathbb{E}\,\Delta^* \leq \mathbb{E} \sup_{g\in\mathcal{G}} \left(\sum_{i=1}^{m_{\min}} \frac{\mathbb{E}_{(Z,Z')\sim\mathcal{D}_Z\times\mathcal{D}'_Z} \phi_\eta(g(Z) - g(Z')) - \phi_\eta(g(Z_i) - g(Z'_i))}{m_{\min}} \right).$$

The pairs $\{(Z_i, Z'_i)\}_{i=1}^{m_{\min}}$ are i.i.d, and we can now apply symmetrization inequality (1.9). Letting

$$\tilde{\mathcal{G}} \coloneqq \{ \text{ all functions of the form "}(z, z') \in \mathcal{Z} \times \mathcal{Z} \mapsto \phi_\eta(g(z) - g(z')) - \phi_\eta(0)\text{" w/ } g \in \mathcal{G}\},$$

we obtain:

$$\mathbb{E}\,\Delta^* \leq\ 2\,\mathbb{E} \sup_{\tilde{g}\in\tilde{\mathcal{G}}} \left(\sum_{i=1}^{m_{\min}} \frac{\sigma_i\, \tilde{g}(Z_i, Z'_i)}{m_{\min}} \right)$$

where the σ_i are i.i.d. uniform over ± 1 and independent from the Z_i and Z'_i. As in the proof of Lemma 2.6, we observe that ϕ_η is $1/\eta$-Lipschitz, and apply item 5 of [BM02, Theorem 12] to obtain:

$$\mathbb{E}\,\Delta^* \leq\ \frac{4}{\eta}\,\mathbb{E} \sup_{g\in\mathcal{G}} \left(\sum_{i=1}^{m_{\min}} \frac{\sigma_i\,(g(Z_i) - g(Z'_i))}{m_{\min}} \right) \leq \frac{4\mathcal{R}_{m_{\min}}(\mathcal{G}) + 4\mathcal{R}'_{m_{\min}}(\mathcal{G})}{\eta}.$$

Combining this bound with (2.8) and (2.7) gives the Lemma. □

Other auxiliary results

Proposition 2.9. *If $\mathcal{H}$ consists of binary functions with VC dimension bounded by d, then $\mathcal{R}_n(\mathcal{H}) \leq C\sqrt{d/n}$ and $\mathcal{R}_{n,y}(\mathcal{H}) \leq C\sqrt{d/n_y}$ (conditionally on $n_y > 0$) for some universal, distribution-independent constant $C > 0$. If $\mathcal{H} = \mathsf{Stumps}$ consists of all stumps over $\mathbb{R}^p$ with coefficients in $[-1, 1]$, then $\mathcal{R}_n(\mathsf{Stumps}) \leq C\sqrt{\log p/n}$ and $\mathcal{R}_{n,y}(\mathsf{Stumps}) \leq C\sqrt{\log p/n_y}$ (conditionally on $n_y > 0$), with $C > 0$ universal.*

Proof of Proposition 2.9. The first statement is [BM02, Theorem 6, Lemma 4]. The second results from the following steps. Given a coordinate $j \in [p]$, use $x^{(j)}$ to denote the j-th coordinate of x. Let

Stumps_j denote the set of all functions of the form

$$x \in \mathbb{R}^p \mapsto a\mathbf{1}_{[x^{(j)} \leq \xi]} + b\mathbf{1}_{[x^{(j)} > \xi]}, \text{ with } a, b \in [-1, 1], \xi \in \mathbb{R}.$$

Each $f \in \mathsf{Stumps}_j$ is a convex combination of the 0 function and functions of the form $\pm 2\mathbf{1}_{[x^{(j)} \leq \xi]}$, $\pm 2\mathbf{1}_{[x^{(j)} > \xi]}$. For each j, each family $\{\mathbf{1}_{[x^{(j)} \leq \xi]}\} \cup \{\mathbf{1}_{[x^{(j)} > \xi]}\} \cup \{0\}$ comprises 0/1-valued functions with VC dimension bounded by an absolute constant. From [BM02, Theorem 6, Lemma 4], their Rademacher complexities are $O(1/\sqrt{n})$, which doesn't change when these functions are multiplied by 2. Moreover, passing to the convex hull does not change the Rademacher complexity, as shown in [BM02, Theorem 12, items 3 and 7]. We deduce that $\mathcal{R}_n(\mathsf{Stumps}_j) = O(1/\sqrt{n})$. Now,

$$\mathcal{R}_n(\mathsf{Stumps}) - \max_{j \in [p]} \mathcal{R}_n(\mathsf{Stumps}_j) \leq \mathbb{E} \max_{j \in [p]} \left[\sup_{h \in \mathsf{Stumps}_j} \frac{1}{n} \sum_{i=1}^n \sigma_i h(X_i) - \mathbb{E}\left(\sup_{h \in \mathsf{Stumps}_j} \frac{1}{n} \sum_{i=1}^n \sigma_i h(X_i) \right) \right].$$

The random variables inside the supremum in the RHS have zero mean. By McDiarmid's inequality [McD98], they are also sub-Gaussian with variance proxies $O(1/n)$. By [Ver18, Exercise 2.5.10], the expectation of the maximum satisfies:

$$\mathbb{E} \max_{j \in [p]} \left(\sup_{h \in \mathsf{Stumps}_j} \frac{1}{n} \sum_{i=1}^n \sigma_i h(X_i) - \mathcal{R}_n(\mathsf{Stumps}_j) \right) \leq C \sqrt{\frac{\log p}{n}}, \text{ with } C > 0 \text{ universal.}$$

This implies $\mathcal{R}_n(\mathsf{Stumps}) \leq C\sqrt{\log p / n}$, with a potentially larger (but still universal) $C > 0$. The bounds for $\mathcal{R}_{n,y}(\mathsf{Stumps})$ follow similarly once we condition on the number of examples with the two labels. □

2.4.2 Proof of Theorem 2.1 As we explain below, the proof is a direct application of Lemma 2.8 to the two distributions $\mathcal{D}_1 = \mathcal{D}_Z$ and $\mathcal{D}_0 = \mathcal{D}'_Z$ with $\mathcal{G} = \mathrm{conv}(\mathcal{H})$, with $\eta = \theta$. Importantly, the Rademacher complexities of $\mathcal{G}$ and $\mathcal{H}$ are equal [BM02, Theorem 12].

The only slightly subtle aspect in our argument, which will also come up in later proofs, is the following. We wish to control the probability of an event E given by "the inequality for AUC(S) in Theorem 2.1 holds for all S in the convex hull of $\mathcal{H}$." Now consider what happens when one conditions on specific (non-random) values $n_0 = m_0 > 0$ and $n_1 = m_1 = n - m_0 > 0$; that is, $m_0, m_1 = n - m_0$ are fixed (non-random) positive integers such that $\mathbb{P}(n_0 = m_0, n_1 = m_1) > 0$. Crucially, under this conditioning, the subsamples $\mathbf{X}_1 = \{X_i : y_i = 1\}$ and $\mathbf{X}_0\{X_i : y_i = 0\}$ corresponding to 1- and 0-labelled examples (respectively) are i.i.d. with respective laws $\mathcal{D}_1$ and $\mathcal{D}_0$, and independent from one another. Under this conditioning, Lemma 2.8 gives that E holds with probability $\geq 1 - \delta$. This is irrespective of the

choice of $m_0, m_1 = n - m_0 > 0$. Therefore, we discover that

$$\begin{aligned}\mathbb{P}(E \mid \min\{n_0, n_1\} > 0) = &\sum_{m_0=1}^{n-1} \mathbb{P}(E \mid n_0 = m_0, n_1 = n - m_0)\, \mathbb{P}(n_0 = m_0, n_1 = n - m_0 \mid \min\{n_0, n_1\} > 0) \\ \geq &1 - \delta.\end{aligned}$$

Remark 2.10. The same reasoning we gave above shows that, for any event E,

$$\mathbb{P}(E \mid \min\{n_0, n_1\} > 0) \geq \min\{\mathbb{P}(E \mid n_0 = m_0, n_1 = n - m_0), 1 \leq m_0 \leq n - 1\}.$$

Moreover, under the conditioning in the RHS, the subsamples $\mathbf{X}_1 = \{X_i : y_i = 1\}$ and $\mathbf{X}_0\{X_i : y_i = 0\}$ corresponding to 1- and 0-labelled examples (respectively) are i.i.d. with respective laws $\mathcal{D}_1$ and $\mathcal{D}_0$, and independent from one another. In later proofs, we will abuse notation slightly and compute $\mathbb{P}(E)$ assuming that n_0 and n_1 are fixed positive constants, as all bounds on $\mathbb{P}(E \mid n_0 = m_0, n_1 = n - m_0)$ we obtain are uniform in the choice of $0 < m_0 < n$.

2.4.3 Proof of Theorem 2.2 We want to prove that, with probability $\geq 1 - \delta$, conditionally on $\min\{n_0, n_1\} > 0$, for all S in the convex hull of $\mathcal{H}$,

$$\mathrm{KS}(S) \leq \widehat{\mathrm{KS}}_\theta(S) + \frac{8}{\theta}\zeta_{\mathrm{KS}}(\mathcal{H}) + \sqrt{\frac{\log(2/\delta)}{2}}\left(\frac{1}{\sqrt{n_0}} + \frac{1}{\sqrt{n_1}}\right), \tag{2.9}$$

where

$$\zeta_{\mathrm{KS}}(\mathcal{H}) = \mathcal{R}_{n_0,0}(\mathcal{H}) + \mathcal{R}_{n_1,1}(\mathcal{H}) + \frac{1}{\sqrt{n_0}} + \frac{1}{\sqrt{n_1}}.$$

To this end, we apply Lemma 2.6 from Section 2.4.1 to the two subsamples $\mathbf{X}_1$ and $\mathbf{X}_0$, with $\eta = \theta/2$, $\delta/2$ replacing δ, and $\mathcal{G} = \mathrm{conv}(\mathcal{H})$ equal to the convex hull of $\mathcal{H}$. As described in Remark 2.10 above, we abuse notation slightly and treat n_0, n_1 as fixed (non-random) positive integers in what follows; that is, n_0, n_1 represent specific values of these random variables. Under this (implicit) conditioning, the subsamples $\mathbf{X}_1 = \{X_i : y_i = 1\}$ and $\mathbf{X}_0\{X_i : y_i = 0\}$ corresponding to 1- and 0-labelled examples (respectively) are i.i.d. with respective laws $\mathcal{D}_1$ and $\mathcal{D}_0$, and independent from one another. Thus Lemma 2.6 indeed applies.

To continue, we recall that the Rademacher complexities of $\mathcal{G}$ and $\mathcal{H}$ are the same. same (cf. [BM02, Theorem 12]). Therefore, Lemma 2.6 allows us to deduce that, conditionally on specific values of $n_0, n_1 > 0$, with probability at least $1 - \delta$, the following two inequalities hold simultaneously for all

$S \in \mathrm{conv}(\mathcal{H})$ and $t \in \mathbb{R}$:

$$\mathbb{P}_{X\sim\mathcal{D}_1}\{S(X) \le t\} \le \frac{1}{n_1}\sum_{i\,:\,y_i=1} \mathbf{1}_{[S(X_i)\le t+\frac{\theta}{2}]} + \varepsilon_1,$$
$$\frac{1}{n_0}\sum_{i\,:\,y_i=0} \mathbf{1}_{[S(X_i)\le t-\frac{\theta}{2}]} \le \mathbb{P}_{X\sim\mathcal{D}_0}\{S(X) \le t\} + \varepsilon_0,$$

where, for $y = 0, 1$:

$$\varepsilon_y \coloneqq \frac{8\mathcal{R}_{n_y,y}(\mathcal{G}) + \frac{8}{\sqrt{n_y}}}{\theta} + \sqrt{\frac{\log(2/\delta)}{2n_y}}.$$

Now notice that, when these two inequalities hold, we also have

$$\begin{aligned}
\mathrm{KS}(S) - 1 &= \inf_{t\in\mathbb{R}} (\mathbb{P}_{X\sim\mathcal{D}_1}\{S(X) \le t\} - \mathbb{P}_{X\sim\mathcal{D}_0}\{S(X) \le t\}) \\
&\le \inf_{t\in\mathbb{R}} \left(\frac{1}{n_1}\sum_{i\,:\,y_i=1} \mathbf{1}_{[S(X_i)\le t+\frac{\theta}{2}]} - \frac{1}{n_0}\sum_{i\,:\,y_i=0} \mathbf{1}_{[S(X_i)\le t-\frac{\theta}{2}]} \right) + \varepsilon_0 + \varepsilon_1 \\
&= \widehat{\mathrm{KS}}_\theta(S) - 1 + \varepsilon_0 + \varepsilon_1,
\end{aligned}$$

which inspection reveals to be the same inequality as (2.9). Therefore, the probability of (2.9) holding is also at least $1 - \delta$ (conditionally on $n_0, n_1 > 0$).

2.4.4 Proof of Theorem 2.3 This proof is somewhat more complex than preceding examples. As before, let $\mathcal{G} \coloneqq \mathrm{conv}(\mathcal{H})$ denote the convex hull of $\mathcal{H}$. We will use below that the Rademacher complexities of $\mathcal{G}$ and $\mathcal{H}$ are always equal.

For convenience, we define

$$\Gamma \coloneqq \sup_{S\in\mathcal{G}} \left(\frac{1}{n_1}\sum_{i\,:\,y_i=1} \mathbf{1}_{[S(X_i)\ge \widehat{t}_\alpha(S)+\theta]} - \mathbb{P}_{X\sim\mathcal{D}_1}\{S(X) \ge t_\alpha(S)\} \right), \tag{2.10}$$

so that we can write, for any $S \in \mathcal{G}$:

$$\begin{aligned}
\mathrm{P@k}(S) - \widehat{\mathrm{P@k}}_\theta(S) &\le \mathbb{P}_{(X,y)\sim\mathcal{D}}\{y = 1\}\,\Gamma && (2.11)\\
&\quad + \left(\frac{1}{n_1}\sum_{i\,:\,y_i=1} \mathbf{1}_{[S(X_i)\ge \widehat{t}_\alpha(S)+\theta]} \right) \left(\frac{n_1}{n} - \mathbb{P}_{(X,y)\sim\mathcal{D}}\{y = 1\} \right) && (2.12)\\
&\le \mathbb{P}_{(X,y)\sim\mathcal{D}}\{y = 1\}\,\Gamma + \max\left\{ \left(\frac{n_1}{n} - \mathbb{P}_{(X,y)\sim\mathcal{D}}\{y = 1\} \right), 0 \right\}. && (2.13)
\end{aligned}$$

If we define an event,

$$C = \left\{ \frac{n_1}{n} \leq \mathbb{P}_{(X,y)\sim\mathcal{D}}\{y = 1\} + \sqrt{\frac{\log(3/(\delta - \delta^2))}{2n}} \right\}, \tag{2.14}$$

it is clear that $\mathbb{P}(C) \geq 1 - \delta/3 + \delta^2/3$ due to a simple application of Hoeffding's inequality. Since $\mathbb{P}(\min\{n_0, n_1\} > 0) \geq 1 - \delta$,

$$\mathbb{P}(C \mid \min\{n_0, n_1\} > 0) \geq 1 - \frac{\mathbb{P}(C^c)}{\mathbb{P}(\min\{n_0, n_1\} > 0)} \geq 1 - \delta/3.$$

Now consider another event D defined as follows: either $\min\{n_0, n_1\} = 0$, or

$$\Gamma \leq \frac{\overline{\eta}_n(\mathcal{H})}{\mathbb{P}_{(X,y)\sim\mathcal{D}}\{y = 1\}} + \frac{4\mathcal{R}_{n_1,1}(\mathcal{H}) + \frac{4}{\sqrt{n_1}}}{\theta - 2\bar{\eta}_n(\mathcal{H})} + \sqrt{\frac{\log(3/\delta)}{2n_1}}. \tag{2.15}$$

We see from the above that, if $D \cap C$ holds, then (2.13) implies that either $\min\{n_0, n_1\} = 0$, or the inequality on $\mathrm{P@k}(S) - \widehat{\mathrm{P@k}}_\theta(S)$ claimed in the statement of the Theorem holds for all $S \in \mathcal{G}$. Therefore, we will be done once we show that $\mathbb{P}(D \cap C \mid \min\{n_0, n_1\} > 0) \geq 1 - \delta$. In fact, since $\mathbb{P}(C \mid \min\{n_0, n_1\} > 0) \geq 1 - \delta/3$, it suffices to show $\mathbb{P}(D \mid \min\{n_0, n_1\} > 0) \geq 1 - 2\delta/3$. This will be our goal for the remainder of the proof.

To continue, we define a third event which we use to control $t_\alpha(S)$, $\widehat{t}_\alpha(S)$ and related quantities. Define

$$E \coloneqq \left\{ \forall S \in \mathcal{G},\, \forall t \in \mathbb{R} : \mathbb{P}_{X\sim\mathcal{D}}\{S(X) \geq t\} \leq \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{[S(X_i)\geq t - \bar{\eta}_n(\mathcal{H})]} + \bar{\eta}_n(\mathcal{H}) \right\}. \tag{2.16}$$

This is the kind of event controlled by Corollary 2.7, except that we have $S(X_i) \geq t$ and $S(X) \geq t - \theta$ as opposed to "$\leq$" inequalities. However, the corollary still applies if we consider the functions $-S$ as S ranges over $\mathcal{G}$. This is tantamount to applying the corollary to the family of functions $-\mathcal{G} = \{-S : S \in \mathcal{G}\}$. Since $-\mathcal{G}$ has the same Rademacher complexity as $\mathcal{G}$ and $\mathcal{H}$, we obtain $\mathbb{P}(E) \geq 1 - \delta/3 + \delta^2/3$. As noted in the case of C, we obtain that $\mathbb{P}(E \mid \min\{n_0, n_1\} > 0) \geq 1 - \delta/3$.

We now claim the following.

Claim. *When E holds,*

$$\mathbb{P}_{X\sim\mathcal{D}_1}\{S(X) \geq t_\alpha(S)\} \geq \mathbb{P}_{X\sim\mathcal{D}_1}\{S(X) \geq \widehat{t}_\alpha(S) + 2\bar{\eta}_n(\mathcal{H})\} - \frac{\bar{\eta}_n(\mathcal{H})}{\mathbb{P}_{(X,y)\sim\mathcal{D}}\{y = 1\}}. \tag{2.17}$$

Indeed, the claim is trivial if $t_* :- \widehat{t}_\alpha(S) + 2\bar{\eta}_n(\mathcal{H}) \geq t_\alpha(S)$. Otherwise,

$$\begin{aligned}\mathbb{P}_{X\sim\mathcal{D}_1}\{S(X) \geq t_*\} - \mathbb{P}_{X\sim\mathcal{D}_1}\{S(X) \geq t_\alpha(S)\} &= \frac{\mathbb{P}_{(X,y)\sim\mathcal{D}}\{y=1, t_* \leq S(X) < t_\alpha(S)\}}{\mathbb{P}_{(X,y)\sim\mathcal{D}}\{y=1\}} \\ &\leq \frac{\mathbb{P}_{(X,y)\sim\mathcal{D}}\{t_* \leq S(X) < t_\alpha(S)\}}{\mathbb{P}_{(X,y)\sim\mathcal{D}}\{y=1\}}\end{aligned}$$

Since we know from the definition of $t_\alpha(S)$ that $\mathbb{P}_{(X,y)\sim\mathcal{D}}\{t_\alpha(S) \leq S(X)\} \geq \alpha$, we will be done if we show $\mathbb{P}_{X\sim\mathcal{D}}\{S(X) \geq t_*\} \leq \alpha + \bar{\eta}_n(\mathcal{H})$ whenever D holds. To do this, take $t = t_*$ in the definition of E. Since $t - \bar{\eta}_n(\mathcal{H}) > \widehat{t}_\alpha(S)$, and the latter is a $(1-\alpha)$-quantile for S, under the sample distribution, we obtain that:

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{[S(X_i) \geq t_* - \bar{\eta}_n(\mathcal{H})]} \leq \alpha,$$

and so, when E holds,

$$\mathbb{P}_{X\sim\mathcal{D}}\{S(X) \geq t_*\} \leq \alpha + \bar{\eta}_n(\mathcal{H}).$$

This gives us the claim.

To continue, we go back to the definition of Γ in (2.10) and notice that, by the Claim, when E holds,

$$\Gamma \leq \frac{\bar{\eta}_n(\mathcal{H})}{\mathbb{P}_{(X,y)\sim\mathcal{D}}\{y=1\}} + \Gamma^*,$$

where we define

$$\Gamma^* \coloneqq \sup_{S\in\mathcal{G}, t\in\mathbb{R}} \left(\frac{1}{n_1} \sum_{i\,:\,y_i=1} \mathbf{1}_{[S(X_i)\geq t]} - \mathbb{P}_{X\sim\mathcal{D}_1}\{S(X) \geq t - (\theta - 2\bar{\eta}_n(\mathcal{H}))\} \right).$$

Recall that our goal is to show that the probability $\mathbb{P}(D \mid \min\{n_0, n_1\} > 0)$ above is at least $1 - 2\delta/3$. By the above reasoning, we see that $D \supset E \cap F$, where

$$F \coloneqq \left\{ \Gamma^* \leq \frac{4\mathcal{R}_{n_1,1}(\mathcal{H}) + \frac{4}{\sqrt{n_1}}}{\theta - 2\bar{\eta}_n(\mathcal{H})} + \sqrt{\frac{\log(3/\delta)}{2n_1}} \right\}.$$

Since we know already that $\mathbb{P}(E \mid \min\{n_0, n_1\} > 0) \geq 1 - \delta/3$, we will be done once we show that $\mathbb{P}(F \mid \min\{n_0, n_1\} > 0) \geq 1 - \delta/3$, which (as seen above) will follow from $\mathbb{P}(F) \geq 1 - \delta/3 + \delta^2/3$.

At this last step, we will apply the reasoning in Remark 2.10 above: that is, we treat n_0 and n_1 as fixed constants and the subsamples $\mathbf{X}_0, \mathbf{X}_1$ as i.i.d. and independent. Under this (implicit) conditioning, Γ^* is almost the kind of quantity to which Lemma 2.6 applies, with $\eta = \theta - 2\bar{\eta}_n(\mathcal{H}) > 0$. The differences

one notices is that there is a minus sign in front of η, and there are " $\geq$" signs where "$\leq$" should be. As we observed following (2.16), one can circumvent this by applying the Lemma to $-\mathcal{G}$. If we do that (with $\delta/3 - \delta^2/3$ replacing δ), we obtain that the event satisfies $\mathbb{P}(F) \geq 1 - \delta/3 + \delta^2/3$, as desired.

2.4.5 Proof of Proposition 2.4 The idea of the proof is that any observation from the original sample is close (the precise meaning of this statement will be defined below) to some observation in the subsample with high probability. Moreover, the better is such approximation, the lower is the impact on the minimization of the target loss.

First, consider $\widehat{L} = \widehat{\mathrm{KS}}$. For each $j \in [p]$, let

$$f_{j,\leq}(a,\xi) \coloneqq \sum_{i=1}^{n} \rho_i \, \mathbf{1}_{[a \leq t_i, X_{i,(j)} \leq \xi]};$$
$$f_{j,>}(b,\xi) \coloneqq \sum_{i=1}^{n} \rho_i \, \mathbf{1}_{[b \leq t_i, X_{i,(j)} > \xi]},$$

Note that our objective function $\widehat{\mathrm{KS}}(S+h)$ is one minus the sum of $f_{j,\leq}$ and $f_{j,>}$ where h is a stump with parameters (a,b,j,ξ). Hence, our problem is equivalent to maximizing $f_{j,\leq} + f_{j,>}$. We also use $t_i = \widehat{t}(S) - S(X_i)$. For convenience, we assume the sample has been ordered so that $t_1 \leq t_2 \leq \cdots \leq t_n$.

Now imagine $a = t_i$ is changed to $a' = t_{i'}$ with $i' \leq i$. Notice that:

$$f_{j,\leq}(t_i,\xi) - f_{j,\leq}(t_{i'},\xi) = \sum_{\ell=i'}^{i-1} \rho_i \, \mathbf{1}_{[X_{i,(j)} \leq \xi]} \in \left[-\frac{\mathrm{pos}(i,i')}{n_1}, \frac{\mathrm{neg}(i,i')}{n_0} \right],$$

where $\mathrm{pos}(i,i')$ and $\mathrm{neg}(i,i')$ count the number of positive and negative examples between t_i and $t_{i'}$, including the largest of the two extreme points (these are well-defined even if $i' > i$). Therefore,

$$\|f_{j,\leq}(t_i,\xi) - f_{j,\leq}(t_{i'},\xi)\| \leq \max\left\{ \frac{\mathrm{pos}(i,i')}{n_1}, \frac{\mathrm{neg}(i,i')}{n_0} \right\} \tag{2.18}$$

If we like, we can say that the above implies that $f_{j,\leq}(t_i,\xi)$ is a 1-Lipschitz function of i in the pseudometric:

$$d(i,i') \coloneqq \max\left\{ \frac{\mathrm{pos}(i,i')}{n_1}, \frac{\mathrm{neg}(i,i')}{n_0} \right\}.$$

A similar property holds for the $f_{>,j}$ function.

Now let (a_*, b_*, j_*, ξ_*) be the parameters of the optimal h_*. Say $a_* = t_{i_*}$ and $b_* = t_{j_*}$ for indices $i_*, j_* \in [n]$. We consider a modified function $\tilde{h}$ where a_*, b_* are replaced by points $t_{\tilde{i}}, t_{\tilde{j}}$ with $\tilde{i}, \tilde{j} \in I$

chosen to minimize $d(i_*, \tilde{i}) + d(j_*, \tilde{j})$. Notice that:

$$\widehat{\mathrm{KS}}(S + h_R) \leq \widehat{\mathrm{KS}}(S + \tilde{h})$$

because $\tilde{h}$ is feasible for the optimization problem of which h_R achieves the minimum. Therefore,

$$\begin{aligned}
\mathbb{E}[\widehat{\mathrm{KS}}(S + h_R)] &\leq \mathbb{E}[\widehat{\mathrm{KS}}(S + \tilde{h})] \\
&\leq \widehat{\mathrm{KS}}(S + h_*) - \mathbb{E}[\widehat{\mathrm{KS}}(S + h_*) - \widehat{\mathrm{KS}}(S + \tilde{h})] \\
&\leq \widehat{\mathrm{KS}}(S + h_*) + \mathbb{E}[d(i_*, \tilde{i}) + d(j_*, \tilde{j})],
\end{aligned}$$

where the last step uses the Lipschitz property.

To finish, we bound the expected distances in the RHS.

Let $\ell \in \mathbb{R}$. Suppose there are at least $\lfloor \ell n_1 \rfloor$ positive examples to the right of t_{i_*}, denoted $t_{i_1}, \ldots, t_{i_{\lfloor \ell n_1 \rfloor}}$, and at least $\lfloor \ell n_0 \rfloor$ negative examples to the right of t_{i_*}, denoted $t_{j_1}, \ldots, t_{j_{\lfloor \ell n_0 \rfloor}}$. If $t_{i_{\lfloor \ell n_1 \rfloor}} \leq t_{j_{\lfloor \ell n_0 \rfloor}}$, then for any $k \leq \lfloor \ell n_1 \rfloor$ with $i_k \in I_1$, we have $d(i_*, \tilde{i}) \leq \ell$. To see this, note that

$$d(i_*, \tilde{i}) \leq d(i_*, i_k) = \max\left\{ \frac{\mathrm{pos}(i_*, i_k)}{n_1}, \frac{\mathrm{neg}(i_*, i_k)}{n_0} \right\} \leq \max\left\{ \frac{\ell n_1}{n_1}, \frac{\ell n_0}{n_0} \right\}.$$

Then,

$$\begin{aligned}
\mathbb{P}\big[d(i_*, \tilde{i}) > \ell\big] &\leq \mathbb{P}\Big[I_1 \cap \{t_{i_1}, \ldots, t_{i_{\lfloor \ell n_1 \rfloor}}\} = \emptyset\Big] \\
&\leq \left(1 - \frac{\lfloor \ell n_1 \rfloor}{n_1}\right)^k \leq \exp\left(-k \frac{\lfloor \ell n_1 \rfloor}{n_1}\right) \\
&\leq \exp\left(-k\left(\frac{\ell n_1}{n_1} - \frac{1}{n_1}\right)\right) = \exp\left(-k\ell\right) \exp\left(k/n_1\right).
\end{aligned}$$

Note that the same reasoning works even if there are less than $\lfloor \ell n_1 \rfloor$ positive examples.

Similarly, if $t_{i_{\lfloor \ell n_1 \rfloor}} > t_{j_{\lfloor \ell n_0 \rfloor}}$ and some $i_k \in I_0$ for $k \leq \lfloor \ell n_0 \rfloor$, $\mathbb{P}\big[d(i_*, \tilde{i}) > \ell\big] \leq \exp\left(-k\ell\right) \exp\left(k/n_0\right)$. Then

$$\mathbb{E}[d(i_*, \tilde{i})] \leq \int_0^\infty \mathbb{P}\left[d(i_*, \tilde{i}) > \ell\right] d\ell \leq \max\left\{\mathrm{e}^{k/n_1}, \mathrm{e}^{k/n_0}\right\} \int_0^\infty \mathrm{e}^{-k\ell} d\ell = \frac{\max\left\{\mathrm{e}^{k/n_1}, \mathrm{e}^{k/n_0}\right\}}{k}$$

And if $k \leq \min\{n_1, n_0\}$, we bound

$$\mathbb{E}[d(i_*, \tilde{i})] \leq \frac{\mathrm{e}}{k},$$

and we are done.

Now, let $\widehat{L} = \widehat{\text{AUC}}$. We'll apply the same strategy as above. As with the KS loss, the optimal stump coefficients can be searched on a finite set. In this case, we have $\{t_{ij} : t_{ij} = S(X_i) - S(X_j) \text{ with } i, j \in [n]\}$. For ease of calculation, consider some stump $h(X) = t_{pq}\mathbf{1}_{[X_{(m)} \leq \xi]}$. Then,

$$\begin{aligned}
\widehat{\text{AUC}}(S + h) &= 1 - \frac{1}{n_0 n_1} \sum_{\{i:y_i=1\}} \sum_{\{j:y_j=0\}} \mathbf{1}_{[S(X_i)+h(X_i) > S(X_j)+h(X_j)]} \\
&= 1 - \sum_{i=1}^{n} \sum_{j=1}^{n} \rho_{ij} \mathbf{1}_{[t_{ij}+h(X_i)-h(X_j)>0]} \\
&= 1 - \sum_{i=1}^{n} \sum_{j=1}^{n} \rho_{ij} \mathbf{1}_{[t_{ij}+h(X_i)-h(X_j)>0]}
\end{aligned}$$

where $\rho_{ij} = \frac{1}{n_0 n_1} \mathbf{1}_{[y_i=1]} \mathbf{1}_{[y_j=0]}$. Note that

$$h(X_i) - h(X_j) = t_{pq}(\mathbf{1}_{[X_{i,(m)} \leq \xi]} - \mathbf{1}_{[X_{j,(m)} \leq \xi]}).$$

If t_{pq} is changed to some $t_{p'q'} \leq t_{pq}$ so that $h'(X) = t_{p'q'}\mathbf{1}_{[X_{(m)} \leq \xi]}$, we have

$$\mathbf{1}_{[t_{ij}+h(X_i)-h(X_j)>0]} - \mathbf{1}_{[t_{ij}+h'(X_i)-h'(X_j)>0]} = \begin{cases} \mathbf{1}_{[t_{ij}+t_{pq}>0]} - \mathbf{1}_{[t_{ij}+t_{p'q'}>0]}, \text{ if } X_{i,(m)} \leq \xi < X_{j,(m)} \\ \mathbf{1}_{[t_{ij}-t_{pq}>0]} - \mathbf{1}_{[t_{ij}-t_{p'q'})>0]}, \text{ if } X_{i,(m)} > \xi \geq X_{j,(m)} \\ 0, \text{ if } X_{i,(m)} \leq \xi, \text{ and } X_{j,(m)} \leq \xi \\ 0, \text{ if } X_{i,(m)} > \xi, \text{ and } X_{j,(m)} > \xi \end{cases}$$

Therefore,

$$\widehat{\text{AUC}}(S + h') - \widehat{\text{AUC}}(S + h) \in \left[-\frac{\#J_-((p,q),(p',q'))}{n_0 n_1}, \frac{\#J_+((p,q),(p',q'))}{n_0 n_1} \right]$$

where

$$\begin{aligned}
J_-((p,q),(p',q')) &= \{(i,j) : y_i = 1,\ y_j = 0,\ -t_{p'q'} > t_{ij} > -t_{pq}\} \\
J_+((p,q),(p',q')) &= \{(i,j) : y_i = 1,\ y_j = 0,\ t_{p'q'} < t_{ij} < t_{pq}\}.
\end{aligned}$$

Therefore,

$$\left\|\widehat{\mathrm{AUC}}(S+h) - \widehat{\mathrm{AUC}}(S+h')\right\| \leq \max\left\{\frac{\#J_-((p,q),(p',q'))}{n_0 n_1}, \frac{\#J_+((p,q),(p',q'))}{n_0 n_1}\right\}$$
$$\leq \frac{\#J((p,q),(p',q'))}{n_0 n_1},$$

where $J((p,q),(p',q')) = J_-((p,q),(p',q')) \cup J_+((p,q),(p',q'))$. The rest of the proof follows the same strategy used in the $\widehat{\mathrm{KS}}$ loss, replacing the pseudometric d with $\tilde{d}$, where

$$\tilde{d}((p,q),(p',q')) = \frac{\#J((p,q),(p',q'))}{n_0 n_1}.$$

Now let the optimal stump be $h_*(x) = t_{p_*q_*}\mathbf{1}_{[x_{(m_*)} \leq \xi_*]}$ and, again, consider a modified function $\tilde{h}$ where $t_{p_*q_*}$ is replaced by a point $t_{\tilde{p}\tilde{q}}$ with $\tilde{p}, \tilde{q} \in I$ chosen to minimize $\tilde{d}((p_*,q_*),(\tilde{p},\tilde{q}))$. Recall that the optimal stump over the reduced sample, h_R, satisfies

$$\widehat{\mathrm{AUC}}(S+h_R) \leq \widehat{\mathrm{AUC}}(S+\tilde{h})$$

and therefore,

$$\mathbb{E}\Big[\widehat{\mathrm{AUC}}(S+h_R)\Big] \leq \mathbb{E}\Big[\widehat{\mathrm{AUC}}(S+\tilde{h})\Big]$$
$$\leq \widehat{\mathrm{AUC}}(S+h_*) - \mathbb{E}\Big[\widehat{\mathrm{AUC}}(S+h_*) - \widehat{\mathrm{AUC}}(S+\tilde{h})\Big]$$
$$\leq \widehat{\mathrm{AUC}}(S+h_*) + \mathbb{E}\Big[\tilde{d}((p_*,q_*),(\tilde{p},\tilde{q}))\Big]$$

And finally, we bound the expected distance on the RHS. Let $\ell \in \mathbb{R}$. Suppose there are at least $r = \lfloor \ell n_1 n_0 \rfloor$ pairs $(i,j) \in J((p,q),(p',q'))$ such that $t_{ij} \leq t_{p_*q_*}$, denoted $t_{i_1j_1}, \ldots, t_{i_rj_r}$. Then, $\tilde{d}((p_*,q_*),(\tilde{p},\tilde{q})) \leq \ell$. To verify this, note that for any pair (p,q) with $t_{pq} \leq t_{i_rj_r}$ such that $p \in I_1, q \in I_0$, we have

$$\tilde{d}((p_*,q_*),(\tilde{p},\tilde{q})) \leq \tilde{d}((p_*,q_*),(p,q))$$
$$= \frac{\#J((p_*,q_*),(p,q))}{n_0 n_1}$$
$$\leq \frac{\ell n_1 n_0}{n_1 n_0} = \ell$$

Moreover, note that $r \leq r_1 r_0$ where r_1 is the number of distinct indices i_s, $s \leq r$, such that $y_{i_s} = 1$

and r_0 is the number of distinct indices j_s, $s \leq r$, such that $y_{j_s} = 0$. Then,

$$\begin{aligned}
\mathbb{P}\left[\tilde{d}((p_*, q_*), (\tilde{p}, \tilde{q})) > \ell\right] &\leq \mathbb{P}\left[I_1 \times I_0 \cap \{(i_1, j_1), \ldots, (i_r, j_r)\} = \emptyset\right] \\
&\leq \left(1 - \frac{r_1}{n_1}\right)^k \left(1 - \frac{r_0}{n_0}\right)^k \\
&\leq \left(1 - \frac{r_1 r_0}{n_1 n_0}\right)^k \\
&\leq \left(1 - \frac{\lfloor \ell n_0 n_1 \rfloor}{n_0 n_1}\right)^k \\
&\leq \exp\left(-k \frac{\lfloor \ell n_0 n_1 \rfloor}{n_0 n_1}\right) \\
&\leq \exp\left(-k \left(\frac{\ell n_0 n_1}{n_0 n_1} - \frac{1}{n_0 n_1}\right)\right) \\
&= \exp(-k\ell) \exp(k/(n_0 n_1))
\end{aligned}$$

where the third inequality follows from the fact that $r_0 \leq n_0$ and $r_1 \leq n_1$. Then,

$$\begin{aligned}
\mathbb{E}\left[\tilde{d}((p_*, q_*), (\tilde{p}, \tilde{q}))\right] &\leq \int_0^\infty \mathbb{P}\left[\tilde{d}((p_*, q_*), (\tilde{p}, \tilde{q})) > \ell\right] d\ell \\
&\leq \mathrm{e}^{k/(n_0 n_1)} \int_0^\infty \mathrm{e}^{-k\ell} d\ell = \frac{\mathrm{e}^{k/(n_0 n_1)}}{k} \leq \frac{\mathrm{e}}{k}.
\end{aligned}$$

2.4.6 Proof of Proposition 2.5 For any loss $\widehat{L}$, ExactBoost obtains a sequence of score functions with decreasing values of $\widehat{L}$. Therefore, the loss of S_* is upper bounded by that of $S_{*,1}$, the stump function obtained in the first round of ExactBoost.

Now take any $1 \leq m \leq M$ and $t \in \mathbb{R}$ and consider the stump function $h_{m,t} : \mathbb{R}^M \to \mathbb{R}$, defined via $h_{m,t}(z) = \mathbf{1}_{[z^{(m)} \geq t]}$, where $z^{(m)}$ denotes the mth entry of z. Since $S_{*,1}$ has the smallest loss over training data of all stumps, for all $t \in \mathbb{R}$ and $1 \leq m \leq M$, it holds that:

$$\widehat{L}_{(Z_i, y_i)_{i=1}^n}(S_*) \leq \widehat{L}_{(Z_i, y_i)_{i=1}^n}(S_{*,1}) \leq \widehat{L}_{(Z_i, y_i)_{i=1}^n}(h_{m,t}). \tag{2.19}$$

The remainder of the proof consists of applying (2.19) judiciously. First, consider the $\widehat{\mathrm{KS}}$ loss. To estimate the $\widehat{\mathrm{KS}}$ loss for $h_{m,t}$, let n_0, n_1 denote the numbers of 0- and 1-labelled examples in (X_i, y_i). Then

$$\widehat{\mathrm{KS}}_{(Z_i, y_i)_{i=1}^n}(h_{m,t}) = \inf_{s \in \mathbb{R}} \left(\frac{1}{n_1} \sum_{i: y_i = 1} \mathbf{1}_{[h_{m,t}(Z_i) \leq s]} + \frac{1}{n_0} \sum_{i: y_i = 0} \mathbf{1}_{[h_{m,t}(Z_i) > s]} \right).$$

In particular, taking the specific value $s = 0$ instead of the infimum in the right-hand side gives an upper bound for the $\widehat{\mathrm{KS}}$ losses of S_*, $S_{*,1}$ and $h_{m,t}$. Since $\mathbf{1}_{[h_{m,t}(Z_i) \leq 0]} = \mathbf{1}_{[S_m(X_i) \leq t]}$ and $\mathbf{1}_{[h_{m,t}(Z_i) > 0]} =$

$\mathbf{1}_{[S_m(X_i)>t]}$, from (2.19) it follows that, for all $t \in \mathbb{R}$ and $1 \leq m \leq M$,

$$\widehat{\mathrm{KS}}_{(Z_i,y_i)_{i=1}^n}(S_*) \leq \frac{1}{n_1} \sum_{i:y_i=1} \mathbf{1}_{[S_m(X_i)\leq t]} + \frac{1}{n_0} \sum_{i:y_i=0} \mathbf{1}_{[S_m(X_i)>t]}.$$

Minimizing the right-hand side over t for a given m shows that

$$\widehat{\mathrm{KS}}_{(Z_i,y_i)_{i=1}^n}(h_{m,t}) \leq \widehat{\mathrm{KS}}_{(X_i,y_i)_{i=1}^n}(S_m),$$

and taking the minimum over m finishes the proof in the case $\widehat{L} = \widehat{\mathrm{KS}}$.

Now consider the metric $\widehat{\mathrm{P@k}}$. For each $1 \leq m \leq M$, let $\widehat{t}_\alpha(S_m)$ denote the $(1-\alpha)$-quantile of the score S_m on the dataset $(X_i, y_i)_{i=1}^n$. Apply (2.19) to each m and to values $t < \widehat{t}_\alpha(S_m)$. To compute $\widehat{\mathrm{P@k}}_{(Z_i,y_i)}(h_{m,t})$, note that, for $0 \leq s < 1$, $h_{m,t}(Z_i) \leq s$ if and only if $Z_i^{(m)} = S_m(X_i) < t$. Since t is smaller than the $(1-\alpha)$-quantile, for all $0 \leq s < 1$:

$$\frac{1}{n} \sum_{i=1}^n \mathbf{1}_{[h_{m,t}(Z_i)\leq s]} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{[S_m(X_i)\leq t]} < 1 - \alpha.$$

Since $h_{m,t}$ takes binary values, the $(1-\alpha)$-quantile of the vector $(h_{m,t}(Z_i))_{i=1}^n$ is 1, and from (2.19) it follows that for any $1 \leq m \leq M$ and $t < \widehat{t}_\alpha(S_m)$,

$$\widehat{\mathrm{P@k}}_{(Z_i,y_i)_{i=1}^n}(S_*) \leq \widehat{\mathrm{P@k}}_{(Z_i,y_i)_{i=1}^n}(h_{m,t}) = 1 - \frac{1}{n_1} \sum_{i\,:\,y_i=1} \mathbf{1}_{[h_{m,t}(Z_i)\geq 1]} = 1 - \frac{1}{n_1} \sum_{i\,:\,y_i=1} \mathbf{1}_{[S_m(X_i)\geq t]}.$$

When $t \nearrow \widehat{t}_\alpha(S_m)$, it holds that $\mathbf{1}_{[S_m(X_i)\geq t]} \to \mathbf{1}_{[S_m(X_i)\geq \widehat{t}_\alpha(S_m)]}$, so, for all $1 \leq m \leq M$,

$$\widehat{\mathrm{P@k}}_{(Z_i,y_i)_{i=1}^n}(S_*) \leq 1 - \frac{1}{n_1} \sum_{i\,:\,y_i=1} \mathbf{1}_{[S_m(X_i)\geq \widehat{t}_\alpha(S_m)]} = \widehat{\mathrm{P@k}}_{(X_i,y_i)_{i=1}^n}(S_m).$$

Minimizing the right-hand side over m finishes the proof.

Chapter 3

Learning to hash via boosting

Submission in preparation. We thank Lucas Nissenbaum, Alex Akira Okuno and Rodrigo Schuller for help with the experiments.

3.1 Introduction

Given databases $\mathcal{A} := \{A_\ell\}_{\ell=1}^{N_\mathcal{A}}$ and $\mathcal{B} := \{B_r\}_{r=1}^{N_\mathcal{B}}$, such that $\mathcal{A}, \mathcal{B} \subset \mathcal{X}$ and a notion of similarity between items $\sim_R$, we want to find pairs (A_ℓ, B_r) such that $A_\ell \sim_R B_r$. For example, A_ℓ and B_r may correspond to the same person in different databases. Our goal is to build a hash table for these items so that, given an item A_ℓ, one can find $B_r \sim_R A_\ell$ with as few table lookups as possible. To this end, we suppose we have access to a training sample,

$$\mathcal{S}_{\text{train},n} := \{((a_i, b_i), y_i) \in \mathcal{A} \times \mathcal{B} \times \{-1, 1\}, i \in [n]\}$$

such that, $y_i = 1$ if $a_i \sim_R b_i$ and -1 otherwise. This sample will be used in a training stage so our algorithm can learn a similarity classifier via a sample of similar/dissimilar items, via boosting and margin maximization techniques.

Our method will consist of two stages that are described in what follows.

§3.1.1 Learn functions $\{k_t^*\}_{t=1}^T$ from a family of binary classifiers $\mathcal{K}$ over $\mathcal{X}$, and convex weights $\{\alpha_t\}_{t=1}^T$ for these classifiers. We do this via a variant of the AdaBoost algorithm, where at each step we minimize a weighted average of $y_i k_t^*(a_i) k_t^*(b_i)$ over the training sample.

§3.1.2 Obtain, from the functions and weights from item 1, one-bit hash functions that correlate with our similarity relation. Build the hash code via these functions.

3.1.1 Learning classifiers and weights via boosting Fix a family $\mathcal{K}$ of binary classifiers $k : \mathcal{X} \to \{-1, +1\}$. To find the functions $\{k_t^*\}_{t=1}^T \in \mathcal{K}$ and the convex weights $\{\alpha_t\}_{t=1}^T$ we use the following adaptation of the AdaBoost algorithm over our training sample $\mathcal{S}_{\text{train},n} = ((a_i, b_i), y_i)_{i=1}^n$:

Algorithm 4 Boosting algorithm

Require: $\mathcal{S}_{\text{train},n} = ((a_i, b_i), y_i)_{i=1}^n$, number of iterations $T \in \mathbb{N}$, binary family $\mathcal{K}$

1: **for** $i \leftarrow 1$ to n **do**
2: $\quad Q_1(i) \leftarrow \frac{1}{n}$
3: **end for**
4: **for** $t \leftarrow 1$ to T **do**
5: $\quad k_t^* \leftarrow$ classifier in $\mathcal{K}$ with error $\varepsilon_t = \sum_{i=1}^n Q_t(i)\mathbf{1}_{[y_i k_t^*(a_i)k_t^*(b_i)<0]} < 1/2$
6: $\quad \alpha_t' \leftarrow \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) > 0$
7: $\quad Z_t \leftarrow 2\left[\varepsilon_t(1-\varepsilon_t)\right]^{1/2}$
8: $\quad$ **for** $i \leftarrow 1$ to n **do**
9: $\quad\quad Q_{t+1}(i) \leftarrow \frac{Q_t(i)\exp(-\alpha_t' y_i k_t^*(a_i)k_t^*(b_i))}{Z_t}$
10: $\quad$ **end for**
11: **end for**
12: **for** $t \leftarrow 1$ to T **do**
13: $\quad \alpha_t^* \leftarrow \frac{\alpha_t'}{\sum_{s=1}^T \alpha_s'}$
14: **end for**
15: **return** $(\alpha_t^*)_{t=1}^T$, $(k_t^*)_{t=1}^T$

Line 5 of the algorithm requires that one always find a classifier k_t^* such that

$$\varepsilon_t = \sum_{i=1}^n Q_t(i)\mathbf{1}_{[y_i k_t^*(a_i)k_t^*(b_i)<0]} < 1/2,$$

but in practice, one can add a stopping condition if it is not possible to find such a classifier. The main difference with ordinary AdaBoost is that we are optimizing a function that is *quadratic* over the choice of classifier. This is crucial for the second stage of the algorithm, described in §3.1.2. Importantly, there are simple and effective classifier families for which this quadratic optimization problem is feasible. One example is stump functions as defined in (2.3): for $x \in \mathbb{R}^d$, if $X_{(j)}$ indicates the projection of $X \in \mathbb{R}^d$ in the j coordinate, then

$$\mathcal{H} = \left\{\pm\mathbf{1}_{[X_{(j)}\le\xi]} \pm \mathbf{1}_{[X_{(j)}>\xi]} \ : \ \xi \in \mathbb{R}, j \in [p]\right\},$$

The intuition for our procedure is similar to that of AdaBoost. We expect that given items $A \in \mathcal{A}$ and

$B \in \mathcal{B}$, the function

$$f^*(A,B) = \sum_{t=1}^{T} \alpha_t^* k_t^*(A) k_t^*(B). \tag{3.1}$$

to be a good measure of similarity between A and B so that $f^*(A,B)$ is large and positive when $A \sim_R B$, and large and negative otherwise. Notice that larger weights α_t are given to the classifiers k_t^* that achieve the smallest values of ε_t. That is, our algorithm naturally gives more weight to functions $k_t^*(A)k_t^*(B)$ that correlate more strongly with the similarity relation.

3.1.2 Building the hash tables We now use the convex weights $(\alpha_t^*)_{t=1}^T$ and the functions $(k_t^*)_{t=1}^T$, to construct hash functions that correlate with our similarity relation. We do this via the following algorithm:

Algorithm 5 Hash algorithm

Require: $k, L \in \mathbb{N}$, convex weights$(\alpha_t^*)_{t=1}^T$, classifiers $(k_t^*)_{t=1}^T$

1: **for** $i \leftarrow 1$ to L **do**
2: **for** $j \leftarrow 1$ to k **do**
3: $g_{i,j} \leftarrow k_t^*$ with probability α_t^*
4: **end for**
5: $g_i \leftarrow (g_{i,1}, \ldots, g_{i,k})$
6: **end for**
7: $g \leftarrow (g_1, \ldots, g_L)$
8: **return** g

Finally, we carry out a brute-force search for pairs (A,B) if, and only if, there exists $i \in \{1,\ldots,L\}$ such that $g_i(A) = g_i(B)$. The following simple lemma explain why we expect that the single-bit hash functions $g_{i,j}$ correlate with the similarity relation.

Lemma 3.1. *Let f^* be as in (3.1). Then for any $(A,B) \in \mathcal{A} \times \mathcal{B}$, and any function $g_{i,j}$ as above,*

$$\mathbb{P}_{g_{i,j}}\left[g_{i,j}(A) = g_{i,j}(B)\right] = \frac{1 + f^*(A,B)}{2},$$

where the probability is over the choice of $g_{i,j}$.

Proof. Since $g_{i,j}$ is $\{-1,+1\}$-valued,

$$\mathbb{P}_{g_{i,j}}\left[g_{i,j}(A) = g_{i,j}(B)\right] = \mathbb{E}_{g_{i,j}}\left[\frac{1 + g_{i,j}(A)g_{i,j}(B)}{2}\right].$$

Now recall that $g_{i,j} = k_t^*$ with probability α_t for each $t \in [T]$. □

The upshot of Lemma 3.1 is this: if $f^*(A, B)$ correlates positively with our similarity notion, it is expected that $\mathbb{P}_{g_{i,j}}[g_{i,j}(A) = g_{i,j}(B)] \geq p_1 > 1/2$ for "most" similar pairs, whereas $\mathbb{P}[g_{i,j}(A) = g_{i,j}(B)] \leq p_2 < 1/2$ for "most" dissimilar pairs. This will be made precise in Section 3.2, where values p_1 and p_2 will be derived from a margin property of the function f^*. The parameters k and L are used to amplify the gap between the values p_1 and p_2. Theoretical and practical methods for choosing these hyperparameters will be discussed in the next sections.

3.1.3 Performance metrics The goal of our method is to ensure that, for each $A \in \mathcal{A}$, one can find all similar $B \in \mathcal{B}$ while doing as few pairwise comparisons as possible. This is made precise by the Recall and Reduction Ratio (RR) metrics which are standard in the literature of Record Linkage [SVSF14, SS18, Chr12]:

$$\text{Recall} := \frac{1}{|\mathcal{M}|} \sum_{(\ell,r)\in\mathcal{M}} \mathbf{1}_{[\exists i\in\{1,\ldots,L\}, g_i(A_\ell)=g_i(B_r)]}; \tag{3.2}$$

$$\text{RR} := 1 - \frac{1}{N_{\mathcal{A}} \cdot N_{\mathcal{B}}} \sum_{(\ell,r)\in[N_{\mathcal{A}}]\times[N_{\mathcal{B}}]} \mathbf{1}_{[\exists i\in\{1,\ldots,L\}, g_i(A_\ell)=g_i(B_r)]}, \tag{3.3}$$

where $\mathcal{M}$ denotes the set of matching pairs:

$$\mathcal{M} := \{(\ell, r) \in [N_{\mathcal{A}}] \times [N_{\mathcal{B}}], A_\ell \sim_R B_r, (A_\ell, B_r) \in \mathcal{A} \times \mathcal{B}\}. \tag{3.4}$$

The Recall metric measures the proportion of similar pairs that are matched by our method, whereas RR measures what proportion of the $N_{\mathcal{A}} \cdot N_{\mathcal{B}}$ potential pairwise comparisons are avoided by the method. Ideally, we expect to find as many as possible matching pairs (Recall close to 1), while avoiding as many comparisons as possible (RR close to 1).

3.2 Theoretical results

We now present an idealized set of conditions under which our method provably achieves high values of Recall and RR (see above). These conditions are encapsulated in the following assumption.

Assumption 3.2. $\mathcal{A} := \{A_\ell\}_{\ell=1}^{N_{\mathcal{A}}}$ and $\mathcal{B} := \{B_r\}_{r=1}^{N_{\mathcal{B}}}$ are both contained in a set $\mathcal{X}$. A notion of similarity $\sim_R$ between items (A_ℓ, B_r) is given. The training sample:

$$\mathcal{S}_{\text{train},n} := \{((a_i, b_i), y_i) \in \mathcal{A} \times \mathcal{B} \times \{-1, 1\}\}_{i=1}^n$$

is obtained via n independent draws of the following type:

With probability 1/2, set $y_i = +1$ and choose $(A_\ell, B_r) \in \mathcal{A} \times \mathcal{B}$ uniformly at random from the set of pairs satisfying $A_\ell \sim_R B_r$. With the remaining probability, set $y_i = -1$ and choose $(A_\ell, B_r) \in \mathcal{A} \times \mathcal{B}$ uniformly at random from the set of dissimilar pairs $A_\ell \not\sim_R B_r$.

We let P denote the distribution of one random draw as above.

In this idealized scenario, the similar pairs in our training sample are chosen uniformly at random. This might be too strong an assumption in practice. For instance, when matching two movie databases (e.g., IMDB and TMDB), it is natural to use a training set consisting of movies that can be surely matched. It is likely that this training set thus correlates with popularity or other movie characteristics. Still, our Assumption can be viewed as a natural first step towards a fuller analysis of hashing in the setting of record linkage.

The following condition will play a central role in our analysis. It corresponds to our intuition in the previous section that f^* should correlate with the similarity relation. Specifically, the condition requires that good classification of similar pairs happens with a good margin.

Condition 1. Classifiers $(k_t^*)_{t=1}^T$, convex weights α_t are given so that, for a given $\theta > 0$, the following holds with probability at least $1 - \varepsilon$ over the choice of $(A, B, y) \sim P$,

$$y\, f^*(A, B) > \theta, \tag{3.5}$$

where f^* is as in (3.1).

Note that, given a triple $((A, B), y) \sim P$, we can rewrite (3.5) as

$$\begin{aligned} f^*(A, B) &> +\theta, \text{ if } y = +1 \\ f^*(A, B) &< -\theta, \text{ if } y = -1 \end{aligned}$$

Our first theorem gives a sufficient condition for the above condition to be achieved by the first stage of our procedure (cf. §3.1.1) given a tranining sample $\mathcal{S}_{\text{train},n} = ((a_i, b_i), y_i)_{i=1}^n$. The value of ε depends on the Rademacher complexity of the base classifiers $\mathcal{K}$ and the projected samples $\mathcal{S}_{\mathcal{A},n} := \{a_i\}_{i=1}^n$ and $\mathcal{S}_{\mathcal{B},n} := \{b_i\}_{i=1}^n$ given by

$$\begin{aligned} \mathcal{R}_{\mathcal{S}_{\mathcal{A},n}}(\mathcal{K}) &= \frac{1}{n}\mathbb{E}_\sigma \left[\sup_{k \in \mathcal{K}} \sum_{i=1}^n \sigma_i y_i k(a_i) \right] \\ \mathcal{R}_{\mathcal{S}_{\mathcal{B},n}}(\mathcal{K}) &= \frac{1}{n}\mathbb{E}_\sigma \left[\sup_{k \in \mathcal{K}} \sum_{i=1}^n \sigma_i y_i k(b_i) \right], \end{aligned}$$

as defined in Section 1.2.2.

Theorem 3.3. *The following holds with probability $\geq 1-\delta$: if $\theta > 0$ is given, then the function f^* corresponding to the output of Algorithm 4 satisfies Condition 1 with the value of $\varepsilon = \varepsilon_{\text{train}}(f^*, \mathcal{S}_{\text{train},n}, \theta, \delta)$ given by:*

$$\varepsilon_{\text{train}}(f^*, \mathcal{S}_{\text{train},n}, \theta, \delta) := 2^T \prod_{t=1}^{T} \varepsilon_t^{1/2-\theta}(1-\varepsilon_t)^{\theta-1/2} \tag{3.6}$$

$$+ \frac{8}{\theta}\left(\mathcal{R}_{\mathcal{S}_{\mathcal{A},n}}(\mathcal{K}) + \mathcal{R}_{\mathcal{S}_{\mathcal{B},n}}(\mathcal{K})\right) + \sqrt{\frac{\log(1/\delta)}{2n}}. \tag{3.7}$$

Furthermore, if there exists $\gamma > 0$ such that for all $t \in [T]$, $\gamma \leq (1/2 - \varepsilon_t)$ and $\theta \leq 2\gamma$, then the term in (3.6) decreases exponentially with T.

To get some intuition for this bound, observe that the product term in the definition of $\varepsilon_{\text{train}}(f, \mathcal{S}_{\text{train},n}, \theta, \delta)$ is a margin bound for AdaBoost over the training data [BFLS98]. In the ideal "weak learning" scenario where $\varepsilon_t \leq 1/2 - \eta$ for all t, this training error will decay exponentially fast in T for suitably small margin parameters θ. The other terms in $\varepsilon_{\text{train}}(f, \mathcal{S}_{\text{train},n}, \theta, \delta)$ correspond to a generalization bound used for bounding the test error. The proof of Theorem 3.3 is an adaptation to our setting of the arguments by Bartlett et al. [BM02].

Our second result shows that, when Condition 1 holds, then the hash construction in §3.1.2 gives high values of the Recall and RR metrics, with a suitable choice of parameters k and L.

Theorem 3.4. *Consider databases $\mathcal{A}$ and $\mathcal{B}$ such that $|\mathcal{A}| = N_{\mathcal{A}}$ and $|\mathcal{B}| = N_{\mathcal{B}}$. If Condition 1 holds for the output f^* of Algorithm 4 for a given $\theta > 0$, $\gamma \in (0,1)$ is given, and we set:*

$$\rho := \frac{\log\left(\frac{2}{1+\theta}\right)}{\log\left(\frac{2}{1-\theta}\right)} \in [0,1),\ k := \lceil \log_{\frac{2}{1+\theta}} N_{\mathcal{A}} \cdot N_{\mathcal{B}} \rceil \text{ and } L := \left\lceil \frac{2(N_{\mathcal{A}} \cdot N_{\mathcal{B}})^{\rho} \log(1/\gamma)}{1+\theta} \right\rceil,$$

then Algorithm 5 achieves the following expected values for the Recall and RR metrics defined in (3.2) and (3.3):

$$\mathbb{E}\left[\text{Recall}\right] \geq (1-\gamma)(1-\varepsilon)$$
$$\mathbb{E}\left[\text{RR}\right] \geq \left(1 - \frac{|\mathcal{M}| + L}{N_{\mathcal{A}} \cdot N_{\mathcal{B}}}\right)(1-\varepsilon).$$

Both expectations are with respect to the randomness in the hash code.

Note that $\rho < 1$ and in linkage problems is usually the case that $|\mathcal{M}| \ll N_{\mathcal{A}} \cdot N_{\mathcal{B}}$ so the expected

RR is close to 1, meaning that we have to make only few comparisons. Theorem 3.4 may be viewed as a "proof of concept" that our method has large values of Recall and RR for some choice of the hyperparameters k and L. In practice, we expect that these choices are quite conservative, especially because the bound in Theorem 3.3 is probably loose. A simple solution to choose k and L is just to consider a validation set to test several possible combinations and choose one that achieves as large Recall and RR as possible.

3.3 Experiments

To test the performance of our method, we compare it against two algorithms in three different Record Linkage datasets varying several possible hyperparamters, as we describe next.

3.3.1 Datasets We consider three standard datasets for Record Linkage models [SVSF14, SS18]. The first two are `RLDATA500` and `RLDATA10000`, which consists of 500 and 10000 artificial textual personal information, respectively. The third dataset is `Restaurant`, which consists of 865 textual names and addresses of restaurants, some of them being duplicates.

RLDATA							
first_name_1	first_name_2	last_name_1	last_name_2	birth_year	birth_month	birth_day	full_name
GERD		BAUER		1968	7	27	GERD BAUER
WOLFGANG		ENGEL		1936	12	27	WOLFGANG ENGEL
HARALD		WEBER		1977	6	1	HARALD WEBER
GERD		BAUERH		1968	7	27	GERD BAUERH

Table 3.1: Example of entries in `RLDATA500` and `RLDATA10000` datasets. In yellow we have duplicated entries.

Restaurant			
name	address	location	cuisine
arnie morton's of chicago	435 s. la cienega blv.	los angeles	american
nate 'n' al's	414 n. beverly dr.	los angeles	american
schatzi on main	3110 main st.	los angeles	continental
arnie morton's of chicago	435 s. la cienega blvd.	los angeles	steakhouses

Table 3.2: Example of entries in `Restaurant` dataset. In yellow we have duplicated entries.

3.3.2 Vectorization We follow the setting described in [SVSF14, SS18]. We first apply the shingling technique to construct a sparse numerical representation of our textual data. Here, each

string is mapped to k contiguous sub-strings known as "k-grams", "shingles", or "tokens". For example, the string "TORONTO" yields the bag of length-two shingles "TO", "OR", "RO", "ON", "NT", "TO". (note "TO" appears twice.). Then, we apply minhash [BCFM00] to transform our sparse numerical information into a dense one while still preserving similarity between the original data.

3.3.3 Benchmark models Our method is compared to two LSH-based blocking methods [AI06, HPIM12]. The first one is the Transitive Locality Sensitive Hashing (TLSH) [OCC13], which uses a community detection technique to find similar records. The next one is the K-Means Locality Sensitive Hashing (KLSH) [PJA10] which makes use of k-means algorithm to construct a low-dimensional projection of the data. These baselines are considered in classical Record Linkage works such as [SVSF14, SS18].

3.3.4 Hyperparameters Baseline models were trained with the original implementation hyperparameters. For the training stage in Algorithm 4 we take $\mathcal{K}$ as the family of Stumps functions defined in (2.3). In order to determine optimal hyperparameters for each of the benchmarked models and each of the selected metrics, and to provide a large enough seed variation to produce reliable and reproducible results, more than 300k experiments were done in total. Among the variations tested, shingle sizes of the vectorization process ranged from 1 to 6 and each of the model-specific parameters — such as $k, L \in \mathbb{N}$ for our model.

3.3.5 Performance in record-linkage applications As we described in Section 3.1.3, we are interested in models with high Reduction Ratio (RR) and Recall. So to be able to rank the tested models, we simply consider its RR plus its Recall. We run each model with several combinations of hyperparameters over 8 different seeds, then we take the average performance and its standard deviation. Finally, we choose the best based on the previous criteria for each method. In Table 3.3 we show the result for the dataset `RLDATA500`. Our model shows the best RR metric and it is able to find every matching pair in this dataset, since it resulting recall equals one. The hyperparameters used in Algorithm 5 that led to the best performance for our model were $k = 50$ and $L = 140$.

`RLDATA500`		
model	recall	reduction ratio
our model	**1.0 ±0**	**0.9973 ± 0.0001**
TLSH	**1.0 ± 0**	0.9969 ± 0.0005
KLSH	**1.0 ± 0**	0.9870± 0.0030

Table 3.3

In Table 3.4 we exhibit the best performance of each model for the dataset `RLDATA10000` based on

the previous criteria. Our model shows the best Recall metric and its RR metric is very close to the best one, given by TLSH. The hyperparameters used in Algorithm 5 that led to the best performance for our model were $k = 20$ and $L = 110$.

RLDATA10000		
model	recall	reduction ratio
our model	**0.9925 ±0.0055**	0.9916 ± 0.0009
TLSH	0.9862 ± 0.0079	**0.9990 ± 0.0003**
KLSH	0.9340 ± 0.0152	0.9872± 0.0004

Table 3.4

In Table 3.5 we exhibit the best performance of each model for the dataset `Restaurant` based on the previous criteria. Our model shows a RR metric that is reasonably close to the other models, while our Recall is superior to the TLSH. The hyperparameters used in Algorithm 5 that led to the best performance for our model were $k = 20$ and $L = 120$.

Restaurant		
model	recall	reduction ratio
our model	0.9296 ±0.07.03	0.9800 ± 0.0046
TLSH	0.7977 ± 0.0862	**0.9903 ± 0.0056**
KLSH	**0.9872 ± 0.0284**	0.9869± 0.0017

Table 3.5

Note that even though there are cases where our method is not the best, our model always has a Recall and RR metric very close to the best one, unlike KLSH which has the worst Recall and RR in `RLDATA10000` and TLSH which has the worst Recall in `Restaurant`.

3.4 Proofs and technical results

3.4.1 Proof of Theorem 3.3 To show that the output of Algorithm 4 indeed satisfies Condition 1 we first prove a stronger result. First, we need to define the following sets:

$$\mathcal{K}_2 := \{f_k : (A, B) \mapsto k(A)k(B), k \in \mathcal{K}\},$$

and the convex hull $\text{conv}(\mathcal{K}_2)$ of $\mathcal{K}_2$ given by,

$$\begin{aligned}\text{conv}(\mathcal{K}_2) &:= \left\{ f : (A,B) \mapsto \sum_{t=1}^{T} \alpha_t f_{k_t}(A,B) : T \geq 1, \alpha_t \geq 0, f_{k_t} \in \mathcal{K}_2, \sum_{t=1}^{T} \alpha_t = 1 \right\} \\ &= \left\{ f : (A,B) \mapsto \sum_{t=1}^{T} \alpha_t k_t(A) k_t(B) : T \geq 1, \alpha_t \geq 0, k_t \in \mathcal{K}, \sum_{t=1}^{T} \alpha_t = 1 \right\}.\end{aligned}$$

Theorem 3.5. *Consider an iid sample $\mathcal{S}_{\text{train},n} = ((a_i,b_i),y_i)_{i=1}^n$ with $((a_i,b_i),y_i)$ drawn from P. Then, given $\theta \in (0,1)$ and $\delta \in (0,1)$, with probability at least $1-\delta$, for any $f \in conv(\mathcal{K}_2)$*

$$P[yf(A,B) \leq \theta] \leq \varepsilon_{\text{train}}(f, \mathcal{S}_{\text{train},n}, \theta, \delta),$$

where

$$\varepsilon_{\text{train}}(f, \mathcal{S}_{\text{train},n}, \theta, \delta) := \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{[y_i f(a_i,b_i) \leq 2\theta]} + \frac{8}{\theta} \mathcal{R}_{\mathcal{S}_{\text{train},n}}(\mathcal{K}) + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Proof. First, consider the surrogate margin loss function given by:

$$\varphi_\theta(x) = \min\left(1, \max\left(1 - \frac{x}{\theta}, 0\right)\right).$$

and the following set:

$$\Phi_\theta := \{\varphi_{\theta,f} : ((a,b),y) \mapsto \varphi_\theta(yf(a,b) - \theta) : f \in \text{conv}(\mathcal{K}_2)\}.$$

By Rademacher Inequality 1.3, we have that with probability at least $1-\delta$, for all $f \in \text{conv}(\mathcal{K}_2)$:

$$\mathbb{E}\left[\varphi_\theta(yf(A,B) - \theta)\right] \leq \frac{1}{n} \sum_{i=1}^{n} \varphi_\theta(y_i f(a_i,b_i) - \theta) + 2\mathcal{R}_{\mathcal{S}_{\text{train},n}}(\Phi_\theta) + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Using the fact that $\mathbf{1}_{[x \leq \theta]} \leq \varphi_\theta(x-\theta)$, we have that with probability at least $1-\delta$, for all $f \in \text{conv}(\mathcal{K}_2)$

$$P[yf(a,b) \leq \theta] \leq \frac{1}{n} \sum_{i=1}^{n} \varphi_\theta(y_i f(a_i,b_i) - \theta) + 2\mathcal{R}_{\mathcal{S}_{\text{train},n}}(\Phi_\theta) + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Since φ_θ is $1/\theta$-Lipschitz, by Talagrand's Lemma and the fact that $\mathcal{R}_{\mathcal{S}_{\text{train},n}}(\text{conv}(\mathcal{K}_2)) = \mathcal{R}_{\mathcal{S}_{\text{train},n}}(\mathcal{K}_2)$ [BM02, Theorem 12], we have with probability at least $1-\delta$, for all $f \in \text{conv}(\mathcal{K}_2)$:

$$P[yf(A,B) \leq \theta] \leq \frac{1}{n} \sum_{i=1}^{n} \varphi_\theta(y_i f(a_i,b_i) - \theta) + \frac{2}{\theta} \mathcal{R}_{\mathcal{S}_{\text{train},n}}(\mathcal{K}_2) + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Using the fact that $\varphi_\theta(x-\theta) \leq \mathbf{1}_{[x\leq 2\theta]}$, with probability at least $1-\delta$, for all $f \in \text{conv}(\mathcal{K}_2)$:

$$P[yf(A,B) \leq \theta] \leq \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{[y_i f(a_i,b_i)\leq 2\theta]} + \frac{2}{\theta}\mathcal{R}_{\mathcal{S}_{\text{train},n}}(\mathcal{K}_2) + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Now we just need to bound $\mathcal{R}_{\mathcal{S}_{\text{train},n}}(\mathcal{K}_2)$ in terms of $\mathcal{R}_{\mathcal{S}_{\text{train},n}}(\mathcal{K})$ so our final result depends only on the Rademacher complexity of the family $\mathcal{K}$ which is usually known. But note that

$$\begin{aligned}
\mathcal{R}_{\mathcal{S}_{\text{train},n}}(\mathcal{K}_2) &= \frac{1}{n}\mathbb{E}_\sigma\left[\sup_{k\in\mathcal{K}_2}\sum_{i=1}^{n}\sigma_i y_i k(a_i)k(b_i)\right] \\
&= \frac{1}{n}\mathbb{E}_\sigma\left[\sup_{k\in\mathcal{K}}\sum_{i=1}^{n}\sigma_i y_i k(a_i)k(b_i)\right] \\
&\leq \frac{1}{n}\mathbb{E}_\sigma\left[\sup_{k_1,k_2\in\mathcal{K}}\sum_{i=1}^{n}\sigma_i y_i k_1(a_i)k_2(b_i)\right] \\
&= \frac{1}{n}\mathbb{E}_\sigma\left[\sup_{k_1,k_2\in\mathcal{K}}\sum_{i=1}^{n}\sigma_i k_1(a_i)k_2(b_i)\right] \\
&= \mathbb{E}_\sigma\left[\sup_{k_1,k_2\in\mathcal{K}}\sum_{i=1}^{n}\sigma_i\left(1-\frac{(k_1(a_i)-k_2(b_i))^2}{2}\right)\right] \\
&= 0 + \frac{1}{n}\mathbb{E}_\sigma\left[\sup_{k_1,k_2\in\mathcal{K}}\sum_{i=1}^{n}\sigma_i\frac{(k_1(a_i)-k_2(b_i))^2}{2}\right] \\
&= \frac{1}{2n}\mathbb{E}_\sigma\left[\sup_{k_1,k_2\in\mathcal{K}}\sum_{i=1}^{n}\sigma_i L\left(k_1(a_i)-k_2(b_i)\right)\right]
\end{aligned}$$

where,

$$L(x) = \begin{cases} x^2, \text{ if } x\in[-2,2] \\ 4, \text{ otherwise.} \end{cases}$$

Since L is 4-Lipschitz, by Talagrand's Lemma, we have that, with probability at least $1-\delta$, for all $f \in \text{conv}(\mathcal{K}_2)$

$$P[yf(A,B) \leq \theta] \leq \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{[y_i f^*(a_i,b_i)\leq 2\theta]} + \frac{8}{\theta}\left(\mathcal{R}_{\mathcal{S}_{\mathcal{A},n}}(\mathcal{K}) + \mathcal{R}_{\mathcal{S}_{\mathcal{B},n}}(\mathcal{K})\right) + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

□

Note that, the output f^* of Algorithm 4 satisfies $f^* \in \text{conv}(\mathcal{K}_2)$, so using Theorem 3.5, to finish the proof of Theorem 3.3 we only need to bound $\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{[y_i f(a_i,b_i)\leq 2\theta]}$. By the definition of Q_t in

Algorithm 4 in line 9, we have that

$$Q_{T+1}(i) = \frac{\exp\left(-y_i f(a_i, b_i) \sum_{t=1}^{T} \alpha'_t\right)}{n \prod_{t=1}^{T} Z_t},$$

therefore, using the fact that for $x \in \mathbb{R}$ we have that $\mathbf{1}_{[x \leq 0]} \leq \exp(-x)$,

$$\begin{aligned}
\frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{[y_i f(a_i,b_i) \leq 2\theta]} &= \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{[y_i f(a_i,b_i) \sum_{t=1}^{T} \alpha'_t \leq 2\theta \sum_{t=1}^{T} \alpha'_t]} \\
&\leq \frac{1}{n} \exp\left(2\theta \sum_{t=1}^{T} \alpha'_t\right) \sum_{i=1}^{n} \exp\left(-y_i f(a_i, b_i) \sum_{t=1}^{T} \alpha'_t\right) \\
&= \exp\left(2\theta \sum_{t=1}^{T} \alpha'_t\right) \prod_{t=1}^{T} Z_t \sum_{i=1}^{n} Q_{T+1}(i) \\
&= \exp\left(2\theta \sum_{t=1}^{T} \alpha'_t\right) \prod_{t=1}^{T} Z_t \cdot 1.
\end{aligned}$$

But note that, by definition of ε_t and the definition of $\alpha'_t = \log((1-\varepsilon_t)/\varepsilon_t)/2$ (cf. Algorithm 4),

$$\begin{aligned}
Z_t &= \sum_{i=1}^{n} Q_t(i) \exp\left(-\alpha'_t k^*_t(a_i) k^*_t(b_i)\right) \\
&= (1-\varepsilon_t) \exp(-\alpha_t) + \varepsilon_t \exp(\alpha'_t) \\
&= 2\sqrt{\varepsilon_t(1-\varepsilon_t)},
\end{aligned}$$

which is exactly the value we use in line 7 of Algorithm 4. Moreover,

$$\begin{aligned}
\exp\left(2\theta \sum_{t=1}^{T} \alpha'_t\right) &= \prod_{t=1}^{T} \exp\left(2\theta \alpha'_t\right) \\
&= \prod_{t=1}^{T} \exp\left(\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)\right)^{\theta} \\
&= \prod_{t=1}^{T} \left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)^{\theta}
\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{[y_i f(a_i,b_i)\leq 2\theta]} &\leq \exp\left(2\theta\sum_{t=1}^{T}\alpha_t'\right)\prod_{t=1}^{T}Z_t \\ &= \prod_{t=1}^{T}\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)^{\theta} 2\sqrt{\varepsilon_t(1-\varepsilon_t)}\end{aligned}$$

finishing the first part of the proof.

The proof that (3.6) decreases exponentially in T is a straightforward calculation and can be found in [MRT12, Theorem 7.7].

3.4.2 Proof of Theorem 3.4 This proof is an adaptation of [HPIM12]. Since f^* satisfies Condition 1 for $\theta > 0$, we know by Lemma 3.1 that for all A, B in a set $\mathcal{E}$ of P-measure $\geq 1-\varepsilon$

$$\begin{aligned}&\text{if } A \sim_R B, \text{then } \mathbb{P}_{g_{i,j}}\left[g_{i,j}(A) = g_{i,j}(B)\right] \geq \frac{1+\theta}{2} = p_1 \\ &\text{if } A \not\sim_R B, \text{then } \mathbb{P}_{g_{i,j}}\left[g_{i,j}(A) = g_{i,j}(B)\right] \leq \frac{1-\theta}{2} = p_2,\end{aligned}$$

where P is as described in Assumption 3.2. For our next calculations assume we are conditioned to this event. Fix $k = \lceil \log_{1/p_2} N_{\mathcal{A}} \cdot N_{\mathcal{B}} \rceil$ and let $\mathcal{M}$ be the set of matching pairs as defined in (3.4). We split the proof in the following steps:

- **Probability of finding correct matches.** Suppose that $A \sim_R B$ and $(A, B) \in \mathcal{E}$. By independence, for $i \in [L]$

$$\begin{aligned}\mathbb{P}_{g_i}\left[g_i(A) = g_i(B)\right] &= \mathbb{P}_{g_{i,j}}\left[g_{i,j}(A) = g_{i,j}(B)\right]^k \\ &\geq p_1^k \\ &\geq p_1^{\log_{1/p_2}(N_{\mathcal{A}}\cdot N_{\mathcal{B}})+1} \\ &= p_1 p_1^{\log_{1/p_2}(N_{\mathcal{A}}\cdot N_{\mathcal{B}})} \\ &= p_1 (N_{\mathcal{A}} \cdot N_{\mathcal{B}})^{-\rho},\end{aligned}$$

  where in the last equality we used a simple logarithm change of basis. That is,

$$\mathbb{P}_{g_i}\left[g_i(A) \neq g_i(B)\right] \leq 1 - p_1 (N_{\mathcal{A}} \cdot N_{\mathcal{B}})^{-\rho},$$

Thus, the probability of finding the correct match is

$$\begin{aligned}\mathbb{P}\left[\exists i \in \{1,\dots,L\},\ g_i(A) = g_i(B)\right] &= 1 - \mathbb{P}\left[\forall i \in \{1,\dots,L\},\ g_i(A) \neq g_i(B)\right] \\ &= 1 - \mathbb{P}_{g_i}\left[g_i(A) \neq g_i(B)\right]^L \\ &\geq 1 - \left(1 - p_1(N_{\mathcal{A}} \cdot N_{\mathcal{B}})^{-\rho}\right)^L\end{aligned}$$

hence, by setting $L = \frac{\log(1/\gamma)(N_{\mathcal{A}} \cdot N_{\mathcal{B}})^{\rho}}{p_1}$ for $\gamma \in (0,1)$, we have that

$$\begin{aligned}\mathbb{P}\left[\exists i \in \{1,\dots,L\},\ g_i(A) = g_i(B)\right] &\geq 1 - \left(1 - p_1(N_{\mathcal{A}} \cdot N_{\mathcal{B}})^{-\rho}\right)^L \\ &\geq 1 - e^{-\log(1/\gamma)} \\ &= 1 - \gamma.\end{aligned}$$

- **Expected Recall.** By the previous item, we have

$$\begin{aligned}\mathbb{E}\left[\text{Recall}\right] &\geq \frac{1}{|\mathcal{M}|} \sum_{(\ell,r)\in\mathcal{M}} \mathbb{P}\left[\exists i \in \{1,\dots,L\}, g_i(A_\ell) = g_i(B_r) | (A,B) \in \mathcal{E}\right] P[\mathcal{E}] \\ &\geq (1-\gamma)(1-\varepsilon).\end{aligned}$$

- **Probability of finding wrong matches.** Suppose that $A \not\sim_R B$ and $(A,B) \in \mathcal{E}$. Then, for $i \in [L]$:

$$\begin{aligned}\mathbb{P}_{g_i}\left[g_i(A) = g_i(B)\right] &= \mathbb{P}_{g_{i,j}}\left[g_{i,j}(A) = g_{i,j}(B)\right]^k \\ &\leq p_2^k \\ &\leq \frac{1}{N_{\mathcal{A}} \cdot N_{\mathcal{B}}},\end{aligned}$$

  by our choice of k.

- **Expected number of wrong matches.** By the previous item, conditioned to $(A,B) \in \mathcal{E}$, the random variable that counts the number wrong matches found by g_i

$$C(g_i) = \sum_{(\ell,r)\notin\mathcal{M}} \mathbf{1}_{[g_i(A_\ell)=g_i(B_r)]}$$

  follows a binomial distribution with parameter $\left(N_{\mathcal{A}} \cdot N_{\mathcal{B}} - |\mathcal{M}|, \frac{1}{N_{\mathcal{A}} \cdot N_{\mathcal{B}}}\right)$, hence

$$\mathbb{E}_{g_i}\left[C(g_i)\right] \leq 1,$$

  therefore the number of total wrong collisions for g_i is at most 1 and the number of total wrong collisions for all g_i for $i \in \{1,\dots,L\}$ is at most L.

- **Expected RR.** By the previous item and the fact that Condition 1 holds with probability $\geq 1-\varepsilon$, the expected number of comparisons is

$$\begin{aligned}
\mathbb{E}\left[\#\text{ comparisons}\right] \leq & \sum_{(\ell,r)\in[N_{\mathcal{A}}]\times[N_{\mathcal{B}}]} \mathbb{P}\left[\exists i \in \{1,\dots,L\}, g_i(A_\ell) = g_i(B_r) | (A,B)\in\mathcal{E}\right] P[(A,B)\in\mathcal{E}] \\
& + P[(A,B)\notin\mathcal{E}] \\
& \leq \sum_{(\ell,r)\in\mathcal{M}} \mathbb{P}\left[\exists i \in \{1,\dots,L\}, g_i(A_\ell) = g_i(B_r) | (A,B)\in\mathcal{E}\right](1-\varepsilon) \\
& + \sum_{(\ell,r)\notin\mathcal{M}} \mathbb{P}\left[\exists i \in \{1,\dots,L\}, g_i(A_\ell) = g_i(B_r) | (A,B)\in\mathcal{E}\right](1-\varepsilon) + \varepsilon \\
& \leq (|\mathcal{M}|+L)(1-\varepsilon)+\varepsilon.
\end{aligned}$$

  Therefore, the expected RR satisfies

$$\mathbb{E}\left[\text{RR}\right] \geq 1-\varepsilon-\left(\frac{|\mathcal{M}|+L}{N_{\mathcal{A}}\cdot N_{\mathcal{B}}}\right)(1-\varepsilon).$$

Chapter 4

Split conformal prediction for dependent data

The results in this section were obtained in a joint paper [OORR22] with João Vitor Romano, Roberto I. Oliveira and Paulo Orenstein. The author of this thesis is responsible for all the theory and mathematical proofs.

4.1 Introduction

We denote by $(X_i, Y_i)_{i=1}^n$ a random sample of n random covariate/response pairs with stationary marginals: $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are measurable spaces. An additional random pair $(X_*, Y_*) \in \mathcal{X} \times \mathcal{Y}$, independent from the sample $(X_i, Y_i)_{i=1}^n$, will also be considered, and we assume $(X_i, Y_i) \sim (X_*, Y_*)$ for all $i \in [n]$, where $[n] \coloneqq \{1, \ldots, n\}$.

The data indices can be partitioned $[n] = I_{\text{train}} \sqcup I_{\text{cal}} \sqcup I_{\text{test}}$, where $n = n_{\text{train}} + n_{\text{cal}} + n_{\text{test}}$ and $I_{\text{train}} \coloneqq [n_{\text{train}}]$ corresponds to the training data, $I_{\text{cal}} \coloneqq [n_{\text{train}} + n_{\text{cal}}] \backslash [n_{\text{train}}]$ corresponds to calibration data, and $I_{\text{test}} \coloneqq [n] \backslash [n_{\text{train}} + n_{\text{cal}}]$ corresponds to test data.

For any function $s : (\mathcal{X} \times \mathcal{Y})^{n_{\text{train}}+1} \to \mathbb{R}$ and datapoint $(x, y) \in \mathcal{X} \times \mathcal{Y}$, denote a nonconformity score as

$$\widehat{s}_{\text{train}}(x, y) \coloneqq s((X_i, Y_i)_{i \in I_{\text{train}}}, (x, y)),$$

corresponding to the values of s when the first n_{train} pairs in the input are the training data. Intuitively, the role of $\widehat{s}_{\text{train}}$ is to measure how discrepant a prediction based on x_i is compared to the true y_i; e.g., $\widehat{s}_{\text{train}}(x, y) = |y - \widehat{\mu}(x)|$, where $\widehat{\mu}$ is some regression model trained on $(X_i, Y_i)_{i \in I_{\text{train}}}$. Several choices have been proposed in the literature [LGR+18, HPW19, RPC19, ABJM21].

Given $\phi \in [0,1)$, let $\widehat{q}_{\phi,\text{cal}}$ denote the empirical ϕ-quantile of $\widehat{s}_{\text{train}}(X_i, Y_i)$ over I_{cal}; that is:

$$\widehat{q}_{\phi,\text{cal}} \coloneqq \inf \left\{ t \in \mathbb{R} \,:\, \frac{1}{n_{\text{cal}}} \sum_{i \in I_{\text{cal}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \le t]} \ge \phi \right\}. \tag{4.1}$$

For $x \in \mathcal{X}$, the predictive sets are then defined via:

$$C_\phi(x) \coloneqq \{y \in \mathcal{Y} \,:\, \widehat{s}_{\text{train}}(x,y) \le \widehat{q}_{\phi,\text{cal}}\}. \tag{4.2}$$

Also, given $q : (\mathcal{X} \times \mathcal{Y})^{n_{\text{train}}} \to \mathbb{R}$, which is assumed measurable, define the quantile $q_{\text{train}} \coloneqq q((X_i, Y_i)_{i \in I_{\text{train}}})$ and the probability

$$P_{q,\text{train}} \coloneqq \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \le q_{\text{train}} \mid (X_i, Y_i)_{i \in I_{\text{train}}}]. \tag{4.3}$$

Finally, denote the conditional version of the above quantities as follows. Let $\mathcal{A}$ denote a family of measurable subsets of $\mathcal{X}$. Given $\phi \in [0,1)$ and $A \in \mathcal{A}$, let $I_{\text{cal}}(A) \coloneqq \{i \in I_{\text{cal}} \,:\, X_i \in A\}$ and $n_{\text{cal}}(A) \coloneqq \#I_{\text{cal}}(A)$. Denote the empirical ϕ-quantile of $\widehat{s}_{\text{train}}(X_i, Y_i)$ over $i \in I_{\text{cal}}$ as:

$$\widehat{q}_{\phi,\text{cal}}(A) \coloneqq \inf \left\{ t \in \mathbb{R} \,:\, \frac{1}{n_{\text{cal}}(A)} \sum_{i \in I_{\text{cal}}(A)} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \le t]} \ge \phi \right\},$$

and, for $x \in A$, define the predictive set:

$$C_\phi(x; A) \coloneqq \{y \in \mathcal{Y} \,:\, \widehat{s}_{\text{train}}(x,y) \le \widehat{q}_{\phi,\text{cal}}(A)\}.$$

For $A \in \mathcal{A} \subset \mathcal{X}$ with $\mathbb{P}[X \in A] > 0$, let

$$P_{q,\text{train}}(A) \coloneqq \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \le q_{\text{train}} \mid (X_i, Y_i)_{i \in I_{\text{train}}}, X_* \in A]. \tag{4.4}$$

4.1.1 Marginal and empirical guarantees This subsection details how marginal and empirical guarantees (1.5) and (1.6) can be extended when the data is not exchangeable. Some basic assumptions are needed, though they are satisfied by large classes of processes. Section 4.2 shows that is the case for stationary β-mixing data, and Section 4.3 details further extensions.

First, it is necessary to have some form of concentration over the calibration data, as well as a degree of marginal decoupling over the test data. We will assume there exist $\varepsilon_{\text{cal}} \in (0,1)$ and $\delta_{\text{cal}} \in (0,1)$

such that

$$\mathbb{P}\left[\left|\frac{1}{n_{\text{cal}}}\sum_{i\in I_{\text{cal}}}\mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i)\leq q_{\text{train}}]}-P_{q,\text{train}}\right|\leq\varepsilon_{\text{cal}}\right]\geq 1-\delta_{\text{cal}}, \tag{4.5}$$

where $P_{q,\text{train}}$ is defined as in (4.3). Further, we assume that there exists $\varepsilon_{\text{test}}$ such that, for $i \in I_{\text{test}}$,

$$|\mathbb{P}[\widehat{s}_{\text{train}}(X_i,Y_i)\leq q_{\text{train}}]-\mathbb{E}[P_{q,\text{train}}]|\leq\varepsilon_{\text{test}}. \tag{4.6}$$

Under these conditions, the usual marginal coverage guarantees can be recovered for split conformal prediction.

Theorem 4.1 (Marginal coverage over test data)**.** *Given $\alpha \in (0,1)$ and $\delta_{\text{cal}} > 0$, if conditions (4.5) and (4.6) hold, then, for all $i \in I_{\text{test}}$:*

$$\mathbb{P}[Y_i\in C_{1-\alpha}(X_i)]\geq 1-\alpha-\varepsilon_{\text{cal}}-\delta_{\text{cal}}-\varepsilon_{\text{test}}. \tag{4.7}$$

Additionally, if $\widehat{s}_{\text{train}}(X_*,Y_*)$ almost surely has a continuous distribution conditionally on the training data, then

$$|\mathbb{P}[Y_i\in C_{1-\alpha}(X_i)]-(1-\alpha)|\leq\varepsilon_{\text{cal}}+\delta_{\text{cal}}+\varepsilon_{\text{test}}.$$

To also guarantee empirical coverage, suppose that instead of the decoupling assumption (4.6), there exists concentration of the empirical c.d.f. of the nonconformity score over the test data, that is, there exist $\varepsilon_{\text{test}}, \delta_{\text{test}} \in (0,1)$ such that

$$\mathbb{P}\left[\left|P_{q,\text{train}}-\frac{1}{n_{\text{test}}}\sum_{i\in I_{\text{test}}}\mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i)\leq q_{\text{train}}]}\right|\leq\varepsilon_{\text{test}}\right]\geq 1-\delta_{\text{test}}. \tag{4.8}$$

Theorem 4.2 (Empirical coverage over test data)**.** *Given $\alpha \in (0,1)$, $\delta_{\text{cal}} > 0$ and $\delta_{\text{test}} > 0$, if (4.5) and (4.8) hold, then:*

$$\mathbb{P}\left[\frac{1}{n_{\text{test}}}\sum_{i\in I_{\text{test}}}\mathbf{1}_{[Y_i\in C_{1-\alpha}(X_i)]}\geq 1-\alpha-\eta\right]\geq 1-\delta_{\text{cal}}-\delta_{\text{test}},$$

where $\eta = \varepsilon_{\text{cal}} + \varepsilon_{\text{test}}$. Additionally, if $\widehat{s}_{\text{train}}(X_*,Y_*)$ almost surely has a continuous distribution condi-

tionally on the training data, then:

$$\mathbb{P}\left[\left|\frac{1}{n_{\text{test}}}\sum_{i\in I_{\text{test}}}\mathbf{1}_{[Y_i\in C_{1-\alpha}(X_i)]}-(1-\alpha)\right|\leq\eta\right]\geq 1-2\delta_{\text{cal}}-2\delta_{\text{test}}.$$

While the purpose of the above theorems is to extend split conformal guarantees to non-exchangeable data, they also readily apply to the iid case. In the Section 4.5, we show that, in such case, it suffices to take $\varepsilon_{\text{cal}}=\sqrt{(2n_{\text{cal}})^{-1}\log(2/\delta_{\text{cal}})}$ and $\varepsilon_{\text{test}}=\sqrt{(2n_{\text{test}})^{-1}\log(2/\delta_{\text{test}})}$.

4.1.2 Conditional guarantees Obtaining a conditional version of (1.5) and (1.6) is of interest in many cases. Experiments have confirmed that, to achieve an unconditional coverage at level $1-\alpha$, coverage might be better than $1-\alpha$ for certain values of X_i and much worse for others [CGD21]. [BCRT20] prove that coverage is not generally attainable, even for iid data. On the positive side, they show that by conditioning on sets of finite VC dimension that are not too small, conditional guarantees can be achieved. Our goal is to show these also hold for split conformal prediction under non-exchangeable data.

First, conditional versions of assumptions (4.5) and (4.6) are needed. For concentration over the calibration data, suppose there exist δ_{cal} and $\varepsilon_{\text{cal}}\in(0,1)$ such that, for $P_{q,\text{train}}(A)$ as in (4.4),

$$\mathbb{P}\left[\sup_{A\in\mathcal{A}}\left|\frac{1}{n_{\text{cal}}(A)}\sum_{i\in I_{\text{cal}}(A)}\mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i)\leq q_{\text{train}}]}-P_{q,\text{train}}(A)\right|\leq\varepsilon_{\text{cal}}\right]\geq 1-\delta_{\text{cal}}. \tag{4.9}$$

For a conditional version of marginal decouping, assume there exists $\varepsilon_{\text{test}}\in(0,1)$ such that

$$|\mathbb{P}[\widehat{s}_{\text{train}}(X_k,Y_k)\leq q_{\text{train}}\mid X_k\in A]-\mathbb{E}[P_{p,\text{train}}(A)]|\leq\varepsilon_{\text{test}}. \tag{4.10}$$

These conditions suffice for conditional marginal coverage.

Theorem 4.3 (Conditional coverage over test data)**.** *Given $\alpha\in(0,1)$ and $\delta_{\text{cal}}>0$, if (4.9) and (4.10) hold, then, for each $A\in\mathcal{A}\subset\mathcal{X}$ and any $i\in I_{\text{test}}$:*

$$\mathbb{P}[Y_i\in C_{1-\alpha}(X_i;A)\mid X_i\in A]\geq 1-\alpha-\varepsilon_{\text{cal}}-\delta_{\text{cal}}-\varepsilon_{\text{test}}.$$

Additionally, if $\widehat{s}_{\text{train}}(X_*,Y_*)$ almost surely has a continuous distribution conditionally on the training data, then:

$$|\mathbb{P}[Y_i\in C_{1-\alpha}(X_i;A)\mid X_i\in A]-(1-\alpha)|\leq\varepsilon_{\text{cal}}+\delta_{\text{cal}}+\varepsilon_{\text{test}}.$$

For a conditional version of (4.8), suppose there exist δ_{test} and $\varepsilon_{\text{test}} \in (0,1)$ such that

$$\mathbb{P}\left[\sup_{A\in\mathcal{A}}\left|P_{q,\text{train}}(A) - \frac{1}{n_{\text{test}}(A)}\sum_{i\in I_{\text{test}}(A)}\mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i)\le q_{\text{train}}]}\right| \le \varepsilon_{\text{test}}\right] \ge 1-\delta_{\text{test}}, \tag{4.11}$$

where $P_{q,\text{train}}(A)$ is defined as in (4.4). This suffices for empirical conditional coverage.

Theorem 4.4 (Empirical conditional coverage over test data)**.** *Given $\alpha \in (0,1)$, $\delta_{\text{cal}} > 0$ and $\delta_{\text{test}} > 0$, if (4.9) and (4.11) hold, then for each $A \in \mathcal{A}$:*

$$\mathbb{P}\left[\inf_{A\in\mathcal{A}}\frac{1}{n_{\text{test}}(A)}\sum_{i\in I_{\text{test}}(A)}\mathbf{1}_{[Y_i\in C_{1-\alpha}(X_i;A)]} \ge 1-\alpha-\eta\right] \ge 1-\delta_{\text{cal}}-\delta_{\text{test}},$$

where $\eta = \varepsilon_{\text{cal}} + \varepsilon_{\text{test}}$. Additionally, if $\widehat{s}_{\text{train}}(X_*, Y_*)$ almost surely has a continuous distribution conditionally on the training data, then:

$$\mathbb{P}\left[\sup_{A\in\mathcal{A}}\left|\frac{1}{n_{\text{test}}(A)}\sum_{i\in I_{\text{test}}(A)}\mathbf{1}_{[Y_i\in C_{1-\alpha}(X_i;A)]} - (1-\alpha)\right| \le \eta\right] \ge 1-2\delta_{\text{cal}}-2\delta_{\text{test}}.$$

The results above are directly applicable in the iid case. As we show in the Section 4.5, (4.10) holds with $\varepsilon_{\text{test}} = 0$ and, if the family $\mathcal{A}$ has finite VC dimension $\text{VC}(\mathcal{A}) = d$ and $\mathbb{P}[A] > \gamma$ for some $\gamma > 0$ and all $A \in \mathcal{A}$, it suffices to take $\varepsilon_{\text{cal}} = \gamma^{-1}(4\sqrt{\log(2(n+1)^d)/n} + 2\sqrt{\log(4/\delta)/(2n)})$.

4.2 Stationary β-mixing data

We now apply the framework from Section 4.1 to the class of stationary β-mixing processes. This class of non-exchangeable data is broad enough to cover many important applications, such as hidden Markov models and Markov chains [Dou12] as well as ARMA and GARCH models [CC02, Mok88], while still providing explicit error terms in the bounds of Theorems 4.1, 4.2, 4.3 and 4.4.

Recall a sequence of random variables $\{Z_t\}_{t=-\infty}^{\infty}$ is said to be stationary if its finite-dimensional distributions are time-invariant; that is, for any $t \in \mathbb{Z}$ and $m, k \in \mathbb{N}$,

$$Z_{t:(t+m)} = (Z_, \ldots, Z_{t+m}) \overset{d}{=} (Z_{t+k}, \ldots, Z_{t+m+k}) = Z_{(t+k):(t+m+k)}.$$

Furthermore, for a stationary stochastic process $\{Z_t\}_{t=-\infty}^{\infty}$ and index $a \in \mathbb{N}$, the β-mixing coefficient of the process at a is defined as

$$\beta(a) = \|\mathbb{P}_{-\infty:0,a:\infty} - \mathbb{P}_{-\infty:0} \otimes \mathbb{P}_{a:\infty}\|_{\text{TV}}.$$

where $\|\cdot\|_{\mathrm{TV}}$ denotes the total variation norm, and $\mathbb{P}_{-\infty:0,a:\infty}$ is the joint distribution of the blocks $(Z_{-\infty:0}, Z_{a:\infty})$. The process is said to be β-mixing if $\beta(a) \to 0$ when $a \to \infty$.

The β-mixing condition allows us to replace independence with asymptotic independence and still retain some important concentration results. In particular, the so-called Blocking Technique [Yu94, MR10, KM17] allows one to compare a β-mixing process with another process made of independent blocks. The results below generally follow from combining the Blocking Technique with decoupling arguments and Bernstein's concentration inequality.

4.2.1 Standard coverage guarantees We now show how the framework from Section 4.1 yields explicit coverage bounds for stationary β-mixing processes. As is standard with the Blocking Technique, the error bounds obtained will depend on an optimization of block sizes, though note this is purely a mathematical device. The split CP method itself is not dependent on this optimization or even the definition of block sizes (unlike other CP variants, such as [CWZ18]).

The sets of parameters we optimize over are defined as follows:

$$F_{\mathrm{cal}} = \{(a, m, r) \in \mathbb{N}_{>0}^3 \ : \ 2ma = n_{\mathrm{cal}} - r + 1, \ \delta_{\mathrm{cal}} > 4(m-1)\beta(a) + \beta(r)\}$$

and

$$F_{\mathrm{test}} = \{(a, m, s) \in \mathbb{N}_{>0}^2 \times \mathbb{N} \ : \ 2ma = n_{\mathrm{test}} - s, \ \delta_{\mathrm{test}} > 4(m-1)\beta(a) + \beta(n_{\mathrm{cal}})\}.$$

These two sets correspond to block size choices in the calibration and test sets, respectively. For the calibration set, define the error term as follows:

$$\begin{aligned} \varepsilon_{\mathrm{cal}} \coloneqq \inf_{(a,m,r)\in F_{\mathrm{cal}}} \Bigg\{ & \widetilde{\sigma}(a)\sqrt{\frac{4}{n_{\mathrm{cal}} - r + 1} \log\left(\frac{4}{\delta_{\mathrm{cal}} - 4(m-1)\beta(a) - \beta(r)}\right)} \\ & + \frac{1}{3m} \log\left(\frac{4}{\delta_{\mathrm{cal}} - 4(m-1)\beta(a) - \beta(r)}\right) + \frac{r-1}{n_{\mathrm{cal}}} \Bigg\}, \end{aligned} \tag{4.12}$$

where

$$\widetilde{\sigma}(a) = \sqrt{\frac{1}{4} + \frac{2}{a}\sum_{j=1}^{a-1}(a-j)\beta(j)}. \tag{4.13}$$

Similarly, we define the test error correction factor for a stationary β-mixing process as

$$\varepsilon_{\text{test}} = \inf_{(a,m,s)\in F_{\text{test}}} \left\{ \widetilde{\sigma}(a)\sqrt{\frac{4}{n_{\text{test}}}\log\left(\frac{4}{\delta_{\text{test}} - 4(m-1)\beta(a) - \beta(n_{\text{cal}})}\right)} + \frac{1}{3m}\log\left(\frac{4}{\delta_{\text{test}} - 4(m-1)\beta(a) - \beta(n_{\text{cal}})}\right) + \frac{s}{n_{\text{test}}} \right\}. \tag{4.14}$$

With ε_{cal} as above, Theorem 4.1 yields the following result for stationary β-mixing processes:

Theorem 4.5 (Marginal coverage: stationary β-mixing processes)**.** *Suppose that $(X_i, Y_i)_{i=1}^n$ is stationary β-mixing. Then given $\alpha \in (0,1)$ and $\delta_{\text{cal}} > 0$, for $i \in I_{\text{test}}$,*

$$\mathbb{P}\left[Y_i \in C_{1-\alpha}(X_i)\right] \geq 1 - \alpha - \eta,$$

with $\eta = \varepsilon_{\text{cal}} + \varepsilon_{\text{train}} + \delta_{\text{cal}}$, where ε_{cal} is as in (4.12) and $\varepsilon_{\text{train}} = \beta(i - n_{\text{train}})$. Additionally, if $\widehat{s}_{\text{train}}(X_*, Y_*)$ almost surely has a continuous distribution conditionally on the training data:

$$\left|\mathbb{P}[Y_i \in C_{1-\alpha}(X_i)] - (1-\alpha)\right| \leq \eta.$$

Under certain assumptions over the dependence of the processes, the stationary β-mixing bounds given by (4.12) are of the same asymptotic order as the corresponding iid bounds. Indeed, if $\beta(k) \leq k^{-b}$ and $\delta \geq n_{\text{cal}}^{-c}$ for $b > 1, c > 0$, with $1 + 2c < b$, as long as $m = o(n_{\text{cal}}^{(b-c)/(b+1)})$ and $\sqrt{n_{\text{cal}}\log(n_{\text{cal}})} = o(m)$, the bounds are of the same order. This is satisfied, for example, if $m = n_{\text{cal}}^{\lambda}$, $a = n_{\text{cal}}^{1-\lambda}/2$ with $1/2 < \lambda < (b-c)/(b+1)$.

Additionally, with $\varepsilon_{\text{test}}$ as above, Theorem 4.2 yields the following:

Theorem 4.6 (Empirical coverage: stationary β-mixing processes)**.** *Suppose that $(X_i, Y_i)_{i=1}^n$ is stationary β-mixing. Then given $\alpha \in (0,1)$,$\delta_{\text{cal}} > 0$ and $\delta_{\text{test}} > 0$*

$$\mathbb{P}\left[\frac{1}{n_{\text{test}}}\sum_{i\in I_{\text{test}}} \mathbf{1}_{[Y_i \in C_{1-\alpha}(X_i)]} \geq 1 - \alpha - \eta\right] \geq 1 - \delta_{\text{cal}} - \delta_{\text{test}},$$

with $\eta = \varepsilon_{\text{cal}} + \varepsilon_{\text{test}}$, and ε_{cal} and $\varepsilon_{\text{test}}$ defined in (4.12) and (4.14). Additionally, if $\widehat{s}_{\text{train}}(X_*, Y_*)$ almost surely has a continuous distribution conditionally on the training data, then:

$$\mathbb{P}\left[\left|\frac{1}{n_{\text{test}}}\sum_{i\in I_{\text{test}}} \mathbf{1}_{[Y_i \in C_{1-\alpha}(X_i)]} - (1-\alpha)\right| \leq \eta\right] \geq 1 - 2\delta_{\text{cal}} - 2\delta_{\text{test}}.$$

We note in passing that the expression in (4.12) follows from a stationary β-mixing version of Bernstein's inequality, proved in the Section 4.5, which might be of independent interest.

4.2.2 Conditional guarantees To obtain conditional guarantees for stationary β-mixing processes, we need to specify a family $\mathcal{A}$ of Borel measurable sets in $\mathcal{X}$ satisfying certain conditions. In particular, we will assume that, for a fixed value $\gamma > 0$, the family $\mathcal{A}$ of Borel measurable sets in $\mathcal{X}$ has finite VC dimension $\mathrm{VC}(\mathcal{A}) = d$ and $\mathbb{P}[X_* \in A] > \gamma$ for all $A \in \mathcal{A}$.

Then, given $\delta_{\mathrm{cal}} > 0$ and $\alpha \in (0,1)$, define the calibration error correction factor for a stationary β-mixing process conditioned to the family $\mathcal{A}$ as

$$\varepsilon_{\mathrm{cal}} = \inf_{(a,m,r)\in G_{\mathrm{cal}}} \left\{ \frac{1}{\gamma} \left(\frac{\kappa(m,r)}{n_{\mathrm{cal}}} + \sqrt{\frac{2}{m} \log\left(\frac{16}{\delta_{\mathrm{cal}} - 16(m-1)\beta(a) - \beta(r)} \right)} \right) \right\} \tag{4.15}$$

where $\kappa(m,r) = 4n_{\mathrm{cal}}\sqrt{\log(2(m+1)^d)/m} + 2(r-2)$ and

$$G_{\mathrm{cal}} = \{(a,m,r) \in \mathbb{N}^3_{>0} \ : \ 2ma = n_{\mathrm{cal}} - r + 1, \delta_{\mathrm{cal}} > 16(m-1)\beta(a) + \beta(r)\}.$$

Note the factor $1/\gamma$ in $\varepsilon_{\mathrm{cal}}$: for η to be small, we need $\varepsilon_{\mathrm{cal}}$ to be small and consequently m has to be large. This is quite natural, since if γ is too small, the probability $\mathbb{P}[X_* \in A]$ can be close to zero, and thus a larger sample is necessary to estimate the empirical quantile well.

Then, Theorem 4.3 yields the following.

Theorem 4.7 (Conditional coverage: stationary β-mixing processes)**.** *Suppose that $(X_i, Y_i)_{i=1}^n$ is stationary β-mixing. Then given $\alpha \in (0,1)$, $\gamma > 0$ and $\delta_{\mathrm{cal}} > 0$, for each $A \in \mathcal{A}$ and any $i \in I_{\mathrm{test}}$*

$$\mathbb{P}[Y_i \in C_{1-\alpha}(X_i; A) \mid X_i \in A] \geq 1 - \alpha - \eta,$$

with $\eta = \varepsilon_{\mathrm{cal}} + \varepsilon_{\mathrm{test}}$, where $\varepsilon_{\mathrm{cal}}$ is as in (4.15) and $\varepsilon_{\mathrm{test}} = \beta(i - n_{\mathrm{train}})$.

Additionally, if $\widehat{s}_{\mathrm{train}}(X_*, Y_*)$ almost surely has a continuous distribution conditionally on the training data, then:

$$|\mathbb{P}[Y_i \in C_{1-\alpha}(X_i; A) \mid X_i \in A] - (1-\alpha)| \leq \varepsilon_{\mathrm{cal}} + \delta_{\mathrm{cal}} + \varepsilon_{\mathrm{test}}.$$

Now, denote the test error correction factor for a stationary β-mixing process conditioned to the family $\mathcal{A}$ as

$$\varepsilon_{\mathrm{test}} = \inf_{(a,m,s)\in G_{\mathrm{test}}} \left\{ \frac{1}{\gamma} \left(\frac{\tilde{\kappa}(m,r)}{n_{\mathrm{test}}} + \sqrt{\frac{2}{m} \log\left(\frac{8}{\delta_{\mathrm{test}} - 8(m-1)\beta(a) - \beta(n_{\mathrm{cal}})} \right)} \right) \right\}, \tag{4.16}$$

where $\tilde{\kappa}(m,r) = 4n_{\text{test}}\sqrt{\log(2(m+1)^d)/m} + 2s$ and

$$G_{\text{test}} = \{(a,m,s) \in \mathbb{N}_{>0}^2 \times \mathbb{N} \ : \ 2ma = n_{\text{test}} - s, \delta_{\text{test}} > 8(m-1)\beta(a) + \beta(n_{\text{cal}})\}.$$

The following result then follows from Theorem 4.4.

Theorem 4.8 (Empirical conditional coverage: stationary β-mixing processes)**.** *Suppose that $(X_i, Y_i)_{i=1}^n$ is stationary β-mixing, then given $\alpha \in (0,1)$, $\gamma > 0$, $\delta_{\text{cal}} > 0$ and $\delta_{\text{test}} > 0$, for each $A \in \mathcal{A}$:*

$$\mathbb{P}\left[\inf_{A\in\mathcal{A}} \frac{1}{n_{\text{test}}(A)} \sum_{i\in I_{\text{test}}(A)} \mathbf{1}_{[Y_i \in C_{1-\alpha}(X_i;A)]} \geq 1 - \alpha - \eta\right] \geq 1 - \delta_{\text{cal}} - \delta_{\text{test}},$$

where $\eta = \varepsilon_{\text{cal}} + \varepsilon_{\text{test}}$, for ε_{cal} as in (4.15) and $\varepsilon_{\text{test}}$ as in (4.16).

Additionally, if $\widehat{s}_{\text{train}}(X_*, Y_*)$ almost surely has a continuous distribution conditionally on the training data, then:

$$\mathbb{P}\left[\sup_{A\in\mathcal{A}} \left| \frac{1}{n_{\text{test}}(A)} \sum_{i\in I_{\text{test}}(A)} \mathbf{1}_{[Y_i \in C_{1-\alpha}(X_i;A)]} - (1-\alpha) \right| \leq \eta\right] \geq 1 - 2\delta_{\text{cal}} - 2\delta_{\text{test}}.$$

4.3 Extensions

4.3.1 Risk-controlling prediction sets Risk-controlling prediction sets (RCPS), introduced by [BAL+21], give a general methodology for CP that applies in a variety of settings, including regression, multiclass classification and image segmentation. Importantly, RCPS does not involve nonconformity scores, but rather, the construction of nested sets. While the original theory of RCPS assumes independent data, we now show it also applies within our framework.

Suppose $\mathcal{Y}'$ is a family of sets, $\Lambda \subset \mathbb{R} \cup \{+\infty\}$ is a closed set, and a map $\mathcal{T} : (\mathcal{X} \times \mathcal{Y})^{n_{\text{train}}} \times \mathcal{X} \times \Lambda \to \mathcal{Y}'$ is given with the following property: for all choices of $(x_i, y_i)_{i=1}^{n_{\text{cal}}} \in (\mathcal{X} \times \mathcal{Y})^{n_{\text{cal}}}$, $x \in \mathcal{X}$ and $\lambda_1, \lambda_2 \in \Lambda$: if $\lambda_1 \leq \lambda_2$, then $\mathcal{T}((x_i, y_i)_{i=1}^{n_{\text{cal}}}, x, \lambda_1) \subset \mathcal{T}((x_i, y_i)_{i=1}^{n_{\text{cal}}}, x, \lambda_2)$.

For $(x, \lambda) \in \mathcal{X}$, we use the notation

$$\widehat{\mathcal{T}}_{\lambda,\text{train}}(x) \coloneqq \mathcal{T}((X_i, Y_i)_{i\in I_{\text{train}}}, x, \lambda)$$

to denote the values of $\mathcal{T}$ when the first n_{train} pairs in the input correspond to the training data. We call $\widehat{\mathcal{T}}_{\lambda,\text{train}}(\cdot)$ a trained tolerance region. Finally, $L : \mathcal{Y} \times \mathcal{Y}' \to \mathbb{R}$ is a loss function that is decreasing in

the $\mathcal{Y}'$ component. The goal of RCPS is to compute a value $\widehat{\lambda}$ from the calibration data that achieves (conditional) risk smaller than a prespecified level $\alpha > 0$.

To define the conditional risk, first assume that the map $\lambda \in \Lambda \mapsto \mathbb{E}[L(Y_*, \widehat{\mathcal{T}}_\lambda(X_*)) \mid (X_i, Y_i)_{i \in I_{\text{train}}}]$ is almost surely continuous. Moreover, given a measurable $\ell : (\mathcal{X} \times \mathcal{Y})^{n_{\text{train}}} \to \Lambda$, we let $\ell_{\text{train}} \coloneqq \ell((X_i, Y_i)_{i \in I_{\text{train}}})$, define the conditional expected risk as

$$R(\ell) \coloneqq \mathbb{E}[L\left(Y_*, \widehat{\mathcal{T}}_{\ell_{\text{train}}}(X_*)\right) \mid (X_i, Y_i)_{i \in I_{\text{train}}}].$$

Also, define the empirical risk over the calibration data as

$$\widehat{R}_{\text{cal}}(\lambda) \coloneqq \frac{1}{n_{\text{cal}}} \sum_{i \in I_{\text{cal}}} L(Y_i, \mathcal{T}_\lambda(X_i)).$$

Now, a threshold $\widehat{\lambda}$ must be chosen from calibration data to control the risk. In [BAL$^+$21], this requires finding a function $\lambda \mapsto \widehat{R}_{\text{UCB}}(\lambda)$ that gives a pointwise high-probability upper bound on $R(\lambda)$. In our case, we can allow for a $\widehat{R}(\lambda)$ that bounds the risk up to a small error; for us, the empirical risk will play this role. Thus, consider the empirical threshold

$$\widehat{\lambda}_{\alpha,\text{cal}} \coloneqq \inf\left\{\lambda \in \Lambda \,:\, \forall \lambda' \in \Lambda, \lambda' > \lambda \Rightarrow \widehat{R}_{\text{cal}}(\lambda) < \alpha\right\}.$$

Finally, we give conditions that guarantee that $\widehat{\lambda}_{\alpha,\text{cal}}$ controls the risk with high probability. First, assume that there exist $\varepsilon_{\text{cal}} > 0$, $\delta_{\text{cal}} \in (0,1)$ such that, for any $\ell, \ell_{\text{train}}$,

$$\mathbb{P}\left[|\widehat{R}_{\text{cal}}(\ell_{\text{train}}) - R(\ell)| \le \varepsilon_{\text{cal}}\right] \ge 1 - \delta_{\text{cal}}. \tag{4.17}$$

Also, assume there exists a $\varepsilon_{\text{test}}$ such that for all $i \in I_{\text{test}}$ and all ℓ,

$$|\mathbb{E}[L(Y_i, \mathcal{T}_{\ell_{\text{train}}}(Y_*))] - \mathbb{E}[R(\ell)]| \le \varepsilon_{\text{test}}. \tag{4.18}$$

Then, the following result on the performance of RCPS over a single test point holds.

Theorem 4.9 (Approximate risk control for $\widehat{\lambda}_{1-\alpha,\text{cal}}$)**.** *Assume (4.17) and (4.18). Then,*

$$\mathbb{E}[L(Y_*, \mathcal{T}_{\widehat{\lambda}_{1-\alpha,\text{cal}}}(X_*))] \le \alpha + \varepsilon_{\text{cal}}] \ge 1 - \delta_{\text{cal}}.$$

Moreover, if L is uniformly bounded, we have the following for all $i \in I_{\text{test}}$:

$$\mathbb{E}[L(Y_i, \mathcal{T}_{\widehat{\lambda}_{1-\alpha,\text{cal}}}(X_i))] \leq \alpha + \varepsilon_{\text{test}} + \varepsilon_{\text{cal}} + 2\|L\|_\infty \, \delta_{\text{test}}.$$

Thus the expected loss at any test point is controlled by α plus an error term that can be shown to be small, even for non-exchangeable data. Importantly, the result is achieved via assumptions that only bound the behavior of the loss over individual thresholds ℓ_{train} obtained from the training data. In particular, there is no need to require uniform control of the loss over a range of ℓ, which would require stronger (and looser) concentration bounds. The uniform bound on L can be replaced by a moment assumption, at the cost of a messier bound.

4.3.2 Non-stationary data In this subsection, we sketch how to analyze split CP in a setting where there is dependent data with a marginal distribution changing slowly over time. For brevity, we focus on marginal coverage. Our analysis is partly inspired by the recent work of [BCRT22].

Let empirical quantiles and predictive sets still be as in (4.1) and (4.2), so the method is still split CP, but replace the pair (X_*, Y_*) with an *auxiliary process* $(X_{*,i}, Y_{*,i})_{i\in[n]}$ that is an independent copy of the original data $(X_i, Y_i)_{i\in[n]}$. Let N_{cal} be a random number, uniformly distributed over I_{cal}, independently of the problem data and auxiliary process. For $j \in I_{\text{test}}$, the quantity

$$\delta^{(j)} := \|\text{Law}(X_j, Y_j) - \text{Law}(X_{N_{\text{cal}}}, Y_{N_{\text{cal}}})\|_{\text{TV}}$$

measures how far the distribution of (X_j, Y_j) is to that of a randomly chosen point in the calibration dataset. This can be taken as a measure of distributional drift.

For marginal coverage, we take the random variable q_{train} as before, but replace (4.3) by a time-inhomogeneous version for $j \in I_{\text{cal}} \sqcup I_{\text{test}}$

$$P^{(j)}_{q,\text{train}} := \mathbb{P}[\widehat{s}_{\text{train}}(X_{*,j}, Y_{*,j}) \leq q_{\text{train}} \mid (X_i, Y_i)_{i\in I_{\text{train}}}]$$

Furthermore, (4.5) and (4.6) are also replaced with time-inhomogeneous versions:

$$\mathbb{P}\left[\left|\frac{1}{n_{\text{cal}}} \sum_{i\in I_{\text{cal}}} (\mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i)\leq q_{\text{train}}]} - P^{(i)}_{q,\text{train}})\right| \leq \varepsilon_{\text{cal}}\right] \geq 1 - \delta_{\text{cal}},$$

and, for $j \in I_{\text{test}}$,

$$|\mathbb{P}[\widehat{s}_{\text{train}}(X_j, Y_j) \leq q_{\text{train}}] - \mathbb{E}[P^{(j)}_{q,\text{train}}]| \leq \varepsilon_{\text{test}}.$$

With these assumptions, marginal coverage result holds: for any $i \in I_{\text{test}}$,

$$\mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq \widehat{q}_{1-\alpha,\text{cal}}] \geq 1 - \alpha - \eta - \delta^{(i)},$$

where $\eta = \varepsilon_{\text{cal}} + \delta_{\text{cal}} + \varepsilon_{\text{test}}$ is the same error term appearing in Theorem 4.1. In particular, we recover that theorem up to an error depending on how much distributional drift there is between i and the calibration set. This is similar to the main result in [BCRT22], except that there the authors consider weighted calibration sets. Finally, it is possible to show that the assumptions above hold for the nonstationary β-mixing case.

In practice, a reasonable way to deal with distribution drift is to occasionally (or always) update the training and calibration sets. Experiments in Section 4.4 showcase online CP. Our finite-sample methods can also be used to analyze this case when the β-mixing coefficients decay fast enough, but the results in this subsection do not assume stationarity.

4.3.3 Rank-one-out conformal prediction Rank-one-out (ROO) conformal prediction, introduced by [LGR$^+$18], is different from split CP in that the method calibrates the quantile used for each test data point by looking at the remaining test points.

This requires adapting the above setup as follows: partition the data indices as $[n] = I_{\text{train}} \sqcup I_{\text{test}}$, and for each $i \in I_{\text{test}}$ the calibration set is $I_{\text{cal}}^{(i)} = I_{\text{test}} \setminus \{i\}$. Also, define the empirical quantiles as follows: given $\phi \in [0, 1)$ and $i \in I_{\text{test}}$, let $\widehat{q}_{\phi,\text{cal}}^{(i)}$ denote the empirical ϕ-quantile

$$\widehat{q}_{\phi,\text{cal}}^{(i)} \coloneqq \inf \left\{ t \in \mathbb{R} \,:\, \frac{1}{n_{\text{test}} - 1} \sum_{j \in I_{\text{cal}}^{(i)}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_j, Y_j) \leq t]} \geq \phi \right\}.$$

For $x \in \mathcal{X}$, the rank-one-out predictive set for $i \in I_{\text{test}}$ is then defined via:

$$C_\phi^{(i)}(x) \coloneqq \{y \in \mathcal{Y} \,:\, \widehat{s}_{\text{train}}(x, y) \leq \widehat{q}_{\phi,\text{cal}}^{(i)}\}.$$

We can then adapt the concentration and decoupling hypotheses. Indeed, we assume there exist $\varepsilon_{\text{test}} \in (0, 1)$, $\{\varepsilon_{\text{test}}(i)\}_{i \in I_{\text{test}}} \subset (0, 1)$ and $\delta_{\text{test}} \in (0, 1)$ such that, for any $i \in I_{\text{test}}$,

$$|\mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq q_{\text{train}}] - \mathbb{E}[P_{q,\text{train}}]| \leq \varepsilon_{\text{test}}(i), \tag{4.19}$$

and, moreover,

$$\mathbb{P}\left[\left| \frac{1}{n_{\text{test}}} \sum_{i \in I_{\text{test}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i, Y_i) \leq q_{\text{train}}]} - P_{q,\text{train}} \right| \leq \varepsilon_{\text{test}} \right] \geq 1 - \delta_{\text{test}}. \tag{4.20}$$

Then, the analogue of Theorems 4.1 and 4.2 still hold for ROO.

Theorem 4.10 (Marginal and empirical coverage over test data for ROO)**.** *Given $\alpha \in (0,1)$, if (4.19) and (4.20) hold, then, for all $i \in I_{\text{test}}$:*

$$\mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq \widehat{q}_{1-\alpha,\text{cal}}] \geq 1 - \alpha - \varepsilon_{\text{test}}(i) - \varepsilon_{\text{test}} - \delta_{\text{cal}} - \frac{1}{n_{\text{test}}}.$$

Moreover, it holds that

$$\mathbb{P}\left[\frac{1}{n_{\text{test}}} \sum_{i \in I_{\text{test}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \leq \widehat{q}^{(i)}_{1-\alpha,\text{cal}}]} \geq 1 - \alpha - \varepsilon_{\text{test}} - \frac{1}{n_{\text{test}}}\right] \geq 1 - 2\delta_{\text{test}}.$$

One can adapt the analysis in Section 4.2 to bound the parameters δ_{test}, $\varepsilon_{\text{test}}$ and $\varepsilon_{\text{test}}(i)$ for β-mixing data. In particular, one may take $\varepsilon_{\text{test}}(i) = \beta(i - n_{\text{cal}})$, and $\varepsilon_{\text{test}}, \delta_{\text{test}}$ equal to the respective parameters $\varepsilon_{\text{cal}}, \delta_{\text{cal}}$ in that section, but with n_{test} replacing n_{cal}. This reflects the fact that the calibration set for each point of rank-one-out is essentially equal to the test set.

On the other hand, we note that marginal coverage might suffer somewhat over the first few test data, since $\varepsilon_{\text{test}}(i) = \beta(i - n_{\text{cal}})$ may be large for small values $i - n_{\text{cal}}$. In contrast to split CP, there is no gap in ROO between training and test data so the first test points may be strongly correlated with the training data.

4.4 Experiments

This section studies split CPs empirical performance in four numerical experiments. The first two involve synthetic simulations where the bounds can be calculated explicitly, while the last two show that split CP's guarantees work with real data, even when exchangeability is clearly violated. In all examples, we employ split conformal quantile regression [RPC19].

Example 1 (Two-state hidden Markov model) Let $(W_0, W_1, W_2, \ldots)$ be a Markov chain with state space $\mathcal{W} = \{0, 1\}$, probabilities $\mathbb{P}[W_t = 1 | W_{t-1} = 0] = p$ and $\mathbb{P}[W_t = 0 | W_{t-1} = 1] = q$, following the stationary distribution with $\pi = \begin{bmatrix} \frac{q}{p+q} & \frac{p}{p+q} \end{bmatrix}$, with $p, q \in (0,1)$ and $p + q > 0$. This data is stationary β-mixing, and the mixing coefficients can be found explicitly [MSS15]. When $p = q = 0.5$, $\beta(r) \equiv 0$ for all $r \in \mathbb{N}_{>0}$, so the Markov chain reduces to a sequence of iid Bernoulli trials. On the other hand, as p and q tend towards zero, $\beta(r)$ becomes large for every r and dependence increases. We construct a hidden Markov model by adding a Gaussian noise with zero mean and variance of 10^{-6} to the Markov chain with $p = q$, and consider predicting a single data point by using the 11 preceding elements in the series as features.

Figure 4.1 shows how marginal coverage (4.7), calculated through 10 000 simulations, is affected by increasing levels of dependence $1-p$ for three different models (boosting, neural network and random forest) and $n_{\text{train}} = 1000, n_{\text{cal}} = 500$ and $n_{\text{test}} = 1$. Marginal coverage observed is close to nominal iid value of 90% for the independent case ($p = 0.5$) and weak to medium dependence, measured by the probabilities $1-p$ of repeating the previous state. Coverage remains above 89% even for large values of dependence, and falls below 88% only after $1-p = 0.999$. Also, Figure 4.2a shows how the correction $\varepsilon_{\text{cal}} + \delta_{\text{cal}} + \varepsilon_{\text{test}}$ in Theorem 4.1 depends on the calibration set sizes, quickly converging to the iid limit even for moderately dependent data.

Further, Figure 4.2b shows the empirical coverage for a gradient boosting model with $n_{\text{train}} = 1000$ and $n_{\text{cal}} = n_{\text{test}} = 15\,000$, so $\delta_{\text{cal}} = \delta_{\text{test}} = 0.005$. Note the empirical coverage revolves around the prescribed iid level 0.9, and it remains above the worst-case theoretical bound, which decreases with the dependence level $1-p$.

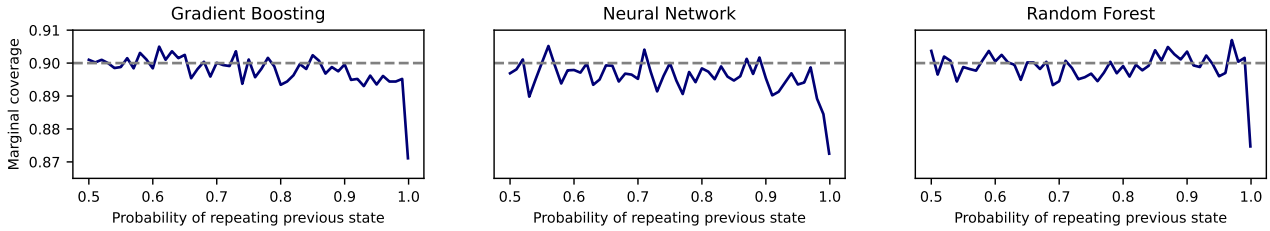
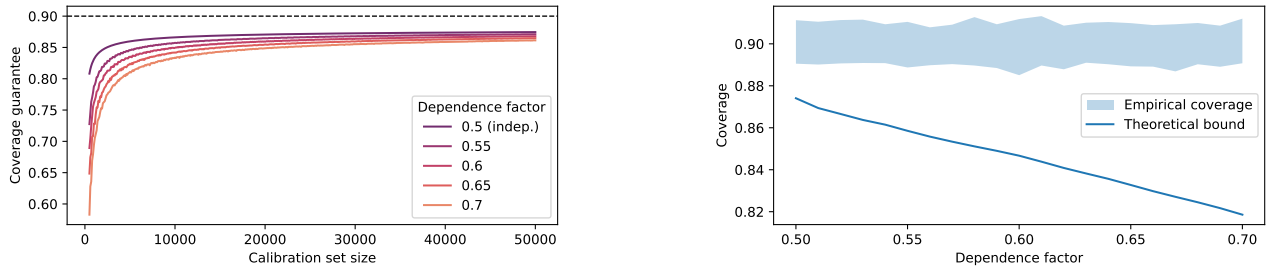


Figure 4.1: Marginal coverage for two-state hidden Markov model (solid) and nominally prescribed 90% target (dashed) for different levels of dependence and three different models. Coverage is above 89% for all but very extreme levels.

Example 2 (Autoregressive process) Consider the autoregressive process of order one (AR(1)), defined by the recurrence $W_t = \lambda W_t + \varepsilon_t$, with $t \in \mathbb{N}$, $\lambda \in \mathbb{R}$ and $\varepsilon_t \sim N(0,1)$ independently. This sequence



(a) Marginal coverage for varying parameters.

(b) Empirical coverage.

Figure 4.2: Theoretical bounds in two-state hidden Markov model data. Left: Marginal coverage for different calibration set sizes and dependence levels; the guarantees under dependence converges to the iid case with larger calibration sizes. Right: Empirical coverage bound; while the worst-case bound decreases with dependence, empirical coverage remains close to the iid level.

is stationary as long as $|\lambda| < 1$. Figure 1.1 shows that marginal coverage remains very close to the prescribed nominal level $1 - \alpha = 0.9$ even as λ increases from 0 (iid) to 1. Autoregressive coefficients up to $\lambda = 0.99$ achieve coverage higher than 89%, and a significant loss of coverage only occurs when $\lambda = 0.999$.

Example 3 (Financial time series) We study split CP's performance on three real-world time series: the euro spot exchange rate (`eurusd`), Brent crude oil future (`bcousd`) and S&P 500 stock index future (`spxusd`). We compute minute-by-minute linear returns by dividing a price at minute t by the price at minute $t-1$ and subtracting 1. Due to market closures, Fridays and Sundays were discarded. We use gradient boosting to predict the price at time t using the prices at times $t-10, \ldots, t-1$. Then, we apply online conformal prediction over a sliding window of 1000 training points, 500 calibration points and 1 single test point for the entire year of 2021.

Figure 4.3 shows the daily coverage of split CP. The dashed black line represents the iid nominal coverage of 90% and the dashed orange one the marginal coverage over the entire year. Marginal coverage is slightly below 90%, but never drastically so. It is possible to use the guarantees provided by Theorem 4.1 to adjust split CP's quantile to achieve the desired nominal level.

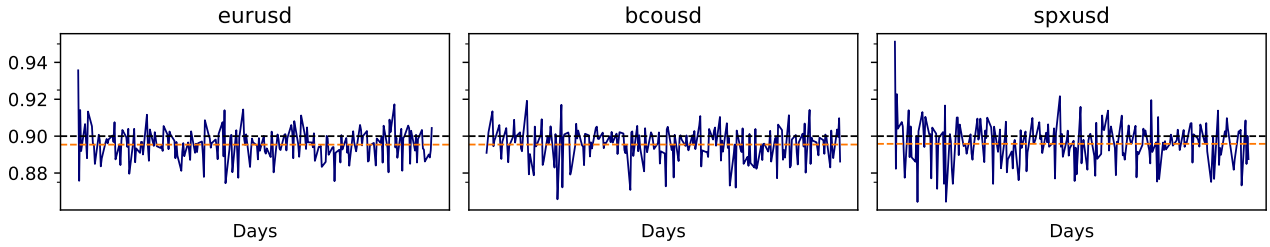


Figure 4.3: Daily marginal coverages of minute-by-minute online prediction for financial time series `eurusd`, `bcousd` and `spxusd` (solid blue), prescribed iid levels of $1 - \alpha = 0.9$ (dashed black) and observed marginal coverages over the entire year (dashed orange), above 0.895 in all cases.

Table 4.1 presents the conditional coverage (4.7) on four events of interest for all three financial datasets. Uptrend (respectively, downtrend) stands for two consecutive observations of positive (negative) returns. High (low) volatility events are taken to be those in which the standard deviation of the previous 10 returns observed is above (below) a given threshold. Note that conditioning on all such events still yields coverage close to the nominal iid level, on all three datasets. As previously noted, larger calibration sets have an important effect in improving coverage.

Dataset	Cal. set size	Conditional coverage			
		Uptrend	Downtrend	High vol.	Low vol.
eurusd	500	88.76%	88.82%	87.64%	90.07%
	1000	89.19%	89.17%	88.38%	90.19%
	5000	90.03%	89.98%	89.85%	90.08%
bcousd	500	88.94%	88.72%	87.10%	89.43%
	1000	89.35%	89.04%	87.65%	89.95%
	5000	89.78%	89.77%	89.33%	89.98%
spxusd	500	89.12%	89.01%	88.87%	89.68%
	1000	89.53%	89.48%	88.84%	90.03%
	5000	90.04%	89.73%	89.53%	90.30%

Table 4.1: Conditional coverage for distinct trend and volatility events and varying calibration set size (before conditioning). Note that empirical coverage is generally close to nominal iid level $1 - \alpha = 0.9$ and results improve given more calibration points.

4.5 Proofs and technical results

4.5.1 Proofs of Section 4.1 For the proofs below, we need to introduce certain population quantiles for $\widehat{s}_{\text{train}}(X_*, Y_*)$ conditionally on the training data.

Definition 4.11 (Conditional ϕ-quantile of the conformity score)**.** Given $\phi \in [0, 1)$, let $q_{\phi,\text{train}}$ denote the ϕ-quantile of $\widehat{s}_{\text{train}}(X_*, Y_*)$ conditioned on the training data; that is:

$$q_{\phi,\text{train}} \coloneqq \inf\{t \in \mathbb{R} : \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \leq t \mid \{(X_i, Y_i)\}_{i \in I_{\text{train}}}] \geq \phi\}.$$

Alternatively, define, for a deterministic $(x_i, y_i)_{i=1}^{n_{\text{train}}} \in (\mathcal{X} \times \mathcal{Y})^{n_{\text{train}}}$, the ϕ-quantile:

$$q_\phi((x_i, y_i)_{i=1}^{n_{\text{train}}}) \coloneqq \inf\{t \in \mathbb{R} : \mathbb{P}[s((x_i, y_i)_{i=1}^{n_{\text{train}}}, (X_*, Y_*)) \leq t] \geq \phi\},$$

and set $q_{\phi,\text{train}} \coloneqq q_\phi((X_i, Y_i)_{i \in I_{\text{train}}})$. We also define:

$$p_{\phi,\text{train}} \coloneqq \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \leq t \mid (X_i, Y_i)_{i \in I_{\text{train}}}]$$

Remark 4.12. When the conditional law of $\widehat{s}_{\text{train}}(X_*, Y_*)$ given the training data is continuous, we have $p_{\phi,\text{train}} = \phi$. Otherwise, it only holds that $p_{\phi,\text{train}} \geq \phi$.

Proof of Theorem 4.1. First we show that the event

$$F = \{q_{1-\alpha-\varepsilon_{\text{cal}},\text{train}} \leq \widehat{q}_{1-\alpha,\text{cal}}\},$$

satisfies $\mathbb{P}[F] \geq 1 - \delta_{\text{cal}}$. Indeed, by Definitions 4.1 and 4.11, if condition (4.5) holds, for any $\ell \in \mathbb{N}_{>0}$, with probability at least $1 - \delta_{\text{cal}}$,

$$\begin{aligned}\frac{1}{n_{\text{cal}}} \sum_{i \in I_{\text{cal}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \leq q_{1-\alpha-\varepsilon_{\text{cal}},\text{train}} - 1/\ell]} &\leq \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \leq q_{1-\alpha-\varepsilon_{\text{cal}},\text{train}} - 1/\ell] + \varepsilon_{\text{cal}} \\ &< 1 - \alpha - \varepsilon_{\text{cal}} + \varepsilon_{\text{cal}} = 1 - \alpha \\ &\leq \frac{1}{n_{\text{cal}}} \sum_{i \in I_{\text{cal}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \leq \widehat{q}_{1-\alpha,\text{cal}}]}.\end{aligned}$$

This implies that the event

$$E_\ell = \left\{ \frac{1}{n_{\text{cal}}} \sum_{i \in I_{\text{cal}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \leq q_{1-\alpha-\varepsilon_{\text{cal}},\text{train}} - 1/\ell]} < \frac{1}{n_{\text{cal}}} \sum_{i \in I_{\text{cal}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \leq \widehat{q}_{1-\alpha,\text{cal}}]} \right\}$$

satisfies $P[E_\ell] \geq 1 - \delta_{\text{cal}}$ for all $\ell \in \mathbb{N}_{>0}$, and since $E_{\ell+1} \subset E_\ell$, we have

$$1 - \delta_{\text{cal}} \leq \lim_{\ell \to \infty} \mathbb{P}[E_\ell] = \mathbb{P}[E_\infty],$$

where,

$$E_\infty = \left\{ \frac{1}{n_{\text{cal}}} \sum_{i \in I_{\text{cal}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \leq q_{1-\alpha-\varepsilon_{\text{cal}},\text{train}}]} \leq \frac{1}{n_{\text{cal}}} \sum_{i \in I_{\text{cal}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \leq \widehat{q}_{1-\alpha,\text{cal}}]} \right\},$$

proving that $\mathbb{P}[F] \geq 1 - \delta_{\text{cal}}$. Therefore, given $i \in I_{\text{test}}$, using the fact that the function $t \mapsto \mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq t]$ is increasing,

$$\begin{aligned}\mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq \widehat{q}_{1-\alpha,\text{cal}}] &\geq \mathbb{P}[\{\widehat{s}_{\text{train}}(X_i, Y_i) \leq \widehat{q}_{1-\alpha,\text{cal}}\} \cap F] \\ &\geq \mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq q_{1-\alpha-\varepsilon_{\text{cal}},\text{train}}] - \delta_{\text{cal}}.\end{aligned}$$

Hence, by condition (4.6) and a conditioning argument,

$$\begin{aligned}\mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq \widehat{q}_{1-\alpha,\text{cal}}] &\geq \mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq q_{1-\alpha-\varepsilon_{\text{cal}},\text{train}}] - \delta_{\text{cal}} \\ &\geq \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \leq q_{1-\alpha-\varepsilon_{\text{cal}},\text{train}}] - \varepsilon_{\text{test}} - \delta_{\text{cal}} \\ &\geq 1 - \alpha - \varepsilon_{\text{cal}} - \varepsilon_{\text{test}} - \delta_{\text{cal}},\end{aligned} \tag{4.21}$$

proving the first part of the theorem.

For the second part, note that by Definition 4.11 and condition (4.5) we have with probability at least

$1 - \delta_{\text{cal}}$,

$$\begin{aligned}\frac{1}{n_{\text{cal}}} \sum_{i \in I_{\text{cal}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i, Y_i) \leq q_{1-\alpha+\varepsilon_{\text{cal}},\text{train}}]} &\geq \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \leq q_{1-\alpha+\varepsilon_{\text{cal}},\text{train}}] - \varepsilon_{\text{cal}} \\ &\geq 1 - \alpha,\end{aligned}$$

and since $\widehat{q}_{1-\alpha-\varepsilon_{\text{cal}},\text{cal}}$ is the smallest possible value satisfying the expression above, the event

$$G = \{\widehat{q}_{1-\alpha,\text{cal}} \leq q_{1-\alpha+\varepsilon_{\text{cal}},\text{train}}\}$$

satisfies $\mathbb{P}[G] \geq 1 - \delta_{\text{cal}}$. Then,

$$\begin{aligned}\mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq \widehat{q}_{1-\alpha,\text{cal}}] &\leq \mathbb{P}[\{\widehat{s}_{\text{train}}(X_i, Y_i) \leq \widehat{q}_{1-\alpha,\text{cal}}\} \cap G] + \delta_{\text{cal}} \\ &\leq \mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq q_{1-\alpha+\varepsilon_{\text{cal}},\text{train}}] + \delta_{\text{cal}}.\end{aligned}$$

Hence, if $\widehat{s}_{\text{train}}(X_*, Y_*)$ almost surely has a continuous distribution conditionally on the training data, by condition (4.6)

$$\begin{aligned}\mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq \widehat{q}_{1-\alpha,\text{cal}}] &\leq \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \leq q_{1-\alpha+\varepsilon_{\text{cal}},\text{train}}] + \varepsilon_{\text{test}} + \delta_{\text{cal}} \\ &= 1 - \alpha + \varepsilon_{\text{cal}} + \varepsilon_{\text{test}} + \delta_{\text{cal}}.\end{aligned}$$

Putting this together with (4.21) concludes the second part. □

Proof of Theorem 4.2. Assuming that conditions (4.5) and (4.8) hold and using a similar argument as we did in the proof of Theorem 4.1, it is easy to show that the event

$$F = \{\widehat{q}_{1-\alpha-\varepsilon_{\text{cal}}-\varepsilon_{\text{test}},\text{test}} \leq \widehat{q}_{1-\alpha,\text{cal}}\}$$

satisfies $\mathbb{P}[F] \geq 1 - \delta_{\text{cal}} - \delta_{\text{test}}$. But then,

$$\begin{aligned}&\mathbb{P}\left[\frac{1}{n_{\text{test}}} \sum_{i \in I_{\text{test}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i, Y_i) \leq \widehat{q}_{1-\alpha,\text{cal}}]} \geq 1 - \alpha - \varepsilon_{\text{cal}} - \varepsilon_{\text{test}}\right] \\ &\geq \mathbb{P}\left[\frac{1}{n_{\text{test}}} \sum_{i \in I_{\text{test}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i, Y_i) \leq \widehat{q}_{1-\alpha-\varepsilon_{\text{cal}}-\varepsilon_{\text{test}},\text{test}}]} \geq 1 - \alpha - \varepsilon_{\text{cal}} - \varepsilon_{\text{test}}\right] - \mathbb{P}[F^c] \\ &\geq 1 - \delta_{\text{cal}} - \delta_{\text{test}},\end{aligned}$$

proving the first part. For the second part, note that

$$G = \{\widehat{q}_{1-\alpha+\varepsilon_{\text{cal}}+\varepsilon_{\text{test}},\text{test}} \geq \widehat{q}_{1-\alpha,\text{cal}}\}$$

also has probability at least $1 - \delta_{\text{cal}} - \delta_{\text{test}}$, therefore the event

$$F \cap G = \{\widehat{q}_{1-\alpha+\varepsilon_{\text{cal}}+\varepsilon_{\text{test}},\text{test}} \geq \widehat{q}_{1-\alpha-\varepsilon_{\text{cal}}-\varepsilon_{\text{test}},\text{test}}\}$$

has probability at least $1 - 2\delta_{\text{cal}} - 2\delta_{\text{test}}$

Hence, if $\widehat{s}_{\text{train}}(X_*, Y_*)$ almost surely has a continuous distribution conditionally on the training data, using the same argument as we did in the proof of Theorem 4.1 concludes the theorem. □

Proof of Theorem 4.3. Following the same strategy as in Theorem 4.1, we have that, with probability at least $1 - \delta_{\text{cal}}$, the event

$$F_{\text{cal}} = \{q_{1-\alpha-\varepsilon_{\text{cal}}}(A) \leq \widehat{q}_{1-\alpha,\text{cal}}(A),\ \forall A \in \mathcal{A}\}$$

satisfies $\mathbb{P}[F_{\text{cal}}] \geq 1 - \delta_{\text{cal}}$. Now, using the fact that the function

$$t \mapsto \mathbb{P}[\widehat{s}_{\text{train}}(X_k, Y_k) \leq t \mid X_k \in A]$$

is increasing, for any $k \in I_{\text{test}}$ and all $A \in \mathcal{A}$

$$\begin{aligned}
\mathbb{P}[\widehat{s}_{\text{train}}(X_k, Y_k) \leq \widehat{q}_{1-\alpha,\text{cal}}(A) \mid X_k \in A] &\geq \mathbb{P}[\{\widehat{s}_{\text{train}}(X_k, Y_k) \leq \widehat{q}_{1-\alpha,\text{cal}}(A)\} \cap F_{\text{cal}} \mid X_k \in A] \\
&\geq \mathbb{P}[\{\widehat{s}_{\text{train}}(X_k, Y_k) \leq q_{1-\alpha-\varepsilon_{\text{cal}}}(A)\} \cap F_{\text{cal}} \mid X_k \in A] \\
&\geq \mathbb{P}[\widehat{s}_{\text{train}}(X_k, Y_k) \leq q_{1-\alpha-\varepsilon_{\text{cal}}}(A) \mid X_k \in A] - \delta_{\text{cal}}.
\end{aligned}$$

Then,

$$\begin{aligned}
&\mathbb{P}[\widehat{s}_{\text{train}}(X_k, Y_k) \leq \widehat{q}_{1-\alpha,\text{cal}}(A) \mid X_k \in A] \\
&\geq \mathbb{P}[\widehat{s}_{\text{train}}(X_k, Y_k) \leq q_{1-\alpha-\varepsilon_{\text{cal}}}(A) \mid X_k \in A] - \delta_{\text{cal}} \\
&\geq \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \leq q_{1-\alpha-\varepsilon_{\text{cal}}}(A) \mid X_* \in A] - \varepsilon_{\text{train}} - \delta_{\text{cal}} \\
&= \mathbb{E}[\mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \leq q_{1-\alpha-\varepsilon_{\text{cal}}}(A) \mid (X_i, Y_i)_{i \in I_{\text{train}}}, X_* \in A]] - \varepsilon_{\text{train}} - \delta_{\text{cal}} \\
&\geq 1 - \alpha - \varepsilon_{\text{cal}} - \varepsilon_{\text{train}} - \delta_{\text{cal}},
\end{aligned}$$

proving the theorem. □

Proof of Theorem 4.4. Just as in Theorem 4.2, we have that the event

$$F = \{\widehat{q}_{1-\alpha-\varepsilon_{\text{cal}}-\varepsilon_{\text{test}},\text{test}}(A) \leq \widehat{q}_{1-\alpha,\text{cal}}(A),\ \forall A \in \mathcal{A}\}$$

satisfies $\mathbb{P}[F] \geq 1 - \delta_{\text{cal}} - \delta_{\text{test}}$ and the remaining of the proof is just a direct application of the definition

of conditional empirical quantile calibrated over the test data. □

Proof of application to the iid case. First, note that, in the iid case, when $i \in I_{\text{test}}$,

$$\mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq q_{\text{train}}] = \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \leq q_{\text{train}}],$$

showing that condition (4.6) holds with $\varepsilon_{\text{test}} = 0$.

Moreover, using the fact that $(\mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \leq q_{\text{train}}]})_{i=1}^n$ is an iid sample of bounded random variables, by Hoeffding's inequality, with probability at least $1 - \delta$,

$$\left| \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \leq q_{\text{train}}]} - P_{q,\text{train}} \right| \leq \sqrt{\frac{1}{2n} \log\left(\frac{2}{\delta}\right)}.$$

Therefore, taking

$$\varepsilon_{\text{cal}} = \sqrt{\frac{1}{2n_{\text{cal}}} \log\left(\frac{2}{\delta_{\text{cal}}}\right)} \quad \text{and} \quad \varepsilon_{\text{test}} = \sqrt{\frac{1}{2n_{\text{test}}} \log\left(\frac{2}{\delta_{\text{test}}}\right)} \tag{4.22}$$

proves conditions (4.5) and (4.8).

For conditional guarantees, note that, as in the marginal case, when $i \in I_{\text{test}}(A)$,

$$\mathbb{P}[\widehat{s}_{\text{train}}(X_i, Y_i) \leq q_{\text{train}}(A) \mid X_i \in A] = \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \leq q_{\text{train}}(A) \mid X_* \in A],$$

proving condition (4.10).

Next, suppose the family $\mathcal{A}$ has finite VC dimension $\text{VC}(\mathcal{A}) = d$. Now, if for some $\gamma > 0$, $\mathbb{P}[A] > \gamma$ for all $A \in \mathcal{A}$, it is straightforward to show that

$$\sup_{A \in \mathcal{A}} \left| P_{q,\text{train}}(A) - \frac{1}{n(A)} \sum_{i \in I(A)} \mathbf{1}_{[s(X_i,Y_i) \leq q_{\text{train}}]} \right| \leq \varepsilon,$$

where

$$\varepsilon = \frac{1}{\gamma} \left(4\sqrt{\frac{\log(2(n+1)^d)}{n}} + 2\sqrt{\frac{1}{2n} \log\left(\frac{4}{\delta}\right)} \right).$$

Thus, it is possible to pick n and δ to guarantee conditions (4.9) and (4.11). □

4.5.2 Proofs of Section 4.2.1 Our goal is to check conditions (4.5), (4.6) and (4.8) when $(X_i, Y_i)_{i=1}^n$ is a stationary β-mixing process. As stated in the main text, the main tool we will use is the so-called Blocking Technique [Yu94, MR10, KM17]. It allows one to measure the difference in expectation between a function of a β-mixing process and the same function over an independent pro-

cess, thereby transforming the original dependent problem into an independent one with the addition of a penalty factor.

Proposition 4.13 (Blocking Technique)**.** *Let $\{Z_t\}_{t=1}^T$ be a sample of a stationary β-mixing process. Split the sample into $2m$ interleaved blocks, with even blocks of size a and odd blocks of size b, such that $T = m(a+b)$. Denote each block by $B_j = \{Z_i\}_{i=l(j)}^{u(j)}$, where $l(j) = 1 + \lceil (j-2)/2 \rceil a + \lfloor j/2 \rfloor b$ and $u(j) = \lfloor j/2 \rfloor a + \lceil j/2 \rceil b$, so the set of odd blocks, each of size b, is given by $B_{\text{odd}} = (B_1, B_3, \ldots, B_{2m-1})$. Consider also the set $B^*_{\text{odd}} = (B^*_1, B^*_3, \ldots, B^*_{2m-1})$ where B^*_j are independent for $j = 1, 3, \ldots, 2m-1$, and $B^*_j \overset{d}{=} B_j$. If $h : \mathbb{R}^{mb} \to \mathbb{R}$ is a Borel-measurable function with $|h| \leq M$ for some $M > 0$, then*

$$|\mathbb{E}[h(B_{\text{odd}})] - \mathbb{E}[h(B^*_{\text{odd}})]| \leq 2M(m-1)\beta(a),$$

where $\beta(a)$ is the β-mixing coefficient of $\{Z_t\}_{t=1}^T$.

Using the Blocking Technique, we can prove that up to a error correction factor, we can transform our stationary β-mixing problem into a iid one:

Lemma 4.14 ([MR09])**.** *Let $Z_1, \ldots, Z_n$ be a sample drawn from a stationary β-mixing distribution. Split the sample into $2m$ blocks, with blocks of size a with $n = 2ma$. Denote the blocks by $B_j = \{Z_i\}_{i=l(j)}^{u(j)}$ where $l(j) = 1 + (j-1)a$ and $u(j) = ja$, with $B_{\text{odd}} = (B_1, B_3, \ldots, B_{2m-1})$. Call the independent version of B_{odd} by $B^*_{\text{odd}} = (B^*_1, B^*_3, \ldots, B^*_{2m-1})$, where B^*_j are independent with $B^*_j \overset{d}{=} B_j$, and let $\mathbb{P}_*$ be their law. Then,*

$$\mathbb{P}\left[\left|\mathbb{E}[Z_1] - \frac{1}{n}\sum_{i=1}^n Z_i\right| > \varepsilon\right] \leq 2\mathbb{P}_*\left[\left|\mathbb{E}\left[Z_1\right] - \frac{1}{ma}\sum_{Z_j \in B^*_{\text{odd}}} Z_j\right| > \varepsilon\right] + 4(m-1)\beta(a).$$

Finally, using Lemma 4.14 and Bernstein's Inequality 1.2, we are ready to prove a concentration inequality for stationary β-mixing processes.

Lemma 4.15. *Let $Z_1, \ldots, Z_n$ be a sample drawn from a stationary β-mixing distribution with $Z_1 \in [0,1]$ and $\text{Var}[Z_1] = v < \infty$. Then, for any $m, a, s \in \mathbb{N}_+$ with $m > 1$, $n = 2ma + s$ and $\delta > 4(m-1)\beta(a))$, with probability at least $1 - \delta$ it holds that*

$$\left|\mathbb{E}\left[Z_1\right] - \frac{1}{n}\sum_{i=1}^n Z_i\right| \leq \varepsilon,$$

where

$$\varepsilon = \widetilde{\sigma}(a)\sqrt{\frac{4}{n}\log\left(\frac{4}{\delta - 4(m-1)\beta(a)}\right)} + \frac{1}{3m}\log\left(\frac{4}{\delta - 4(m-1)\beta(a)}\right) + \frac{s}{n},$$

and

$$\widetilde{\sigma}(a) = \sqrt{v + \frac{2}{a}\sum_{k=1}^{a-1}(a-k)\beta(k)}.$$

Proof. By an application of Lemma 4.14 and Bernstein's inequality over the m independent blocks, with probability at least $1-\delta$

$$\left|\mathbb{E}\left[Z_1\right] - \frac{1}{n}\sum_{i=1}^{n} Z_i\right| \leq \frac{2ma}{n}\left|\mathbb{E}\left[Z_1\right] - \frac{1}{2ma}\sum_{i=1}^{2ma} Z_i\right| + \frac{s}{n} \tag{4.23}$$

$$\leq \sigma\sqrt{\frac{2}{m}\log\left(\frac{4}{\delta - 4(m-1)\beta(a)}\right)} \tag{4.24}$$

$$+ \frac{1}{3m}\log\left(\frac{4}{\delta - 4(m-1)\beta(a)}\right) + \frac{s}{n},$$

where $\sigma^2 = \mathrm{Var}\left[\frac{1}{a}\sum_{i:Z_i\in B_j} Z_i\right]$. To estimate σ^2, note that by stationarity,

$$\mathrm{Var}\left[\frac{1}{a}\sum_{i=1}^{a} Z_i\right] = \frac{1}{a}\mathbb{E}\left[Z_1^2\right] - \mathbb{E}[Z_1]^2 + \frac{1}{a^2}\sum_{k=1}^{a-1}\sum_{|i-j|=k}^{a}\mathbb{E}\left[Z_i Z_j\right].$$

Now, using the fact that $\{Z_i\}_{i\in B_j}$ is β-mixing, we have

$$\begin{aligned}
\mathrm{Var}\left[\frac{1}{a}\sum_{i=1}^{a} Z_i\right] &\leq \frac{1}{a}\mathbb{E}\left[Z_1^2\right] - \mathbb{E}[Z_1]^2 + \frac{1}{a^2}\sum_{k=1}^{a-1}\sum_{|i-j|=k}^{a}\Big(\mathbb{E}\left[Z\right]^2 + \beta(k)\Big) \\
&= \frac{1}{a}\mathbb{E}\left[Z_1^2\right] - \mathbb{E}[Z_1]^2 + \frac{1}{a^2}\sum_{k=1}^{a-1}\sum_{|i-j|=k}^{a}\mathbb{E}\left[Z\right]^2 + \frac{1}{a^2}\sum_{k=1}^{a-1}\sum_{|i-j|=k}^{a}\beta(k) \\
&= \frac{1}{a}\mathrm{Var}[Z_1] + \frac{1}{a^2}\sum_{k=1}^{a-1}\sum_{|i-j|=k}\beta(k) \\
&= \frac{1}{a}\left(\mathrm{Var}[Z_1] + \frac{2}{a}\sum_{k=1}^{a-1}(a-k)\beta(k)\right).
\end{aligned}$$

That is,

$$\sigma \leq \sqrt{\frac{1}{a}\left(v + \frac{2}{a}\sum_{k=1}^{a-1}(a-k)\beta(k)\right)}.$$

Plugging the above expression in (4.23) and using the fact that $2ma = n$, yields the result. □

Now we are ready to prove conditions (4.5), (4.6) and (4.8) for the stationary β-mixing case.

Proposition 4.16. *If $(X_i, Y_i)_{i=1}^n$ is stationary β-mixing, then condition (4.5) holds with*

$$\varepsilon_{\text{cal}} = \inf_{(a,m,r)\in F_{\text{cal}}} \left\{ \widetilde{\sigma}(a)\sqrt{\frac{4}{n_{\text{cal}}-r+1}\log\left(\frac{4}{\delta_{\text{cal}}-4(m-1)\beta(a)-\beta(r)}\right)} + \frac{1}{3m}\log\left(\frac{4}{\delta_{\text{cal}}-4(m-1)\beta(a)-\beta(r)}\right) + \frac{r-1}{n_{\text{cal}}} \right\}$$

for

$$F_{\text{cal}} = \{(a,m,r)\in\mathbb{N}_{>0}^3 \ : \ 2ma = n_{\text{cal}}-r+1,\ \delta_{\text{cal}} > 4(m-1)\beta(a)+\beta(r)\},$$

where

$$\widetilde{\sigma}(a) = \sqrt{\frac{1}{4}+\frac{2}{a}\sum_{k=1}^{a-1}(a-k)\beta(k)}.$$

Proof. We want to use Lemma 4.15 for the random variables

$$\left\{\mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i)\le q_{\text{train}}]}\right\}_{i\in I_{\text{cal}}},$$

however, since the random variables $(X_i, Y_i)_{i\in I_{\text{cal}}}$ are dependent to $\widehat{s}_{\text{train}}$ and the quantile q_{train}, we cannot simply apply the result. To fix this problem, it will be necessary to create a gap between our training and calibration data and use the Blocking Technique, Proposition 4.13, to transpose our problem to an independent setting.

For $\varepsilon > 0$ and $r \in \{1, \dots, n_{\text{cal}}\}$, let $I_{\text{cal},r} = \{n_{\text{train}}+r, \dots, n_{\text{train}}+n_{\text{cal}}\}$ and define the event

$$E(r,\varepsilon) = \left\{ \left| \frac{1}{n_{\text{cal}}-r+1} \sum_{i\in I_{\text{cal},r}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i)\le q_{\text{train}}]} - P_{q,\text{train}} \right| > \varepsilon \right\},$$

we want to show that there exists $\varepsilon > 0$ such that $\mathbb{P}[E(1,\varepsilon)] \le \delta$. Note that if $E(1,\varepsilon)$ holds, then

$$\left| \frac{1}{n_{\text{cal}}} \sum_{i\in I_{\text{cal},r}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i)\le q_{\text{train}}]} - P_{q,\text{train}} \right| > \varepsilon - \frac{r-1}{n_{\text{cal}}},$$

and since $n_{\text{cal}} \ge n_{\text{cal}} - r + 1$,

$$\left| \frac{1}{n_{\text{cal}}-r+1} \sum_{i\in I_{\text{cal},r}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i)\le q_{\text{train}}]} - P_{q,\text{train}} \right| > \varepsilon - \frac{r-1}{n_{\text{cal}}},$$

that is, $E(1,\varepsilon) \subset E(r, \varepsilon - (r-1)/n_{\text{cal}})$. Now, define

$$\mathbb{P}_* = \mathbb{P}_1^{n_{\text{train}}} \otimes \mathbb{P}_{n_{\text{train}}+r}^{n_{\text{train}}+n_{\text{cal}}},$$

so under $\mathbb{P}_*$, we have that $(X_i, Y_i)_{i \in I_{\text{train}}}$ and $(X_i, Y_i)_{i \in I_{\text{cal}}, r}$ are independent. Then, by Proposition 4.13,

$$\begin{aligned}
\mathbb{P}[E(1,\varepsilon)] &\leq \mathbb{P}[E(r, \varepsilon - (r-1)/n_{\text{cal}})] \\
&\leq \mathbb{P}_*[E(r, \varepsilon - (r-1)/n_{\text{cal}})] + \beta(r) \\
&= \mathbb{E}_*[\mathbb{P}_*[E(r, \varepsilon - (r-1)/n_{\text{cal}}) \mid (X_i, Y_i)_{i \in I_{\text{train}}}]] + \beta(r).
\end{aligned}$$

Note that by Lemma 4.15, for any $m, a \in \mathbb{N}_+$ with $n_{\text{cal}} - (r+s) + 1 = 2ma$ and $\delta_{\text{cal}} > 4(m-1)\beta(a))$, using the fact that $\text{Var}\left[\mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \leq q_{\text{train}}]}\right] \leq 1/4$, taking

$$\begin{aligned}
\varepsilon = \widetilde{\sigma}(a) \sqrt{\frac{4}{n_{\text{cal}} - r + 1} \log\left(\frac{4}{\delta - 4(m-1)\beta(a)}\right)} \\
+ \frac{1}{3m} \log\left(\frac{4}{\delta - 4(m-1)\beta(a)}\right) + \frac{r-1}{n_{\text{cal}}},
\end{aligned}$$

implies

$$\mathbb{P}_*[E(r, \varepsilon - (r-1)/n_{\text{cal}}) \mid (X_i, Y_i)_{i \in I_{\text{train}}}] \leq \delta_{\text{cal}},$$

hence,

$$\mathbb{P}[E(1,\varepsilon)] \leq \delta_{\text{cal}} + \beta(r).$$

which is equivalent to

$$\mathbb{P}[E(1,\varepsilon')] \leq \delta_{\text{cal}}$$

if we take

$$\begin{aligned}
\varepsilon' = \widetilde{\sigma}(a) \sqrt{\frac{4}{n_{\text{cal}} - r + 1} \log\left(\frac{4}{\delta - 4(m-1)\beta(a) - \beta(r)}\right)} \\
+ \frac{1}{3m} \log\left(\frac{4}{\delta - 4(m-1)\beta(a) - \beta(r)}\right) + \frac{s + r - 1}{n_{\text{cal}}}.
\end{aligned}$$

Finally, since this is true for any choice of $a, m, r \in \mathbb{N}_{>0}$ and $s \in \mathbb{N}$ with $s + 2ma = n_{\text{cal}} - r + 1$ and $\delta > 4(m-1)\beta(a) + \beta(r)$, we can choose a, m, r optimally such that the value of ε' is minimized and there is no need to optimize in s in this case. □

Proposition 4.17. *If $(X_i, Y_i)_{i=1}^n$ is stationary β-mixing, then condition (4.6) holds with*

$$\varepsilon_{\text{train}} = \beta(k - n_{\text{train}}).$$

Moreover, since $\beta(k - n_{\text{train}}) \leq \beta(1 - n_{\text{train}})$, it is possible to find $\varepsilon_{\text{train}}$ not depending on k.

Proof. Given $k \in I_{\text{test}}$, define

$$\mathbb{P}_* = \mathbb{P}_1^{n_{\text{train}}} \otimes \mathbb{P}_k^k,$$

so under $\mathbb{P}_*$ the random variable (X_k, Y_K) is independent of the training data $(X_i, Y_i)_{i \in I_{\text{train}}}$. Then, if $\beta_k = \beta(k - n_{\text{train}})$ we have,

$$\begin{aligned}
\beta_k &\geq |\mathbb{P}[\widehat{s}_{\text{train}}(X_k, Y_k) \leq q_{\text{train}}] - \mathbb{P}_*[\widehat{s}_{\text{train}}(X_k, Y_k) \leq q_{\text{train}}]| \\
&= |\mathbb{P}[\widehat{s}_{\text{train}}(X_k, Y_k) \leq q_{\text{train}}] - \mathbb{E}_*[\mathbb{P}_*[\widehat{s}_{\text{train}}(X_k, Y_k) \leq q_{\text{train}} \mid (X_i, Y_i)_{i \in I_{\text{train}}}]]| \\
&= |\mathbb{P}[\widehat{s}_{\text{train}}(X_k, Y_k) \leq q_{\text{train}}] - \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \leq q_{\text{train}} \mid (X_i, Y_i)_{i \in I_{\text{train}}}]|\,,
\end{aligned}$$

where the β_k penalty follows from Proposition 4.13. Note that the larger the k the smaller the penalty incurred by the dependence in the β-mixing process. Moreover, since

$$\beta_k \leq \beta_1,$$

it is possible to define $\varepsilon_{\text{train}} = \beta_1$ not depending on $k \in I_{\text{test}}$. □

Proposition 4.18. *If $(X_i, Y_i)_{i=1}^n$ is stationary β-mixing, then condition (4.8) holds with*

$$\begin{aligned}
\varepsilon_{\text{test}} = \inf_{(a,m) \in F_{\text{test}}} \Bigg\{ &\widetilde{\sigma}(a) \sqrt{\frac{4}{n_{\text{test}}} \log\left(\frac{4}{\delta_{\text{test}} - 4(m-1)\beta(a) - \beta(n_{\text{cal}})}\right)} \\
&+ \frac{1}{3m} \log\left(\frac{4}{\delta_{\text{test}} - 4(m-1)\beta(a) - \beta(n_{\text{cal}})}\right) + \frac{s}{n_{\text{test}}} \Bigg\}
\end{aligned}$$

for

$$F_{\text{test}} = \{(a, m, s) \in \mathbb{N}_{>0}^2 \times \mathbb{N} \ : \ s + 2ma = n_{\text{test}},\ \delta > 4(m-1)\beta(a) + \beta(n_{\text{cal}})\},$$

and

$$\widetilde{\sigma}(a) = \sqrt{\frac{1}{4} + \frac{2}{a} \sum_{k=1}^{a-1} (a - k)\beta(k)}.$$

Proof. The proof is similar to the proof of Proposition 4.16. Let the event $E(\varepsilon)$ be

$$E(\varepsilon) = \left\{ \left| P_{q,\text{train}} - \frac{1}{n_{\text{test}}} \sum_{i \in I_{\text{test}}} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \leq q_{\text{train}}]} \right| > \varepsilon \right\}.$$

Define,

$$\mathbb{P}_* = \mathbb{P}_1^{n_{\text{train}}} \otimes \mathbb{P}_{n_{\text{train}}+n_{\text{cal}}}^{n_{\text{train}}+n_{\text{cal}}+n_{\text{test}}},$$

so under $\mathbb{P}_*$ we have that $(X_i, Y_i)_{i \in I_{\text{train}}}$ and $(X_i, Y_i)_{i \in I_{\text{test}}}$ are independent. By Proposition 4.13 we have

$$\begin{aligned} \mathbb{P}[E(\varepsilon)] &\leq \mathbb{P}_*[E(\varepsilon)] + \beta(n_{\text{cal}}) \\ &= \mathbb{E}_*[\mathbb{P}_*[E(\varepsilon) \mid (X_i, Y_i)_{i \in I_{\text{train}}}]] + \beta(n_{\text{cal}}). \end{aligned}$$

Now we can apply Lemma 4.15 and conclude that, just as we did in Proposition 4.16, that for any $m, a \in \mathbb{N}_+$, $s \in \mathbb{N}$ with $n_{\text{test}} = 2ma - s$ and $\delta_{\text{test}} > 4(m-1)\beta(a)) + \beta(n_{\text{cal}})$, it is true that $\mathbb{P}[E(\varepsilon)] \leq \delta$, where

$$\begin{aligned} \varepsilon = \widetilde{\sigma}(a) \sqrt{\frac{4}{n_{\text{test}}} \log\left(\frac{4}{\delta_{\text{test}} - 4(m-1)\beta(a) - \beta(n_{\text{cal}})} \right)} \\ + \frac{1}{3m} \log\left(\frac{4}{\delta_{\text{test}} - 4(m-1)\beta(a) - \beta(n_{\text{cal}})} \right) + \frac{s}{n_{\text{test}}}, \end{aligned}$$

and

$$\widetilde{\sigma}(a) = \sqrt{\frac{1}{4} + \frac{2}{a} \sum_{k=1}^{a-1} (a-k)\beta(k)}.$$

Finally, since this is true for any choice of $a, m \in \mathbb{N}_{>0}$ and $s \in \mathbb{N}$, with $s + 2ma = n_{\text{test}}$ and $\delta_{\text{test}} > 4(m-1)\beta(a) + \beta(n_{\text{cal}})$, we can choose a, m, s optimally such that the value of ε is minimized. □

4.5.3 Proofs of Section 4.2.2 The proofs in this section are very similar to the proofs in Section 4.2.2, however, since we are dealing with a family of Borel measurable sets $\mathcal{A}$, we will need concentration results that allow us to uniformly control certain quantities over the family $\mathcal{A}$. First, we state such classical results for iid sequences.

Theorem 4.19 (Sauer-Shelah)**.** *Let $\mathcal{F}$ be a class of functions from $\mathcal{X}$ to $\{0,1\}$ with finite* VC *dimension* $\text{VC}(\mathcal{F}) = d$*. Then, for any integer $n \geq 1$,*

$$\mathcal{S}_{\mathcal{F}}(n) \leq (n+1)^d.$$

Theorem 4.20. *Let $Z_1, \ldots, Z_n$ be iid random variables taking values on $\mathcal{X}$ and $\mathcal{F}$ be a class of functions from $\mathcal{X}$ to $\{0,1\}$. Then*

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}}\left|\mathbb{E}f(Z) - \frac{1}{n}\sum_{i=1}^{n} f(Z_i)\right|\right] \leq 2\sqrt{\frac{\log(2\mathcal{S}_{\mathcal{F}}(n))}{n}}.$$

Using the Blocking Technique, we can prove that up to a error correction factor, we can transform our stationary β-mixing problem into a iid one. For example,

Lemma 4.21 ([MR09])**.** *Let $Z_1, \ldots, Z_n$ be a sample drawn from a stationary β-mixing distribution and $\mathcal{F}$ be a class of functions from $\mathcal{X}$ to $\{0,1\}$. Split the sample into $2m$ blocks, with blocks of size a with $n = 2ma$. Denote the blocks by $B_j = \{Z_i\}_{i=l(j)}^{u(j)}$ where $l(j) = 1 + (j-1)a$ and $u(j) = ja$, with $B_{\text{odd}} = (B_1, B_3, \ldots, B_{2m-1})$. Call the independent version of B_{odd} by $B^*_{\text{odd}} = (B^*_1, B^*_3, \ldots, B^*_{2m-1})$, where B^*_j are independent with $B^*_j \overset{d}{=} B_j$, and let $\mathbb{P}_*$ be their law. Then,*

$$\mathbb{P}\left[\sup_{f\in\mathcal{F}}\left|\mathbb{E}[f(Z_1)] - \frac{1}{n}\sum_{i=1}^{n} f(Z_i)\right| > \varepsilon\right] \leq 2\mathbb{P}_*\left[\sup_{f\in\mathcal{F}}\left|\mathbb{E}\left[f(Z_1)\right] - \frac{1}{ma}\sum_{Z_j\in B^*_{\text{odd}}} f(Z_j)\right| > \varepsilon\right] + 4(m-1)\beta(a).$$

Corollary 4.22. *Let $Z_1, \ldots, Z_n$ be a sample drawn from a stationary β-mixing distribution and $\mathcal{F}$ be a class of functions from $\mathcal{X}$ to $\{0,1\}$. Then, for any $a, m, s \in \mathbb{N}_+$, with $m > 1$, $n = 2ma + s$ and $\delta > 4(m-1)\beta(a)$, it holds that*

$$\mathbb{P}\left[\sup_{f\in\mathcal{F}}\left|\mathbb{E}\left[f(Z_1)\right] - \frac{1}{n}\sum_{i=1}^{n} f(Z_i)\right| \leq \varepsilon_0(a, m, \delta)\right] \geq 1 - \delta,$$

where

$$\varepsilon_0(a, m, s, \delta) = 2\sqrt{\frac{\log(2\mathcal{S}_{\mathcal{F}}(m))}{m}} + \sqrt{\frac{1}{2m}\log\left(\frac{4}{\delta - 4(m-1)\beta(a)}\right)} + \frac{s}{n}. \tag{4.25}$$

Proof. By an application of Lemma 4.21 and McDiarmids's inequality over the m independent blocks, it follows that

$$\mathbb{P}\left[\sup_{f\in\mathcal{F}}\left|\mathbb{E}[f(Z_1)] - \frac{1}{n}\sum_{i=1}^{n} f(Z_i)\right| > \varepsilon\right] \leq 4(e^{-2m\varepsilon'^2} + (m-1)\beta(a)). \tag{4.26}$$

where

$$\varepsilon' = \varepsilon - \mathbb{E}_* \left[\sup_{f \in \mathcal{F}} \left| \mathbb{E}[f(Z_1)] - \frac{1}{m} \sum_{j: B_j \in B^*_{\text{odd}}} \left(\frac{1}{a} \sum_{i: Z_i \in B_j} f(Z_i) \right) \right| \right] - \frac{s}{n}. \tag{4.27}$$

Denote by $Z_j^{(i)}$ the ith random variable of the jth block $B_j \in B^*_{\text{odd}}$, therefore the expectation in (4.27) can be written as

$$\begin{aligned}
&\mathbb{E}_* \left[\sup_{f \in \mathcal{F}} \left| \mathbb{E}[f(Z_1)] - \frac{1}{m} \sum_{j: B_j \in B^*_{\text{odd}}} \left(\frac{1}{a} \sum_{i: Z_i \in B_j} f(Z_i) \right) \right| \right] \\
&= \mathbb{E}_* \left[\sup_{f \in \mathcal{F}} \left| \mathbb{E}[f(Z_1)] - \frac{1}{a} \sum_{j=1}^{a} \left(\frac{1}{m} \sum_{i=1}^{m} f(Z_j^{(i)}) \right) \right| \right] \\
&\leq \frac{1}{a} \sum_{j=1}^{a} \mathbb{E}_* \left[\sup_{f \in \mathcal{F}} \left| \mathbb{E}[f(Z_1)] - \frac{1}{m} \sum_{i=1}^{m} f(Z_j^{(i)}) \right| \right],
\end{aligned}$$

where the inequality comes from the triangular inequality and the monotonicity of the supremum.

Note that in $\frac{1}{m} \sum_{i=1}^{m} f(Z_j^{(i)})$ we are considering only one element of each independent block $B_j \in B^*_{\text{odd}}$, therefore this is a sum over iid random variables. Hence, by Theorem 4.20

$$\mathbb{E}_* \left[\sup_{f \in \mathcal{F}} \left| \mathbb{E}[f(Z_1)] - \frac{1}{m} \sum_{j: B_j \in B^*_{\text{odd}}} \left(\frac{1}{a} \sum_{i: Z_i \in B_j} f(Z_i) \right) \right| \right] \leq 2 \sqrt{\frac{\log(2\mathcal{S}_{\mathcal{F}}(m))}{m}}.$$

That is,

$$\varepsilon' > \varepsilon - 2 \sqrt{\frac{\log(2\mathcal{S}_{\mathcal{F}}(m))}{m}} - \frac{s}{n}.$$

So taking $\delta > 4(e^{-2m\varepsilon'^2} + (m-1)\beta(a))$ and $\varepsilon = \varepsilon_0(a, m, \delta)$ yields

$$\mathbb{P} \left[\sup_{f \in \mathcal{F}} \left| \mathbb{E}[f(Z_1)] - \frac{1}{n} \sum_{i=1}^{n} f(Z_i) \right| > \varepsilon_0(a, m, \delta) \right] \leq \delta.$$

□

Corollary 4.23. *Let $(X_*, Y_*), \ldots, (X_n, Y_n)$ be a sample drawn from a stationary β-mixing distribution, $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be any deterministic function and $\mathcal{A}$ be a family of Borel measurable sets in $\mathcal{X}$ with finite* VC *dimension* $\text{VC}(\mathcal{A}) = d$.

Then, for any $m, a \in \mathbb{N}_+$ with $m > 1$, $n = 2ma$ and $\delta > 4(m-1)\beta(a)$, it holds that

$$\mathbb{P}\left[\sup_{A\in\mathcal{A}}\left|\mathbb{P}\left[X_* \in A\right] - \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{[X_i\in A]}\right| \leq \varepsilon_0(a,m,\delta)\right] \geq 1-\delta,$$

where $\varepsilon_0(a,m,\delta)$ is as defined in (4.25).

Proof. Taking $\mathcal{F}$ as

$$\mathcal{F} = \left\{x \mapsto \mathbf{1}_{[X_*\in A]} \ : \ A \in \mathcal{A}\right\},$$

in Corollary 4.22 and using Sauer-Shelah Theorem 4.19 yields the result. □

Lemma 4.24. *Let $X_1, \ldots, X_n$ be a sample drawn from a stationary β-mixing distribution, $\gamma \in (0,1)$ and $\mathcal{A}$ be a family of Borel measurable sets in $\mathcal{X}$ with finite* VC *dimension* $\mathrm{VC}(\mathcal{A}) = d$ *such that $\mathbb{P}[X_* \in A] > \gamma$ for all $A \in \mathcal{A}$. For $m, a \in \mathbb{N}_+$ with $m > 1$, $n = 2ma$ and $\delta > 4(m-1)\beta(a)$ suppose that $\frac{2}{\gamma}\varepsilon_0(a,m,\delta) < 1$, with $\varepsilon_0(a,m,\delta)$ as in (4.25). Then,*

$$\mathbb{P}\left[\inf_{A\in\mathcal{A}}\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{[X_i\in A]} > \frac{\gamma}{2}\right] \geq 1-\delta.$$

Proof. By Corollary 4.23, for any $m, a \in \mathbb{N}_+$ with $m > 1$, $n = 2ma$ and $\delta > 4(m-1)\beta(a)$, using the fact that $\varepsilon_0(a,m,\delta) < \gamma/2$,

$$\begin{aligned}
\mathbb{P}\left[\inf_{A\in\mathcal{A}}\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{[X_i\in A]} \leq \frac{\gamma}{2}\right] &= \mathbb{P}\left[\sup_{A\in\mathcal{A}}\gamma - \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{[X_i\in A]} \geq \frac{\gamma}{2}\right] \\
&\leq \mathbb{P}\left[\sup_{A\in\mathcal{A}}\left|\mathbb{P}[X_* \in A] - \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{[X_i\in A]}\right| \geq \frac{\gamma}{2}\right] \\
&\leq \mathbb{P}\left[\sup_{A\in\mathcal{A}}\left|\mathbb{P}[X_* \in A] - \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{[X_i\in A]}\right| \geq \varepsilon_0(a,m,\delta)\right] \\
&\leq \delta.
\end{aligned}$$

□

Lemma 4.25. *Let $(X_*, Y_*), \ldots, (X_n, Y_n)$ be a sample drawn from a stationary β-mixing distribution, $s : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ be a deterministic function, $\gamma \in (0,1)$ and $\mathcal{A}$ be a family of Borel measurable sets in $\mathcal{X}$ with finite* VC *dimension* $\mathrm{VC}(\mathcal{A}) = d$ *such that $\mathbb{P}[X_* \in A] > \gamma$ for all $A \in \mathcal{A}$. For $m, a \in \mathbb{N}_+$ with*

$m > 1$, $n = 2ma$ and $\delta > 8(m-1)\beta(a)$, if

$$\varepsilon := \frac{2}{\gamma}\varepsilon_0(a, m, \delta/2) < 1,$$

then with probability at least $1 - \delta$

$$\sup_{\substack{A\in\mathcal{A}\\ t\in\mathbb{R}}} \left| \frac{\mathbb{P}[s(X_*, Y_*) \le t, X_* \in A]}{\mathbb{P}[X_* \in A]} - \frac{\sum_{i=1}^n \mathbf{1}_{[s(X_i,Y_i)\le t]}\mathbf{1}_{[X_i\in A]}}{\sum_{i=1}^n \mathbf{1}_{[X_i\in A]}} \right| \le \varepsilon,$$

where $\varepsilon_0(a, m, \delta/2)$ as in (4.25).

Proof. Define ε as in the lemma statement. We want to show that:

$$C = \left\{ \sup_{\substack{A\in\mathcal{A}\\ t\in\mathbb{R}}} \left| \frac{\mathbb{P}[s(X_*, Y_*) \le t, X_* \in A]}{\mathbb{P}[X_* \in A]} - \frac{\sum_{i=1}^n \mathbf{1}_{[s(X_i,Y_i)\le t]}\mathbf{1}_{[X_i\in A]}}{\sum_{i=1}^n \mathbf{1}_{[X_i\in A]}} \right| > \varepsilon \right\}$$

has probability at most δ. To this end, we define the following auxiliary event, which controls the random denominator term in C:

$$B = \left\{ \inf_{A\in\mathcal{A}} \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{[X_i\in A]} > \frac{\gamma}{2} \right\}.$$

By Lemma 4.24, $\mathbb{P}[B^c] < \delta/2$, so it suffices to show that:

$$\textbf{Goal: } \mathbb{P}[E] \le \frac{\delta}{2}, \quad \text{where } E := C \cap B. \tag{4.28}$$

Note that, if E holds, then the quotient $\frac{\sum_{i=1}^n \mathbf{1}_{[s(X_i,Y_i)\leq t]}\mathbf{1}_{[X_i\in A]}}{\sum_{i=1}^n \mathbf{1}_{[X_i\in A]}}$ is well defined and

$$\begin{aligned}
\varepsilon &< \sup_{\substack{A\in\mathcal{A}\\ t\in\mathbb{R}}} \left| \frac{\mathbb{P}[s(X_*,Y_*)\leq t, X_*\in A]}{\mathbb{P}[X_*\in A]} - \frac{\sum_{i=1}^n \mathbf{1}_{[s(X_i,Y_i)\leq t]}\mathbf{1}_{[X_i\in A]}}{\sum_{i=1}^n \mathbf{1}_{[X_i\in A]}} \right| \\
&\leq \sup_{\substack{A\in\mathcal{A}\\ t\in\mathbb{R}}} \left| \frac{\mathbb{P}[s(X_*,Y_*)\leq t, X_*\in A] - \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{[s(X_i,Y_i)\leq t]}\mathbf{1}_{[X_i\in A]}}{\mathbb{P}[X_*\in A]} \right| \\
&+ \sup_{\substack{A\in\mathcal{A}\\ t\in\mathbb{R}}} \left| \frac{\sum_{i=1}^n \mathbf{1}_{[s(X_i,Y_i)\leq t]}\mathbf{1}_{[X_i\in A]}\left(\mathbb{P}[X_*\in A] - \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{[X_i\in A]}\right)}{\mathbb{P}[X_*\in A]\sum_{i=1}^n \mathbf{1}_{[X_i\in A]}} \right| \\
&\leq \sup_{\substack{A\in\mathcal{A}\\ t\in\mathbb{R}}} \left| \frac{\mathbb{P}[s(X_*,Y_*)\leq t, X_*\in A] - \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{[s(X_i,Y_i)\leq t]}\mathbf{1}_{[X_i\in A]}}{\gamma} \right| \\
&+ \sup_{A\in\mathcal{A}} \left| \frac{\mathbb{P}[X_*\in A] - \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{[X_i\in A]}}{\gamma} \right|,
\end{aligned}$$

Moreover, for any $A\in\mathcal{A}$:

$$\begin{aligned}
&\left| \frac{\mathbb{P}[X_*\in A] - \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{[X_i\in A]}}{\gamma} \right| \\
&= \lim_{t\to+\infty} \left| \frac{\mathbb{P}[s(X_*,Y_*)\leq t, X_*\in A] - \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{[s(X_i,Y_i)\leq t]}\mathbf{1}_{[X_i\in A]}}{\gamma} \right|
\end{aligned}$$

We deduce that:

$$E \text{ holds } \Rightarrow \varepsilon < 2 \sup_{A\in\mathcal{A}} \left| \frac{\mathbb{P}[X_*\in A] - \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{[X_i\in A]}}{\gamma} \right|. \tag{4.29}$$

By Corollary 4.23, we know that, with our choice of ε:

$$\mathbb{P}\left\{ \sup_{A\in\mathcal{A}} \left| \frac{\mathbb{P}[X_*\in A] - \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{[X_i\in A]}}{\gamma} \right| > \frac{\varepsilon}{2} \right\} \leq \frac{\delta}{2}.$$

By (4.29), we also have $\mathbb{P}[E]\leq\delta/2$. This finishes the proof by (4.28). □

Proposition 4.26. *Define*

$$\varepsilon = \inf_{(a,m,r)\in G_{\text{cal}}} \left\{ \frac{2}{\gamma}\left(\varepsilon_0\left(a, m, \frac{\delta-\beta(r)}{4}\right) + \frac{2(r-1)}{n_{\text{cal}}} \right) \right\}$$

where $\varepsilon_0(a,m,\delta/2)$ as in (4.25) and

$$G_{\text{cal}} = \{(a,m,r)\in\mathbb{N}_{>0}^3 \ : \ 2ma = n_{\text{cal}} - r + 1, \delta > 16(m-1)\beta(a) + \beta(r)\}.$$

If $\varepsilon < 1$, then condition (4.9) holds with $\varepsilon_{\text{cal}} = \varepsilon$.

Proof. The proof is similar to the proof of Proposition 4.16. For $\varepsilon > 0$ and $r \in \{1, \ldots, n_{\text{cal}}\}$, let

$$I_{\text{cal},r} = \{n_{\text{train}} + r, \ldots, n_{\text{train}} + n_{\text{cal}}\}$$

and

$$I_{\text{cal},r}(A) = \{i \in I_{\text{cal},r} \ : \ X_i \in A\}.$$

Define the events

$$E(r, \varepsilon') = \left\{ \inf_{A \in \mathcal{A}} \left| \frac{\sum_{i \in I_{\text{cal},r}(A)} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i, Y_i) \leq q_{\text{train}}]}}{\# I_{\text{cal},r}(A)} - P_{q,\text{train}}(A) \right| > \varepsilon' \right\},$$

and

$$C = \left\{ \inf_{A \in \mathcal{A}} \frac{1}{n_{\text{cal}}} \sum_{i \in I_{\text{cal}}} \mathbf{1}_{[X_i \in A]} > \frac{\gamma}{2} \right\},$$

and $B(r, \varepsilon') = E(r, \varepsilon') \cap C$. We want to show that there exists $\varepsilon' > 0$ such that if $\varepsilon' < 1$ then $\mathbb{P}[E(1, \varepsilon')] \leq \delta$.

Note that if $B(1, \varepsilon')$ holds, then for all $A \in \mathcal{A}$

$$\left| \frac{\sum_{i \in I_{\text{cal},r}(A)} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i, Y_i) \leq q_{\text{train}}]}}{\# I_{\text{cal},r}(A)} - P_{q,\text{train}}(A) \right| > \varepsilon' - \frac{2(r-1)}{\gamma n_{\text{cal}}}.$$

That is, $B(1, \varepsilon') \subset B\left(r, \varepsilon' - \frac{2(r-1)}{\gamma n_{\text{cal}}}\right)$. Now, define

$$\mathbb{P}_* = \mathbb{P}_1^{n_{\text{train}}} \otimes \mathbb{P}_{n_{\text{train}}+r}^{n_{\text{train}}+n_{\text{cal}}},$$

so under $\mathbb{P}_*$ we have that $(X_i, Y_i)_{i \in I_{\text{train}}}$ and $(X_i, Y_i)_{i \in I_{\text{cal},r}}$ are independent. By Proposition 4.13 we have

$$\begin{aligned} \mathbb{P}[B(1, \varepsilon')] &\leq \mathbb{P}\left[B\left(r, \varepsilon' - \frac{2(r-1)}{\gamma n_{\text{cal}}}\right)\right] \\ &\leq \mathbb{P}_*\left[B\left(r, \varepsilon' - \frac{2(r-1)}{\gamma n_{\text{cal}}}\right)\right] + \beta(r). \end{aligned}$$

But this implies that

$$\mathbb{P}[E(1, \varepsilon')] \leq \mathbb{P}_*\left[B\left(r, \varepsilon' - \frac{2(r-1)}{\gamma n_{\text{cal}}}\right)\right] + \beta(r) + 1 - \mathbb{P}[C].$$

For any $m, a \in \mathbb{N}_+$ with $n_{\text{cal}} - r + 1 = 2ma$ and $\delta_{\text{cal}} > 8(m-1)\beta(a)$, if we take

$$\varepsilon' = \frac{1}{\gamma}\left(4\sqrt{\frac{\log(2(m+1)^d)}{m}} + 2\sqrt{\frac{1}{2m}\log\left(\frac{8}{\delta - 8(m-1)\beta(a)}\right)} + \frac{2(r-1)}{n_{\text{cal}}}\right),$$

and assume that $\varepsilon' < 1$ then

$$\frac{1}{\gamma}\left(4\sqrt{\frac{\log(2(m+1)^d)}{m}} + 2\sqrt{\frac{1}{2m}\log\left(\frac{8}{\delta - 8(m-1)\beta(a)}\right)}\right) < 1$$

so Lemma 4.25 tells us that

$$\mathbb{E}_*\left[\mathbb{P}_*\left[B\left(r, \varepsilon' - \frac{2(r-1)}{\gamma n_{\text{cal}}}\right) \mid (X_i, Y_i)_{i \in I_{\text{train}}}\right]\right] \leq \delta$$

and Lemma 4.24 tells us $1 - \mathbb{P}[C] \leq \delta$. That is,

$$\mathbb{P}[E(1, \varepsilon')] \leq 2\delta + \beta(r),$$

which is equivalent to say that

$$\mathbb{P}[E(1, \varepsilon)] \leq \delta,$$

if ε is as in the proposition statement. □

Proposition 4.27. *Condition (4.10) holds with*

$$\varepsilon_{\text{train}} = \beta(k - n_{\text{train}}).$$

Proof. Given $k \in I_{\text{test}}$, note that we can decompose

$$\mathbb{P}_* = \mathbb{P}_1^{n_{\text{train}}} \otimes \mathbb{P}_k^k,$$

so under $\mathbb{P}_*$ we have that (X_k, Y_k) is independent of $(X_i, Y_i)_{i \in I_{\text{train}}}$. Then, defining $\beta_k = \beta(k - n_{\text{train}})$ we have for all $A \in \mathcal{A}$,

$$\beta_k \geq |\mathbb{P}[\widehat{s}_{\text{train}}(X_k, Y_k) \leq q_{\text{train}}(A), X_k \in A] - \mathbb{P}_*[\widehat{s}_{\text{train}}(X_k, Y_k) \leq q_{\text{train}}(A), X_k \in A]|$$

where the β_k penalty follows from Proposition 4.13. But then, by a conditioning argument,

$$\beta_k \geq |\mathbb{P}[\widehat{s}_{\text{train}}(X_k, Y_k) \leq q_{\text{train}}(A), X_k \in A] - \mathbb{P}[\widehat{s}_{\text{train}}(X_*, Y_*) \leq q_{\text{train}}(A), X_* \in A]|\,.$$

Finally, dividing by $\mathbb{P}[X_k \in A] = \mathbb{P}[X_* \in A]$ and using the fact that

$$\frac{\beta_k}{\mathbb{P}[X_k \in A]} \geq \beta_k$$

yields the result. □

Proposition 4.28. *Define*

$$\varepsilon = \inf_{(a,m,s)\in G_{\text{test}}} \left\{ \frac{2}{\gamma} \left(\varepsilon_0 \left(a, m, \frac{\delta - \beta(n_{\text{cal}})}{2} \right) \right) + \frac{s}{n_{\text{test}}} \right\}$$

where

$$G_{\text{test}} = \{(a,m) \in \mathbb{N}_{>0}^2 \ : \ s + 2ma = n_{\text{test}}, \delta > 8(m-1)\beta(a) + \beta(n_{\text{cal}})\}.$$

If $\varepsilon < 1$, then condition (4.11) holds with $\varepsilon_{test} = \varepsilon$.

Proof. The proof is similar to the proof of Proposition 4.26. Let the event $E(\varepsilon)$ be

$$E(\varepsilon) = \left\{ \inf_{A\in\mathcal{A}} \left| P_{q,\text{train}}(A) - \frac{\sum_{i\in I_{\text{test}}(A)} \mathbf{1}_{[\widehat{s}_{\text{train}}(X_i,Y_i) \leq q_{\text{train}}]]}}{n_{\text{test}}(A)} \right| > \varepsilon \right\},$$

Define,

$$\mathbb{P}_* = \mathbb{P}_1^{n_{\text{train}}} \otimes \mathbb{P}_{n_{\text{train}}+n_{\text{cal}}}^{n_{\text{train}}+n_{\text{cal}}+n_{\text{test}}},$$

so under $\mathbb{P}_*$ we have that $(X_i, Y_i)_{i\in I_{\text{train}}}$ and $(X_i, Y_i)_{i\in I_{\text{test}}}$ are independent. By Proposition 4.13 we have

$$\begin{aligned} \mathbb{P}[E(\varepsilon)] &\leq \mathbb{P}_*[E(\varepsilon)] + \beta(n_{\text{cal}}) \\ &= \mathbb{E}_*[\mathbb{P}_*[E(\varepsilon) \mid (X_i, Y_i)_{i\in I_{\text{train}}}]] + \beta(n_{\text{cal}}). \end{aligned}$$

Now we can apply Lemma 4.25 and conclude that if $\varepsilon < 1$, for any $m, a \in \mathbb{N}_+$ with $n_{\text{test}} = 2ma$ and $\delta_{\text{test}} > 8(m-1)\beta(a)) + \beta(n_{\text{cal}})$, it is true that $\mathbb{P}[E] \leq \delta$, where

$$\varepsilon = \frac{1}{\gamma} \left(4\sqrt{\frac{\log(2(m+1)^d)}{m}} + 2\sqrt{\frac{1}{2m} \log\left(\frac{8}{\delta - 8(m-1)\beta(a) - \beta(n_{\text{cal}})} \right)} + \frac{s}{n_{\text{test}}} \right).$$

Finally, since this is true for any choice of $a, m, s \in \mathbb{N}_{>0}$ with $s + 2ma = n_{\text{test}}$ and $\delta_{\text{test}} > 8(m-1)\beta(a) + \beta(n_{\text{cal}})$, we can choose a, m, s optimally such that the value of ε is minimized. □

4.5.4 Proofs of Section 4.3

Proof of Theorem 4.10. This proof is similar to the proof of Theorem 4.1. Indeed, if we consider $I_{\text{cal}} \coloneqq I_{\text{test}}$ in the proof of Theorem 4.1, the event

$$F = \{q_{1-\alpha-\varepsilon_{\text{cal}},\text{train}} \leq \widehat{q}_{1-\alpha,\text{cal}}\},$$

satisfies $\mathbb{P}[F] \geq 1 - \delta_{\text{cal}}$. But since,

$$\widehat{q}^{(i)}_{1-\alpha,\text{cal}} \geq \widehat{q}_{1-\alpha-1/n_{\text{test}},\text{cal}},$$

the following event

$$F' = \{q_{1-\alpha-\varepsilon_{\text{cal}}-1/n_{\text{test}},\text{train}} \leq \widehat{q}^{(i)}_{1-\alpha,\text{cal}}\},$$

also satisfies $\mathbb{P}[F'] \geq 1 - \delta_{\text{cal}}$. The rest of the proof follows the same strategy as in Theorem 4.1 using $\widehat{q}^{(i)}_{1-\alpha,\text{cal}}$ instead of $\widehat{q}_{1-\alpha,\text{cal}}$. □

Chapter 5

Conclusion

Concentration of measure and boosting techniques are powerful tools to build theoretical analyses of modern statistical learning models. In this thesis we have shown how one can apply such methods in three different Machine Learning sub-areas: optimization of non-decomposable metrics, hashing methods for Record Linkage, and the construction of predictive intervals.

In the optimization of non-decomposable metrics scenario, we introduced ExactBoost, a method which directly optimizes combinatorial and non-decomposable losses, instead of making use of surrogate functions as is often the case in standard boosting methods. We have shown in this thesis how it is possible to extend some concepts of concentration of measure and margin theory for this setting. More precisely, we prove bounds for the generalization error of our method for many important metrics with different levels of non-decomposability.

For the Record Linkage problem, we proposed a method that uses a variant of AdaBoost to learn a large-margin similarity classifier via a sample of similar/dissimilar items. This classifier can be used to build hash codes that significantly speed up searches for similar items in databases.

Finally, we combined concentration of measure and split conformal prediction, a popular tool to obtain predictive intervals for general statistical algorithms under exchangeable data assumptions. Then we shown how this theory can be used to obtain finite-sample marginal, empirical and conditional guarantees for large classes of non-exchangeable data.

Bibliography

[AAB+15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[AB21a] Alexandr Andoni and Daniel Beaglehole. Learning to hash robustly, guaranteed, 2021.

[AB21b] Anastasios Nikolas Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

[ABJM21] Anastasios Nikolas Angelopoulos, Stephen Bates, Michael Jordan, and Jitendra Malik. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*, 2021.

[Aga13] Shivani Agarwal. Surrogate regret bounds for the area under the roc curve via strongly proper losses. In *Conference on Learning Theory*, pages 338–353. PMLR, 2013.

[AI06] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 459–468, 2006.

[AMŠP20] Lukáš Adam, Václav Mácha, Václav Šmídl, and Tomáš Pevnỳ. General framework for binary classification on top samples. *arXiv preprint arXiv:2002.10923*, 2020.

[BAL+21] Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan.

Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, 68(6):1–34, 2021.

[BCFM00] Andrei Z Broder, Moses Charikar, Alan M Frieze, and Michael Mitzenmacher. Min-wise independent permutations. *Journal of Computer and System Sciences*, 60(3):630–659, 2000.

[BCMR12] Stephen Boyd, Corinna Cortes, Mehryar Mohri, and Ana Radovanovic. Accuracy at the top. In *Advances in neural information processing systems*, pages 953–961, 2012.

[BCRT20] Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 08 2020.

[BCRT22] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *arXiv preprint arXiv:2202.13415*, 2022.

[Ber] Serge Bernstein. Sur l'extension du théoréme limite du calcul des probabilités aux sommes de quantités dépendantes. *Mathematische Annalen*, 97:1–59.

[BFLS98] Peter Bartlett, Yoav Freund, Wee Sun Lee, and Robert E. Schapire. Boosting the margin: a new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651 – 1686, 1998.

[BJM06] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[BLM13] S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. OUP Oxford, 2013.

[BM02] Peter Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[Bra05] Richard C. Bradley. Basic Properties of Strong Mixing Conditions. A Survey and Some Open Questions. *Probability Surveys*, 2(none):107–144, 2005.

[BV10] Nitin Bhatia and Vandana. Survey of nearest neighbor techniques. 2010.

[CC02] Marine Carrasco and Xiaohong Chen. Mixing and moment properties of various garch and stochastic volatility models. *Econometric Theory*, 18(1):17–39, 2002.

[CG16] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[CGD21] M Cauchois, S Gupta, and J. Duchi. Knowing what you know: valid and validated confidence sets in multiclass and multilabel prediction. *Journal of machine learning research*, 22(81), 2021.

[Cha02] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, page 380–388, New York, NY, USA, 2002. Association for Computing Machinery.

[Chr12] Peter Christen. A survey of indexing techniques for scalable record linkage and deduplication. *IEEE Transactions on Knowledge and Data Engineering*, 24(9):1537–1555, 2012.

[Cla04] D E Clark. Practical introduction to record linkage for injury research. *Injury Prevention*, 10(3):186–191, 2004.

[CM03] Corinna Cortes and Mehryar Mohri. AUC optimization vs. error rate minimization. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pages 313–320, Cambridge, MA, USA, 2003. MIT Press.

[CPR+22] Daniel Csillag, Carolina Piazza, Thiago Ramos, João Vitor Romano, Roberto I. Oliveira, and Paulo Orenstein. Exactboost: Directly boosting the margin in combinatorial and non-decomposable metrics. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 9017–9049. PMLR, 28–30 Mar 2022.

[CWZ18] Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. Exact and robust conformal inference methods for predictive machine learning with dependent data. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 732–749. PMLR, 2018.

[CWZ21] Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. Distributional conformal prediction. *Proceedings of the National Academy of Sciences*, 118(48):e2107794118, 2021.

[DKKN17] Krzysztof Dembczyński, Wojciech Kotłowski, Oluwasanmi Koyejo, and Nagarajan Natarajan. Consistency analysis for binary classification revisited. In *International Conference on Machine Learning*, pages 961–969. PMLR, 2017.

[DLLG01] L. Devroye, G.Ł. Lugosi, G. Lugosi, and L. Gábor. *Combinatorial Methods in Density Estimation*. Springer Series in Statistics. Springer New York, 2001.

[Dou12] Paul Doukhan. *Mixing: properties and examples*, volume 85. Springer Science & Business

Media, 2012.

[ECPC19] Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. Sodeep: a sorting deep net to learn ranking loss surrogates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10792–10801, 2019.

[EIV07] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):1–16, 2007.

[FC19] Fang Fang and Yuanyuan Chen. A new approach for credit scoring by directly maximizing the kolmogorov–smirnov statistic. *Computational Statistics & Data Analysis*, 133:180–194, 2019.

[FFHO02] César Ferri, Peter Flach, and José Hernández-Orallo. Learning decision trees using the area under the ROC curve. In *Proceedings of the 19th International Conference on Machine Learning*, volume 2, pages 139–146, 2002.

[FISS03] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4(null):933–969, December 2003.

[FK20] Rizal Fathony and Zico Kolter. Ap-perf: Incorporating generic performance metrics in differentiable learning. In *International Conference on Artificial Intelligence and Statistics*, pages 4130–4140. PMLR, 2020.

[Fri01] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5):1189–1232, 10 2001.

[FS69] Ivan P. Fellegi and Alan B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[FS97] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997.

[FS99] Yoav Freund and Robert E. Schapire. A short introduction to boosting. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kaufmann, 1999.

[GC21] Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[GSST20] Josif Grabocka, Randolf Scholz, and Lars Schmidt-Thieme. Learning surrogate losses, 2020.

[GZPA19] Xi Gao, Han Zhang, Aliakbar Panahi, and Tom Arodz. Differentiable combinatorial losses through generalized gradients of linear programs. *arXiv preprint arXiv:1910.08211*, 2019.

[HJvE01] T. Hickey, Qun Ju, and Maarten van Emden. Interval arithmetic: From principles to implementation. *J. ACM*, 48:1038–1068, 2001.

[HPIM12] Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of Computing*, 8(14):321–350, 2012.

[HPW19] Yotam Hechtlinger, Barnabas Poczos, and Larry Wasserman. Cautious deep learning. *arXiv preprint arXiv:1805.09460*, 2019.

[JAN+20] Qijia Jiang, Olaoluwa Adigun, Harikrishna Narasimhan, Mahdi Milani Fard, and Maya Gupta. Optimizing black-box metrics with adaptive surrogates. In *International Conference on Machine Learning*, pages 4784–4793. PMLR, 2020.

[JBA22] Vilde Jensen, Filippo Maria Bianchi, and Stian Norman Anfinsen. Ensemble conformalized quantile regression for probabilistic time series forecasting. *arXiv preprint arXiv:2202.08756*, 2022.

[JH06] Jeff Jonas and Jim C. Harper. Effective counterterrorism and the limited role of predictive data mining. 2006.

[Joa05] Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pages 377–384, New York, NY, USA, 2005. Association for Computing Machinery.

[JQK14] Ke Jiang, Qichao Que, and Brian Kulis. Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval. 11 2014.

[KBH02] Chris Kelman, John Bass, and C.D.J. Holman. Research use of linked health data - a best practice protocol. *Australian and New Zealand journal of public health*, 26:251–5, 02 2002.

[KD09] Brian Kulis and Trevor Darrell. Learning to hash with binary reconstructive embeddings. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

[KG09] Brian Kulis and Kristen Grauman. Kernelized locality-sensitive hashing for scalable image search. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2130–2137, 2009.

[KM17] Vitaly Kuznetsov and Mehryar Mohri. Generalization bounds for non-stationary mixing processes. *Machine Learning*, 106(1):93–117, 2017.

[KNJ14] Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Online and stochastic gradient methods for non-decomposable loss functions. In *Advances in Neural Information Processing Systems*, pages 694–702, 2014.

[KNJ15] Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Surrogate functions for maximizing precision at the top. In *International Conference on Machine Learning*, pages 189–198. PMLR, 2015.

[KNRD14] Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S Dhillon. Consistent binary classification with generalized performance metrics. In *NIPS*, volume 27, pages 2744–2752. Citeseer, 2014.

[KP02] V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Ann. Statist.*, 30(1):1–50, 02 2002.

[KYK20] Sunwoo Kim, Haici Yang, and Minje Kim. Boosted locality sensitive hashing: Discriminative binary codes for source separation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 106–110, 2020.

[LGR+18] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

[LJZ14] Nan Li, Rong Jin, and Zhi-Hua Zhou. Top rank optimization in linear time. In *Advances in neural information processing systems*, pages 1502–1510, 2014.

[LY18] Siwei Lyu and Yiming Ying. A univariate bound of area under ROC. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 43–52. AUAI Press, 2018.

[McD98] Colin McDiarmid. *Concentration*, pages 195–248. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[Mok88] Abdelkader Mokkadem. Mixing properties of arma processes. *Stochastic processes and their applications*, 29(2):309–315, 1988.

[MR09] Mehryar Mohri and Afshin Rostamizadeh. Rademacher complexity bounds for non-i.i.d. processes. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in*

Neural Information Processing Systems, volume 21. Curran Associates, Inc., 2009.

[MR10] Mehryar Mohri and Afshin Rostamizadeh. Stability bounds for stationary φ-mixing and β-mixing processes. *Journal of Machine Learning Research*, 11(2), 2010.

[MRT12] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2 edition, 2018.

[MSS15] Daniel J. McDonald, Cosma Rohilla Shalizi, and Mark Schervish. Estimating beta-mixing coefficients via histograms. *Electronic Journal of Statistics*, 9:2855–2883, 2015.

[OCC13] Jonathan Oliver, Chun Cheng, and Yanggui Chen. Tlsh – a locality sensitive hash. In *2013 Fourth Cybercrime and Trustworthy Computing Workshop*, pages 7–13, 2013.

[OORR22] Roberto I. Oliveira, Paulo Orenstein, Thiago Ramos, and João Vitor Romano. Split conformal prediction for dependent data, 2022.

[PJA10] Loïc Paulevé, Hervé Jégou, and Laurent Amsaleg. Locality sensitive hashing: a comparison of hash function types and querying mechanisms. *Pattern Recognition Letters*, 31(11):1348–1358, August 2010.

[PP20] Marc Pfetsch and Sebastian Pokutta. Ipboost–non-convex boosting via integer programming. In *International Conference on Machine Learning*, pages 7663–7672. PMLR, 2020.

[PPVG02] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alexander Gammerman. Inductive confidence machines for regression. In *Proceedings of the 13th European Conference on Machine Learning*, ECML '02, pages 345–356. Springer-Verlag, 2002.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RPC19] Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[SF13] Robert E Schapire and Yoav Freund. Boosting: Foundations and algorithms. *Kybernetes*,

2013.

[SS18] Rebecca C. Steorts and Anshumali Shrivastava. Probabilistic blocking with an application to the syrian conflict. 2018.

[SV07] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *CoRR*, abs/0706.3188, 2007.

[SVSF14] Rebecca C. Steorts, Samuel L. Ventura, Mauricio Sadinle, and Stephen E. Fienberg. A comparison of blocking methods for record linkage, 2014.

[Tas18] Dirk Tasche. A plug-in approach to maximising precision at the top and recall at the top. *CoRR*, abs/1804.03077, 2018.

[TBCR19] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[Vap98] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[Ver18] R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

[VGS05] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer-Verlag, 2005.

[Win04] William E. Winkler. Methods for evaluating and creating data quality. *Information Systems*, 29(7):531–550, 2004. Data Quality in Cooperative Information Systems.

[Win06] William E Winkler. Overview of record linkage and current research directions. Technical report, BUREAU OF THE CENSUS, 2006.

[WTF08] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

[WZs+18] Jingdong Wang, Ting Zhang, jingkuan song, Nicu Sebe, and Heng Tao Shen. A survey on learning to hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):769–790, 2018.

[XX21] Chen Xu and Yao Xie. Conformal prediction interval for dynamic time-series. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on*

Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, pages 11559–11569. PMLR, 2021.

[Yu94] Bin Yu. Rates of Convergence for Empirical Processes of Stationary Mixing Sequences. *The Annals of Probability*, 22(1):94–116, 1994.

[ZXTW13] Shaodan Zhai, Tian Xia, Ming Tan, and Shaojun Wang. Direct 0-1 loss minimization and margin maximization with boosting. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 872–880. Curran Associates, Inc., 2013.