

Hierarchies of Relaxations for Online Prediction Problems with Evolving Constraints

Alexander Rakhlin

University of Pennsylvania

RAKHLIN@WHARTON.UPENN.EDU

Karthik Sridharan

Cornell University

SRIDHARAN@CS.CORNELL.EDU

Abstract

We study online prediction where regret of the algorithm is measured against a benchmark defined via evolving constraints. This framework captures online prediction on graphs, as well as other prediction problems with combinatorial structure. A key aspect here is that finding the optimal benchmark predictor (even in hindsight, given all the data) might be computationally hard due to the combinatorial nature of the constraints. Despite this, we provide polynomial-time *prediction* algorithms that achieve low regret against combinatorial benchmark sets. We do so by building improper learning algorithms based on two ideas that work together. The first is to alleviate part of the computational burden through random playout, and the second is to employ Lasserre semidefinite hierarchies to approximate the resulting integer program. Interestingly, for our prediction algorithms, we only need to compute the values of the semidefinite programs and not the rounded solutions. However, the integrality gap for Lasserre hierarchy *does* enter the generic regret bound in terms of Rademacher complexity of the benchmark set. This establishes a trade-off between the computation time and the regret bound of the algorithm.

1. Introduction

To motivate the general setting of the paper, let us start with an example. Consider the problem of node label prediction in an evolving social network. At each round, a new user joins the network and makes connections to some existing users. The observable part of a user's type is represented by a covariate vector (or, *side information*) that may consist of gender, age, education level, and other revealed characteristics. Suppose we are tasked with developing a system that predicts a "label" for the user, in a possible set of outcomes. For instance, our goal might be to conduct a successful marketing campaign; here, the unseen labels could stand for the type of product the user will buy. Having made the prediction, we observe the actual behavior of the person (such as a purchase) and suffer a cost if the prediction was wrong.

We would like to devise a framework for developing prediction algorithms for this problem. Several aspects require careful consideration. First, how do we phrase the goal of the forecaster? Second, how do we model the evolution of the graph, arrival of users, and users' covariate vectors? Third, how can we leverage global information dispersed in the network in order to make good predictions on the individual level? Last but not least, how do we develop computationally feasible prediction methods?

We depart from the usual regret minimization framework in several ways. First, instead of restricting the set of possible labelings based solely on the graph structure and edge weights, we model

the set \mathcal{F} through the number of satisfied constraints. To this extent, a graph structure is just a particular set of constraints that involve *pairs* of nodes (which we shall interchangeably call “items” or “individuals”). A more general constraint might involve groups of individuals, and this gives greater flexibility in modeling the overall interaction between the nodes. Formally, a constraint is an arbitrary binary or real-valued function from assignments of labels for a subset of nodes to $\mathbb{R}_{\geq 0}$. Within theoretical computer science, constraint satisfaction problems (CSPs) are a natural umbrella for such combinatorial problems as Max Cut, Unique Games, and Max k -SAT. Furthermore, under the Unique Games Conjecture, semidefinite relaxations are providing an optimal approximation ratio for every CSP (Raghavendra, 2008; Raghavendra and Steurer, 2009). One of the goals of his paper is to apply semidefinite relaxation techniques to the problem of online prediction with combinatorial constraints.

The second way in which we depart from the traditional work on online learning is in allowing constraints to be revealed in an online manner. For the example of a graph-based constraints, this means that the graph can be revealed to the forecaster sequentially. Moreover, we can think of the graph as *evolving* in time since identities of the nodes have little significance, except for being arguments to constraints. We assume that the probability distribution that governs this evolution is known to the forecaster. As a particular case, the distribution may put all the mass on the revelation of all the constraints at the first round, in which case the constraints (or, the graph) are “known ahead of time.” More generally, one may take graph evolution models studied in probability theory and in social networks research, and use these for the prediction problem. In addition to the evolution of constraints, we allow the forecaster to observe side information about the new node. This side information is, once again, stochastic and follows a distribution jointly with constraints and node identities.

While the constraints and side information are stochastic, the label is chosen in an adversarial way. We have in mind the situation where we can model the network structure and the distribution of people types, but the label (or, action) of the person is not easily modeled. Instead, this behavior can be best understood through *global information* within the network, not the local information. Such a global coherence of labels and the constraints is modeled through the comparator class \mathcal{F} .

It would appear that the overall framework involving constraints, side information, and adversarially chosen labels cannot yield computationally tractable algorithms. Yet we show that by moving to improper prediction algorithms one can develop computationally efficient methods for the problem with only slight worsening of the regret guarantees. As a first step towards developing efficient methods, we show that the knowledge of the overall distribution governing the presentation of constraints and the side information allows us to define a randomized method with a provable guarantee on prediction error. We analyze “random payout,” a method that simulates future constraints and side information and uses these hallucinated values in place of missing information. We show that such an algorithm (which arises from the relaxation framework in (Rakhlin et al., 2012)) has regret that is bounded by classical Rademacher complexity of \mathcal{F} given the constraints and side information.

The last missing piece in this story is how to calculate the next prediction given the random payout. Here, we show that the forecaster needs to compute a value with conditional Rademacher complexity as part of the objective. In general, the computation of Rademacher complexity is not a feasible task for the types of combinatorial constraints we have in mind. However, the online relaxation framework suggests that we may take a superset of \mathcal{F} (given the constraints and side information) and suffer regret of Rademacher complexity of this larger set. We propose to use

semidefinite hierarchies for this task. In particular, we define Lasserre hierarchy (Lasserre, 2001; Parrilo, 2003) to obtain polynomial-time prediction methods with a “knob” (level of the hierarchy) that trades off computational time and prediction performance as measured by the regret.

In this paper, two distinct uses of the word “relaxation” come together. *Online relaxations* are upper bounds on the minimax value of the multistage prediction problem (Rakhlin et al., 2012). One of a number of approaches for obtaining online relaxations is to increase the set of benchmark solutions. The latter is a relaxation in the sense of optimization, as we show in the paper. Indeed, in this case, online relaxations and optimization relaxations are put on the same footing, and any distinction between the two should be clear from the context.

We use semidefinite relaxations in a somewhat unconventional way because the end goal is the problem of *prediction*. The online relaxation requires us to compute the *value* of the relaxed objective rather than the integer solution. Sidestepping the need to round the solution is a nice feature of “improper” prediction methods. The integrality gap still comes into the picture, as it effectively quantifies the increase of Rademacher complexity for the larger set. Yet, the regret bound only requires *existence* of a rounding procedure with a given guarantee and not its implementation. Crucially, the multiplicative increase due to the integrality gap is a constant that enters the regret bound only, leaving the constant in front of the comparator (OPT) to be one! The way in which the power of semidefinite relaxations fuses with the power of online relaxations is rather fortuitous.

The statements proved in this paper have an interesting “modularity” property. As soon as one finds a rounding procedure with a smaller integrality gap, this gap can be immediately inserted in the regret upper bound of our method. The prediction algorithm itself does not change, as it does not need to round the solution. Further, since Lasserre hierarchies we are employing are known to be tighter than LP-based and other hierarchies, the integrality gap can be proved for these weaker approximation methods.

We remark that it has been noted in the literature by various authors that the problem of prediction can be solved in situations when the offline solution is NP-hard (see e.g. (Hazan et al., 2012; Christiano, 2014; Abernethy, 2010)). Our work can be seen as formally extending this statement to approximation schemes, with an additional knob for the computation-prediction tradeoff. We also remark that ideas similar in spirit have been proposed in (Chandrasekaran et al., 2012; Chandrasekaran and Jordan, 2013), among others, in the statistical (rather than online) setting. In particular, the recent paper of Barak and Moitra (2015) gives very strong guarantees for learning third-order tensors using the 6th level of the sum-of-squares hierarchy. The authors compute a tight bound on the Rademacher complexity of the relaxed norm. In fact in Kakade et al. (2009) approximation algorithms are used for the problem of online linear optimization problems of combinatorial nature. However in the regret bounds in this work the loss of the algorithm is not compared with one times the optimal action but rather approximation factor times the optimal.

In summary, our contribution involves a framework for online prediction of labels for individuals that appear in a streaming fashion, with side information about individuals and constraints being also revealed in an online manner. The labels themselves can be adversarially chosen, while we assume that the stochastic model of the constraints and side information is known a priori. We propose a general method that is based on random playout, and further propose a semidefinite relaxation for the resulting CSP-like problem. We prove several regret bounds for the prediction method in terms of integrality gaps. The method allows for a trade-off between computation time and performance guarantee.

This paper is organized as follows. After describing the setting in the next section, we present in Section 3 the formalism of online relaxations and state a generic random-payout algorithm with a regret guarantee in terms of the expected relaxation. In Section 4 we show that the relaxation based on classical Rademacher averages is “admissible”, and we state the computationally-difficult problem. In Section 5 we relax the problem in the SDP language of Lasserre hierarchy. Section 6 makes the connection between the integrality gap and the regret bound of the r -th level in the hierarchy. The main result here is Theorem 4 which gives a regret bound in terms of the Rademacher complexity and the integrality gap. We turn to an alternative “Lagrangian” form of the optimization problem in Section A and prove a regret bound for the r -th level of this form of relaxation (Theorem 7). Several examples are discussed in Section 7, and the paper is concluded with a lower bound in Section B which shows near-optimality of our methods in terms of prediction performance.

Notation We use the following shorthand notation: let $[n] \triangleq \{1, \dots, n\}$, $a_{1:t} \triangleq (a_1, \dots, a_t)$, $(a, b)_{1:t} = (a_1, b_1, \dots, a_t, b_t)$. We denote by $\Delta(A)$ the set of distributions on the set A .

2. Setting

On each round $t = 1, \dots, V$, the forecaster observes a new item along with side information $x_t \in \mathcal{X}_t \subseteq \mathcal{X}$ and a set \mathcal{C}_t of constraints. The forecaster then makes a prediction $\hat{y}_t \in \{1, \dots, \kappa\} \triangleq [\kappa]$ and observes the label $y_t \in [\kappa]$. The side information set \mathcal{X}_t may be time-varying, but is known to the forecaster. Each constraint $c \in \mathcal{C}_t$ is represented by a pair (S_c, R_c) where $S_c \subseteq \mathcal{V}$ and $R_c : [\kappa]^{S_c} \mapsto \mathbb{R}_{\geq 0}$. For an assignment $g \in [\kappa]^V$, we write $c(g)$ or $R_c(g)$ for the value of R_c on $g(S_c)$. To lighten the notation, let us introduce a shorthand $\mathcal{I}_t = (\mathcal{C}_t, x_t)$ for the associated constraints and the side information for the item.

Example 1 Let $\kappa = 2$ and let $g \in \{1, 2\}^V$ be an assignment of binary labels to vertices of an unweighted graph $G = (\mathcal{V}, E)$. Define a constraint c for each edge $(u, v) \in E$ by taking $S_c = (u, v)$ and $R_c(g_u, g_v) = \mathbf{1}\{g_u \neq g_v\}$. Any labeling g defines a partition of G , and the size of the cut is precisely $\sum_c c(g)$.

Let \mathcal{F} be a class of functions $\mathcal{X} \rightarrow [\kappa]$. Each $f \in \mathcal{F}$ gives rise to a vector $(f(x_1), \dots, f(x_V))$ of labelings of the items. Given x_1, \dots, x_V , each $f \in \mathcal{F}$ induces an assignment vector $[f(x_j)]_{j=1}^V \in [\kappa]^V$, and now $c([f(x_j)]_{j=1}^V)$ represents the value of the constraint c on this assignment.

Let $\cup \mathcal{C}_t = \cup_{t=1}^V \mathcal{C}_t$ denote the union of all the constraint sets. Given this union, as well as $x_{1:V}$, we define the subset of those functions that do not violate more than K constraints as

$$\mathcal{F}_K[\mathcal{I}_{1:V}] = \{f \in \mathcal{F} : \sum_{c \in \cup \mathcal{C}_t} c([f(x_1), \dots, f(x_V)]) \leq K\} \quad (1)$$

for some given $K \geq 0$.

Example 2 Continuing with Example 1, let $\mathcal{F} = \{f(x) = \mathbf{1}\{\langle w, x \rangle > \gamma\} + 1 : w \in \mathbb{R}^d\}$. The set in (1) is then the set of homogenous hyperplanes that classify the vertices of the graph with a margin γ in such a way that the cut is at most of size K .

Let $\ell(\hat{y}_t, y_t) = \mathbf{1}\{\hat{y}_t \neq y_t\}$ be the indicator loss function. The goal of the forecaster is phrased as minimization of regret

$$\mathbf{Reg} = \sum_{t=1}^V \ell(\hat{y}_t, y_t) - \inf_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \ell(f(x_t), y_t) \quad (2)$$

with respect to the (data-dependent) subset of \mathcal{F} . This definition forces the forecaster to perform nearly as well as the benchmark that satisfies the constraints up to a certain threshold.

We remark that the class \mathcal{F} is “pruned” as more information about the constraints arrives over time. This pruning in effect captures the global information in the network, which requires adherence of labelings (given locally by values of f on the side information) to the global structure of constraints. It is important to recognize that the forecaster faces a difficulty: the “pruned” set (1) of comparators can only be calculated in hindsight.

We assume that the constraints and side information are drawn from a distribution known to the forecaster. That is, given $\mathcal{I}_{1:t-1}$, we assume that the forecaster is able to draw samples from the conditional distributions $p(\mathcal{C}_t, x_t | \mathcal{I}_{1:t-1})$.

Example 3 (Preferential Attachment) *In the preferential attachment model, the set \mathcal{C}_t of constraints corresponds to a set of new edges connected to previously revealed nodes. The edges are drawn according to the node degree given by the set of edges $\mathcal{C}_{1:t-1}$. In this example, the distribution does not depend on side-information.*

Example 4 (Geometric Random Graphs) *We may allow x_t 's to be drawn from some fixed distribution that does not depend on the constraints. In turn, the constraints can be formed according to the side information. One example is a geometric random graph, where pairwise constraints (graph edges) are formed according to distances from the new random point which may be given by the distance between the side information vectors. It is known that such graphs have better spectral properties (Barak et al., 2011). The result in this paper indeed employ an average (rather than the worst-case) integrality gap and can take advantage of “nice” graphs.*

Example 5 (Unlabeled Data) *Rather than assuming the knowledge of the distribution of x_t 's, the random play-out algorithm introduced in the paper may tap into a pool of unlabeled data.*

Other examples of distributions include a variant of the stochastic block model (SBM). This generative process provides the simplest model of group formation (though we remark that we are not aiming to recovery a hidden labeling, which is the focus of much research on SBM).

The upper bounds on regret obtained in this paper will also hold for an intermediate time horizon $n \leq V$. This “anytime” property follows from the fact that constraints are only added, and not deleted. If one is only concerned with regret at time V , the deletion is easy to incorporate in the model.

Finally, let us mention that much of prior literature on online prediction on graphs requires the knowledge of the graph from the beginning. When the order in which nodes are presented is given to us in advance the problem is readily modeled by our setting via $\mathcal{X}_t = \{t\}$. We then write $f(x_t) = f(t)$, precisely the notation for a static expert (Cesa-Bianchi and Lugosi, 2006). On the other hand, the case when nodes are presented to us in adversarial fashion is not directly modeled by the presented setting. However, the algorithms presented here can be easily extended to such a scenario. Indeed, at every round t , we simply pick some prefixed order for remaining unseen nodes and make predictions assuming this is the order in which nodes will be presented. On similar lines as the inductive proof in (Cesa-Bianchi and Shamir, 2011), we can show that the algorithm enjoys the same regret against an adversarial ordering of nodes as the algorithm would for the case when the order is known in advance.

In summary, we presented a flexible problem definition that models the arrival of items and the evolution of constraints. The model encapsulates local information about the items. The goal of the forecaster is phrased as a *global* measure of coherence given all the information at the end of the day. The rest of the paper is focused on exhibiting randomized methods that provably minimize regret in this general framework efficiently.

3. Online Relaxations

The idea of online relaxations was studied in (Rakhlín et al., 2012) as a generic recipe for deriving prediction algorithms. The basic technique for our context is as follows. Consider for a moment the problem that does not involve constraints, and suppose x_1, \dots, x_V are provided to the forecaster ahead of time. At time t , the forecaster predicts $\hat{y}_t \in \mathcal{Y}$ and observes $y_t \in \mathcal{Y}$. Furthermore, suppose the comparator set \mathcal{G} of functions $\mathcal{X} \rightarrow \mathcal{Y}$ in the regret definition is fixed. Given a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$, an online relaxation **Rel** is a sequence of functions that satisfies two conditions. First is the dominance condition: for any sequence of instances $x_{1:V}$ and $y_{1:V}$,

$$\mathbf{Rel}(\mathcal{G} | y_{1:V}) \geq - \inf_{f \in \mathcal{G}} \sum_{t=1}^V \ell(f(x_t), y_t). \quad (3)$$

Second is the recursive condition: for any $t \in [V]$,

$$\inf_{q_t \in \Delta(\mathcal{Y})} \sup_{y_t \in \mathcal{Y}} \{ \mathbb{E}_{\hat{y}_t \sim q_t} [\ell(\hat{y}_t, y_t)] + \mathbf{Rel}(\mathcal{G} | y_{1:t}) \} \leq \mathbf{Rel}(\mathcal{G} | y_{1:t-1}). \quad (4)$$

A relaxation that satisfies these conditions is termed *admissible*. Given a relaxation **Rel** for a class \mathcal{G} , define an online learning algorithm which at time t , given instances $y_{1:t-1}$ and $x_{1:V}$, makes the random prediction \hat{y}_t by drawing from the distribution $q_t \in \Delta(\mathcal{Y})$ either given by

$$q_t = \operatorname{argmin}_{q \in \Delta(\mathcal{Y})} \sup_{y_t \in \mathcal{Y}} \{ \mathbb{E}_{\hat{y}_t \sim q} [\ell(\hat{y}_t, y_t)] + \mathbf{Rel}(\mathcal{G} | y_{1:t}) \},$$

or by any other choice that ensures admissibility of the relaxation. It can be easily shown that regret of such a strategy is upper bounded (in expectation and with high probability) by $\mathbb{E} [\mathbf{Rel}(\mathcal{G} | \emptyset)]$.

We now turn to the case of side-information and constraints being revealed to the forecaster sequentially. We would like to “lift” the admissibility technique to this situation. To start, assume that we have a relaxation that is admissible for any class $\mathcal{G} = \mathcal{F}_K[\mathcal{I}_{1:V}]$. We propose the following simple randomized strategy.

At time t , given $\mathcal{I}_{1:t} = (\mathcal{C}_s, x_s)_{s=1}^t$, draw $\mathcal{I}_{t+1:V} = (\mathcal{C}, x)_{t+1:V}$ from the known distribution \mathbf{p} . Pick distribution q_t over \mathcal{Y} as follows

$$\hat{q}_t(\mathcal{I}_{t+1:V}) = \operatorname{argmin}_{q \in \Delta(\mathcal{Y})} \sup_{y_t \in \mathcal{Y}} \{ \mathbb{E}_{\hat{y}_t \sim q} [\ell(\hat{y}_t, y_t)] + \mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:t}) \} \quad (5)$$

and make a randomized prediction according to $\hat{q}_t(\mathcal{I}_{t+1:V})$.

As mentioned in the introduction, the above randomized method is of a “random playout” style. The forecaster simulates future draws to solve the (otherwise difficult) problem in expectation. The next lemma guarantees a bound on the expected regret in terms of expected Rademacher complexity of the data-dependent class. The upper bound behaves as if the forecaster were able to integrate over the complete distribution \mathbf{p} on each round, despite the fact that the method only draws one sample.

Lemma 1 *Suppose \mathbf{Rel} is an admissible relaxation for any $\mathcal{F}_K[\mathcal{I}_{1:V}]$. Then the randomized algorithm given in (5) enjoys the performance guarantee*

$$\mathbb{E}[\mathbf{Reg}] \leq \mathbb{E}_{(\mathcal{C}, x)_{1:V}} [\mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] \mid \emptyset)]$$

The proof of this lemma is postponed to the appendix. We refer to (Rakhlin et al., 2012) for more details of the technique.

Of course, the question remains: how do we come up with admissible relaxations required by Lemma 1. This is the subject of the next section.

4. Rademacher-Based Relaxations

The previous section presented a generic randomized prediction algorithm when the forecaster can sample from the distribution \mathbf{p} that generates the constraint sets and the side information. In this section, we provide a specific form of the relaxation we can use, along with the corresponding regret bound. The forecaster will be required to solve κ optimization problems per round to obtain the randomized prediction for that round.

Let \mathcal{M} be a set of $V \times \kappa$ matrices such that for any $M \in \mathcal{M}$, every $t \in [V]$ and $k \in [\kappa]$, $M_{t,k} \in [0, 1]$ and $\sum_{k=1}^{\kappa} M_{t,k} \leq 1$. Given any class \mathcal{G} of functions $\mathcal{X} \rightarrow [\kappa]$ and side information $x_{1:V}$, we define a set of matrices $\mathcal{M}_{\mathcal{G}}$ as

$$\mathcal{M}_{\mathcal{G}} = \{M_f : f \in \mathcal{G}, M_{t,k} = \mathbf{1}\{f(x_t) = k\}\}.$$

If $\kappa = 2$, each M_f can be simply represented by a vector of binary labels that f assigns to x_1, \dots, x_V .

Lemma 2 *For any class \mathcal{G} of predictors, if $\mathcal{M}_{\mathcal{G}} \subseteq \mathcal{M}$, then the following relaxation is admissible for prediction with respect to class \mathcal{G} :*

$$\mathbf{Rel}(\mathcal{G} \mid y_{1:t}) = \mathbb{E}_{\epsilon_{t+1:V}} \left[\sup_{M \in \mathcal{M}} \left\{ 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} + \sum_{i=1}^t M_{i,y_i} \right\} \right] - t.$$

Here, each ϵ_j is a vector of independent Rademacher random variables and $\epsilon_{j,k}$ stands for the k^{th} coordinate of this vector. Further, the randomized strategy corresponding to the above relaxation is given by first drawing $\epsilon_{t+1:V}$ Rademacher vectors and then predicting \hat{y}_t according to

$$\hat{q}_t(\epsilon_{t+1:V}) = \underset{q \in \Delta([\kappa])}{\operatorname{argmin}} \sup_{y_t \in [\kappa]} \left\{ 1 - q[y_t] + \sup_{M \in \mathcal{M}} \left\{ 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} + \sum_{s=1}^t M_{s,y_s} \right\} - t \right\}.$$

Recall that Lemma 1 provides a generic randomized strategy that, at round t , generates the future instances $\mathcal{I}_{t+1:V}$ and then uses as a black box an admissible relaxation for function classes $\mathcal{F}_K[\mathcal{I}_{1:V}]$. By combining Lemma 2 and Lemma 1, we get the following randomized prediction strategy:

At time t , given side information $x_{1:t}$, constraint sets $\mathcal{C}_{1:t}$ and past labels $y_{1:t-1}$, draw $\mathcal{I}_{t+1:V}$ from \mathbf{p} . Next, draw Rademacher vectors $\epsilon_{t+1:V}$ and compute, for each $o \in [\kappa]$, the value

$$R_t(o) = \sup_{M \in \mathcal{M}(\mathcal{I}_{1:V})} \left\{ 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} + M_{t,o} + \sum_{s=1}^{t-1} M_{s,y_s} \right\} \quad (6)$$

where $\mathcal{M}(\mathcal{I}_{1:V})$ is some set of matrices such that $\mathcal{M}_{\mathcal{F}_K[\mathcal{I}_{1:V}]} \subseteq \mathcal{M}(\mathcal{I}_{1:V})$. Finally, predict \hat{y}_t by simply drawing from the randomized strategy

$$\hat{q}_t(\epsilon_{t+1:V}) = \operatorname{argmin}_{q \in \Delta_\kappa} \max_{o \in [\kappa]} \{1 - q[o] + R(o)\}. \quad (7)$$

Note that the step of solving for $\hat{q}_t(\epsilon_{t+1:V})$ can be done efficiently by first sorting $R_t(1), \dots, R_t(\kappa)$'s in descending order and then using a simple water filling argument to find $\hat{q}_t(\epsilon_{t+1:V})$.

For the algorithm outlined above, in view of Lemma 1, the expected regret is upper-bounded as:

$$\mathbb{E}[\mathbf{Reg}] \leq 2 \mathbb{E}_{(\mathcal{C}, x)_{1:V}} \mathbb{E}_{\epsilon_{1:V}} \left[\sup_{M \in \mathcal{M}(\mathcal{I}_{1:V})} \sum_{j=t}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} \right]. \quad (8)$$

Of course one could use $\mathcal{M}(\mathcal{I}_{1:V}) = \mathcal{M}_{\mathcal{F}_K[\mathcal{I}_{1:V}]}$. However in many prediction problems of interest, solving the optimization problem (i.e., computing $R_t(o)$) for this class might be computationally hard. Hence, for computational efficiency we shall use a superset of $\mathcal{M}_{\mathcal{F}_K[\mathcal{I}_{1:V}]}$. We pay for computational efficiency by having a worse regret bound given by the Rademacher complexity over the larger set $\mathcal{M}(\mathcal{I}_{1:V})$, rather than $\mathcal{M}_{\mathcal{F}_K[\mathcal{I}_{1:V}]}$. We investigate this topic in the next two sections.

5. Prediction Based on Lasserre SDP Hierarchy

In the previous section we provided a randomized prediction strategy based on any class of matrices $\mathcal{M}(\mathcal{I}_{1:V})$ that is a superset of $\mathcal{M}_{\mathcal{F}_K[\mathcal{I}_{1:V}]}$. In this section we will employ Semidefinite Programming and Lasserre hierarchies to solve for the values $R(o)$, defined in (6).

Let us begin with $\mathcal{M}_{\mathcal{F}_K[\mathcal{I}_{1:V}]}$ and relax the problem. By the definition of $\mathcal{M}_{\mathcal{F}_K[\mathcal{I}_{1:V}]}$, we can write down the optimization problem for each $o \in [\kappa]$ as

$$\begin{aligned} & \max_{M \in \mathcal{M}_{\mathcal{F}_K[\mathcal{I}_{1:V}]}} \left\{ 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} + M_{t,o} + \sum_{s=1}^{t-1} M_{s,y_s} \right\} \\ & = \max \left\{ 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} + M_{t,o} + \sum_{s=1}^{t-1} M_{s,y_s} \right\} \quad \text{s.t.} \quad \sum_{c \in \cup \mathcal{C}_t} c(M) \leq K, \quad M \in \mathcal{F}_{x_{1:V}} \end{aligned}$$

where, $\mathcal{F}_{x_{1:V}} = \{M \in \{0, 1\}^{V \times \kappa} : M_{t,i} = \mathbf{1}\{f(x_t) = i\}, f \in \mathcal{F}, t \in [V], i \in [\kappa]\}$. We shall assume throughout this section that for any $x_{1:V}$, the set $\mathcal{F}_{x_{1:V}}$ can be represented as $\{0, 1\}^{V \times \kappa} \cap \mathcal{P}^{x_{1:V}}$ where $\mathcal{P}^{x_{1:V}} \subset \mathbb{R}^{V \times \kappa}$ can be represented by linear constraints efficiently. The superscript with side information is to remind us that the constraints can depend on the side information presented. To best match semidefinite formulations found in the literature, we assume, $\mathcal{P}^{x_{1:V}} = \{M \in \mathbb{R}^{V \times \kappa} : \forall j \in [d], M^\top B^j \leq c_j\}$, an intersection of d linear constraints. (Henceforth, whenever we refer to a matrix M as a vector, we mean the vectorized form.) The reason for the assumption is that we would like to apply Lasserre Hierarchy to represent $\{0, 1\}^{V \times \kappa} \cap \mathcal{P}$. As an example, for the case of all possible static experts, we are interested in predicting as well as any labeling that violates at most K constraints and, hence, $\mathcal{P}^{x_{1:V}}$ is simply $[0, 1]^{V \times \kappa}$.

Given $y_{1:t-1}$, $o \in [\kappa]$, and a draw of $\epsilon_{t+1:V}$, we define the $V \times \kappa$ dimensional vector $Y^t(o)$ as $Y_{s,j}^t(o) = \mathbf{1}\{j = y_s\}$ for $s < t$, $j \in [\kappa]$, $Y_{s,j}^t(o) = 2\epsilon_{s,j}$ for $s > t$, $j \in [\kappa]$, and $Y_{s,j}^t(o) = \mathbf{1}\{j = o\}$ for $s = t$, $j \in [\kappa]$. With this notation, we write the linear objective as

$$2 \sum_{s=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{s,k} M_{s,k} + M_{t,o} + \sum_{s=1}^{t-1} M_{s,y_s} = M^\top Y^t(o).$$

We are now ready to write down the SDP relaxation that we shall solve for every round t and every $o \in [\kappa]$ (these are the $R_t(o)$'s from (6)). The optimization problem is based on the r^{th} level of Lasserre SDP relaxation, written in the vector form as follows. First, we introduce a vector $\mathbf{U}_{S,\alpha}$ for every $S \subset [V]$ with $|S| \leq r$ and every $\alpha \in [\kappa]^S$. The optimization problem is now written as

$$\text{SDP}_r^{1\text{st}}(Y, K) = \max A \quad (9)$$

$$\text{s.t. } \sum_{c \in \mathcal{U}^{\mathcal{C}_t}} \sum_{\alpha \in [q]^{S_c}} R_c(\alpha) \|\mathbf{U}_{(S_c, \alpha)}\|^2 \leq K \quad (10)$$

$$\langle \mathbf{U}_{(S_1, \alpha_1)}, \mathbf{U}_{(S_2, \alpha_2)} \rangle = 0 \quad \forall \alpha_1(S_1 \cap S_2) \neq \alpha_2(S_1 \cap S_2)$$

$$\langle \mathbf{U}_{(S_1, \alpha_1)}, \mathbf{U}_{(S_2, \alpha_2)} \rangle = \langle \mathbf{U}_{(S_3, \alpha_3)}, \mathbf{U}_{(S_4, \alpha_4)} \rangle \quad \forall S_1 \cup S_2 = S_3 \cup S_4, \alpha_1 \circ \alpha_2 = \alpha_3 \circ \alpha_4$$

$$\sum_{k=1}^{\kappa} \|\mathbf{U}_{(\{i\}, k)}\|^2 = 1, \quad \|\mathbf{U}_{\emptyset, \emptyset}\|^2 = 1 \quad \forall i \in [V]$$

$$\langle \mathbf{U}_{(S_1, \alpha_1)}, \mathbf{U}_{(S_2, \alpha_2)} \rangle \geq 0 \quad \forall S_1, S_2, \alpha_1, \alpha_2$$

$$\sum_{\substack{v \in \mathcal{V} \\ \beta \in [\kappa]^v}} \|\mathbf{U}_{(S \cup \{v\}, \alpha \circ \beta)}\|^2 B_{(v, \beta)}^j \leq c_j \|\mathbf{U}_{(S, \alpha)}\|^2 \quad \forall S, \alpha, j \in [d]$$

$$\sum_{\substack{v \in \mathcal{V} \\ \beta \in [\kappa]^v}} \|\mathbf{U}_{(S \cup \{v\}, \alpha \circ \beta)}\|^2 Y_{(v, \beta)} \geq A \|\mathbf{U}_{(S, \alpha)}\|^2 \quad \forall S, \alpha \quad (11)$$

where in the above $R_c \in [\kappa]^{S_c}$ is the constraint violation mapping corresponding to constraint c . The first constraint in the above program is the requirement that cumulative constraint violation does not exceed K . The rest of the constraints are standard (the notation $\alpha_1 \circ \alpha_2$ denotes the concatenated assignment of labels whenever the assignments don't have a mismatch on the common entries). The above formulation is similar to the formulation for CSP's using Lasserre hierarchy, and we refer to (Tulsiani, 2009; Raghavendra and Steurer, 2009; Guruswami and Sinop, 2013; Schoenebeck, 2008) for a more detailed treatment of the semidefinite relaxation technique.

In the above optimization problem, maximizing over A can be performed efficiently as follows. First for a given A , we assume that we can solve the following optimization problem:

$$\text{SDP}_r^{2\text{nd}}(Y, A) = \min \sum_{c \in \mathcal{U}^{\mathcal{C}_t}} \sum_{\alpha \in [q]^{S_c}} R_c(\alpha) \|\mathbf{U}_{S_c, \alpha}\|^2 \quad (12)$$

$$\text{under the constraints of } \text{SDP}_r^{1\text{st}} \text{ excluding constraint (10)} \quad (13)$$

To find the solution to the maximization problem in (9) we simply perform a binary search over A to find the largest A for which the value of the solution of (12) is smaller than K .

On each round $t \in [V]$ and for each $o \in [\kappa]$, we find the value of $\text{SDP}_r^{1\text{st}}(Y^t(o), K)$. This gives $R_t(o)$ in (6), and, consequently, the randomized prediction obtained from (7). One can think of the solution in $\mathcal{M}(\mathcal{I}_{1:V})$ as the projected solution from the r^{th} level Lasserre hierarchy SDP. Specifically think of $\mathcal{M}(\mathcal{I}_{1:V})$ as being described by set of vectors \mathbf{U} that satisfy the constraints of the SDP and $M_{j,k}$ as $\|\mathbf{U}_{(\{j\}, k)}\|^2$. It is important to note that for any constant level r , we obtain a poly-time algorithm. In the next session, we shall provide an analysis of the bound on the expected regret of this randomized strategy using the generic upper bound from (8).

6. Regret Bounds Based on *Existence* of Rounding Strategies

Let us define solutions to two other optimization problems in addition to the solution to $\text{SDP}^{1\text{st}}(Y, K)$. These programs are defined for the purposes of analysis only, and will serve as a step to upper bounding Rademacher complexity of the relaxed set. To this end, define:

$$\text{OPT}^{2\text{nd}}(Y, A) = \min_{c \in \mathcal{U}^{\mathcal{C}_t}} c(M) \quad \text{subject to} \quad Y^\top M \geq A, \quad M \in \mathcal{F}_{x_{1:V}} \quad (14)$$

$$\text{OPT}^{1\text{st}}(Y, K) = \max F^\top Y \quad \text{subject to} \quad \sum_{c \in \mathcal{U}^{\mathcal{C}_t}} c(M) \leq K, \quad M \in \mathcal{F}_{x_{1:V}} \quad (15)$$

Definition 3 Given $\mathcal{I}_{1:V} = (\mathcal{C}_t, x_t)_{1:V}$, we define the gap between the Lasserre SDP solution at level r in (12) and the optimization problem in (14) as $\text{gap}(r; \mathcal{I}_{1:V}) := \sup_{\epsilon, D \in [-V, V]} \frac{\text{OPT}^{2\text{nd}}(\epsilon, D)}{\text{SDP}_r^{2\text{nd}}(\epsilon, D)}$. Whenever the context of $\mathcal{C}_{1:V}, x_{1:V}$ is clear we will simply use $\text{gap}(r)$.

The following theorem provides a bound on the expected regret of the proposed randomized strategy based on gap. Observe that the regret bound only gains a multiplicative factor $\text{gap}(r)$ in the constraint K , as compared to the original class. Below we prove our main theorem providing a bound on the expected regret of the proposed strategy in terms of the Rademacher complexity of the original class with its violation budget K enlarged. Given sequence $(\mathcal{C}, x)_{1:V}$ and any $K > 0$, let

$$\text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}]) := \mathbb{E}_{\epsilon_{1:V}} \left[\sup_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{j=1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} \mathbf{1}\{f(x_j) = k\} \right]$$

The following theorem is a performance guarantee for the proposed prediction strategy.

Theorem 4 *If we use the r^{th} level Lasserre hierarchy and use the randomized strategy obtained from the solutions via (7), the bound on the expected regret of the forecaster is given by*

$$\mathbb{E}[\mathbf{Reg}] \leq 2 \mathbb{E}_{\mathcal{I}_{1:V}} \text{Rad}_V(\mathcal{F}_{\text{gap}(r) \cdot K}[\mathcal{I}_{1:V}])$$

A few remarks are in order. First, since in the above $\text{gap}(r)$ really refers to $\text{gap}(r; \mathcal{C}_{1:V}, x_{1:V})$, for $\mathcal{C}_{1:V}, x_{1:V}$ drawn from the known generation process, bounds can often be improved: the behavior is given by the *average* case gap rather than the worst case gap.

Second, we would like to stress that while the bounds in this section are provided in terms of integrality gaps, for the actual prediction algorithm we never require a rounding strategy. We only need existence of a rounding strategy with some integrality gap to provide bounds on the expected regret in terms of Rademacher complexity of the original class.

Third, as already mentioned in the introduction, the approximation factor multiplies the regret bound rather than the cumulative loss of the benchmark predictor. That is, regret is still with respect to $1 \times \text{OPT}$. As long as the integrality gap is not too large for $r = O(1)$ of the Lasserre hierarchy, we obtain polynomial-time algorithms even when the problem of finding the optimal benchmark predictor given all the instances and constraints might be computationally hard. This is due to the improper nature of the prediction algorithm.

The Lasserre hierarchy is known to be more powerful than the Sherali-Adams and Lovasz-Schrijver hierarchies. This means that if we use for our prediction strategy some $r \in \mathbb{N}$, then the $\text{gap}(r)$ we obtain is smaller than approximation guarantees provided by algorithms using Sherali-Adams or Lovász-Schrijver hierarchies at around the same r . Also clearly $\text{gap}(r) \leq \text{gap}(r')$ for

any $r' < r$. Thus we can use approximation guarantees proved for the same problems based on algorithms that use LP hierarchies at level r or smaller. In summary, a result about an integrality gap for any weaker relaxation has immediate implication for the regret bound.

So far, we considered the problem where the benchmark was minimizing the number of violated constraints. Alternatively one could think of \mathcal{F} being restricted across items by requiring that at least K constraints need to be satisfied. Much of the machinery presented here including the application of rounding results to obtain bounds on the expected regret can easily be extended to such problems (which consist of typical CSP type problems) and in these cases the SDP optimization problems we solve on every step would be replaced by maximization versions of the SDP relaxations with the appropriate level of Lasserre hierarchy.

7. Examples

We illustrate the results of this paper on two examples. The first uses the SDP formulation in Section 5, while the second example uses the penalized version of Section A.

Binary Classification of Nodes with Cut Constraints Let us consider a weighted version of the problem discussed in the introduction. Suppose we are given a weighted graph $G = (\mathcal{V}, E, W)$ where $W : E \mapsto [-1, 1]$. Let us consider the case of $x_t = t$ and there is no other side information. The benchmark class of predictors is all binary labelings of the nodes from the set $\mathcal{F}_K = \{f \in [2]^V : f^\top L f \leq K\}$, where L is graph Laplacian. This problem can be formulated easily in the generic form specified in this paper by adding one constraint c per edge $(u, v) \in E$ with $S_c = \{u, v\}$. The cost of the constraint violation is given by $R_c(\alpha) = 1 - W(e_{u,v})(2 \mathbf{1}\{\alpha(u) = \alpha(v)\} - 1)$. These constraints can in fact be rewritten as quadratic constraints and Lasserre SDP at level r for the $\text{SDP}_r^{2\text{nd}}$ problem in Eq. (12) is in fact the r^{th} level SDP relaxation to the quadratic integer programming with a single linear constraint given by the labels (corresponding to (11) in $\text{SDP}_r^{1\text{st}}$).

It is shown in (Guruswami and Sinop, 2013) that the value of a rounded solution with $O(r)$ levels of Lasserre hierarchy is no more than $2/\lambda_r(L)$ times OPT. Furthermore, the rounding is faithful, and hence concentration bounds hold for linear constraints (Guruswami and Sinop, 2013, Thm 6.1). Since the linear constraints are given by Rademacher random variables, standard concentration results tell us that de-randomization does not violate the constraints by more than $O(\sqrt{V})$. By tracing through the proof of Theorem 4, one can see that this extra $O(\sqrt{V})$ factor comes out additively in the final bound on Rademacher complexity. Since this factor is of smaller order than Rademacher complexity itself, the bound is not affected. We conclude that $\mathbb{E}[\mathbf{Reg}] \leq O\left(\mathbb{E}_{(\mathcal{C}, x)_{1:V}} \text{Rad}_V\left(\mathcal{F}_{\frac{2}{\min\{1, \lambda_r\}} \cdot K}[\mathcal{I}_{1:V}]\right)\right)$ where λ_r is the r^{th} smallest eigenvalue of the normalized Laplacian of the graph, and the algorithm runs in time $n^{O(r)}$. If the graph generation process is well behaved in terms of spectral values of the Laplacian—like in a preferential attachment model for the graph—then the bound we obtain is near optimal. As a crude upper bound on $\text{Rad}_V\left(\mathcal{F}_{\frac{2}{\min\{1, \lambda_r\}} \cdot K}[\mathcal{I}_{1:V}]\right)$ one can use $\sqrt{KV \max\{1, \lambda_r^{-1}\} \log V}$.

Beyond the binary prediction considered above, one can also analyze the problem of predicting one of $[\kappa]$ labels for each node of a graph. As an interesting set of constraints, one can consider the Unique-Games-type constraints for labelings of edges in the graph. As a benchmark we compare our cumulative loss to the cumulative loss of the labelings that violate at most K of the labeling constraints on edges. Similar to the previous example, this problem can also be written with quadratic

form for constraints. The integrality gap from (Guruswami and Sinop, 2013) yields a bound on regret in terms of Rademacher complexity of the original class where the constraint K is enlarged by factor of order $\max\{1, \lambda_r^{-1}\}$. Here, the de-randomization procedure incurs an additional $O(\sqrt{\kappa V})$ violation of the constraints, which, again, does not affect the final bound of Rademacher complexity.

Online Prediction with Metric Labeling Constraints In the metric labeling problem (Kleinberg and Tardos, 2002), one aims to assign one of κ labels to each of the V items, minimizing a combinatorial objective function consisting of two parts: assignment costs per item and separation costs based on pairs of items. This model subsumes MAP estimation in a Markov random field model.

More precisely, let $G = (\mathcal{V}, E, W)$ be a weighted graph with $W : E \rightarrow [0, 1]$. The cost of an assignment $g \in [\kappa]^V$ is written as

$$\sum_{v \in [V]} d_1(v, g_v) + \sum_{(u,v) \in E} W(u, v) d_2(g_u, g_v) \quad (16)$$

where $d_2 : [\kappa] \times [\kappa] \rightarrow \mathbb{R}_{\geq 0}$ is a metric on the space of labels and $d_1 : [V] \times [\kappa] \rightarrow \mathbb{R}_{\geq 0}$ is a cost of assigning a particular label to the node. d_2 is a metric on the space of labels, and this distance is multiplied by the edge weight, encouraging “similar” items to pay more for disagreeing labels.

To map this setting into our notation, we define two types of constraints. The first type of a constraint c is associated to a singleton set $S_c = \{v\}$ and cost $R_c(g) = d_1(v, g_v)$, for $g \in [\kappa]^V$. The second type corresponds to separation costs, and we define it through $S_c = \{u, v\}$ and $R_c(g) = W(u, v) d_2(g_u, g_v)$ if (u, v) is an edge, and 0 otherwise.

To exhibit a polynomial-time method with a provable regret bound, we turn to the penalized version of SDP, developed in Section A. We observe that both (Kleinberg and Tardos, 2002) and (Chekuri et al., 2004) study linear relaxations of the integer program and prove integrality gaps which are based on the separation costs. Specifically, (Chekuri et al., 2004) use a simple LP relaxation for the problem, and since Lassere hierarchy at any level $r \geq 1$ is strictly stronger than this Linear program, we can directly use the integrality gap from (Chekuri et al., 2004) to obtain our regret bound. More precisely, (Chekuri et al., 2004) shows that the integrality gap for the separation costs is $O(\log \kappa)$, while the assignment costs are exact and have no integrality gap (gap of 1). The overall integrality gap is then stated as $O(\log \kappa)$ by combining the two parts. However, for our purposes, it is important that the assignment costs are exact. To invoke the integrality gap result, we write the objective in (17) as (negative of) the total cost (16) with the linear part involving Y being incorporated into the assignment costs (per item). Since the values of Y could be negative, we may only appeal to Theorem 7 if there is no gap for the assignment costs. This is the case for the proof in (Chekuri et al., 2004), and we conclude that $\widetilde{\text{gap}}(r) = O(\log \kappa)$. Theorem 7 then ensures a regret bound of Rademacher complexity of the class, increased multiplicatively by $O(\log \kappa)$.

We remark that the metric labeling objective subsumes Multiway Cut, among other problems. The objective also subsumes the energy function of the Ising model. Constraints based on Multiway Cut and Ising model appear to be well-suited for modeling global information dispersed throughout the graph. Furthermore, as soon as a better integrality gap is proved for a particular instance of a problem (such as, say, a known constant integrality gap for metric labeling on planar graphs), it can be immediately used in the regret bound without changing the algorithm.

Acknowledgments

We thank David Steurer for many helpful discussions. We gratefully acknowledge the support of NSF under grants CAREER DMS-0954737 and CCF-1116928, ONR BRC Program on Decentralized Online Optimization, as well as Dean’s Research Fund.

References

- J. Abernethy. Can we learn to gamble efficiently? In *COLT*, pages 318–319, 2010.
- B. Barak and A. Moitra. Tensor prediction, Rademacher complexity and random 3-XOR. *arXiv preprint arXiv:1501.06521*, 2015.
- B. Barak, M. Hardt, T. Holenstein, and D. Steurer. Subsampling mathematical relaxations and average-case complexity. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 512–531. SIAM, 2011.
- N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- N. Cesa-Bianchi and O. Shamir. Efficient online learning via randomized rounding. In *Advances in Neural Information Processing Systems*, pages 343–351, 2011.
- V. Chandrasekaran and M.I. Jordan. Computational and statistical tradeoffs via convex relaxation. *Proceedings of the National Academy of Sciences*, 110(13):E1181–E1190, 2013.
- V. Chandrasekaran, B. Recht, P.A. Parrilo, and A.S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12(6):805–849, 2012.
- C. Chekuri, S. Khanna, J. Naor, and L. Zosin. A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM Journal on Discrete Mathematics*, 18(3):608–625, 2004.
- P. Christiano. Online local learning via semidefinite programming. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 468–474. ACM, 2014.
- V. Guruswami and A. K. Sinop. Rounding Lasserre SDPs using column selection and spectrum-based approximation schemes for graph partitioning and Quadratic IPs. *arXiv preprint arXiv:1312.3024*, 2013.
- E. Hazan, S. Kale, and S. Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. *CoRR*, abs/1204.0136, 2012. URL <http://arxiv.org/abs/1204.0136>.
- S. Kakade, A. Kalai, and K. Ligett. Playing games with approximation algorithms. *SIAM J. Comput.*, 39(3):1088–1106, 2009.
- J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Journal of the ACM (JACM)*, 49(5):616–639, 2002.
- J. B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.
- P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320, 2003.
- P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 245–254. ACM, 2008.
- P. Raghavendra and D. Steurer. How to round any CSP. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 586–594. IEEE, 2009.
- A. Rakhlin, O. Shamir, and K. Sridharan. Relax and randomize: From value to algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2150–2158, 2012.
- G. Schoenebeck. Linear level lasserre lower bounds for certain k-CSPs. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 593–602. IEEE, 2008.
- M. Tulsiani. CSP gaps and reductions in the Lasserre hierarchy. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing, STOC '09*, pages 303–312, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-506-2. doi: 10.1145/1536414.1536457. URL <http://doi.acm.org/10.1145/1536414.1536457>.

Appendix A. Penalized Version of Relaxation

In this section we consider a penalized version of the relaxation, putting the “ $\leq K$ ” constraint into the objective. We use the Lasserre hierarchy to solve the penalized version of the optimization problem. Let us write down the SDP corresponding to the r^{th} level of Lasserre hierarchy. To this end, we introduce a vector $\mathbf{U}_{S,\alpha}$ for every $S \subset [V]$ with $|S| \leq r$ and every $\alpha \in [\kappa]^S$. The optimization problem is written as

$$\begin{aligned} \text{SDP}_r^\lambda(Y, \lambda) = \min & \left\{ \lambda \sum_{c \in \mathcal{U}^{\mathcal{C}_t}} \sum_{\alpha \in [q]^{S_c}} R_c(\alpha) \|\mathbf{U}_{S_c, \alpha}\|^2 - \sum_{\substack{v \in \mathcal{V} \\ \beta \in [\kappa]^v}} \|\mathbf{U}_{(\{v\}, \beta)}\|^2 Y_{(v, \beta)} \right\} \quad (17) \\ \text{s.t. } & \langle \mathbf{U}_{(S_1, \alpha_1)}, \mathbf{U}_{(S_2, \alpha_2)} \rangle = 0 & \forall \alpha_1(S_1 \cap S_2) \neq \alpha_2(S_1 \cap S_2) \\ & \langle \mathbf{U}_{(S_1, \alpha_1)}, \mathbf{U}_{(S_2, \alpha_2)} \rangle = \langle \mathbf{U}_{(S_3, \alpha_3)}, \mathbf{U}_{(S_4, \alpha_4)} \rangle & \forall S_1 \cup S_2 = S_3 \cup S_4, \alpha_1 \circ \alpha_2 = \alpha_3 \circ \alpha_4 \\ & \sum_{k=1}^{\kappa} \|\mathbf{U}_{(\{i\}, k)}\|^2 = 1, \|\mathbf{U}_{\emptyset, \emptyset}\|^2 = 1 & \forall i \in [V] \\ & \langle \mathbf{U}_{(S_1, \alpha_1)}, \mathbf{U}_{(S_2, \alpha_2)} \rangle \geq 0 & \forall S_1, S_2, \alpha_1, \alpha_2 \\ & \sum_{\substack{v \in \mathcal{V} \\ \beta \in [\kappa]^v}} \|\mathbf{U}_{(S \cup \{v\}, \alpha \circ \beta)}\|^2 B_{(v, \beta)}^j \leq c_j \|\mathbf{U}_{(S, \alpha)}\|^2 & \forall S, \alpha, j \in [d] \end{aligned}$$

This SDP should be compared to $\text{SDP}_r^{\text{1st}}$. Notice that the constraint (10) now appears in the objective. We now prove a “penalized version” of Lemma 2. We will also provide an appropriate relaxation from which an efficient prediction strategy follows.

Let us define a slightly modified version of gap between the SDP solution and integral solution to the penalized optimization problem as follows. Define the optimization problem

$$\text{OPT}^\lambda(Y, \lambda) = \min \lambda \sum_{c \in \mathcal{U}^{\mathcal{C}_t}} c(M) - Y^\top M \quad \text{s.t. } M \in \mathcal{F}_{x_{1:V}} \quad (18)$$

Definition 5 Given $(\mathcal{C}_{1:V}, x_{1:V})$, we define the gap between the Lasserre SDP solution at level r in (17) and the optimization problem in (18) as

$$\widehat{\text{gap}}(r; \mathcal{C}_{1:V}, x_{1:V}) := \min \left\{ a : \forall \epsilon \in \{-1, 1\}^{V \times \kappa}, \text{SDP}_r^\lambda(Y, \lambda) \geq \text{OPT}^\lambda(Y, \lambda/a) \right\}$$

Whenever the context of $\mathcal{C}_{1:V}, x_{1:V}$ is clear we will simply write $\widehat{\text{gap}}(r)$.

That is the factor by which we only scale down the constraint costs but not the linear part.

Lemma 6 Given $(\mathcal{C}, x)_{1:V}$, let $\mathcal{G} = \mathcal{F}_K[\mathcal{I}_{1:V}]$, and fix any $\lambda > 0$. Let $\mathcal{L}\text{as}(r, \mathcal{F}_{x_{1:V}})$ denote the set of vectors \mathbf{U} ’s corresponding to the r th level Lasserre hierarchy—that is, vectors satisfying the constraints of the SDP in Eq. (17). The following relaxation is admissible for prediction with respect to \mathcal{G} :

$$\begin{aligned} \mathbf{Rel}(\mathcal{G} | y_{1:t}) = \mathbb{E}_{\epsilon_{t+1:V}} \sup_{\mathbf{U} \in \mathcal{L}\text{as}(r, \mathcal{F}_{x_{1:V}})} & \left\{ 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} \|\mathbf{U}_{(\{j\}, k)}\|^2 + \sum_{s=1}^t \|\mathbf{U}_{(\{s\}, y_s)}\|^2 \right. \\ & \left. - \lambda \sum_{c \in \mathcal{C}_{1:V}} \sum_{\alpha \in [q]^{S_c}} R_c(\alpha) \|\mathbf{U}_{(S_c, \alpha)}\|^2 \right\} - t + \lambda K \end{aligned}$$

Further, the randomized strategy corresponding to the above relaxation is given by first drawing $\epsilon_{t+1:V}$ Rademacher vectors and then predicting \hat{y}_t according to

$$\begin{aligned} \hat{q}_t(\epsilon_{t+1:V}) = \operatorname{argmin}_{q \in \Delta([\kappa])} \sup_{y_t \in [\kappa]} & \left\{ \sup_{\mathbf{U} \in \mathcal{L}\text{as}(r, \mathcal{F}_{x_{1:V}})} \left\{ 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} \|\mathbf{U}_{(\{j\}, k)}\|^2 + \sum_{s=1}^t \|\mathbf{U}_{(\{s\}, y_s)}\|^2 \right. \right. \\ & \left. \left. - \lambda \sum_{c \in \mathcal{C}_{1:V}} \sum_{\alpha \in [q]^{S_c}} R_c(\alpha) \|\mathbf{U}_{(S_c, \alpha)}\|^2 \right\} - q[y_t] \right\}. \end{aligned}$$

As before, each ϵ_j is a vector of independent Rademacher random variables and $\epsilon_{j,k}$ stands for the k^{th} coordinate of this vector.

Let us bound the regret of the algorithm. To this end, assume we have a bound on the gap for the penalized SDP.

Theorem 7 *Suppose that for any $c \geq 1$,*

$$\text{Rad}_V(\mathcal{F}_{cK}[\mathcal{I}_{1:V}]) \leq c^p \text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}])$$

for some $p \leq 1$. With the notation of Lemma 6, if we choose

$$\lambda^* = \sup \left\{ \lambda : \lambda K \leq \mathbb{E}_{\epsilon_{1:V}} \sup_{\mathbf{U} \in \mathcal{L}_{\text{as}}(r, \mathcal{F}_{x_{1:V}})} \left\{ 2 \sum_{t=1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} \|\mathbf{U}_{\{j\},k}\|^2 - \lambda \sum_{\substack{c \in \cup_t \mathcal{C}_t \\ \alpha \in [q]^{S_c}}} R_c(\alpha) \|\mathbf{U}_{(S_c, \alpha)}\|^2 \right\} \right\} \quad (19)$$

the final relaxation is upper bounded by

$$\mathbf{Rel}(G \mid \emptyset) \leq 4 \widehat{\text{gap}}(r) \text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}]).$$

In view of Lemma 1, the expected regret of the strategy described in (5) is upper bounded as

$$\mathbb{E}[\mathbf{Reg}] \leq 4 \widehat{\text{gap}}(r) \mathbb{E}_{\mathcal{I}_{1:V}} \text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}]).$$

To estimate λ^* in (19), we use the concentration property of Rademacher complexity. We sample Rademacher random variables, constraints, and side information. Next we optimize over the Lasserre SDP at level r multiple times to find the maximal λ that satisfies the inequality

$$\lambda K \leq \sup_{\mathbf{U} \in \mathcal{L}_{\text{as}}(r, \mathcal{F}_{x_{1:V}})} \left\{ 2 \sum_{t=1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} \|\mathbf{U}_{\{t\},k}\|^2 - \lambda \sum_{c \in \cup_t \mathcal{C}_t} \sum_{\alpha \in [q]^{S_c}} R_c(\alpha) \|\mathbf{U}_{(S_c, \alpha)}\|^2 \right\}.$$

Appendix B. A Lower Bound

In this short section we prove a lower bound, showing that the algorithms we developed are near-optimal in terms of regret guarantees. We first consider the case of binary classification with $\kappa = 2$. We show a simple lower bound on the expected regret in terms of the Rademacher complexity of the constrained set of predictors. Next, we use the binary case lower bound to obtain a lower bound for the general case when $\kappa > 2$. We show that the worst case regret of any prediction strategy is lower bounded by $1/\kappa$ times the Rademacher complexity. In summary, as long as the integrality gap is of constant order, and the Rademacher complexity of the class only depends polynomially on K , the upper bounds we obtained are optimal up to a constant factor indicated by the gap.

Proposition 8 *For any K , any generating process that produces $(x_t, \mathcal{C}_t)_{t=1}^V$ and any class of benchmark predictors $\mathcal{F} \subset [2]^{\mathcal{X}}$, there exists a strategy of labelings such that the following bound on the expected regret holds for any prediction algorithm:*

$$\mathbb{E}[\mathbf{Reg}] \geq \frac{1}{2} \mathbb{E}_{(\mathcal{C}, x)_{1:V}} \text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}])$$

Corollary 9 *For any K , any generating process that produces $(x_t, \mathcal{C}_t)_{t=1}^V$ and any class of benchmark predictors $\mathcal{F} \subset [\kappa]^{\mathcal{X}}$, there exists a strategy of labelings such that the following bound on the expected regret holds for any prediction algorithm:*

$$\mathbb{E}[\mathbf{Reg}] \geq \frac{1}{\kappa} \mathbb{E}_{(\mathcal{C}, x)_{1:V}} \text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}])$$

Appendix C. Proofs

Proof [Proof of Lemma 1] At time t , given $\{\mathcal{C}_s, x_s\}_{s=1}^t$, let q_t be a strategy defined by first drawing the random variables $\mathcal{I}_{t+1:V} = (\mathcal{C}, x)_{t+1:V}$ and then solving for the randomized strategy \hat{q}_t defined in (5). We shall first prove the following inequality for any $t \in [V]$:

$$\mathbb{E}_{\mathcal{C}_t, x_t} \left[\sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} [\ell(\hat{y}_t, y_t)] + \mathbb{E}_{\mathcal{I}_{t+1:V}} [\mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:t})] \right\} \right] \leq \mathbb{E}_{\mathcal{I}_{t:V}} [\mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:t-1})] \quad (20)$$

We will prove the above statement for any $t \in [V]$ by first starting from base case $t = V$ and then working backward inductively. To this end consider the very last step. Given $\mathcal{C}_{1:V}, x_{1:V}, y_{1:V-1}$,

$$\begin{aligned} & \sup_{y_V} \left\{ \mathbb{E}_{\hat{y}_V \sim q_V} [\ell(\hat{y}_V, y_V)] + \mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:V}) \right\} \\ &= \inf_{q_V} \sup_{y_V} \left\{ \mathbb{E}_{\hat{y}_V \sim q_V} [\ell(\hat{y}_V, y_V)] + \mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:V}) \right\} \leq \mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:V-1}) \end{aligned}$$

where the last inequality is by admissibility condition of the relaxation. Hence, we conclude that

$$\mathbb{E}_{\mathcal{C}_V, x_V} \left[\sup_{y_V} \left\{ \mathbb{E}_{\hat{y}_V \sim q_V} [\ell(\hat{y}_V, y_V)] + \mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:V}) \right\} \right] \leq \mathbb{E}_{\mathcal{C}_V, x_V} [\mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:V-1})]$$

This proves the base case. Now assume the statement holds for any $\tau > t$ and let us conclude the statement for t . For the t^{th} round, given $\mathcal{C}_{1:t}, x_{1:t}, y_{1:t-1}$,

$$\begin{aligned} & \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} [\ell(\hat{y}_t, y_t)] + \mathbb{E}_{\mathcal{I}_{t+1:V}} [\mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:t})] \right\} \\ &= \sup_{y_t} \left\{ \mathbb{E}_{\mathcal{I}_{t+1:V}} \left[\mathbb{E}_{\hat{y}_t \sim \hat{q}_t(\mathcal{I}_{t+1:V})} [\ell(\hat{y}_t, y_t)] \right] + \mathbb{E}_{\mathcal{I}_{t+1:V}} [\mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:t})] \right\} \\ &\leq \mathbb{E}_{\mathcal{I}_{t+1:V}} \left[\sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim \hat{q}_t(\mathcal{I}_{t+1:V})} [\ell(\hat{y}_t, y_t)] + \mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:t}) \right\} \right] \end{aligned}$$

By definition of \hat{q}_t , the above expression is equal to

$$\begin{aligned} &= \mathbb{E}_{\mathcal{I}_{t+1:V}} \left[\inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} [\ell(\hat{y}_t, y_t)] + \mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:t}) \right\} \right] \\ &\leq \mathbb{E}_{\mathcal{I}_{t+1:V}} [\mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:t-1})] \end{aligned}$$

Thus we can conclude that,

$$\mathbb{E}_{\mathcal{I}_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim \hat{q}_t(\mathcal{I}_{t+1:V})} [\ell(\hat{y}_t, y_t)] + \mathbb{E}_{\mathcal{I}_{t+1:V}} [\mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:t})] \right\} \leq \mathbb{E}_{\mathcal{I}_{t:V}} [\mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:t-1})]$$

This proves (20) via the inductive argument. To conclude the proof of the lemma, note that by the dominance condition,

$$\sum_{t=1}^V \ell(\hat{y}_t, y_t) - \inf_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \ell(f(x_t), y_t) \leq \sum_{t=1}^V \ell(\hat{y}_t, y_t) + \mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:V})$$

Using the above inequality and Eq. (20) we conclude that,

$$\begin{aligned} & \mathbb{E}_{\mathcal{I}_{1:V}} \left[\sum_{t=1}^V \mathbb{E}_{\hat{y}_t \sim \hat{q}_t} [\ell(\hat{y}_t, y_t)] - \inf_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \ell(f(x_t), y_t) \right] \\ &\leq \mathbb{E}_{\mathcal{I}_{1:V}} \left[\sum_{t=1}^V \mathbb{E}_{\hat{y}_t \sim \hat{q}_t} [\ell(\hat{y}_t, y_t)] + \mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:V}) \right] \\ &\leq \mathbb{E}_{\mathcal{I}_{1:V}} \left[\sum_{t=1}^{V-1} \mathbb{E}_{\hat{y}_t \sim \hat{q}_t} [\ell(\hat{y}_t, y_t)] + \mathbb{E}_{\mathcal{C}_V, x_V} [\mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | y_{1:V-1})] \right] \\ &\leq \mathbb{E}_{\mathcal{I}_{1:V}} [\mathbf{Rel}(\mathcal{F}_K[\mathcal{I}_{1:V}] | \cdot)] \end{aligned}$$

This concludes the proof of the lemma. ■

Proof [Proof of Lemma 2] The initial dominance condition is satisfied, since

$$\begin{aligned} \mathbf{Rel}_V(\mathcal{G} | y_{1:V}) &= \sup_{M \in \mathcal{M}} \sum_{s=1}^V M_{s,y_s} - V \geq \sup_{M \in \mathcal{M}_G} \sum_{s=1}^V M_{s,y_s} - V \\ &= \sup_{M \in \mathcal{M}_G} \sum_{s=1}^V (M_{s,y_s} - 1) = - \inf_{f \in \mathcal{G}} \sum_{s=1}^V \mathbf{1}\{f(x_s) \neq y_s\}. \end{aligned}$$

Next we show the recursive admissibility condition for the randomized strategy provided in the lemma. To this end note that,

$$\begin{aligned} &\max_{y_t \in [\kappa]} \{ \mathbb{E}_{\hat{y}_t \sim \hat{q}_t} [\ell(\hat{y}_t, y_t)] + \mathbf{Rel}(\mathcal{G} | y_{1:t}) \} \\ &= \max_{y_t \in [\kappa]} \left\{ 1 - \mathbb{E}_{\hat{y}_t \sim \hat{q}_t} [\mathbf{1}\{y_t = \hat{y}_t\}] + \mathbb{E}_{\epsilon_{t+1:V}} \sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^t M_{s,y_s} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} \right\} - t \right\} \\ &= \max_{y_t \in [\kappa]} \left\{ - \mathbb{E}_{\epsilon_{t+1:V}} \mathbb{E}_{\hat{y}_t \sim \hat{q}_t(\epsilon_{t+1:V})} [\mathbf{1}\{y_t = \hat{y}_t\}] + \mathbb{E}_{\epsilon_{t+1:V}} \sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^t M_{s,y_s} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} \right\} \right\} \\ &\quad - (t-1) \\ &\leq \mathbb{E}_{\epsilon_{t+1:V}} \left[\max_{y_t \in [\kappa]} \left\{ - \mathbb{E}_{\hat{y}_t \sim \hat{q}_t(\epsilon_{t+1:V})} [\mathbf{1}\{y_t = \hat{y}_t\}] + \sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^t M_{s,y_s} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} \right\} \right\} \right] \\ &\quad - (t-1) \end{aligned}$$

By the definition of the randomized strategy, the last expression is equal to

$$\mathbb{E}_{\epsilon_{t+1:V}} \left[\inf_{q_t \in \Delta([\kappa])} \max_{y_t \in [\kappa]} \left\{ - \mathbb{E}_{\hat{y}_t \sim q_t} [\mathbf{1}\{y_t = \hat{y}_t\}] + \sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^t M_{s,y_s} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} \right\} \right\} \right] - (t-1)$$

Using the minimax theorem, we can swap the infimum and supremum, and obtain equality to

$$\begin{aligned} &\mathbb{E}_{\epsilon_{t+1:V}} \left[\sup_{p_t \in \Delta([\kappa])} \min_{\hat{y}_t \in [\kappa]} \mathbb{E}_{y_t \sim p_t} \left[-\mathbf{1}\{y_t = \hat{y}_t\} + \sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^t M_{s,y_s} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} \right\} \right] \right] - (t-1) \\ &= \mathbb{E}_{\epsilon_{t+1:V}} \sup_{p_t \in \Delta([\kappa])} \left\{ - \max_{\hat{y}_t \in [\kappa]} \mathbb{E}_{y_t \sim p_t} [\mathbf{1}\{y_t = \hat{y}_t\}] + \mathbb{E}_{y_t \sim p_t} \sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^t M_{s,y_s} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} \right\} \right\} \\ &\quad - (t-1) \end{aligned}$$

Since

$$\max_{\hat{y}_t \in [\kappa]} \mathbb{E}_{y_t \sim p_t} [\mathbf{1}\{y_t = \hat{y}_t\}] = \max_{i \in [\kappa]} p_t[i] \geq \max_{i \in [\kappa]} p_t[i] \left(\sum_j M_{t,j} \right) \geq \sum_i p_t[i] M_{t,i} = \mathbb{E}_{y'_t \sim p_t} [M_{t,y'_t}],$$

the previous expression can be upper bounded by

$$\begin{aligned} &\mathbb{E}_{\epsilon_{t+1:V}} \sup_{p_t \in \Delta([\kappa])} \left\{ \mathbb{E}_{y_t \sim p_t} \left[\sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^{t-1} M_{s,y_s} + (M_{t,y_t} - \mathbb{E}_{y'_t \sim p_t} [M_{t,y'_t}]) + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} \right\} \right] \right\} \\ &\quad - (t-1) \end{aligned}$$

which is upper bounded by Jensen's inequality by

$$\mathbb{E}_{\epsilon_{t+1:V}} \sup_{p_t \in \Delta([\kappa])} \left\{ \mathbb{E}_{y'_t, y_t \sim p_t} \left[\sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^{t-1} M_{s, y_s} + (M_{t, y_t} - M_{t, y'_t}) + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j, k} M_{j, k} \right\} \right] \right\} - (t-1)$$

Since in above y_t and y'_t are identically distributed, we can introduce an independent Rademacher random variable δ_t . The last expression is equal to

$$\begin{aligned} & \mathbb{E}_{\epsilon_{t+1:V}} \sup_{p_t \in \Delta([\kappa])} \mathbb{E}_{\delta_t, y'_t, y_t \sim p_t} \left[\sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^{t-1} M_{s, y_s} + \delta_t (M_{t, y_t} - M_{t, y'_t}) + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j, k} M_{j, k} \right\} \right] - (t-1) \\ & \leq \mathbb{E}_{\epsilon_{t+1:V}} \sup_{y_t, y'_t \in [\kappa]} \left\{ \mathbb{E}_{\delta_t} \left[\sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^{t-1} M_{s, y_s} + \delta_t (M_{t, y_t} - M_{t, y'_t}) + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j, k} M_{j, k} \right\} \right] \right\} - (t-1) \\ & \leq \mathbb{E}_{\epsilon_{t+1:V}} \sup_{y_t \in [\kappa]} \left\{ \mathbb{E}_{\delta_t} \left[\sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^{t-1} M_{s, y_s} + 2\delta_t M_{t, y_t} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j, k} M_{j, k} \right\} \right] \right\} - (t-1) \end{aligned}$$

Now let $\epsilon_t^{y_t} \in \{\pm 1\}^{\kappa}$ be defined as 1 on coordinate y_t and independent Rademacher variables on the rest. For any $j \neq y_t$, $\mathbb{E}[\epsilon_{t, j}^{y_t}] = 0$ and $\epsilon_{t, y_t}^{y_t} = 1$ and so the preceding expression is equal to

$$\begin{aligned} & \mathbb{E}_{\epsilon_{t+1:V}} \sup_{y_t \in [\kappa]} \left\{ \mathbb{E}_{\delta_t} \left[\sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^{t-1} M_{s, y_s} + 2 \sum_{k=1}^{\kappa} \delta_t M_{t, k} \mathbb{E}[\epsilon_{t, k}^{y_t}] + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j, k} M_{j, k} \right\} \right] \right\} - (t-1) \\ & \leq \mathbb{E}_{\epsilon_{t+1:V}} \sup_{y_t \in [\kappa]} \left\{ \mathbb{E}_{\delta_t, \epsilon_t^{y_t}} \left[\sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^{t-1} M_{s, y_s} + 2 \sum_{k=1}^{\kappa} \delta_t M_{t, k} \epsilon_{t, k}^{y_t} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j, k} M_{j, k} \right\} \right] \right\} - (t-1) \\ & = \mathbb{E}_{\epsilon_{t+1:V}} \left\{ \mathbb{E}_{\epsilon_t} \left[\sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^{t-1} M_{s, y_s} + 2 \sum_{k=1}^{\kappa} M_{t, k} \epsilon_{t, k} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j, k} M_{j, k} \right\} \right] \right\} - (t-1) \\ & = \mathbb{E}_{\epsilon_{t:V}} \left[\sup_{M \in \mathcal{M}} \left\{ \sum_{s=1}^{t-1} M_{s, y_s} + 2 \sum_{j=t}^V \sum_{k=1}^{\kappa} \epsilon_{j, k} M_{j, k} \right\} \right] - (t-1) = \mathbf{Rel}(\mathcal{G} | y_{1:t-1}) \end{aligned}$$

Thus we have shown admissibility of the relaxation and demonstrated that the randomized strategy for the forecaster is given by the one in the lemma. \blacksquare

Proof [Proof of Theorem 4] From the bound in (8) we have that the expected regret of our algorithm is bounded as

$$\mathbb{E}[\mathbf{Reg}] \leq 2 \mathbb{E}_{\mathcal{I}_{1:V}} \mathbb{E}_{\epsilon_{1:V}} \left[\sup_{M \in \mathcal{M}(\mathcal{I}_{1:V})} \sum_{j=t}^V \sum_{k=1}^{\kappa} \epsilon_{j, k} M_{j, k} \right].$$

Let $M \in \mathcal{M}(\mathcal{I}_{1:V})$ to be the projected solutions from the r^{th} level Lasserre hierarchy SDP in the maximization problem in (9). Then for each draw of $\epsilon_{1:V}$, the supremum in the Rademacher complexity term can be replaced by the value of the optimization problem in (9) given by $\text{SDP}_r^{1st}(\epsilon, K)$. This is because we can think of $M_{j, k}$ as corresponding to $\|\mathbf{U}_{\{j, k}\}\|^2$ where vectors \mathbf{U} 's satisfying constraints of the SDP. On the other hand, for a given draw of $\epsilon_{1:V}$, the solution to

$$\sup_{f \in \mathcal{F}_{\text{gap}(r) \cdot K}[\mathcal{I}_{1:V}]} \sum_{j=t}^V \sum_{k=1}^{\kappa} \epsilon_{j, k} \mathbf{1}\{f(x_j) = k\}$$

is exactly the value of $\text{OPT}^{1st}(\epsilon, \text{gap}(r) \cdot K)$. Hence to prove our bound, it suffices to show that for any problem at hand,

$$\text{SDP}_r^{1st}(\epsilon, K) \leq \text{OPT}^{1st}(\epsilon, \text{gap}(r) \cdot K).$$

To do so we go through the problems in Eqns. (12) and (14) and arrive to $\text{OPT}^{1st}(\epsilon, \text{gap}(r) \cdot K)$. Observe that the solution to the optimization problem in (9) is such that it has value $\text{SDP}_r^{1st}(\epsilon, K)$ and violates constraints by less than K . Using this feasible solution in (12) we conclude that,

$$\text{SDP}_r^{2nd}(\epsilon, \text{SDP}_r^{1st}(\epsilon, K)) \leq K$$

However by definition of $\text{gap}(r)$ we can conclude that

$$\text{OPT}^{2nd}(\epsilon, \text{SDP}_r^{1st}(\epsilon, K)) \leq \text{gap}(r) \cdot \text{SDP}_r^{2nd}(\epsilon, \text{SDP}_r^{1st}(\epsilon, K)) \leq \text{gap}(r) \cdot K$$

By the definition of OPT^{2nd} this means that the solution $M \in \mathcal{F}_{x_{1:V}}$ to the optimization problem is such that $\sum_{c \in \mathcal{U}^{\mathcal{C}_t}} c(M) \leq \text{gap}(r) \cdot K$, and simultaneously, since we are considering OPT^{2nd} with second argument as $\text{SDP}_r^{1st}(\epsilon, K)$, $M^\top Y \geq \text{SDP}_r^{1st}(\epsilon, K)$. Thus by using this solution in the optimization problem in Eq. (15) with second argument of $\text{gap}(r) \cdot K$, we conclude:

$$\text{SDP}_r^{1st}(\epsilon, K) \leq \text{OPT}^{1st}(\epsilon, \text{gap}(r) \cdot K)$$

as required. Now since this is true for every ϵ , we have that

$$\mathbb{E}[\mathbf{Reg}] \leq 2 \mathbb{E}_{\mathcal{I}_{1:V}} \mathbb{E}_{\epsilon_{1:V}} \left[\sup_{f \in \mathcal{F}_{\text{gap}(r) \cdot K}[\mathcal{I}_{1:V}]} \sum_{j=t}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} \mathbf{1}\{f(x_j) = k\} \right].$$

■

Proof [Proof of Proposition 8] To prove the lower bound, we simply consider an adversary who picks nodes in the fixed sorted order and at each time step draw \mathcal{C}_t, x_t from the known generating process and finally draw $y_t \sim \text{Unif}([\kappa])$. Now since y_t is drawn independently and uniformly at random on every round, irrespective of how the forecaster picks \hat{y}_t , the expected loss of the forecaster is $\mathbb{E}[\mathbf{1}\{\hat{y}_t \neq y_t\}] = 1/\kappa$. Thus we get the following lower bound on the expected regret.

$$\begin{aligned} \mathbb{E}[\mathbf{Reg}] &\geq \mathbb{E}_{(x_t, \mathcal{C}_t)_{t=1}^V} \left[\mathbb{E}_{y_{1:V} \sim \text{Unif}(\{2\})} \left[V/2 - \inf_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \mathbf{1}\{f(x_t) \neq y_t\} \right] \right] \\ &= \mathbb{E}_{\mathcal{I}_{1:V}} \left[\mathbb{E}_{y_{1:V} \sim \text{Unif}(\{2\})} \left[\sup_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \left(\mathbf{1}\{f(x_t) = y_t\} - \frac{1}{2} \right) \right] \right] \end{aligned}$$

Now for the uniform distribution over y_t 's, since $\mathbf{1}\{f(x_t) = y_t\} - \frac{1}{2}$ and $\frac{1}{2} - \mathbf{1}\{f(x_t) = y_t\}$ are identically distributed we see that,

$$\begin{aligned} \mathbb{E}[\mathbf{Reg}] &\geq \mathbb{E}_{(x_t, \mathcal{C}_t)_{t=1}^V} \left[\mathbb{E}_{y_{1:V} \sim \text{Unif}(\{2\})} \mathbb{E}_{\epsilon} \left[\sup_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \epsilon_t \left(\mathbf{1}\{f(x_t) = y_t\} - \frac{1}{2} \right) \right] \right] \\ &= \mathbb{E}_{\mathcal{I}_{1:V}} \left[\mathbb{E}_{\epsilon} \mathbb{E}_{y_{1:V} \sim \text{Unif}(\{2\})} \left[\sup_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \epsilon_{t,y_t} \mathbf{1}\{f(x_t) = y_t\} \right] \right] \\ &\geq \mathbb{E}_{\mathcal{I}_{1:V}} \left[\mathbb{E}_{\epsilon} \left[\sup_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \mathbb{E}_{y_t \sim \text{Unif}(\{2\})} [\epsilon_{t,y_t} \mathbf{1}\{f(x_t) = y_t\}] \right] \right] \\ &\geq \mathbb{E}_{\mathcal{I}_{1:V}} \left[\mathbb{E}_{\epsilon} \left[\sup_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \frac{1}{2} \sum_{k=1}^2 (\epsilon_{t,k} \mathbf{1}\{f(x_t) = k\}) \right] \right] \\ &= \frac{1}{2} \mathbb{E}_{\mathcal{I}_{1:V}, \epsilon} \left[\sup_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \sum_{k=1}^2 \epsilon_{t,k} \mathbf{1}\{f(x_t) = k\} \right] \end{aligned}$$

where the last line is because for any f and any instance x_t only one of $\mathbf{1}\{f(x_t) = 1\}$ or $\mathbf{1}\{f(x_t) = 2\}$ will be 1 and the other is 0. ■

Proof [Proof of Corollary 9] This corollary follows by using a simple modification to Proposition 8. We shall assume here that κ is even. The simple modification is as follows: the adversary first picks uniformly at random a number R from $[\kappa/2]$. Next the adversary uses exactly the lower bound construction as in Proposition 8 except that instead of picking $y_t \sim \text{Unif}([2])$ the adversary picks $y_t \sim \text{Unif}(\{R, R + \kappa/2\})$. Now notice that given draw of R , this is exactly the binary case with labels R and $R + \kappa/2$. Hence we can use the proposition to bound the expected regret as follows:

$$\begin{aligned}
 \mathbb{E}[\mathbf{Reg}] &\geq \frac{1}{2} \mathbb{E}_{R \sim \text{Unif}([\kappa/2])} \mathbb{E}_{\mathcal{I}_{1:V}, \epsilon} \left[\sup_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \sum_{k \in \{R, R + \kappa/2\}} \epsilon_{t,k} \mathbf{1}\{f(x_t) = k\} \right] \\
 &= \frac{1}{2} \mathbb{E}_{R \sim \text{Unif}([\kappa/2])} \mathbb{E}_{\mathcal{I}_{1:V}, \epsilon} \left[\sup_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \sum_{k=1}^{\kappa} \mathbf{1}\{k \in \{R, R + \kappa/2\}\} \epsilon_{t,k} \mathbf{1}\{f(x_t) = k\} \right] \\
 &\geq \frac{1}{2} \mathbb{E}_{\mathcal{I}_{1:V}, \epsilon} \left[\sup_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \sum_{k=1}^{\kappa} \mathbb{E}_{R \sim \text{Unif}([\kappa/2])} [\mathbf{1}\{k \in \{R, R + \kappa/2\}\}] \epsilon_{t,k} \mathbf{1}\{f(x_t) = k\} \right] \\
 &= \frac{1}{\kappa} \mathbb{E}_{\mathcal{I}_{1:V}, \epsilon} \left[\sup_{f \in \mathcal{F}_K[\mathcal{I}_{1:V}]} \sum_{t=1}^V \sum_{k=1}^{\kappa} \epsilon_{t,k} \mathbf{1}\{f(x_t) = k\} \right]
 \end{aligned}$$

■

Proof [Proof of Lemma 6] The proof closely follows the analogous proof of Lemma 2. Note that we deal directly with the relaxed set of Lasserre's level r . To make the notation simpler, given a Lasserre vector set at level r , say $\mathbf{U} \in \mathcal{Las}(r, \mathcal{F}_{x_{1:V}})$, let $M_{j,k}^{\mathbf{U}} = \|\mathbf{U}_{(\{j\},k)}\|^2$ and also for each t and each constraint $c \in \mathcal{C}_t$ we use the notation

$$c(\mathbf{U}) = \sum_{\alpha \in [q]^{S_c}} R_c(\alpha) \|\mathbf{U}_{(S_c, \alpha)}\|^2$$

Now let us proceed to verify that the initial dominance condition is satisfied by the relaxation. Note that

$$\begin{aligned}
 - \inf_{f \in \mathcal{G}} \sum_{s=1}^V \mathbf{1}\{f(x_s) \neq y_s\} &\leq - \inf_{f \in \mathcal{G}} \left\{ \sum_{s=1}^V \mathbf{1}\{f(x_s) \neq y_s\} + \lambda \sum_{c \in \mathcal{C}_{1:V}} c(f) \right\} + \lambda K \\
 &\leq - \inf_{\mathbf{U} \in \mathcal{Las}(r, \mathcal{F}_{x_{1:V}})} \left\{ \sum_{s=1}^V M_{s, y_s}^{\mathbf{U}} + \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} + \lambda K,
 \end{aligned}$$

where the first inequality holds because functions in $\mathcal{G} = \mathcal{F}_K[\mathcal{I}_{1:V}]$ are required to keep the sum over unsatisfied constraints below K by definition. The second inequality holds because the Lasserre solution is a relaxation of \mathcal{G} and hence larger than the solution within \mathcal{G} . Let us check the recursive admissibility condition. To show that the proposed randomized strategy is admissible, we prove the recursive admissibility condition using this strategy directly:

$$\begin{aligned}
 &\max_{y_t \in [\kappa]} \left\{ \mathbb{E}_{\hat{y}_t \sim \hat{q}_t} [\ell(\hat{y}_t, y_t)] + \mathbf{Rel}(\mathcal{G} | y_{1:t}) \right\} \\
 &= \max_{y_t \in [\kappa]} \left\{ 1 - \mathbb{E}_{\hat{y}_t \sim \hat{q}_t} [\mathbf{1}\{y_t = \hat{y}_t\}] + \mathbb{E}_{\epsilon_{t+1:V}} \sup_{\mathbf{U} \in \mathcal{Las}(r, \mathcal{F}_{x_{1:V}})} \left\{ \sum_{s=1}^t M_{s, y_s}^{\mathbf{U}} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \right\} \\
 &\quad - t + \lambda K \\
 &= \max_{y_t \in [\kappa]} \mathbb{E}_{\epsilon_{t+1:V}} \left[- \mathbb{E}_{\hat{y}_t \sim \hat{q}_t} [\mathbf{1}\{y_t = \hat{y}_t\}] + \sup_{\mathbf{U} \in \mathcal{Las}(r, \mathcal{F}_{x_{1:V}})} \left\{ \sum_{s=1}^t M_{s, y_s}^{\mathbf{U}} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \right] \\
 &\quad - (t-1) + \lambda K \\
 &\leq \mathbb{E}_{\epsilon_{t+1:V}} \max_{y_t \in [\kappa]} \left\{ - \mathbb{E}_{\hat{y}_t \sim \hat{q}_t} [\mathbf{1}\{y_t = \hat{y}_t\}] + \sup_{\mathbf{U} \in \mathcal{Las}(r, \mathcal{F}_{x_{1:V}})} \left\{ \sum_{s=1}^t M_{s, y_s}^{\mathbf{U}} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \right\} \\
 &\quad - (t-1) + \lambda K
 \end{aligned}$$

by the definition of the strategy,

$$= \mathbb{E}_{\epsilon_{t+1:V}} \inf_{\hat{y}_t \in \Delta([\kappa])} \max_{y_t \in [\kappa]} \left\{ -\mathbb{E}_{\hat{y}_t \sim q_t} [\mathbf{1}\{y_t = \hat{y}_t\}] + \sup_{\mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_{1:V}})} \left\{ \sum_{s=1}^t M_{s,y_s}^{\mathbf{U}} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \right\} - (t-1) + \lambda K$$

Using the minimax theorem, the above expression is equal to

$$\begin{aligned} &= \mathbb{E}_{\epsilon_{t+1:V}} \sup_{p_t \in \Delta([\kappa])} \min_{\hat{y}_t \in [\kappa]} \mathbb{E}_{y_t \sim p_t} \left[-\mathbf{1}\{y_t = \hat{y}_t\} + \sup_{\mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_{1:V}})} \left\{ \sum_{s=1}^t M_{s,y_s}^{\mathbf{U}} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \right] \\ &\quad - (t-1) + \lambda K \\ &= \mathbb{E}_{\epsilon_{t+1:V}} \sup_{p_t \in \Delta([\kappa])} \left\{ -\max_{\hat{y}_t \in [\kappa]} \mathbb{E}_{y_t \sim p_t} [\mathbf{1}\{y_t = \hat{y}_t\}] \right. \\ &\quad \left. + \mathbb{E}_{y_t \sim p_t} \sup_{\mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_{1:V}})} \left\{ \sum_{s=1}^t M_{s,y_s}^{\mathbf{U}} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \right\} \\ &\quad - (t-1) + \lambda K \end{aligned}$$

Once again, by the constraint in the SDP that for any t , $\sum_{k=1}^{\kappa} \|\mathbf{U}_{\{t\},k}\|^2 = \sum_{k=1}^{\kappa} M_{t,k}^{\mathbf{U}} = 1$ we can conclude that

$$\max_{\hat{y}_t \in [\kappa]} \mathbb{E}_{y_t \sim p_t} [\mathbf{1}\{y_t = \hat{y}_t\}] = \max_{i \in [\kappa]} p_t[i] \geq \max_{i \in [\kappa]} p_t[i] \left(\sum_j M_{t,j}^{\mathbf{U}} \right) \geq \sum_i p_t[i] M_{t,i}^{\mathbf{U}} = \mathbb{E}_{y'_t \sim p_t} [M_{t,y'_t}^{\mathbf{U}}].$$

Hence, we conclude that,

$$\begin{aligned} &\max_{y_t \in [\kappa]} \left\{ \mathbb{E}_{\hat{y}_t \sim \hat{q}_t} [\ell(\hat{y}_t, y_t)] + \mathbf{Rel}(\mathcal{G} | y_{1:t}) \right\} \\ &\leq \mathbb{E}_{\epsilon_{t+1:V}} \sup_{p_t \in \Delta([\kappa])} \mathbb{E}_{y_t \sim p_t} \sup_{\mathbf{U} \in \mathcal{L}as(r)} \left\{ \sum_{s=1}^{t-1} M_{s,y_s}^{\mathbf{U}} + (M_{t,y_t}^{\mathbf{U}} - \mathbb{E}_{y'_t \sim p_t} [M_{t,y'_t}^{\mathbf{U}}]) + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \\ &\quad - (t-1) + \lambda K \end{aligned}$$

using Jensen's inequality to pull out the expectation,

$$\begin{aligned} &\leq \mathbb{E}_{\epsilon_{t+1:V}} \sup_{p_t \in \Delta([\kappa])} \mathbb{E}_{y'_t, y_t \sim p_t} \sup_{\mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_{1:V}})} \left\{ \sum_{s=1}^{t-1} M_{s,y_s}^{\mathbf{U}} + (M_{t,y_t}^{\mathbf{U}} - M_{t,y'_t}^{\mathbf{U}}) + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \\ &\quad - (t-1) + \lambda K \end{aligned}$$

since y_t and y'_t are identically distributed, we can introduce Rademacher random variable δ_t ,

$$\begin{aligned} &\mathbb{E}_{\epsilon_{t+1:V}} \sup_{p_t \in \Delta([\kappa])} \mathbb{E}_{\delta_t, y'_t, y_t \sim p_t} \sup_{\mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_{1:V}})} \left\{ \sum_{s=1}^{t-1} M_{s,y_s}^{\mathbf{U}} + \delta_t (M_{t,y_t}^{\mathbf{U}} - M_{t,y'_t}^{\mathbf{U}}) + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \\ &\quad - (t-1) + \lambda K \\ &\leq \mathbb{E}_{\epsilon_{t+1:V}} \sup_{y_t, y'_t \in [\kappa]} \left\{ \mathbb{E}_{\delta_t \mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_{1:V}})} \left\{ \sum_{s=1}^{t-1} M_{s,y_s}^{\mathbf{U}} + \delta_t (M_{t,y_t}^{\mathbf{U}} - M_{t,y'_t}^{\mathbf{U}}) + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \right\} \\ &\quad - (t-1) + \lambda K \\ &\leq \mathbb{E}_{\epsilon_{t+1:V}} \sup_{y_t \in [\kappa]} \left\{ \mathbb{E}_{\delta_t} \left[\sup_{\mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_{1:V}})} \left\{ \sum_{s=1}^{t-1} M_{s,y_s}^{\mathbf{U}} + 2\delta_t M_{t,y_t}^{\mathbf{U}} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \right] \right\} \\ &\quad - (t-1) + \lambda K \end{aligned}$$

Let $\epsilon_t^{y_t} \in \{\pm 1\}^\kappa$ be defined as 1 on coordinate y_t and independent Rademacher variables on the rest. For any $j \neq y_t$, $\mathbb{E}[\epsilon_{t,j}^{y_t}] = 0$ and $\epsilon_{t,y_t}^{y_t} = 1$ and so,

$$\begin{aligned}
 &\leq \mathbb{E}_{\epsilon_{t+1:V}} \sup_{y_t \in [\kappa]} \mathbb{E}_{\delta_t \mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_1:V})} \sup \left\{ \sum_{s=1}^{t-1} M_{s,y_s}^{\mathbf{U}} + 2 \sum_{k=1}^{\kappa} \delta_t M_{t,k}^{\mathbf{U}} \mathbb{E}[\epsilon_{t,k}^{y_t}] + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \\
 &\quad - (t-1) + \lambda K \\
 &\leq \mathbb{E}_{\epsilon_{t+1:V}} \sup_{y_t \in [\kappa]} \mathbb{E}_{\delta_t, \epsilon_t^{y_t} \mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_1:V})} \sup \left\{ \sum_{s=1}^{t-1} M_{s,y_s}^{\mathbf{U}} + 2 \sum_{k=1}^{\kappa} \delta_t M_{t,k}^{\mathbf{U}} \epsilon_{t,k}^{y_t} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \\
 &\quad - (t-1) + \lambda K \\
 &= \mathbb{E}_{\epsilon_{t+1:V}} \left\{ \mathbb{E}_{\epsilon_t} \left[\sup_{\mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_1:V})} \left\{ \sum_{s=1}^{t-1} M_{s,y_s}^{\mathbf{U}} + 2 \sum_{k=1}^{\kappa} M_{t,k}^{\mathbf{U}} \epsilon_{t,k} + 2 \sum_{j=t+1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \right] \right\} \\
 &\quad - (t-1) + \lambda K \\
 &= \mathbb{E}_{\epsilon_{t:V}} \left[\sup_{\mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_1:V})} \left\{ \sum_{s=1}^{t-1} M_{s,y_s}^{\mathbf{U}} + 2 \sum_{j=t}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k}^{\mathbf{U}} - \lambda \sum_{c \in \mathcal{C}_{1:V}} c(\mathbf{U}) \right\} \right] - (t-1) + \lambda K \\
 &= \mathbf{Rel}(\mathcal{G} | y_{1:t-1})
 \end{aligned}$$

■

Proof [Proof of Theorem 7] We have

$$\begin{aligned}
 \mathbf{Rel}(\mathcal{G} | \emptyset) &= \lambda^* K + \mathbb{E}_{\epsilon_{1:V}} \left[\sup_{\mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_1:V})} \left\{ 2 \sum_{t=1}^V \sum_{k=1}^{\kappa} \epsilon_{t,k} \|\mathbf{U}_{\{t\},k}\|^2 - \lambda^* \sum_{c \in \cup_t \mathcal{C}_t} \sum_{\alpha \in [q]^{S_c}} R_c(\alpha) \|\mathbf{U}_{(S_c, \alpha)}\|^2 \right\} \right] \\
 &\leq 2 \mathbb{E}_{\epsilon_{1:V}} \left[\sup_{\mathbf{U} \in \mathcal{L}as(r, \mathcal{F}_{x_1:V})} \left\{ 2 \sum_{t=1}^V \sum_{k=1}^{\kappa} \epsilon_{t,k} \|\mathbf{U}_{\{t\},k}\|^2 - \lambda^* \sum_{c \in \cup_t \mathcal{C}_t} \sum_{\alpha \in [q]^{S_c}} R_c(\alpha) \|\mathbf{U}_{(S_c, \alpha)}\|^2 \right\} \right] \\
 &= 2\lambda^* K.
 \end{aligned}$$

Now by definition of $\widehat{\text{gap}}(r)$ we conclude that

$$\mathbf{Rel}(\mathcal{G} | \emptyset) \leq 2 \mathbb{E}_{\epsilon_{1:V}} \left[\sup_{M \in \mathcal{F}_{x_1:V}} \left\{ 2 \sum_{t=1}^V \sum_{k=1}^{\kappa} \epsilon_{t,k} M_{t,k} - \frac{\lambda^*}{\widehat{\text{gap}}(r)} \sum_{c \in \cup_t \mathcal{C}_t} c(M) \right\} \right]$$

Defining $K_i = 2^i$, we get an upper bound,

$$\begin{aligned}
 &\leq 2 \mathbb{E}_{\epsilon_{1:V}} \left[\max_{i \in \mathbb{Z}} \sup_{\substack{M \in \mathcal{F}_{x_1:V} \\ K_{i-1} \leq \sum_{c \in \cup_t \mathcal{C}_t} c(M) \leq K_i}} \left\{ 2 \sum_{t=1}^V \sum_{k=1}^{\kappa} \epsilon_{t,k} M_{t,k} - \frac{\lambda^*}{\widehat{\text{gap}}(r)} \sum_{c \in \cup_t \mathcal{C}_t} c(M) \right\} \right] \\
 &\leq 2 \max_{i \in \mathbb{Z}} \left\{ \mathbb{E}_{\epsilon_{1:V}} \left[\sup_{\substack{M \in \mathcal{F}_{x_1:V} \\ \sum_{c \in \cup_t \mathcal{C}_t} c(M) \leq K_i}} \left\{ 2 \sum_{t=1}^V \sum_{k=1}^{\kappa} \epsilon_{j,k} M_{j,k} \right\} \right] - \frac{\lambda^*}{\widehat{\text{gap}}(r)} K_{i-1} \right\} \\
 &= 2 \max_{i \in \mathbb{Z}} \left\{ \text{Rad}_V(\mathcal{F}_{K_i}[\mathcal{I}_{1:V}]) - \frac{\lambda^*}{\widehat{\text{gap}}(r)} K_{i-1} \right\} \\
 &= 2 \max_{i \in \mathbb{Z}} \left\{ \text{Rad}_V(\mathcal{F}_{K_i}[\mathcal{I}_{1:V}]) - \frac{\lambda^*}{2\widehat{\text{gap}}(r)} K_i \right\} \\
 &\leq 2 \max_{i \in \mathbb{Z}} \left\{ \text{Rad}_V(\mathcal{F}_{\max\{1, \frac{K_i}{2}\}}[\mathcal{I}_{1:V}]) - \frac{\lambda^*}{2\widehat{\text{gap}}(r)} K_i \right\}
 \end{aligned}$$

$$\leq 2 \max_{i \in \mathbb{Z}} \left\{ \max \left\{ 1, \frac{K_i}{K} \right\}^p \text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}]) - \frac{\lambda^*}{2 \widehat{\text{gap}}(r)} K_i \right\}.$$

Now let us split the analysis into two cases. First, if $\lambda^* > \frac{2 \widehat{\text{gap}}(r) \text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}])}{K}$, then

$$\mathbf{Rel}(\mathcal{G} \mid \emptyset) \leq 2 \max_{i \in \mathbb{Z}} \left\{ \max \left\{ 1, \frac{K_i}{K} \right\}^p \text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}]) - \frac{K_i}{K} \text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}]) \right\} \leq 2 \text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}])$$

where the last line is because $p \leq 1$. Next let us consider the case when $\lambda^* \leq \frac{2 \text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}])}{K \widehat{\text{gap}}(r)}$. For this case however, note that we already showed that $\mathbf{Rel}(\mathcal{G} \mid \emptyset) \leq 2\lambda^* K$ and so

$$\mathbf{Rel}(\mathcal{G} \mid \emptyset) \leq 4 \widehat{\text{gap}}(r) \text{Rad}_V(\mathcal{F}_K[\mathcal{I}_{1:V}]).$$

The first statement follows. The second statement of the Theorem is an immediate consequence of Lemma 1. ■