Correlated Rounding of Multiple Uniform Matroids and Multi-Label Classification

Shahar Chen[1], Dotan Di Castro[2], Zohar Karnin[3], Liane Lewin-Eytan[2], Joseph (Seffi) Naor[1], and Roy Schwartz[1]

1 **Computer Science Department, Technion, Haifa 32000, Israel.**
shahar.chen11@gmail.com and {naor,schwartz}@cs.technion.ac.il
2 **Yahoo Labs, Haifa 31905, Israel.**
{dot,liane}@yahoo-inc.com
3 **Amazon, New York 10001, NY, USA.**
zkarnin@amazon.com

Abstract

We introduce *correlated randomized dependent rounding* where, given multiple points $\mathbf{y}^1, \ldots, \mathbf{y}^n$ in some polytope $\mathcal{P} \subseteq [0,1]^k$, the goal is to simultaneously round each $\mathbf{y}^i$ to some integral $\mathbf{z}^i \in \mathcal{P}$ while preserving both marginal values and expected distances between the points. In addition to being a natural question in its own right, the correlated randomized dependent rounding problem is motivated by multi-label classification applications that arise in machine learning, *e.g.*, classification of web pages, semantic tagging of images, and functional genomics. The results of this work can be summarized as follows: (1) we present an algorithm for solving the correlated randomized dependent rounding problem in uniform matroids while losing only a factor of $O(\log k)$ in the distances (k is the size of the ground set); (2) we introduce a novel multi-label classification problem, the *metric multi-labeling* problem, which captures the above applications. We present a (true) $O(\log k)$-approximation for the general case of metric multi-labeling and a tight 2-approximation for the special case where there is no limit on the number of labels that can be assigned to an object.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics, G.2.2 Graph Theory

Keywords and phrases approximation algorithms, randomized rounding, dependent rounding, metric labeling, classification

Digital Object Identifier 10.4230/LIPIcs.ICALP.2017.

1 Introduction

Randomized rounding [32] is a fundamental technique in approximation algorithms. In this approach, given a solution $\mathbf{y} \in \mathbb{R}^k$ to some linear program, each y_i is *independently* rounded into an integral value. Unfortunately, when constraints on the rounded solution are present, randomized rounding does not always produce a feasible solution. Hence, *dependent* rounding schemes were introduced [1, 2, 10, 12, 22, 26, 35]. In general, dependent rounding needs to solve the following problem: given a polytope $\mathcal{P} \subseteq [0,1]^k$ over ground set K of size k and $\mathbf{y} \in \mathcal{P}$, round $\mathbf{y}$ into $\mathbf{z} \in \mathcal{P} \cap \{0,1\}^k$ such that $\mathbb{E}[\mathbf{z}] = \mathbf{y}$. Intuitively, $\mathbf{z}$ has the following two properties: (1) $\mathbf{z}$ is always integral and feasible since $\mathbf{z} \in \mathcal{P} \cap \{0,1\}^k$; and (2) $\mathbf{z}$ preserves the marginal values given by $\mathbf{y}$ for each element in K since $\mathbb{E}[\mathbf{z}] = \mathbf{y}$. The above problem has been extensively studied and was solved for different types of polytopes $\mathcal{P}$, *e.g.*, bipartite

44th International Colloquium on Automata, Languages, and Programming (ICALP 2017).
Editors: Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl;
Article No. ; pp. :1–:15


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

matching and b-matching [22, 26], uniform matroids [35], spanning trees [2], and general matroids [12][1].

In this work we consider a natural extension of dependent rounding in which we are given many points in $\mathcal{P}$ and the goal is to round all the points, while preserving both marginal values and expected distances (up to some loss) between any pair of points. Formally, given a polytope $\mathcal{P} \subseteq [0,1]^k$ over ground set K of size k and $\mathbf{y}^1, \ldots, \mathbf{y}^n \in \mathcal{P}$, we need to round each $\mathbf{y}^i$ to some $\mathbf{z}^i$ such that the following hold: (1) $\mathbf{z}^i \in \mathcal{P} \cap \{0,1\}^k$ for every $i = 1, \ldots, n$; (2) $\mathbb{E}\left[\mathbf{z}^i\right] = \mathbf{y}^i$ for every $i = 1, \ldots, n$; and (3) there exists some loss factor α such that $\mathbb{E}\left[||\mathbf{z}^i - \mathbf{z}^j||_1\right] \leq \alpha ||\mathbf{y}^i - \mathbf{y}^j||_1$ for every $i, j = 1, \ldots, n$. We call this problem *correlated randomized dependent rounding*. Note that requirements (1) and (2) imply that each $\mathbf{z}^i$ is a feasible rounding of $\mathbf{y}^i$ that preserves marginal values, as in the standard dependent rounding setting. The novelty of our problem lies in requirement (3) which states that for all pairs of points the expected distance after the rounding, *i.e.*, $\mathbb{E}\left[||\mathbf{z}^i - \mathbf{z}^j||_1\right]$, is within a factor of α from the original distance between the points, *i.e.*, $||\mathbf{y}^i - \mathbf{y}^j||_1$. Additionally, it will be useful also to consider an extension of the above where each point $\mathbf{y}^i$ (and thus also $\mathbf{z}^i$) is required to be in a different polytope $\mathcal{P}_i$.

Our main reason for introducing the correlated randomized dependent rounding setting originates from multi-label classification problems. In classification problems, one must assign labels to objects given some observed data. In this work we consider classification problems where multiple labels can be assigned to each object. Such problems naturally arise in various settings, *e.g.*, classification of textual data such as web pages [38, 39], semantic tagging of images and videos [7, 30, 42], and functional genomics [4, 5].

The assignment of labels to objects should be done in a manner that is most consistent with the observed data, from which two important ingredients are derived. The first is an *assignment cost* for every (object,label) pair, reflecting a recommendation given by a local learning process which infers label preferences of objects. The second is similarity information on pairs of objects, giving rise to *separation costs* incurred once different label sets are assigned to a pair of similar objects. Our goal is to find a labeling that minimizes a global cost function, while taking into account both local and pairwise information.

To provide some intuition for the formal problem given below and the possible range of its parameters, we provide a concrete example. The objective in the example is that of assigning topics to web pages, where objects are the web pages and labels are the topics. Here, it is very natural for a web page to discuss more than one topic. The assignment cost of a (webpage,topic) pair can be derived from the features associated with a web page, *e.g.*, its words, or shingles, and the domain it is located in. However, consider information from search queries leading to the web page. A specific search query is typically observed only a handful of times, and though features can be extracted from it, a very natural way to use the latter information is by having pairwise similarity relations between web pages, if both were reached by the same search query.

We note that when assigning multiple labels to objects, it is often desirable to bound the number of labels assigned to objects. As a matter of fact, in most of the papers cited above the total number of labels can be in the thousands or even millions, while each object is expected to be assigned only a handful of labels. In particular, in the above example, we expect a single webpage to be assigned only a small fraction of all possible topics. This

[1] In some of the above works, additional properties of $\mathbf{z}$ are required, *e.g.*, concentration of linear functions over $\mathbf{z}$. Since such concentration bounds are not required for the metric multi-labeling (MML) problem, the discussion on this topic is postponed to a full version of the paper.

property imposes further constraints on our objective that we elaborate on below.

We are now ready to introduce the *metric multi-labeling* (MML) problem. In (MML) we are given a set of nodes V, where each node corresponds to an object, and a set of labels $K = \{1, 2, \ldots, k\}$. The pairwise relations are given in the form of an edge set E and a weight function $s : E \to \mathbb{R}^+$, capturing similarity between objects. Additionally, the bound function $b : V \to \mathbb{N}$ specifies how many labels can be assigned to each node. Finally, we are given an assignment cost function $c : V \times K \to \mathbb{R}$. Assignment costs may be either positive or negative, reflecting a recommendation given by a local learning process which infers the label preferences of objects. Intuitively, if $c(v, \ell) \geq 0$ (or $c(v, \ell) < 0$) we say that node v *dislikes* (or *likes*) label ℓ. A detailed explanation as to why assignment costs might be either positive or negative is deferred to a full version of the paper. The learning process determining assignment costs ignores pairwise relations between objects. Clearly, the labeling cost of completely agreeing with this recommendation is the minimum possible, and this is our *benchmark labeling*. We evaluate the assignment cost of a labeling by its deviation from the benchmark labeling.

A feasible multi-labeling $f : V \to 2^K \setminus \emptyset$ is an assignment of at least one label to every node, such that $|f(v)| \leq b_v$, *i.e.*, the number of labels assigned to v is at most b_v. For the special case where $b_v = k$ for every $v \in V$, *i.e.*, there is no upper bound on the number of labels that can be assigned to a node, we denote the problem by (Unbounded-MML).

The cost of a multi-labeling is measured by the sum of two terms: assignment costs and separation costs. Let us first focus on assignment costs, which measure the deviation of f from the benchmark labeling. Specifically, for every node v, the benchmark labeling assigns to v all labels it likes, *i.e.*, labels ℓ for which $c(v, \ell) < 0$, and does not assign to v any of the labels it dislikes, *i.e.*, labels ℓ for which $c(v, \ell) \geq 0$. Thus, focusing on a single label ℓ, f deviates from the benchmark labeling by $c(v, \ell)$ if $\ell \in f(v)$ and ℓ is a label v dislikes, *i.e.*, $c(v, \ell) \geq 0$. Similarly, f deviates from the benchmark labeling by $|c(v, \ell)|$ if $\ell \notin f(v)$ and ℓ is a label v likes, *i.e.*, $c(v, \ell) < 0$. Formally, denote by $K^+(v) \triangleq \{\ell \in K : c(v, \ell) \geq 0\}$ the collection of all labels v dislikes, and by $K^-(v) \triangleq \{\ell \in K : c(v, \ell) < 0\}$ the collection of all labels v likes. Then, the total assignment cost of node v with respect to f is: $\sum_{\ell \in K^+(v)} c(v, \ell) \mathbf{1}_{\{\ell \in f(v)\}} + \sum_{\ell \in K^-(v)} |c(v, \ell)| \, \mathbf{1}_{\{\ell \notin f(v)\}}$.

Let us now focus on the separation costs. The separation cost of edge (u, v) is the number of labels nodes u and v disagree on, *i.e.*, the ℓ_1 distance between the characteristic vectors of $f(u)$ and $f(v)$. Formally, a pair of nodes (u, v), given a multi-labeling f, incurs the following separation cost: $s(u, v) \cdot ||\mathbf{1}_{f(u)} - \mathbf{1}_{f(v)}||_1$. For any subset of labels $S \subseteq K$, $\mathbf{1}_S$ denotes the characteristic vector of S. Summing up over the above we are now ready to provide a formal definition of the (MML) problem: find a feasible multi-labeling f that minimizes

$$\sum_{v \in V} \left(\sum_{\ell \in K^+(v)} c(v, \ell) \mathbf{1}_{\{\ell \in f(v)\}} + \sum_{\ell \in K^-(v)} |c(v, \ell)| \, \mathbf{1}_{\{\ell \notin f(v)\}} \right) + \sum_{(u,v) \in E} s(u, v) ||\mathbf{1}_{f(u)} - \mathbf{1}_{f(v)}||_1 \quad (1)$$

Summarizing, (MML) is a novel classification model in which multiple labels can be assigned to objects. We emphasize that in (MML), obtaining a solution to the (global) optimization objective is decoupled from the local learning process for the objects, thus allowing us to view the output of these processes as part of the input to (MML), and treating them in a "black box" fashion.

Let us now focus on our results. We introduce the correlated randomized dependent rounding problem and the (MML) problem. We tackle the correlated dependent rounding problem for the case of multiple (possibly different) uniform matroids, as summarized in the following theorem.

▶ **Theorem 1.** *Let K be a ground set of size k and $M_1, \ldots, M_n$ be n uniform matroids over K, where $rank(M_i) = b_i$. Additionally, let $\mathbf{y}^i \in \{\mathbf{y} \in [0,1]^k : \sum_{\ell=1}^{k} y_\ell \leq b_i\}$ for every $i = 1, \ldots, n$. Then there is an efficient algorithm for sampling $\mathbf{z}^1, \ldots, \mathbf{z}^n$ s.t.: (1) $\mathbf{z}^i$ is the characteristic vector of an independent set of M_i for every $i = 1, \ldots, n$; (2) $\mathbb{E}\left[\mathbf{z}^i\right] = \mathbf{y}^i$ for every $i = 1, \ldots, n$; and (3) $\mathbb{E}\left[||\mathbf{z}^i - \mathbf{z}^j||_1\right] \leq O(\log k)||\mathbf{y}^i - \mathbf{y}^j||_1$ for every $i, j = 1, \ldots, n$.*

Note that the loss in the distance, *i.e.*, property (3) above, depends only on the size of the ground set k and not on the number of given matroids n.

We use the above to obtain a (true) approximation of $O(\log k)$ for (MML). For the special case of (Unbounded-MML) we present a tight 2-approximation.

▶ **Theorem 2.** *The (MML) problem admits a (true) approximation of $O(\log k)$.*

▶ **Theorem 3.** *The (Unbounded-MML) problem admits an approximation of 2.*

▶ **Theorem 4.** *Assuming the unique games conjecture, the (Unbounded-MML) problem does not admit an approximation better than $2\,(1 - 1/k)$.*

Let us now focus on our approach and techniques. Consider the correlated dependent rounding problem, we now elaborate as to why known techniques fail when applied to it. The problem of rounding of online paging [6] is closely related to correlated dependent rounding. Unfortunately, techniques developed in the paging context allow us to bound distances only between *some* of the pairs of points, *i.e.*, $\mathbb{E}\left[||\mathbf{z}^{i+1} - \mathbf{z}^i||_1\right]$ for every $i = 1, \ldots, n-1$, as opposed to the desired $\mathbb{E}\left[||\mathbf{z}^i - \mathbf{z}^j||_1\right]$ for every $i, j = 1, \ldots, n$. Therefore, a different approach is required.

We note that achieving requirements (1) and (2) alone, *i.e.*, $\mathbf{z}^i \in \mathcal{P} \cap \{0,1\}^k$ and $\mathbb{E}\left[\mathbf{z}^i\right] = \mathbf{y}^i$ for every $i = 1, \ldots, n$, has already been achieved by any of the dependent rounding algorithms that can be applied to a uniform matroid, *e.g.*, [10, 12, 35] (just execute the algorithm independently for each $\mathbf{y}^i$). Obviously, this approach completely fails when considering requirement (3), *i.e.*, $\mathbb{E}\left[||\mathbf{z}^i - \mathbf{z}^j||_1\right] \leq \alpha||\mathbf{y}^i - \mathbf{y}^j||_1$, as α might be unbounded. The reason for the latter is that if $\mathbf{y}^i = \mathbf{y}^j$ for some $i \neq j$, then $||\mathbf{y}^i - \mathbf{y}^j||_1 = 0$ but $\mathbb{E}\left[||\mathbf{z}^i - \mathbf{z}^j||_1\right] > 0$ (as the two executions of dependent rounding, one for $\mathbf{y}^i$ and the other for $\mathbf{y}^j$, are independent).

Our approach to solving the above is to correlate all n executions of dependent rounding, one for each $\mathbf{y}^1, \ldots, \mathbf{y}^n$. Specifically, we execute the randomized dependent rounding algorithm of [35] for each $\mathbf{y}^i$ separately, but use the *same* random bits as input for all n different executions. Remarkably, this simple approach suffices. However, we note that the analysis of our algorithm uses the specific inner-workings of the algorithm of [35]. Hence, it seems that correlated dependent rounding cannot be easily solved through a "black box" application of any dependent rounding algorithm, *e.g.*, [10, 12].

Let us now focus on the special case (Unbounded-MML) and illustrate why known algorithms and techniques fail when applied to it. (Unbounded-MML) is inspired by the *metric labeling* problem, first introduced in full generality by [25]. In the metric labeling problem we are given an edge weighted graph $G = (V, E)$, a collection K of k labels, a *non-negative* assignment cost function $c : V \times K \to \mathbb{R}^+$, and a metric d over K. The goal is to assign a *single* label to each node while minimizing the sum of assignment and separation costs. As in (Unbounded-MML), assignment costs are defined using c, whereas the separation cost of edge (u, v) is the distance in the metric d between the labels assigned to u and v. It is important to note that metric labeling differs from (Unbounded-MML) in two main points: (1) each object can be assigned exactly one label, as opposed to multiple labels

in (Unbounded-MML), and (2) the assignment cost function c is non-negative, whereas in (Unbounded-MML) assignment costs may be either positive or negative.

Consider a further restricted special case of (Unbounded-MML) where all assignment costs are non-negative. If one applies the algorithm of [25] by expanding the label set K to $2^K \setminus \emptyset$ and considering the ℓ_1 metric on the expanded set[2], then this not only results in a large approximation guarantee of $O(k)$, but also the running time of the algorithm scales with 2^k and not k. More generally, we wish to claim that existing techniques and algorithms for the metric labeling problem cannot be directly applied to (Unbounded-MML). Consider a node v which has multiple labels ℓ it likes, *i.e.*, $c(v, \ell) < 0$. Since only a single label is allowed per node in metric labeling, it must be the case that whatever algorithm or technique we use, there is at least one label v likes that ultimately is not assigned to v. Thus, potentially incurring a huge loss in the objective.

We address the above difficulties by employing two approaches. First, we use a *global* charging argument over all labels in K when bounding the separation cost of an edge (u, v). Typically, such global arguments are avoided, *e.g.*, all known algorithm for metric labeling (either with a general or a specific metric) do not employ any type of global argument. Second, we *distort* the optimal marginal probabilities $x_{v,\ell}$ given by the linear programming relaxation for (Unbounded-MML). This enables us to balance both positive and negative assignment costs, along with separation costs.

Let us now mention some related work. An extensively studied topic is that of dependent rounding of fractional solution. A randomized variant of pipage rounding [1] was given by [22] who applied it to assignment polytopes (see also [26, 35]). An approach based on maximum entropy for dependent rounding was introduced by [3] in the context of max-min allocations, and was later extended to spanning trees by [2]. When considering general matroid independence polytopes, [10, 12] provided methods of conducting dependent rounding.

(MML) gets as input costs for assigning labels to objects and a similarity measure between objects. The labeling costs are based on a multi-label learning process (supervised learning) which is applied to a set of instances, each belonging potentially to multiple classes (labels), and predicts a set of class labels given a new instance. *Multi-label classification* has attracted much attention following various real world problems requiring usage of multiple labels [37], and thorough surveys in this area can be found in [34, 36]. The basic approach transforms the original problem into several instances of simpler binary classification problems, where each instance corresponds to a single label. This method is called *binary relevance*, and it assumes that labels are independent of each other, and thus one needs to solve k separate binary-label classification problems, where k denotes the number of labels. Approaches based on classifier chains have been adopted to model interdependencies between labels while maintaining acceptable computational complexity [33].

The *label power set* approach transforms the problem into a multi-class problem [14], where labels in the multi-class problem are a cross product of the original labels (and cover all possible combinations of these labels), resulting in the problem of mapping each data point to a binary vector. The main drawback of this approach is poor scaling in terms of the number of labels (e.g., vision problems where the number of categories may be large). A different approach addresses the problem directly, in its full generality, and is much harder than the traditional binary and multi-class problems, which in fact are special cases of multi-labeling. Some notable examples of multi-label algorithms, which are extensions based on binary

[2] Only the general algorithm of [25] is known for the case of ℓ_1 distances over the k-dimensional hypercube, and it achieves an approximation of $O(k)$. This guarantee is tight as it based on tree metrics.

problems, are adaptations of AdaBoost [21], the *ML-kNN* [41] based on kNN algorithm [20], and *Clare* which is an adapted decision tree algorithm for multi-label classification [31].

Another related machine learning approach is *kernel pairwise classification* [40]. Here, relations between pairs of samples are given using kernels. Supervised pairwise prediction aims to predict such pairwise relationships based on known relationships. Pairwise prediction takes a pair of instances as its input, and outputs the relationship between the two instances. The application of kernel methods to pairwise classification is based on a kernel function between two pairs of instances [24]. The main difference between this approach and our setting is that it does not consider single items, but rather focuses only on pairwise relations.

Metric labeling is an elegant and powerful mathematical model capturing a wide range of classification problems, where information about objects, as well as their pairwise relations, is given. Notice that such a scenario is not captured by neither known multi-class classification techniques, nor by existing pairwise kernel based techniques. The problem was first formulated in full generality by [25], and captures many classification problems that arise in various settings. Specifically, metric labeling has applications in important fields such as Markov theory [13, 27], image processing and computer vision [18, 8], as well as language modeling [29]. In [25], the authors gave an $O(\log k)$-approximation for any metric[3], and a 2-approximation for the uniform metric case. The latter is known to be tight assuming the unique games conjecture [28]. It is worth mentioning that metric labeling is of much importance in the combinatorial optimization setting, as it captures well studied problems such as multiway cut [9, 15, 16, 17, 23] and 0-extension [11, 19].

2 Preliminaries

We formulate the following natural linear programming relaxation for the (MML) problem (similarly to the relaxation given by [25] for uniform metric labeling). Variable $x_{v,\ell}$ is the (fractional) indicator for labeling node v with label ℓ. The first constraint guarantees that each node v receives between 1 and b_v labels. The following two constraints, along with the fact that the problem is a minimization problem, imply that $z_{u,v,\ell} = |x_{u,\ell} - x_{v,\ell}|$, i.e., $z_{u,v,\ell}$ is the separation cost of nodes u and v with respect to label ℓ. Hence, the fourth constraint asserts that $d_{u,v}$ equals $||\mathbf{x}_u - \mathbf{x}_v||_1$, where $\mathbf{x}_u = (x_{u,1}, \ldots, x_{u,\ell})$ for every $u \in V$. The objective of the relaxation follows directly from the definition of (MML) (1).

$$
\begin{aligned}
\min \quad & \sum_{v \in V} \left[\sum_{\ell \in K^+(v)} c(v,\ell) x_{v,\ell} + \sum_{\ell \in K^-(v)} |c(v,\ell)| \, (1 - x_{v,\ell}) \right] + \sum_{u,v \in V} s(u,v) d_{u,v} && \\
s.t. \quad & 1 \leq \sum_{\ell \in K} x_{v,\ell} \leq b_v && \forall v \in V \\
& z_{u,v,\ell} \geq x_{u,\ell} - x_{v,\ell} && \forall u, v \in V \\
& z_{u,v,\ell} \geq x_{v,\ell} - x_{u,\ell} && \forall u, v \in V \\
& d_{u,v} = \sum_{\ell \in K} z_{u,v,\ell} && \forall u, v \in V \\
& 0 \leq x_{v,\ell} \leq 1 && \forall v \in V, \forall \ell \in K
\end{aligned}
$$

The following observation simplifies the analysis of the separation cost considerably.

▶ Observation 5. Without loss of generality we can simply assume that any two adjacent nodes differ in only a single coordinate, by a value $\varepsilon > 0$, which can be made arbitrarily

[3] The metric over the labels determines their pairwise distances and can be arbitrary in general.

small. Specifically, given $(u,v) \in E$ we assume that $\mathbf{x}_u = (x_{u,1}, x_{u,2}, \ldots, x_{u,k})$ and $\mathbf{x}_v = (x_{u,1} + \varepsilon, x_{u,2}, \ldots, x_{u,k})$.

3 Correlated Randomized Dependent Rounding

Denote by M_{K,b_v} the uniform matroid over K of rank b_v, and recall that $\mathcal{P}(M_{K,b_v}) = \{\mathbf{x} \in [0,1]^k : \sum_{\ell=1}^{k} x_\ell \leq b_v\}$ is the standard independent set polytope corresponding to M_{K,b_v}. For completeness, we start by presenting the basic building block of [35] for rounding a single point in $\mathcal{P}_{M_{K,b_v}}$, as we later require its inner-workings.

Let us now focus on rounding a single point in the uniform matroid polytope. The basic building block (Algorithm 1) receives two marginal probabilities, $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$, for the i^{th} and j^{th} labels correspondingly, and randomly updates them. At least one of the updated marginal probabilities, denoted by α' and β', is "rounded" to either 0 or 1. This is done while deterministically preserving the sum of the marginal probabilities, and each of the marginal probabilities is preserved in expectation. Lemma 6 summarizes the above, and its proof is deferred to a full version of the paper. It is important to note that $0 \leq \alpha', \beta' \leq 1$ always, *i.e.*, Algorithm 1 returns valid marginal probabilities.

Algorithm 1 Resolve(i, α, j, β)

draw a threshold $\theta \sim Unif[0,1]$.
if (case (a)) $0 \leq \alpha + \beta \leq 1$ **then**
  if $\theta \leq {}^{\alpha}/_{(\alpha+\beta)}$ **then**
    $\alpha' \leftarrow \alpha + \beta$, $\beta' \leftarrow 0$, and $s \leftarrow j$.
  else
    $\alpha' \leftarrow 0$, $\beta' \leftarrow \alpha + \beta$, and $s \leftarrow i$.
  end if
end if
if (case (b)) $1 < \alpha + \beta \leq 2$ **then**
  if $\theta \leq {}^{(1-\beta)}/_{(2-\alpha-\beta)}$ **then**
    $\alpha' \leftarrow 1$, $\beta' \leftarrow \alpha + \beta - 1$, and $s \leftarrow i$.
  else
    $\alpha' \leftarrow \alpha + \beta - 1$, $\beta' \leftarrow 1$, and $s \leftarrow j$.
  end if
end if
return (i, α', j, β') and declare s as *fixed*.

▶ **Lemma 6.** *Upon the termination of Algorithm 1:*

1. $\mathbb{E}[\alpha'] = \alpha$ *and* $\mathbb{E}[\beta'] = \beta$.
2. $\alpha' + \beta' = \alpha + \beta$ *always.*
3. *One of* i *and* j *is declared* fixed *and its marginal value belongs to* $\{0,1\}$.

Define a *label tree* T of K to be a full binary tree with exactly k leaves, where each leaf corresponds to a distinct label of K. We now describe the rounding procedure which we denote by *label tree rounding*. It receives as input a label tree T, a point $\mathbf{x}_v \in \mathcal{P}(M_{K,b_v})$, a collection of independent random thresholds $\theta_z \sim Unif[0,1]$ for every non-leaf node z of T, and one additional independent random threshold $\theta \sim Unif[0,1]$. The label tree rounding procedure operates as follows:

1. Every leaf of T sends to its parent its label and its marginal value as given by the relaxation, *i.e.*, a leaf that corresponds to label $\ell \in K$ sends to its parent $(\ell, x_{v,\ell})$.
2. Every non-leaf node z of T (that is not the root) receives from its two children (i,α) and (j,β); it executes Algorithm 1 with parameters (i,α,j,β) and θ_z as the random threshold to obtain (α',β'); updates the marginal probabilities of i and j to be α' and β' respectively; and sends to its parent in T the label that was *not* fixed from $\{i,j\}$ along with its newly updated marginal probability.
3. The root r of T operates exactly as any other non-leaf node of T with the following exception: instead of sending the label that was not fixed to its parent along with its newly updated marginal probability, r uses the given random threshold θ to round the label that was not fixed, *i.e.*, after the execution of Algorithm 1 by r if $s \in \{i,j\}$ denotes the label that is not fixed and the newly updated marginal probability of s equals γ, then r sets the marginal of s to be 1 if $\theta \leq \gamma$ and 0 otherwise.

The following lemma summarizes the desired properties of the label tree rounding procedure, and its proof is deferred to a full version of the paper.

▶ **Lemma 7.** *Let $v \in V$, $\mathbf{x}_v \in \mathcal{P}(M_{K,b_v})$, T a label tree of K, and denote by $\tilde{\mathbf{x}}_v$ the vector of marginal probabilities obtained by executing the label tree rounding procedure. Then,*

1. $\tilde{\mathbf{x}}_v \in \{0,1\}^k$.
2. *Let $B_v \triangleq \sum_{\ell \in K} x_{v,\ell}$, then $\lfloor B_v \rfloor \leq \sum_{\ell \in K} \tilde{x}_{v,\ell} \leq \lceil B_v \rceil$ always.*
3. *For every $\ell \in K$: $\Pr[\tilde{x}_{v,\ell} = 1] = x_{v,\ell}$.*

Let us now focus on rounding multiple points in the uniform matroid polytope. In this section we describe how to round multiple points in $\mathcal{P}(M_{K,b_v})$ while: (1) preserving marginal probabilities; and (2) being "faithful" to the original ℓ_1 distances between any pair of points in $\mathcal{P}(M_{K,b_v})$. Our correlated rounding procedure receives as input a fixed label tree T, along with n points $\{\mathbf{x}_v\}_{v\in V}$ in $\mathcal{P}(M_{K,b_v})$. Intuitively, it applies the label tree rounding procedure to all n points simultaneously, while using the same given tree T and the same random thresholds in all executions. A formal description appears in Algorithm 2. As before, we denote by $\tilde{\mathbf{x}}_v$ the output of Algorithm 2 for node $v \in V$.

Algorithm 2 Correlated Rounding$\left(\{\mathbf{x}_v\}_{v\in V}, T\right)$

For every non-leaf node z of T draw an independent threshold $\theta_z \sim Unif[0,1]$.
Draw an independent threshold $\theta \sim Unif[0,1]$ for the root r of T.
$\forall v \in V$: execute label tree rounding with input T, $\mathbf{x}_v$, $\{\theta_z\}_{z \text{ non-leaf node of } T}$, and θ.
Output the resulting $\{\tilde{\mathbf{x}}_v\}_{v\in V}$.

Lemma 8 bounds the expected separation cost of neighbouring nodes u and v. Assuming $\mathbf{x}_u$ and $\mathbf{x}_v$ differ only in label 1 (as Observation 5 states without loss of generality), the executions of Algorithm 1 can differ between u and v only at non-leaf nodes of T that lie on the (single) path from the leaf that represents label 1 and the root r of T. At the heart of the proof lies the following observation: the expected *additive* increase in $||\mathbf{x}_u - \mathbf{x}_v||_2$ is $O(\varepsilon)$ for each of the non-leaf nodes of T that lie on the above mentioned path.

▶ **Lemma 8.** *Let $u,v \in V$ be such that $\mathbf{x}_u$ and $\mathbf{x}_v$ satisfy Observation 5, let $\tilde{\mathbf{x}}_u$ and $\tilde{\mathbf{x}}_v$ be the output of Algorithm 2 for nodes u and v correspondingly, and let δ be the depth of T. Then, $\mathbb{E}[||\tilde{\mathbf{x}}_u - \tilde{\mathbf{x}}_v||_1] \leq O(\delta)\varepsilon$.*

Proof. Recall that Observation 5 states that $\mathbf{x}_u$ and $\mathbf{x}_v$ are identical, except that $x_{v,1} = x_{u,1} + \varepsilon$. Hence, let P be the path from the leaf in T representing label 1 to the root r of T,

and denote the sequence of nodes in this path by $z_1, z_2, z_3 \ldots, z_m$ (where z_1 is the leaf and z_m is the root r). We use the following two assumptions that can be made without loss of generality.

First, as the order of executions of Algorithm 1 at the nodes of T is irrelevant to the outcome of Algorithm 2, as long as execution of Algorithm 1 at some node z of T is performed after all executions of Algorithm 1 at all non-leaf nodes in the induced subtree of T that z is its root. Hence, let us assume without loss of generality that all executions of Algorithm 1 at nodes not in P are performed before any execution of Algorithm 1 at nodes $z_2, z_3, \ldots, z_m$.

Second, note that in every non-leaf node along P, *i.e.*, $z_2, z_3, \ldots, z_m$, exactly one execution of Algorithm 1 is performed for each of the nodes u and v. The execution of Algorithm 1 at some node z_p, $p = 2, \ldots, m$, receives exactly two labels as input, one from the child z_{p-1} (along the path P) and the other from the other child of z_p which we denote by w_{p-1} (not on the path P). It is important to note that each of these two inputs might be random, however, the input received from node w_{p-1} is *always identical* for both u and v. Therefore, let us denote for simplicity of presentation and without loss of generality that the input w_{p-1} sends to the execution of Algorithm 1 at node z_p is label number p with its updated marginal probability γ_p, *i.e.*, (p, γ_p). Thus, we can focus only on the first m labels of K since for labels $m+1, \ldots, k$ nodes u and v will always be identical and their contribution to $||\tilde{\mathbf{x}}_u - \tilde{\mathbf{x}}_v||_1$ will be always 0.

Denote by $\mathbf{x}_u^t \in [0,1]^m$ and $\mathbf{x}_v^t \in [0,1]^m$ the vector of marginal probabilities of the first m labels after performing the execution of Algorithm 1 at node z_t, for nodes u and v respectively. Thus, for example, $\mathbf{x}_u^1 = (u_1, \gamma_2, \gamma_3, \ldots, \gamma_m)$ and $\mathbf{x}_v^1 = (u_1 + \varepsilon, \gamma_2, \gamma_3, \ldots, \gamma_m)$, and $\mathbf{x}_u^m = (\tilde{x}_{u,1}, \tilde{x}_{u,2}, \tilde{x}_{u,3}, \ldots, \tilde{x}_{u,m})$ and $\mathbf{x}_v^m = (\tilde{x}_{v,1}, \tilde{x}_{v,2}, \tilde{x}_{v,3}, \ldots, \tilde{x}_{v,m})$. We prove that:

$$\mathbb{E}\left[||\mathbf{x}_u^t - \mathbf{x}_v^t||_1 - ||\mathbf{x}_u^{t-1} - \mathbf{x}_v^{t-1}||_1\right] \leq 2\varepsilon \qquad \forall t = 2, 3, \ldots, m. \tag{2}$$

The proof of the lemma is completed by summing (2) over all relevant values of t, and recalling that $||\mathbf{x}_u^1 - \mathbf{x}_v^1||_1 = ||\mathbf{x}_u - \mathbf{x}_v||_1 = \varepsilon$. Inequality (2) is proved by examining the joint distribution of Algorithm 1 at node z_t for both u and v. This computation is deferred to a full version of the paper. ◀

Proof (of Theorem 1). Apply Algorithm 2 to the given points $\mathbf{y}^1, \ldots, \mathbf{y}^n$ with a label tree T whose depth is $O(\log k)$. Lemmas 7 and 8 conclude the proof. ◀

4 $O(\log k)$-Approximation for MML

Proof (of Theorem 2). We apply Algorithm 2 to the fractional solution provided by the linear programming relaxation. Starting with assignment costs, Property (3) of Lemma 7 implies that all assignment costs are preserved in expectation. Considering separation costs, one can always choose a label tree T whose depth is $O(\log k)$, and thus Lemma 8 implies a multiplicative loss of $O(\log k)$ in the separation costs. Finally, Property (2) of Lemma 7 guarantees that every node $v \in V$ is assigned at most $\lceil B_v \rceil$ labels, but since $\lceil B_v \rceil \leq b_v$ our algorithm never deviates from the bound on the number of labels. Additionally, it is important to note that Property (2) of Lemma 7 also ensures that every node $v \in V$ is assigned at least one label since $\lfloor B_v \rfloor \geq 1$. ◀

5 A Tight Approximation for Unbounded MML

Let us now focus on the basic building blocks. We use the following two algorithms as basic building blocks for our final algorithm. The first is a simple single threshold algorithm.

Let $h : [0,1] \to [0,1]$ be a monotone non-decreasing *distortion* function. The algorithm applies the distortion function h to each fractional value $x_{v,\ell}$, and then finds a multi-labeling $f_{ST}(v) : V \to 2^K$ by assigning to v all labels ℓ whose *distorted* fractional value is larger than a uniformly random threshold $\theta \in [0,1]$. The choice of an appropriate distortion function h plays a crucial role in obtaining the best possible approximation of 2 for (Unbounded-MML).

Algorithm 3 Single Threshold (ST)

draw a threshold $\theta \sim Unif[0,1]$.
for every $v \in V$ **do**
  $f_{ST}(v) \leftarrow \{\ell : \theta \leq h(x_{v,\ell})\}$.
end for
output f_{ST}.

The second building block we use is due to [25]. It is the 2-approximation they provide for the uniform metric labeling problem.

Algorithm 4 Kleinberg-Tardos (KT)

while $V \neq \emptyset$ **do**
  independently draw a threshold $\theta \sim Unif[0,1]$ and a uniform label $\ell \in K$.
  for every $v \in V$ **do**
    if $\theta \leq x_{v,\ell}$ **then**
      $f_{KT}(v) \leftarrow \{\ell\}$ and $V \leftarrow V \setminus \{v\}$.
    end if
  end for
end while
output f_{KT}.

Our algorithm is a simple "merge" of the two basic building blocks: Algorithms 3 and 4 are run independently and the union of their label assignments is returned.

Algorithm 5 Union

independently run Algorithms 3 and 4 to obtain f_{ST} and f_{KT}.
for every $v \in V$ **do**
  $f(v) \leftarrow f_{ST}(v) \cup f_{KT}(v)$.
end for
output f.

Let us focus on the assignment costs. We denote by $\mathbf{x}_v \in [0,1]^k$ the vector corresponding to v, i.e., $(\mathbf{x}_v)_\ell = x_{v,\ell}$. First we start by stating two immediate observations regarding the assignment probabilities of labels to vertices by the basic building blocks (a full proof is deferred to a full version of the paper).

▶ **Lemma 9.** *For any $v \in V$ and $\ell \in K$ the following two claims hold:*

1. $\Pr[\ell \in f_{ST}(v)] = h(x_{v,\ell})$.
2. $\Pr[f_{KT}(v) = \{\ell\}] = \frac{x_{v,\ell}}{||\mathbf{x}_v||_1}$.

The following corollary states the probability that label ℓ is assigned to vertex v by the Union Algorithm (Algorithm 5), and is used to bound the total labeling cost of Algorithm 5. Its proof is deferred to a full version of the paper.

▶ **Corollary 10.** *For any $v \in V$ and $\ell \in K$,* $\Pr[\ell \in f(v)] = h(x_{v,\ell}) + \frac{x_{v,\ell}}{||\mathbf{x}_v||_1} - h(x_{v,\ell}) \cdot \frac{x_{v,\ell}}{||\mathbf{x}_v||_1}$.

We focus now on the separation cost of the Union Algorithm (Algorithm 5). Note that the expected separation cost of the Union Algorithm (Algorithm 5) equals:

$$\sum_{(u,v)\in E} s(u,v) \cdot \mathbb{E}\left[\left|\left|\mathbf{1}_{f(u)} - \mathbf{1}_{f(v)}\right|\right|_1\right] \ . \tag{3}$$

The next lemma provides all the ingredients required for bounding the expected separation cost (3), its proof is deferred to a full version of the paper.

▶ **Lemma 11.** *For any $u, v \in V$ such that $\mathbf{x}_u = (x_{u,1}, x_{u,2}, \ldots, x_{u,k})$ and $\mathbf{x}_v = (x_{u,1} + \varepsilon, x_{u,2}, \ldots, x_{u,k})$, the following hold:*

1. $\Pr[1 \in f(u), 1 \notin f(v)] = 0$.
2. $\Pr[1 \notin f(u), 1 \in f(v)] = \varepsilon \cdot \left(1 - \frac{x_{u,1}}{||\mathbf{x}_u||_1}\right) \cdot \left(\frac{h(x_{u,1}+\varepsilon) - h(x_{u,1})}{\varepsilon} + \frac{1-h(x_{u,1}+\varepsilon)}{||\mathbf{x}_u||+\varepsilon}\right)$.
3. $\Pr[\ell \in f(u), \ell \notin f(v)] = \varepsilon \cdot \frac{x_{u,\ell}(1-h(x_{u,\ell}))}{||\mathbf{x}_u||_1(||\mathbf{x}_u||_1+\varepsilon)}$ *for every* $\ell \neq 1$.
4. $\Pr[\ell \notin f(u), \ell \in f(v)] = 0$ *for every* $\ell \neq 1$.

In order to bound the expected separation cost of an edge (u, v), as given by (3), we employ a *global* charging argument. Typically, if *local* charging works it is the case that the part of (3) that corresponds to a fixed label ℓ, *i.e.*, $\mathbb{E}\left[\mathbf{1}_{\{\ell \in f(u) \wedge \ell \notin f(v)\}} + \mathbf{1}_{\{\ell \notin f(u) \wedge \ell \in f(v)\}}\right]$ could be upper bounded by $\alpha \cdot z_{u,v,\ell}$ for some constant $\alpha > 0$. Unfortunately, this is not the case as can be seen from Lemma 11. Edge (u, v) satisfies Observation 5, *i.e.*, $\mathbf{x}_u$ and $\mathbf{x}_v$ differ only coordinate 1, and thus without loss of generality $z_{u,v,\ell} = 0$ for all $\ell \neq 1$. However, for example, case (3) of Lemma 11 implies that u and v have a non-zero probability of disagreeing on any label $\ell \neq 1$. Thus, a local charging argument as described above fails and we must resort to a global argument that sums over all possible labels ℓ. The following corollary provides exactly such a global guarantee, and its proof is deferred to a full version of the paper.

▶ **Corollary 12.** *Let $\mathbf{x}_u = (x_{u,1}, x_{u,2}, \ldots, x_{u,k})$ and $\mathbf{x}_v = (x_{u,1} + \varepsilon, x_{u,2}, \ldots, x_{u,k})$ for an edge $(u, v) \in E$. Then,*

$$\mathbb{E}\left[\left|\left|\mathbf{1}_{f(u)} - \mathbf{1}_{f(v)}\right|\right|_1\right] \leq \left(\frac{h(x_{u,1}+\varepsilon) - h(x_{u,1})}{\varepsilon} + \frac{2 - h(x_{u,1}+\varepsilon)}{||\mathbf{x}_u||_1 + \varepsilon}\right) \cdot \left(1 - \frac{x_{u,1}}{||\mathbf{x}_u||_1}\right) \cdot d_{u,v} \ .$$

Let us not focus on how to choose the distortion h. Given a specific choice of a distortion function $h : [0, 1] \to [0, 1]$, Corollaries 10 and 12 determine the approximation guarantee. Specifically, Corollary 10 determines the loss with respect to the labeling cost, and Corollary 12 determines the loss with respect to the separation cost.

The most natural distortion function is the identity, i.e., $h(x) = x$. The next theorem shows that this choice of h yields a 3-approximation for (Unbounded-MML).

▶ **Theorem 13.** *The Union Algorithm provides an approximation of 3 when $h(x) = x$.*

Proof. First, consider the labeling costs. Corollary 10, along with the fact that $||\mathbf{x}_v||_1 \geq 1$, imply:

$$\begin{cases} \Pr[\ell \in f(v)] = x_{v,\ell} + \frac{x_{v,\ell}}{||\mathbf{x}_v||_1} - \frac{x_{v,\ell}^2}{||\mathbf{x}_v||_1} \leq 2x_{v,\ell} \\ \Pr[\ell \notin f(v)] = (1 - x_{v,\ell}) \cdot \left(1 - \frac{x_{v,\ell}}{||\mathbf{x}_v||_1}\right) \leq 1 - x_{v,\ell} \end{cases}$$

Hence, the labeling costs incur a loss of at most a factor of 2. Second, consider the separation costs. Let $(u,v) \in E$ and assume without loss of generality that $\mathbf{x}_u = (x_{u,1}, x_{u,2}, \ldots, x_{u,k})$ and $\mathbf{x}_v = (x_{u,1} + \varepsilon, x_{u,2}, \ldots, x_{u,k})$. Corollary 12, along with the fact that $||\mathbf{x}_v||_1 \geq 1$, imply:

$$\mathbb{E}\left[\left|\left|\mathbf{1}_{f(u)} - \mathbf{1}_{f(v)}\right|\right|_1\right] \leq \left(\frac{\varepsilon}{\varepsilon} + \frac{2 - x_{u,1} - \varepsilon}{||\mathbf{x}_u||_1 + \varepsilon}\right)\left(1 - \frac{x_{u,1}}{||\mathbf{x}_u||_1}\right) d_{u,v} \leq 3d_{u,v} \ .$$

Thus, the separation costs incur a loss of at most a factor of 3, concluding the proof. ◀

We prove that choosing a quadratic distortion, i.e., $h(x) = x^2$, provides a tight approximation of 2 for (Unbounded-MML).We are now ready to prove Theorem 3.

Proof (of Theorem 3). For simplicity we prove the theorem in two phases. In the first phase we show that the quadratic distortion provides an approximation of $(2 + \varepsilon)$. In the second phase we show that, assuming $\varepsilon \leq (8k^4)^{-1}$, the approximation is in fact 2. This concludes the proof since ε can be chosen to be arbitrarily small.

Let us focus on the first phase. When considering the labeling costs, Corollary 10, along with the facts that $||\mathbf{x}_v||_1 \geq 1$ and $0 \leq x_{v,\ell} \leq 1$, imply:

$$\begin{cases} \Pr\left[\ell \in f(v)\right] = x_{v,\ell}^2 + \frac{x_{v,\ell}}{||\mathbf{x}_v||_1} - \frac{x_{v,\ell}^3}{||\mathbf{x}_v||_1} \leq x_{v,\ell} \cdot (1 - x_{v,\ell}) \leq 2x_{v,\ell} \\ \Pr\left[\ell \notin f(v)\right] = \left(1 - x_{v,\ell}^2\right) \cdot \left(1 - \frac{x_{v,\ell}}{||\mathbf{x}_v||_1}\right) \leq (1 - x_{v,\ell}^2) \leq 2(1 - x_{v,\ell}) \end{cases}$$

Hence, the labeling costs incur a loss of at most 2 in the approximation.

When considering the separation costs, let $(u,v) \in E$ and assume without loss of generality that $\mathbf{x}_u = (x_{u,1}, x_{u,2}, \ldots, x_{u,k})$ and $\mathbf{x}_v = (x_{u,1} + \varepsilon, x_{u,2}, \ldots, x_{u,k})$. Corollary (12) implies:

$$\begin{aligned} &\mathbb{E}\left[\sum_{\ell \in K}\left(\mathbf{1}_{\{\ell \in f(u) \wedge \ell \notin f(v)\}} + \mathbf{1}_{\{\ell \notin f(u) \wedge \ell \in f(v)\}}\right)\right] \leq \\ &\leq \left(\frac{(x_{u,1} + \varepsilon)^2 - x_{u,1}^2}{\varepsilon} + \frac{2 - (x_{u,1} + \varepsilon)^2}{||\mathbf{x}_u||_1 + \varepsilon}\right)\left(1 - \frac{x_{u,1}}{||\mathbf{x}_u||_1}\right) d_{u,v} \\ &= \left[\left(2x_{u,1} + \frac{2 - x_{u,1}^2}{||\mathbf{x}_u||_1 + \varepsilon}\right)\left(1 - \frac{x_{u,1}}{||\mathbf{x}_u||_1}\right) + \varepsilon\left(\frac{||\mathbf{x}_u||_1 - 2x_{u,1}}{|\mathbf{x}_u||_1 + \varepsilon}\right)\left(1 - \frac{x_{u,1}}{||\mathbf{x}_u||_1}\right)\right] d_{u,v} \ . \quad (4) \end{aligned}$$

Define the following function $L(z,t) : [0,1] \times [1,\infty) \to \mathbb{R}^+$, $L(z,t) \triangleq \left(2z + \frac{2-z^2}{t}\right) \cdot \left(1 - \frac{z}{t}\right)$. Clearly the maximum value of L upper bounds the left term of (4), when plugging $z = x_{u,1}$ and $t = ||\mathbf{x}_u||_1$. One can verify that $\max_{0 \leq z \leq 1} \max_{t \geq 1} \{L(z,t)\} \leq 2$ (details are deferred to a full version of the paper). Note that the right term of (4) is at most ε, hence the expected separation cost is at most $(2+\varepsilon)d_{u,v}$. This concludes the first phase. The proof of the second phase is deferred to a full version of the paper. ◀

The proof of Theorem 4 is deferred to a full version of the paper.

Acknowledgements. Joseph (Seffi) Naor's work is supported in part by ISF grant 1585/15 and US-Israel BSF grant 2014414 (part of this work was done while visiting the Simons Institute for the Theory of Computing). Roy Schwart'z work is supported by ISF grant 1336/16.

References

1. A. A. Ageev and M. I. Sviridenko. Pipage rounding: a new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8:307–328, 2004.
2. Arash Asadpour, Michel X. Goemans, Aleksander Mądry, Shayan Oveis Gharan, and Amin Saberi. An $O(\log n/\log\log n)$-approximation algorithm for the asymmetric traveling salesman problem. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 379–389, 2010.
3. Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM J. Comput.*, 39(7):2970–2989, 2010.
4. Zafer Barutçuoglu, Robert E. Schapire, and Olga G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.
5. Hendrik Blockeel, Leander Schietgat, Jan Struyf, Saso Dzeroski, and Amanda Clare. Decision trees for hierarchical multilabel classification: A case study in functional genomics. In *PKDD*, pages 18–29, 2006.
6. Avrim Blum, Carl Burch, and Adam Kalai. Finely-competitive paging. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS '99, pages 450–457, 1999.
7. Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.
8. Yuri Boykov, Olga Veksler, and Ramin Zabih. Markov random fields with efficient approximations. In *CVPR*, pages 648–655, 1998.
9. Niv Buchbinder, Joseph (Seffi) Naor, and Roy Schwartz. Simplex partitioning via exponential clocks and the multiway cut problem. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 535–544, 2013.
10. Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrak. Maximizing a submodular set function subject to a matroid constraint. *SIAM Journal on Computing*, 2011.
11. Gruia Călinescu, Howard Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. SODA '01, pages 8–16, 2001.
12. Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 575–584, 2010.
13. R Chellappa and A. Jain. *Markov Random Fields: Theory and Applications*. Academic Press, 1993.
14. Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:265–292, 2002.
15. Gruia Călinescu, Howard Karloff, and Yuval Rabani. An improved approximation algorithm for multiway cut. *J. Comput. Syst. Sci.*, 60(3):564–574, 2000.
16. William H. Cunningham and Lawrence Tang. Optimal 3-terminal cuts and linear programming. IPCO '99, pages 114–125, 1999.
17. E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894, 1994.
18. Richard C. Dubes and Anil K. Jain. Random field models in image analysis. *Journal of Applied Statistics*, 16(2):131–164, 2006.
19. Jittat Fakcharoenphol, Chris Harrelson, Satish Rao, and Kunal Talwar. An improved approximation algorithm for the 0-extension problem. SODA '03, pages 257–265, 2003.
20. Evelyn Fix and Joseph L Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, DTIC Document, 1951.

21 Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

22 Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *J. ACM*, 53(3):324–360, 2006.

23 David R. Karger, Philip N. Klein, Clifford Stein, Mikkel Thorup, and Neal E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *Math. Oper. Res.*, 29(3):436–461, 2004.

24 Hisashi Kashima, Satoshi Oyama, Yoshihiro Yamanishi, and Koji Tsuda. On pairwise kernels: An efficient alternative and generalization analysis. pages 1030–1037, 2009.

25 Jon M. Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *J. ACM*, 49(5):616–639, 2002.

26 V. S. Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan. A unified approach to scheduling on unrelated parallel machines. *J. ACM*, 56(5):28:1–28:31, 2009.

27 Stan Z. Li. *Markov random field modeling in computer vision*. Computer science workbench. Springer, 1995.

28 Rajsekar Manokaran, Joseph (Seffi) Naor, Prasad Raghavendra, and Roy Schwartz. Sdp gaps and ugc hardness for multiway cut, 0-extension, and metric labeling. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, STOC '08, pages 11–20, 2008.

29 Stephen Della Pietra, Vincent J. Della Pietra, and John D. Lafferty. Inducing features of random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4):380–393, 1997.

30 Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang. Correlative multi-label video annotation. In *Proceedings of ACMMM '07*, pages 17–26, 2007.

31 J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

32 Prabhakar Raghavan and Clark D. Tompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

33 Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359, 2011.

34 Mohammad S Sorower. A literature survey on algorithms for multi-label learning. Technical report, 2010.

35 A. Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 588–597, 2001.

36 Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

37 Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *In Data Mining and Knowledge Discovery Handbook*, pages 667–685, 2010.

38 Naonori Ueda and Kazumi Saito. Parametric mixture models for multi-labeled text. In *NIPS*, pages 721–728, 2002.

39 Adriano Veloso, Wagner Meira Jr., Marcos André Gonçalves, and Mohammed Javeed Zaki. Multi-label lazy associative classification. In *PKDD*, pages 605–612, 2007.

40 Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International journal of computer vision*, 73(2):213–238, 2007.

41 Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *PATTERN RECOGNITION*, 40:2007, 2007.

42 Zhi-Hua Zhou and Min-Ling Zhang. Multi-instance multi-label learning with application to scene classification. In *Proceedings of NIPS*, pages 1609–1616, 2006.