Using the Pseudo-Dimension to Analyze Approximation Algorithms for Integer Programming

Philip M. Long

Genome Institute of Singapore
1 Research Link
IMA Building
National University of Singapore
Singapore 117604, Republic of Singapore

Abstract. We prove approximation guarantees for randomized algorithms for packing and covering integer programs expressed in certain normal forms. The bounds are in terms of the pseudo-dimension of the matrix of the coefficients of the constraints and the value of the optimal solution; they are independent of the number of constraints and the number of variables. The algorithms take time polynomial in the length of the representation of the integer program and the value of the optimal solution. We establish a related result for a class we call the mixed covering integer programs, which contains the covering integer programs. We describe applications of these techniques and results to a generalization of Dominating Set motivated by distributed file sharing applications, to an optimization problem motivated by an analysis of boosting, and to a generalization of matching in hypergraphs.

1 Introduction

Raghavan and Thompson [RT87] introduced *randomized rounding*. Roughly, their idea was to construct algorithms for integer programming problems as follows:

- solve a similar problem without the integrality constraint, and
- round each variable up or down, using its fractional part as the probability of rounding up.

This technique provides strong approximation guarantees for polynomial-time algorithms for a class of problems called covering and packing integer programs [Rag88,Sri99]. In a covering integer program, for an $m \times n$ matrix A and column vectors c and b, all with only nonnegative entries, the goal is to find $x \in \mathbf{Z}_+^n$ to minimize $c^T x$ subject to $Ax \geq b$. In a packing integer program, it is also assumed that A, c and b have only nonnegative entries, but the goal is to choose $x \in \mathbf{Z}_+^n$ to maximize $c^T x$ subject to $Ax \leq b$. Raghavan [Rag88] asked whether one could exploit algebraic properties of A such as its rank to obtain stronger approximation guarantees.

In this paper, we report on work along these lines. One can assume without loss of generality that covering and packing integer programs satisfy $A \in [0,1]^{m\times n}$, $c \in [1,\infty)^n$ and $b = (1,1,...,1)^T$ (see [Sri99] and Sections 3 and 5 of this paper). For covering and packing integer programs that are expressed this way, we prove approximation guarantees for efficient algorithms that are independent of the number of variables and constraints, and are in terms of the *pseudo-dimension* [Pol84,Hau92] of A. We also establish a similar result for *mixed covering integer programs*, which are like covering integer programs but without the requirement that the components of A are nonnegative.

The pseudo-dimension can be defined as follows [Hau92]. Say that an $m \times k$ matrix is *full* if the origin in $\mathbf{R}^k$ can be translated so that the rows of the matrix occupy all 2^k orthants. The pseudo-dimension of A is the size of the largest set of columns of A such that the matrix obtained by deleting all other columns is full. A more formulaic definition is given in Section 2.

Since the pseudo-dimension of A is at most its rank [Dud78,Pol84], our analysis implies results like those envisaged by Raghavan. Sometimes, however, the pseudo-dimension of a matrix is much smaller than its rank. For example, the pseudo-dimension of any identity matrix is 1.

Our general results are as follows. All of our algorithms are randomized and, with probability 1/2, achieve the claimed approximations in time polynomial in the number of bits needed to write A and c and the value of the optimal solution. (For many commonly studied combinatorial optimization problems, the value of the optimal solution is bounded by a polynomial in the size of the input [KT94,CK].) The algorithm for covering integer programs outputs a solution whose value is $O(\mathrm{opt}(1+dr\log(r\mathrm{opt})))$, where d is the pseudo-dimension of A and r is the value of the largest entry in A; since $r \leq 1$, the value of the algorithm's solution is also $O(d\mathrm{opt}\log\mathrm{opt})$. For mixed covering integer programs, the bound is $O(\mathrm{opt}(1+dr^2\mathrm{opt}))$; here we cannot assume that $r \leq 1$. For packing integer programs, our algorithm obtains a solution whose value is $\Omega\left(\frac{\mathrm{opt}}{(2r\mathrm{opt})^{krd}}\right)$, for a constant $k > 0$ (here once again $r \leq 1$).

We illustrate the application of our general result about covering integer programs using the B-domination problem [NR95,Sri99], a generalization of Dominating Set motivated by distributed file sharing applications. In the B-domination problem, the goal is to locate as few facilities as possible at the nodes of a network so that each node of the network has at least B facilities within one hop. We give a randomized algorithm that, for graphs of constant genus, with probability 1/2, outputs a solution of size $O\left(\mathrm{opt}\left(1+\frac{\ln\mathrm{opt}}{B}\right)\right)$ in polynomial time.

Our study of mixed covering integer programs was inspired by a learning problem, which can be abstracted as the *minimum majority problem* as follows: given an $m \times n$ matrix A with entries in $\{-1,1\}$, choose $x \in \mathbf{Z}_+^n$ to minimize $\sum_{i=1}^n x_i$ subject to $Ax > 0$. Our algorithm for mixed covering integer programs yields a bound of $O(d\mathrm{opt}^2)$ for this problem. We derive our motivation for this problem from an analysis of the generalization ability of hypotheses output by boosting algorithms [SFBL98]; details are given in Section 6.2.

Our general results about packing integer programs can be applied to simple B-matching [Lov75]. Here, given a family $\mathcal{S}$ of subsets of a finite set X, the goal is to output as many of the sets in $\mathcal{S}$ as possible while ensuring that each element of X is included in at most B of the chosen sets. We give a randomized polynomial-time algorithm for this problem that outputs a solution of size $\Omega((\mathrm{opt}/B)^{1-kd/B})$, where d is the VC-dimension of the dual of the input and $k > 0$ is an absolute constant.

Our work builds on that of Bronnimann and Goodrich [BG95] and Pach and Agarwal [PA95], who established approximation guarantees for polynomial-time algorithms for Set Cover in terms of the VC-dimension of the dual of the input set system. Our analysis of covering integer programs is a generalization of the analysis of Pach and Agarwal. Set Cover can be formulated as a covering integer program, and the pseudo-dimension of the resulting coefficient matrix is the same as the VC-dimension of the dual of the input set system.

Srinivasan [Sri99] showed that if a fractional solution is rounded as originally proposed by Raghavan and Thompson, then the events that the constraints are violated are *positively correlated*, and used this to improve the analysis of randomized rounding for packing and covering integer programs. Recently, he provided *RNC* and *NC* algorithms with the same approximation guarantees [Sri01]. However, his approximation bounds still depend on m.

Baker [Bak94] described a polynomial-time approximation scheme for Dominating Set when the input is restricted to be planar.

It is not hard to see how to use boosting [Sch90,Fre95], together with Lemma 3.3 of [HMP$^+$93], to design an algorithm for the minimum majority problem that outputs a solution with value $O(\mathrm{opt}^2 \log m)$. Since $d \leq \log m$, our bound is never more than a constant factor worse than this, but when $d << \log m$, it is significantly better.

For simple B-matching, the only bounds we know are in terms of opt and $|X|$; the best is $\Omega\left(\frac{\mathrm{opt}}{1+(|X|/\mathrm{opt})^{1/B}}\right)$ [Sri99]. When $d << B << \mathrm{opt} << |X|$ and $d << \log|X|$ (note again that $d \leq \log|X|$), our bound improves on this significantly.

2 Preliminaries

Denote the nonnegative rationals by $\mathbf{Q}_+$, and the nonnegative integers by $\mathbf{Z}_+$.

For a countable set X, a probability distribution D over X, and a predicate ϕ over X, denote by $\mathbf{Pr}_{x \in D}(\phi(x))$ the probability that $\phi(x)$ is true when x is chosen according to D. Define $\mathbf{E}_{x \in D}$ similarly. Denote by D^ℓ the distribution on X^ℓ obtained by sampling ℓ times independently according to D.

For a domain X, and a subset S of X, define χ_S to be the indicator function for S, i.e. function from X to $\{0,1\}$ for which $\chi_S(x) = 1 \Leftrightarrow x \in S$.

For a domain X, say that a set $\mathcal{F}$ of real-valued functions defined on X *shatters* a sequence $x_1, ..., x_d$ of elements of X if there is a sequence $r_1, ..., r_d$ of real thresholds such that for any $b_1, ..., b_d \in \{\text{above}, \text{below}\}$, there is an $f \in \mathcal{F}$ such that for all $i \in \{1, ..., d\}$, $f(x_i) \geq r_i \Leftrightarrow b_i = \text{above}$. Define the *pseudo-dimension* [Pol84] of $\mathcal{F}$, denoted by $\mathrm{Pdim}(\mathcal{F})$, to be the length of the longest sequence shattered by $\mathcal{F}$. The *VC-dimension* [VC71] of a set $\mathcal{F}$ of functions from X to $\{0,1\}$, denoted by VCdim($\mathcal{F}$), is its pseudo-dimension. The VC-dimension of a family $\mathcal{S}$ of subsets of X is the VC-dimension of $\{\chi_S : S \in \mathcal{S}\}$.

For a real matrix A, define the *pseudo-dimension of* A, denoted by $\mathrm{Pdim}(A)$, by thinking of the rows of A as functions and taking the pseudo-dimension of the resulting class of functions. Specifically, if A is an $m \times n$ matrix, for each $i \in \{1,...,m\}$ define $f_{A,i} : \{1,...,n\} \to \mathbf{R}$ by $f_{A,i}(j) = A_{i,j}$ and let $\mathrm{Pdim}(A) = \mathrm{Pdim}(\{f_{A,i} : i \in \{1,...,m\}\})$.

For a family $\mathcal{S}$ of sets define the *dual* of $\mathcal{S}$, denoted by dual($\mathcal{S}$) as follows. For each $x \in \cup_{S\in\mathcal{S}} S$, let $Q_{x,\mathcal{S}} = \{S \in \mathcal{S} : x \in S\}$. Let $\mathrm{dual}(\mathcal{S}) = \{Q_{x,\mathcal{S}} : x \in \cup_{S\in\mathcal{S}} S\}$.

Lemma 1 ([Vap82,Pol84]). *There is a constant $\kappa > 0$ such that for any $r > 0$, any finite set X, any set $\mathcal{F}$ of functions from X to $[0,r]$, any $\epsilon > 0$, and any probability distribution D over X, if $\ell \geq \frac{\kappa r \mathrm{Pdim}(\mathcal{F})}{\epsilon} \ln \frac{r}{\epsilon}$, then*

$$\mathbf{Pr}_{(z_1,\ldots,z_\ell)\in D^\ell}\left(\exists f \in \mathcal{F}, \mathbf{E}_{x\in D}(f(x)) \geq \epsilon \text{ but } \sum_{i=1}^{\ell} f(z_i) < \epsilon\ell/2\right) \leq 1/4.$$

Lemma 2 ([Tal94]). *There is a constant $\kappa > 0$ such that for any real a and b with $a \leq b$ (let $r = b - a$), any finite set X, any set $\mathcal{F}$ of functions from X to $[a,b]$, any $\epsilon > 0$, and any probability distribution D over X, if $\ell \geq \frac{\kappa r^2 \mathrm{Pdim}(\mathcal{F})}{\epsilon^2}$, then*

$$\mathbf{Pr}_{(z_1,\ldots,z_\ell)\in D^\ell}\left(\exists f \in \mathcal{F}, \left|\mathbf{E}_{x\in D}(f(x)) - \frac{1}{\ell}\sum_{i=1}^{\ell} f(z_i)\right| > \epsilon\right) \leq 1/4,$$

Lemma 3. *There is a constant $\kappa > 0$ such that for any $r > 0$, any finite set X, any set $\mathcal{F}$ of functions from X to $[0,r]$, any $\epsilon > 0$, and any probability distribution D over X, and for any $\alpha \geq 1$, if $\ell \geq \frac{\kappa r \mathrm{Pdim}(\mathcal{F})}{\alpha \ln(1+\alpha)\epsilon} \ln \frac{r}{\epsilon}$, then*

$$\mathbf{Pr}_{(z_1,\ldots,z_\ell)\in D^\ell}\left(\exists f \in \mathcal{F}, \mathbf{E}_{x\in D}(f(x)) \leq \epsilon \text{ but } \sum_{i=1}^{\ell} f(z_i) > (1+\alpha)\epsilon\ell\right) \leq 1/4.$$

We are not aware of a reference for Lemma 3. Its proof, whose rough outline follows those of related results (see [Pol84,Hau92,SAB93,AB99]), is omitted due to space contraints.

3 Covering Integer Programs

In a covering integer program, for natural numbers n and m, column vectors $c \in \mathbf{Q}_+^n$ and $b \in \mathbf{Q}_+^m$, and a matrix $A \in \mathbf{Q}_+^{m\times n}$, the goal is to choose $x \in \mathbf{Z}_+^m$ to minimize $c^T x$ subject to $Ax \geq b$.

Srinivasan [Sri99] showed that one can assume without loss of generality that $A \in [0,1]^{m\times n}$ and $b \in [1,\infty)^m$. By dividing each row i of A by b_i, one can further assume w.l.o.g. that each component of b is 1. Furthermore, one can assume that each component of c is positive, since if some $c_j = 0$, one can eliminate the jth variable by deleting all constraints that can be satisfied by making it arbitrarily large. Finally, we can scale c so that its least component is 1. This is summarized in the following.

Definition 1. *A* covering integer program in normal form *is given by a matrix* $A = [0,1]^{m \times n}$ *and a column vector* $c \in [1, \infty)^n$*. The goal is to find a column vector* $x \in \mathbf{Z}^n$ *such that* $x \geq (0, 0, ..., 0)^T$ *and* $Ax \geq (1, 1, ..., 1)^T$ *in order to minimize* $c^T x$*.*

Theorem 1. *There is a polynomial* q *and a randomized algorithm* R *with the following property. For any covering integer program* (A, c) *in normal form, if* $r = \max_{i,j} A_{i,j}$ *and* L *is the number of bits required to write* A *and* c*, then with probability* $1/2$*, Algorithm* R *outputs a feasible solution* x *in* $q(L, \mathrm{opt}(A, c))$ *time whose solution has cost that is* $O(\mathrm{opt}(A, c)(1 + r\mathrm{Pdim}(A) \log(r\mathrm{opt}(A, c))))$*.*

Proof Sketch: For the sake of brevity, we will consider an algorithm (let's call it R') that makes use of the knowledge of $\mathrm{Pdim}(A)$. It is not hard to see how to remove the need for this knowledge. Algorithm R' is as follows.

- Solve the linear program obtained by relaxing the integrality constraint. Call the solution u.
- Set $Z = \sum_{i=1}^n u_i$, and $p = u/Z$. Note that p can be interpreted as a probability distribution on $\{1, ..., n\}$. Note also that $Z \geq 1/r$, since otherwise all constraints would be violated.
- Let κ be as in Lemma 1 and $\ell = \max\{\lceil 2\kappa \mathrm{Pdim}(A) r Z \ln(rZ) \rceil, \lceil 2Z \rceil\}$. Sample ℓ times at random independently according to p, and, for each j, let x_j be the number of times that j occurs.
- Output $x = (x_1, ..., x_n)$.

Choose an input (A, c) and let $r = \max_{i,j} A_{i,j}$, $\mathrm{opt} = \mathrm{opt}(A, c)$, and $d = \mathrm{Pdim}(A)$. Let $a_1, ..., a_m$ be the rows of A. Since $Au \geq (1, 1, ..., 1)^T$, we have $Ap \geq (1/Z)(1, 1, ..., 1)^T$. Thus, for each i, we have $\mathbf{E}_{j \in p}(A_{i,j}) = \mathbf{E}_{j \in p}(f_{A,i}(j)) \geq 1/Z$.

Since, for each i, incrementing x_j has the effect of increasing $a_i \cdot x$ by $A_{i,j} = f_{A,i}(j)$, applying Lemma 1 with $\epsilon = 1/Z$, with probability at least $3/4$, for all i, $a_i \cdot x \geq \ell/(2Z) \geq 1$. Thus,

$$\mathbf{Pr}(x \text{ is not feasible }) \leq 1/4. \tag{1}$$

We have

$$\mathbf{E}(c^T x) = \ell c^T p \leq \ell \mathrm{opt}/Z \leq \frac{\max\{\lceil 2\kappa d r Z \ln(rZ) \rceil, \lceil 2Z \rceil\} \mathrm{opt}}{Z}.$$

Thus, Markov's inequality implies that

$$\mathbf{Pr}\left(c^T x > \frac{4 \max\{\lceil 2\kappa d (rZ) \ln(rZ) \rceil, \lceil 2Z \rceil\} \mathrm{opt}}{Z}\right) \leq 1/4. \tag{2}$$

Since each $c_i \geq 1$, we have $Z = \sum_{i=1}^n u_i \leq \sum_{i=1}^n c_i u_i \leq \mathrm{opt}$. Combining with (1) and (2) completes the proof. □

4 Mixed Covering Integer Programs

In a *mixed covering integer program*, for natural numbers m and n, column vectors $c \in \mathbf{Q}_+^n$ and $b \in \mathbf{Q}_+^m$, and a matrix $A \in \mathbf{Q}^{m \times n}$, the goal is to choose $x \in \mathbf{Z}_+^m$ to minimize $c^T x$ subject to $Ax \geq b$. Note that to be a mixed covering integer program, the entries of A need not be nonnegative. If $b = (1, 1, ..., 1)^T$ and $c \in [1, \infty)^n$, then we say that the mixed covering integer program is in *normal form*. (This can be seen to be without loss of generality as with covering integer programs.) Note however, that here we cannot assume without loss of generality that the entries of A are at most 1.

Theorem 2. *There is a polynomial q and a randomized algorithm R with the following property. For any mixed covering integer program (A, c) in normal form, if $r = \max_{i,j} |A_{i,j}|$ and L is the number of bits required to write A and c, then with probability $1/2$, Algorithm R outputs a feasible solution x in $q(L, \mathrm{opt}(A, c))$ time whose solution has cost that is $O(\mathrm{Pdim}(A) r^2 \mathrm{opt}(A, c)^2 + \mathrm{opt}(A, c))$.*

Proof Sketch: As in the proof of Theorem 1, we will consider an algorithm R' that "knows" $\mathrm{Pdim}(A)$; the algorithm is the same as in that proof, except k_0 is defined as in Lemma 2, $\ell = \max\{\lceil 4\kappa \mathrm{Pdim}(A) r^2 Z^2 \rceil, \lceil 2Z \rceil\}$. We will borrow notation from that proof.

As before, since $Au \geq (1, 1, ..., 1)^T$ and $p = u/Z$, for each $i \in \{1, ..., m\}$, $\mathbf{E}_{j \in p}(f_{A,i}(j)) \geq 1/Z$. Thus

$$\begin{aligned}
\mathbf{Pr}(x \text{ is not feasible}) &\leq \mathbf{Pr}_{(j_1,\ldots,j_\ell) \in p^\ell}\left(\exists i, \mathbf{E}(f_{A,i}) \geq 1/Z \text{ but } \sum_{t=1}^{\ell} f_{A,i}(j_t) < 1\right) \\
&\leq \mathbf{Pr}_{(j_1,\ldots,j_\ell) \in p^\ell}\left(\exists i, \mathbf{E}(f_{A,i}) \geq 1/Z \text{ but } \sum_{t=1}^{\ell} f_{A,i}(j_t) < \frac{\ell}{2Z}\right) \\
&\leq \mathbf{Pr}_{(j_1,\ldots,j_\ell) \in p^\ell}\left(\exists i, \left|\mathbf{E}(f_{A,i}) - \frac{1}{\ell}\sum_{t=1}^{\ell} f_{A,i}(j_t)\right| > \frac{1}{2Z}\right)
\end{aligned}$$

which is at most $1/4$ by Lemma 2. But

$$\mathbf{E}(c^T x) \leq \ell \mathrm{opt}/Z \leq \frac{\max\{\lceil 4\kappa \mathrm{Pdim}(A) r^2 Z^2 \rceil, \lceil 2Z \rceil\} \mathrm{opt}}{Z}.$$

Applying Markov's inequality and the fact that $Z \leq \mathrm{opt}$ as in the proof of Theorem 1 completes the proof. □

5 Packing Integer Programs

In a packing integer program, for natural numbers n and m, column vectors $c \in \mathbf{Q}_+^n$ and $b \in \mathbf{Q}_+^m$, and a matrix $A \in \mathbf{Q}_+^{m \times n}$, the goal is to choose $x \in \mathbf{Z}_+^m$ to maximize $c^T x$ subject to $Ax \leq b$.

Arguing as for covering, one can assume without loss of generality that entries of A are in $[0,1]$ and $b=(1,1,...,1)^T$. Furthermore, one can also assume in this case that each component of c is positive; here if some $c_j=0$, you might as well set $x_j=0$, and thus, the jth variable can be eliminated. Since again we can scale c so that its least component is 1, we arrive at the following.

Definition 2. *A packing integer program in normal form is given by a matrix $A=[0,1]^{m\times n}$ and a column vector $c\in[1,\infty)^n$. The goal is to find a column vector $x\in\mathbf{Z}^n$ such that $x\geq(0,0,...,0)^T$ and $Ax\leq(1,1,...,1)^T$ in order to maximize c^Tx.*

Theorem 3. *There is a constant $k>0$, a randomized polynomial-time algorithm R and a polynomial q with the following property. For any packing integer program (A,c) in normal form, if B is the least integer such that $\max_{i,j}A_{i,j}\leq 1/B$, L is the number of bits in the representation of A and c, and $d=\mathrm{Pdim}(A)$, with probability $1/2$, Algorithm R outputs a feasible solution x in $q(L,\mathrm{opt}(A,c))$ time whose solution has value that is $\Omega\left(\frac{\mathrm{opt}(A,c)}{(\mathrm{opt}(A,c)/B)^{kd/B}}\right)$.*

Proof Sketch: The fact that the entries of A are at most $1/B$ implies that any x with $\sum_{i=1}x_i\leq B$ is feasible. This, together with the fact that each component of c is at least 1, implies that it is trivial to find a solution of value B. Hence, we can assume without loss of generality that $\frac{\mathrm{opt}}{(\mathrm{opt}/B)^{kd/B}}\geq B$ and therefore, since $\mathrm{opt}\geq B$, that $kd/B\leq 1$.

Again, we will consider an algorithm R' that "knows" $\mathrm{Pdim}(A)$:

- Solve the linear program obtained by relaxing the integrality constraint. Call the solution u.
- Set $Z=\sum_{i=1}^n u_i$, and $p=u/Z$. (Note that $Z\geq B$; otherwise, since the entries of A are at most $1/B$, no constraints would be binding, and u could be improved.)
- Let κ be as in Lemma 3, $d=\mathrm{Pdim}(A)$, $\alpha=(Z/B)^{4\kappa d/B}$, $\ell=\left\lceil\frac{\kappa d(Z/B)\ln(Z/B)}{\alpha\ln(1+\alpha)}\right\rceil$. Sample ℓ times at random independently according to p, and, for each j, let x_j be the number of times that j occurs.
- Output $x=(x_1,...,x_n)$.

Choose an input (A,c) and let $a_1,...,a_m$ be the rows of A. Let B, d, α and ℓ be as in the description of Algorithm R', and let $\mathrm{opt}=\mathrm{opt}(A,c)$.

Suppose $\ell=1$. Again, since the entries in A are at most 1, and the constraints are of the form $a_i\cdot x\leq 1$, then since in this case $\sum_{i=1}^n x_i=1$, x is certainly feasible.

Suppose $\ell>1$. Since $Au\leq(1,1,...,1)^T$, for each i, we have $\mathbf{E}(a_i\cdot x)\leq\ell/Z$. Applying part (c) of Lemma 3 (note that since $Z\geq B$, $\alpha\geq 1$), with probability at least $3/4$, for all i,

$$a_i\cdot x\leq(1+\alpha)\ell/Z=\frac{1+\alpha}{Z}\left\lceil\frac{\kappa d(Z/B)\ln(Z/B)}{\alpha\ln(1+\alpha)}\right\rceil\leq\frac{4\kappa d(Z/B)\ln(Z/B)}{Z\ln(1+\alpha)}\leq 1.$$

Thus, whatever the value of ℓ, we have

$$\mathbf{Pr}(x \text{ is not feasible }) \leq 1/4. \tag{3}$$

Applying Chebyshev's inequality yields

$$\mathbf{Pr}\left(c^T x < \frac{\ell c^T u}{Z} - 2\sqrt{\frac{\ell c^T u}{Z}}\right) \leq 1/4. \tag{4}$$

Substituting the value of ℓ and simplifying, we have

$$\frac{\ell c^T u}{Z} \geq \frac{c^T u}{\left(\frac{1}{B}\sum_{i=1}^{n} u_i\right)^{kd/B}} \geq \frac{c^T u}{(c^T u/B)^{kd/B}} \geq \frac{\text{opt}}{(\text{opt}/B)^{kd/B}},$$

since $c^T u \geq \text{opt}$ and $kd/B \leq 1$. Putting this together with (4) and (3) completes the proof. □

6 Applications

In this section, we give examples of the application of our general results.

6.1 Dominating set and extensions

The *B-domination problem* [NR95,Sri99] is defined as follows: given a graph $G = (V, E)$, place as few facilities as possible on the vertices of G in such a way that each vertex has at least B facilities in its neighborhood. The neighborhood of a vertex is defined to consist of the vertex and all vertices sharing an edge with it.

Define $\mathcal{N}(G)$ to be the set system consisting of all the neighborhoods in G, i.e.

$$\mathcal{N}(G) = \{\{w : \{v, w\} \in E\} \cup \{v\} : v \in V\}.$$

Theorem 4. *For each natural number B, there is a polynomial-time algorithm A for the B-domination problem such that for any graph G with optimal solution* $\text{opt}(G, B)$*, algorithm A outputs a solution of size*

$$O\left(\text{opt}(G, B)\left(1 + \frac{\text{VCdim}(\mathcal{N}(G))}{B}\log\frac{\text{opt}(G, B)}{B}\right)\right).$$

Proof: If x_v is the number of facilities located at vertex v, the problem is to minimize $\sum_{v \in V} x_v$ subject to the constraints, one for each vertex v, that

$$\left(\left(\sum_{w:\{v,w\}\in E} x_w\right) + x_v\right)/B \geq 1$$

(and that the x_v's are nonnegative integers). Since $\mathcal{N}(G) = \text{dual}(\mathcal{N}(G))$, and scaling all members of a set of functions by a common constant factor does not change its pseudo-dimension, applying Theorem 1 completes the proof. □

The following is an example of how this can be applied. Recall that the genus of a graph is, informally, the number of "handles" that need to be added to the plane before the graph can be embedded without any edge crossings.

Theorem 5. *Choose a fixed nonnegative integer k.*

For each natural number B, there is a polynomial-time algorithm A for the B-domination problem such that for any graph G of genus at most k with optimal solution $\text{opt}(G, B)$, *algorithm A outputs a solution of size*

$$O\left(\text{opt}(G,B)\left(1+\frac{\log \text{opt}(G,B)}{B}\right)\right).$$

Proof Sketch: We bound the VC-dimension of $\mathcal{N}(G)$ in terms of the genus of G and apply Theorem 4. Details are omitted from this abstract. □

6.2 Sparse majorities of weak hypotheses

The *minimum majority problem* is to, given an $m \times n$ matrix A with entries in $\{-1, 1\}$, choose $x \in \mathbf{Z}_+^n$ to minimize $\sum_{i=1}^n x_i$ subject to $Ax > 0$. In other words, choose as short a sequence $j_1, ..., j_k$ of columns as possible such that for each row i, a majority of $A_{i,j_1}, ..., A_{i,j_k}$ are 1. The following is an immediate consequence of Theorem 2.

Theorem 6. *There is a randomized polynomial time algorithm for the minimum majority problem that, with probability* $1/2$, *outputs a solution of cost* $O(\text{opt}^2 \text{Pdim}(A))$.

The minimum majority problem is a restatement of an optimization problem motivated by learning applications. Many learning problems can be modeled as that of approximating a $\{0, 1\}$-valued function using examples of its behavior when applied to randomly drawn elements of its domain [Val84,Hau92]; the approximation is sometimes called a *hypothesis*. *Boosting* [Sch90,Fre95,FS97] is a method for combining "weak hypotheses", which are correct on only a slight majority of the input examples, into a "strong hypothesis", which outputs a weighted majority vote of the weak hypotheses. The key idea of the most influential analysis of the ability of the strong hypothesis to generalize to unseen domain elements [SFBL98] is to use the fact that it can be approximated by a majority of a few of the weak hypotheses. This suggests an alternative approach to the design of a learning algorithm: try directly to find hypotheses that explain the data well using majorities of as few as possible of a collection of weak hypotheses. This is captured by the minimum majority problem: the columns correspond to examples, the rows to weak hypotheses, and an entry indicates whether a given weak hypothesis is correct on a given example. The goal is to find a small multiset of weak hypotheses whose majority is correct on all of a collection of examples. This direct optimization might provide improved generalization, but even if not, its output should be easier to interpret, which is an important goal for some applications [Qui99].

6.3 Simple B-matching

The problem of simple B-matching [Lov75] is to, given a family $\mathcal{S}$ of subsets of a finite set X, find a large $\mathcal{T} \subseteq \mathcal{S}$ such that each element of X is contained in at most B of the sets in $\mathcal{T}$.

Theorem 7. *There is a constant k such that for all integers $B \geq 1$ and $d \geq 2$, there is a polynomial time algorithm for the simple B matching problem that, for any input $\mathcal{S}$ such that* $\mathrm{VCdim}(\mathrm{dual}(\mathcal{S})) \leq d$*, outputs a solution of size* $\Omega((\mathrm{opt}(\mathcal{S})/B)^{1-kd/B})$.

Proof: Consider the variant of the simple B matching problem in which multiple copies of sets in $\mathcal{S}$ can be included in the output. This problem can be expressed as a packing integer program in normal form as follows. For each $S \in \mathcal{S}$, include a variable x_S indicating the number of copies of S in the output. Then the goal is to maximize $\sum_{S \in \mathcal{S}} x_S$ subject to the constraints, one for each $x \in X$, that $\left(\sum_{S \in \mathcal{S}: x \in S} x_S\right)/B \leq 1$.

Suppose $\mathrm{opt}(\mathcal{S})$ is the optimal value of the objective function for the original simple B-matching problem. Since the optimal value of the objective function for the multiple-copy variant is at least $\mathrm{opt}(\mathcal{S})$, and since, once again, scaling elements of a set of functions by a common constant factor does not affect its pseudo-dimension, Theorem 3 implies that the value of the solution output by the algorithm described above is $\Omega\left(\frac{\mathrm{opt}(\mathcal{S})}{(\mathrm{opt}(\mathcal{S})/B)^{kd/B}}\right)$. Certainly no more than B copies of any set are included, so if we output one copy of all sets for which $x_S > 0$ we get a solution of size $\Omega\left((\mathrm{opt}(\mathcal{S})/B)^{1-kd/B}\right)$. □

7 Concluding remark

Other generalizations of the VC-dimension to real and integer-valued functions have been proposed, and results similar to Lemmas 1 and 2 proved for them (see [Dud78,Nat89,Vap89,KS94,BCHL92,ABCH97,BLW96,BL98]). It is easy to see how to prove analogues of Theorems 1, 2 and 3 for any of these. In some cases, these may provide easier analyses or stronger guarantees.

Acknowledgement

We thank Aravind Srinivasan and an anonymous referee for their helpful comments about an earlier draft of this paper. We gratefully acknowledge the support of National University of Singapore Academic Research Fund Grant RP–252–000–070–107.

References

[AB99] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations.* Cambridge University Press, 1999.

[ABCH97] N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the Association for Computing Machinery*, 44(4):616–631, 1997.

[Bak94] B.S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the Association for Computing Machinery*, 41:153–180, 1994.

[BCHL92] S. Ben-David, N. Cesa-Bianchi, D. Haussler, and P. M. Long. Characterizations of learnability for classes of $\{0, \ldots, n\}$-valued functions. *Journal of Computer and System Sciences*, 50(1):74–86, 1992.

[BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *JACM*, 36(4):929–965, 1989.

[BG95] H. Bronnimann and M. Goodrich. Almost optimal set covers in finite VC-dimension. *Discrete Comput. Geom.*, 14:263–279, 1995.

[BL98] P. L. Bartlett and P. M. Long. Prediction, learning, uniform convergence, and scale-sensitive dimensions. *Journal of Computer and System Sciences*, 56(2):174–190, 1998.

[BLW96] P. L. Bartlett, P. M. Long, and R. C. Williamson. Fat-shattering and the learnability of real-valued functions. *Journal of Computer and System Sciences*, 52(3):434–452, 1996.

[CK] P. Crescenzi and V. Kann. A compendium of NP optimization problems. See http://www.nada.kth.se/viggo/problemlist/compendium.html.

[Dud78] R. M. Dudley. Central limit theorems for empirical measures. *Annals of Probability*, 6(6):899–929, 1978.

[Fel87] M.R. Fellows. The Robertson-Seymour theorems: A survey of applications. *Contemp. Math.*, 89:1–18, 1987.

[Fre95] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

[FS97] Y. Freund and R. 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.

[Hau92] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

[Hau95] D. Haussler. Sphere packing numbers for subsets of the Boolean n-cube with bounded Vapnik-Chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 69(2):217–232, 1995.

[HMP$^+$93] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46:129–154, 1993.

[KS94] M. J. Kearns and R. E. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48(3):464–497, 1994.

[KT94] P.G. Kolaitis and M.N. Thakur. Logical definability of NP optimization problems. *Information and Computation*, 115(2):321–353, 1994.

[LLS00] Y. Li, P. M. Long, and A. Srinivasan. Improved bounds on the sample complexity of learning. *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 309–318, 2000.

[Lov75] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[Nat89] B. K. Natarajan. On learning sets and functions. *Machine Learning*, 4:67–97, 1989.

[NR95] M. Naor and R. M. Roth. Optimal file sharing in distributed networks. *SIAM J. Comput.*, 24(1):158–183, 1995.

[PA95] J. Pach and P.K. Agarwal. *Combinatorial Geometry*. John Wiley and Sons, 1995.

[Pol84] D. Pollard. *Convergence of Stochastic Processes*. Springer Verlag, 1984.

[Qui99] J.R. Quinlan. Some elements of machine learning. *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 523–524, 1999.

[Rag88] P. Raghavan. Probabilistic construction of deterministic algorithms: Approximating packing integer programs. *Journal of Computer and System Sciences*, 37:130–143, 1988.

[Ram30] F.P. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematics Society*, 30(2):264–286, 1930.

[Roy91] S. Roy. Semantic complexity of relational queries and data independent data partitioning. *Proceedings of the ACM SIGACT-SIGART-SIGMOD Annual Symposium on Principles of Database Systems*, 1991.

[RT87] P. Raghavan and C. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

[SAB93] J. Shawe-Taylor, M. Anthony, and N. Biggs. Bounding the sample size with the Vapnik-Chervonenkis dimension. *Discrete Applied Mathematics*, 42:65–73, 1993.

[Sch90] R. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

[SFBL98] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[Sri99] A. Srinivasan. Improved approximation guarantees for packing and covering integer programs. *SIAM Journal on Computing*, 29:648–670, 1999.

[Sri01] A. Srinivasan. New approaches to covering and packing problems. *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2001.

[Tal94] M. Talagrand. Sharper bounds for Gaussian and empirical processes. *Annals of Probability*, 22:28–76, 1994.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Vap82] V. N. Vapnik. *Estimation of Dependencies based on Empirical Data*. Springer Verlag, 1982.

[Vap89] V. N. Vapnik. Inductive principles of the search for empirical dependences (methods based on weak convergence of probability measures). *Proceedings of the 1989 Workshop on Computational Learning Theory*, 1989.

[VC71] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.