PRINCETON UNIVERSITY F'02 COS 597D: A THEORIST'S TOOLKIT

Lecture 4: The Dimension Method

Lecturer: *Sanjeev Arora* Scribe:*Miroslav Dudík*

The "Dimension Method" is Sanjeev's name for elementary linear algebra arguments. This is all the algebra one needs 80% of the time; only occasionally (one example is number theoretic cryptography) does one need anything more powerful than elementary linear algebra.

1 Basics: Fields and Vector Spaces

We recall some basic linear algebra. A *field* is a set closed under addition, subtraction, multiplication and division by nonzero elements. By addition and multiplication, we mean commutative and associative operations which obey distributive laws. The additive identity is called zero, the multiplicative identity is called unity. Examples of fields are reals $\mathbf{R}$, rationals $\mathbf{Q}$, and integers modulo a prime p denoted by $\mathbf{Z}/p$. We will be mostly concerned with finite fields. The cardinality of a finite field must be a power of prime and all finite fields with the same number of elements are isomorphic. Thus for each power p^k there is essentially one field $\mathbb{F}$ with $|\mathbb{F}| = p^k$. We shall denote this field by $\mathrm{GF}(p^k)$.

A *vector space* V over a field $\mathbb{F}$ is an additive group closed under (left) multiplication by elements of $\mathbb{F}$. We require that this multiplication be distributive with respect to addition in both V and $\mathbb{F}$, and associative with respect to multiplication in $\mathbb{F}$.

Vectors $\mathbf{v_1}, \ldots, \mathbf{v_k}$ are said to be *linearly independent* if $\sum_{i=1}^{k} \alpha_i \mathbf{v_i} = \mathbf{0}$ implies that $\alpha_1 = \alpha_2 = \cdots = \alpha_k = 0$. A maximal set of vectors $\{\mathbf{v}_i\}_{i \in I}$ whose every finite subset is linearly independent is called a *basis* of V; all such sets have the same cardinality, called the *dimension* of V (denoted $\dim V$). If V has a finite dimension k and $\{\mathbf{v}_i\}_{i=1}^{k}$ is a basis then every vector $v \in V$ can be uniquely represented as

$$\mathbf{v} = \sum_{i=1}^{k} \alpha_i \mathbf{v}_i,$$

where $\alpha_i \in \mathbb{F}$. Thus all finite-dimensional vector spaces are isomorphic to $\mathbb{F}^k$. If $\mathbb{F}$ is finite then $|V| = |\mathbb{F}|^k$. An example of a vector space over a finite field is the field $\mathrm{GF}(p^k)$ when viewed as a vector space over $\mathrm{GF}(p)$.

Let $\mathbf{A}_{m \times n} = \{a_{ij}\}$ be a matrix over a field $\mathbb{F}$. Rank of $\mathbf{A}$, denoted by $\operatorname{rank} \mathbf{A}$, is the maximum number of linearly independent rows in $\mathbf{A}$. It is equal to the maximum number of linearly independent columns. Hence $\operatorname{rank} \mathbf{A} = \operatorname{rank} \mathbf{A}^T$.

Let $\mathbf{M} = \{m_{ij}\}$ be an n by n matrix. The determinant of $\mathbf{M}$ is defined as follows:

$$\det \mathbf{M} = \sum_{\sigma \in S_n} (-1)^{\pi(\sigma)} \prod_{i=1}^{n} m_{i\sigma(i)},$$

where S_n is the group of permutations over $[n]$, and $\pi(\sigma)$ is the parity of the permutation σ. The matrix $\mathbf{M}_{n \times n}$ has rank n if and only if $\det \mathbf{M} \neq 0$. We will use this fact to prove the following result, which is our first example of the Dimension Method.

THEOREM 1
Let $\mathbf{M}_{n\times n}$ *be a random matrix over GF*(2)*. Then* $\Pr[\det \mathbf{M} \neq 0] \geq 1/4$.

PROOF: Denote the columns of $\mathbf{M}$ by $\mathbf{M}_i$, where $i = 1, 2, \ldots, n$. It suffices to bound the probability that these columns are linearly independent:

$$\begin{aligned}
&\Pr[\mathbf{M}_1, \ldots, \mathbf{M}_n \text{ linearly independent}] \\
&\quad = \prod_{i=1}^{n} \Pr[\mathbf{M}_1, \ldots, \mathbf{M}_i \text{ linearly independent} \mid \mathbf{M}_1, \ldots, \mathbf{M}_{i-1} \text{ linearly independent}] \\
&\quad = \prod_{i=1}^{n} (1 - \Pr[\mathbf{M}_i \in \operatorname{span}(\mathbf{M}_1, \ldots, \mathbf{M}_{i-1}) \mid \mathbf{M}_1, \ldots, \mathbf{M}_{i-1} \text{ linearly independent}].
\end{aligned}$$

Now, if $\mathbf{M}_1, \ldots, \mathbf{M}_{i-1}$ are independent then their span is of dimension $i-1$ and hence it contains 2^{i-1} vectors. The column $\mathbf{M}_i$ is picked uniformly at random from the space of 2^n vectors, independently of $\mathbf{M}_1, \ldots, \mathbf{M}_{i-1}$. Thus the probability that it will lie in their span is $2^{i-1}/2^n$.

$$\begin{aligned}
&= \prod_{i=1}^{n} (1 - 2^{i-1-n}) \geq \prod_{i=1}^{n} \exp\{-2^{i-1-n} \cdot 2 \ln 2\} \\
&\geq \exp\{-2 \ln 2 \sum_{i=1}^{\infty} 2^{i-1-n}\} = \exp\{-2 \ln 2\} = 1/4.
\end{aligned}$$

□

2 Systems of Linear Equations

The system of m linear equations in n unknowns over a field $\mathbb{F}$ can be represented by a matrix $\mathbf{A}_{m\times n}$ and a vector $\mathbf{b}_{m\times 1}$ as

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \qquad (*)$$

where $\mathbf{x}_{n\times 1}$ is the vector of unknowns.

PROPOSITION 2

1. *The system* $(*)$ *is feasible if and only if* $b \in \operatorname{span}(\mathbf{A}_1, \ldots, \mathbf{A}_n)$*, which occurs if and only if* $\operatorname{rank}(\mathbf{A}|\mathbf{b}) = \operatorname{rank} \mathbf{A}$*. (Here* $\mathbf{A}|\mathbf{b}$ *is the matrix whose last column is* $\mathbf{b}$ *and the other columns are from* $\mathbf{A}$*.)*

2. *Suppose* $\mathbb{F}$ *is finite. If* $\operatorname{rank} \mathbf{A} = k$ *then the system* $(*)$ *has either 0 solutions (if infeasible) or* $\mathbb{F}^{n-k}$ *solutions (if feasible). In particular, if* $n = k$ *then the solution is unique if it exists.*

3. *If* $\mathbf{b} = \mathbf{0}$ *then a nontrivial solution exists if and only if* $\operatorname{rank} \mathbf{A} \leq n - 1$*. In particular, if* $n > m$ *then nontrivial solutions always exist.*

EXAMPLE 1 Suppose M is a random matrix over GF(2) and $\mathbf{b}$ is a random $n \times 1$ vector. What is the probability that the system $Mx = \vec{b}$ has a unique solution? By Theorem 1 it is at least 1/4.

THEOREM 3
A nonzero polynomial of degree d has at most d distinct roots.

PROOF: Suppose $p(x) = \sum_{i=0}^{d} c_i x^i$ has $d+1$ distinct roots $\alpha_1, \ldots, \alpha_{d+1}$ in some field $\mathbb{F}$. Then

$$\sum_{i=0}^{d} \alpha_j^i \cdot c_i = p(\alpha_j) = 0,$$

for $j = 1, \ldots, d+1$. This means that the system $\mathbf{Ay} = \mathbf{0}$ with

$$\mathbf{A} = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \ldots & \alpha_1^d \\ 1 & \alpha_2 & \alpha_2^2 & \ldots & \alpha_2^d \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & \alpha_{d+1} & \alpha_{d+1}^2 & \ldots & \alpha_{d+1}^d \end{pmatrix}$$

has a solution $\mathbf{y} = \mathbf{c}$. The matrix $\mathbf{A}$ is a *Vandermonde* matrix, hence

$$\det \mathbf{A} = \prod_{i>j} (\alpha_i - \alpha_j),$$

which is nonzero for distinct α_i. Hence $\operatorname{rank} \mathbf{A} = d+1$. The system $\mathbf{Ay} = \mathbf{0}$ has therefore only a trivial solution — a contradiction to $\mathbf{c} \neq \mathbf{0}$. □

THEOREM 4
For any set of pairs $(a_1, b_1), \ldots, (a_{d+1}, b_{d+1})$ there exists a unique polynomial $g(x)$ of degree at most d such that $g(a_i) = b_i$ for all $i = 1, 2, \ldots, d+1$.

PROOF: The requirements are satisfied by *Lagrange Interpolating Polynomial*:

$$\sum_{i=1}^{d+1} b_i \cdot \frac{\prod_{j \neq i}(x - a_j)}{\prod_{j \neq i}(a_i - a_j)}.$$

If two polynomials $g_1(x), g_2(x)$ satisfy the requirements then their difference $p(x) = g_1(x) - g_2(x)$ is of degree at most d, and is zero for $x = a_1, \ldots, a_{d+1}$. Thus, from the previous theorem, polynomial $p(x)$ must be zero and polynomials $g_1(x), g_2(x)$ identical. □

3 Dispersal of Information Using Polynomials

Polynomials are often useful in situations where information needs to be dispersed in an error-tolerant way, so that it can be reconstructed even if it is partially corrupted or destroyed.

Suppose we want to encode a message into a finite number of packets to be transmitted through a faulty network. This network can drop up to 1/2 of packets, but it does not corrupt the contents of the remaining packets. To achieve a successful transmission, we can use polynomial interpolation:

Encoding. Without loss of generality assume the message is a d+1-tuple $c_0, c_1, \ldots, c_d \in \mathbb{F}$, where $|\mathbb{F}| > 2d$. Take $2d+1$ distinct points $\alpha_1, \ldots, \alpha_{2d+1} \in \mathbb{F}$ and determine values of the polynomial $p(x) = \sum_{i=0}^{d} c_i x^i$ at α_i. Send packets $(\alpha_1, p(\alpha_1))$,...,$(\alpha_{2d+1}, p(\alpha_{2d+1}))$.

Decoding. Packets describe the polynomial p with sufficient redundancy. Even when d packets are dropped, the polynomial p and hence the original message is uniquely determined by the remaining $d+1$ pairs.

Now suppose that the network can corrupt up to 1/4th of the packets. We will use a strategy developed by Berlekamp and Welch in 1985. In order to transmit a message described by a polynomial $p(x)$ of degree d, we will send $20d$ pairs $(\alpha_i, p(\alpha_i))$. Let the received pairs be (α'_i, β'_i) (for missing packets, we can set $\alpha'_i = \beta'_i = 0$). A pair (α'_i, β'_i) will be considered corrupted if $p(\alpha'_i) \neq \beta'_i$. Then there exists a *nonzero* polynomial $e(x)$ of degree at most $5d$, which is zero at all corrupted values α'_i — this is called an *error locator polynomial*.

Lemma 5
There exist nonzero polynomials $e(x)$ and $c(x)$ such that

$$\deg e \leq 5d$$
$$\deg c \leq 6d$$

and $c(\alpha'_i) = \beta'_i e(\alpha'_i)$ for $i = 1, 2, \ldots, 20d$.

Proof: Taking the error locator polynomial $e(x)$ and $c(x) = p(x)e(x)$ we obtain

$$c(\alpha'_i) = p(\alpha'_i)e(\alpha'_i) = \begin{cases} \beta'_i e(\alpha'_i) & \text{if pair } (\alpha'_i, \beta'_i) \text{ is not corrupted} \\ p(\alpha'_i) \cdot 0 = 0 = \beta'_i \cdot 0 & \text{if pair } (\alpha'_i, \beta'_i) \text{ is corrupted.} \end{cases}$$

□

Corollary 6
Polynomials $e(x)$ and $c(x)$ that satisfy conditions of previous lemma can be found in time polynomial in d.

Proof: Equations $c(\alpha'_i) = \beta'_i e(\alpha'_i)$, where coefficients of c and e are unknown, form a system of $20d$ linear equations in $11d+2$ unknowns. The lemma guarantees its feasibility. We can solve for coefficients by Gaussian elimination. □

Theorem 7 (Berlekamp-Welch)
If $c(x), e(x)$ satisfy the conditions of the lemma then $e(x)|c(x)$ and $p(x) = c(x)/e(x)$.

Proof: Consider the polynomial $c(x) - p(x)e(x)$. It has degree at most $6d$, but it has at least $15d$ roots because it is zero on all noncorrupted α'_i's. Therefore, $c(x) - p(x)e(x) \equiv 0$ and $c(x) \equiv p(x)e(x)$. □

Remark 1 This strategy will work whenever a fixed δ-fraction of packets is corrupted, where $\delta < 1/2$. Somebody asked if a scheme is known that recovers the polynomial even if more than 1/2 the packets are corrupted. The answer is Yes, using Sudan's list decoding algorithm. See the homework.

4 Hashing: An introduction

Most schemes for Hashing also rely on a simple dimension argument.

Suppose we want to store n numbers from the set $1, 2, \ldots, q$ with a fast look-up. We will use an array of size p and each element insert into a *bucket* indexed by the *hash function*. Each bucket contains a chained list of elements with the same value of hash function. During a look-up, it suffices to examine contents of a single bucket. If we can guarantee that the number of elements stored in a bucket (the *bucket size*) is small, the operation will be fast.

We will assume that q and p are prime, $p \approx 2n$ and choose a hash function h at random. We pick $a, b \in \mathrm{GF}(q)$ at random and define h as $x \to (ax + b \mod q) \mod p$. The probability of *collisions*, i.e. events when $h(x) = h(y)$ for $x \neq y$, should be low. We might for example require that the family of hash functions be *2-universal*:

$$(\forall x \neq y) \Pr_h[h(x) = h(y)] \leq \frac{2}{p}.$$

It is often possible to prove a stronger statement:

$$(\forall x \neq y)(\forall u, v) \Pr_h[h(x) = u, h(y) = v] = \frac{1}{p^2}.$$

Families satisfying this condition are called *pairwise independent*.

Example 2 Consider a hash function $h : x \mapsto ax+b \mod p$, where a, b are picked randomly from $\mathrm{GF}(p)$. For fixed $x, y, u, v \in \mathrm{GF}(p)$, where $x \neq y$, the system

$$\begin{aligned} ax + b &= u \\ ay + b &= v \end{aligned}$$

has a single solution $a, b \in \mathrm{GF}(p)$. Hence the probability that $h(x) = u, h(y) = v$ is $1/p^2$.

Example 3 *Element Distinctiveness Problem.* We want to determine if there are two identical numbers in a given sequence. We can hash all elements and then examine each bucket separately. We could do it by sorting elements in every bucket, but for simplicity assume that we examine every pair in a given bucket. If the number of buckets is $O(n)$ and the expected number of pairs in a bucket is $O(1)$ then the expected runtime will be $O(n)$.

Suppose we use a hash function $h : \mathcal{X} \to \mathcal{U}$, $|\mathcal{U}| = p \approx 2n$, picked at random from a pairwise independent family. Fix a bucket u and consider random variables

$$X_x = \begin{cases} 1 & \text{if } h(x) = u, \\ 0 & \text{otherwise,} \end{cases}$$

where x is an element of the sequence. By pairwise independence, choosing arbitrary $y \in \mathcal{X}$ such that $y \neq x$, we obtain

$$\Pr[h(x) = u] = \sum_{v \in \mathcal{U}} \Pr[h(x) = u, h(y) = v] = 1/p.$$

The size of bucket u is $S = \sum_x X_x$. Calculate the expectation of S^2:

$$\begin{aligned}\mathbf{E}[S^2] &= \sum_{x,y} \mathbf{E}[X_x X_y] = \sum_x \mathbf{E}[X_x^2] + \sum_{x \neq y} \mathbf{E}[X_x X_y] \\ &= \sum_x \Pr[h(x) = u] + \sum_{x \neq y} \Pr[h(x) = u, h(y) = u] \\ &= n/p + n(n-1)/p^2 \approx 1/2 + 1/4 = O(1).\end{aligned}$$

Since the number of pairs in a bucket is $O(S^2)$, we obtain by linearity of expectation that the expected runtime is

$$O(\sum_u \mathbf{E}[S_u^2]) = O(n).$$

(*Aside:* The element distinctness problem is impossible to solve (even using randomness) in linear time in the comparison model, where the algorithm is only allowed to compare two numbers at every step.

5 Pairwise and k-wise Independent Sampling

Consider a randomized algorithm that uses n random bits and gives a Yes/No answer. Suppose we know that one of the answer happens with probability at least $2/3$ but we do not know which. We can determine that answer with high probability by running the algorithm m times with independent random bits and taking the majority answer; by the Chernoff bounds the error in this estimation will by $\exp(-m)$. Can we do this estimation using fewer than mn random bits? Intuitively speaking, the algorithm converts n random bits into a single random bit (Yes/No) so it has thrown away a lot of randomness. Can we perhaps "reuse" some of it? Later in the course we will see some powerful techniques to do so; here we use more elementary ideas. Here we see a technique that uses $2n$ random bits and its error probability is $1/m$. (We need $m < 2^n$.) The idea is to use random strings that are pairwise independent and use Chebyshev's inequality.

A sequence of random variables $z_1, z_2, z_3, \ldots$ is *pairwise independent* if every pair is independent.

We can construct a sequence of m pairwise independent strings $\{z_i\}, z_i \in \mathrm{GF}(q)$, $q \approx 2^n$ using $2 \log q$ random bits. Let $\{x_i\}, x_i \in \mathrm{GF}(q)$ be any fixed sequence. Pick $a, b \in \mathrm{GF}(q)$ at random and set $z_i = ax_i + b$. Running the algorithm on $z_1, \ldots, z_m$ will guarantee that answers are pairwise independent.

Analogously, we can construct k-wise independent sequences by picking $a_0, \ldots, a_{k-1}$ at random and applying the map $x \mapsto \sum_{j=0}^{k-1} a_j x^j$ to an arbitrary sequence $\{x_i\}, x_i \in \mathrm{GF}(q)$. Chebyshev inequality generalizes to higher moments:

$$\Pr\left\{ \left|X - \mathbf{E}[X]\right| > \gamma \left(\mathbf{E}\left[|X - \mathbf{E}[X]|^k\right] \right)^{1/k} \right\} < \gamma^{-k}.$$

This uses $k \log q$ random bits but the error in the estimation goes down as $1/m^k$.

EXAMPLE 4 *Secret Sharing (A. Shamir,* How to share a secret*, Comm. ACM 1979).* We want to design a scheme for sharing a secret a_0 among m people so that $k+1$ people can recover the secret, but k or fewer people cannot.

If $a_0, \ldots, a_k$ are picked randomly and person i receives the pair $(\alpha_i, p(\alpha_i))$ where $p(x) = \sum a_i x^i$ then any set of k people will receive a random k-tuple of strings, whereas $k+1$ people will be able to recover the polynomial $p(x)$ by interpolation.