COS521, Fall 2024

Spectral Algorithms - Iterative Methods

In this lecture we will see how studying the spectrum of a matrix can help us design algorithms. In particular we will study iterative methods, which approximate the solution to certain problems by repeatedly refining a solution. We will show how these methods behave nicely, by proving guarantees on their convergence based on the spectrum of some matrix. The two problems we will focus on are computing eigenvalues of a matrix and solving linear systems of equations. [1]

Essential Background in Matrix and Spectral Theory

We will start by recalling some important background definitions and results from linear algebra.

Definition 1. A matrix M is positive semi-definite if $x^T M x \geq 0$ for all vectors $x \in \mathbb{R}^n$. For PSD matrices, all eigenvalues are non-negative.

For any positive integer k, the eigenvalues of M^k are the k-th powers of the eigenvalues of M. Formally, if λ is an eigenvalue of M with corresponding eigenvector v, then:

$$M^k v = (M^{k-1} M) v = M^{k-1}(\lambda v) = \lambda M^{k-1} v = \cdots = \lambda^k v$$

Thus, λ^k is an eigenvalue of M^k with the same eigenvector v.

Spectral Theorem for Symmetric Matrices

Theorem (Spectral Theorem)**.** Let M be a n by n real symmetric matrix M. Then, there exist n real numbers $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n$ and n unit vectors $v_1, v_2, \ldots, v_n$ that form an eigenbasis of M. In other words, $M v_i = \lambda_i v_i$ and all the v_i are mutually orthogonal.

[1] These notes are based on notes from `https://lucatrevisan.github.io/books/expanders-2016.pdf` and from the book `http://cs-www.cs.yale.edu/homes/spielman/sagt/sagt.pdf`.

This allows the matrix M to be diagonalized as:

$$M = V\Lambda V^T$$

where V is an orthogonal matrix of eigenvectors, and Λ is a diagonal matrix containing the eigenvalues of M.

Rayleigh Quotient

Definition 2 (Rayleigh Quotient)**.** Given a vector $x \in \mathbb{R}^n$ and a matrix M, the Rayleigh quotient of x is defined as:

$$\frac{x^T M x}{x^T x}$$

For $Mv_i = \lambda v_i$:

$$\frac{v_i^T M v_i}{v_i^T v_i} = \frac{v_i^T \lambda_i v_i}{v_i^T v_i} = \lambda_i.$$

So the Rayleigh quotient of an eigenvector gives us its eigenvalue. But more generally, we can use any vector to find bounds on eigenvalues of a matrix through its Rayleigh quotion. This is formalized by the following well-known theorem, which gives us a variational characterization of eigenvalues:

Theorem (Courant-Fischer Theorem)**.** Let M be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Then,

$$\lambda_k = \max_{S \subseteq \mathbb{R}^n,\ \dim(S)=k} \ \min_{x \in S,\ x \neq 0} \frac{x^T M x}{x^T x} = \min_{T \subseteq \mathbb{R}^n,\ \dim(T)=n-k+1} \ \max_{x \in T,\ x \neq 0} \frac{x^T M x}{x^T x},$$

where the maximization and minimization are over subspaces S and T of $\mathbb{R}^n$.

The following is a special case of the above.

Corollary. Let M be a symmetric matrix with maximum eigenvalue λ_1. Then, for any vector $v \in \mathbb{R}^n$:

$$\frac{v^T M v}{v^T v} \leq \lambda_1.$$

(Operator) Norm of a Matrix

Definition 3 (Operator norm)**.** Let A be an n by n matrix. The *operator norm* of A, denoted by $\|A\|$, is defined as:

$$\|A\| = \sup_{\|x\| \leq 1} \|Ax\|$$

where $\|x\|$ denotes the Euclidean norm of x.

Equivalent Definition: The operator norm can also be expressed in terms of the supremum over all non-zero vectors:

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

Special Case (Spectral Norm): When A is a symmetric matrix, the operator norm induced by the Euclidean norm is equal to the largest absolute eigenvalue of A. Specifically, if A is symmetric and has eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, then:

$$\|A\| = |\lambda_1|$$

Some properties:

- $\|cA\| = |c| \cdot \|A\|$ for any scalar c.
- $\|AB\| \leq \|A\| \cdot \|B\|$.
- $\|Av\| \leq \|A\| \cdot \|v\|$ for any vector $v \in \mathbb{R}^n$.

Approximating Eigenvalues through Iterative Methods

Suppose we are given an n by n PSD matrix M and we wish to compute its largest eigenvalue. Note that since this matrix is PSD, it is a symmetric matrix and all of it's eigenvalues are real and nonnegative. Using the same notation as before, assume $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ are the n eigenvalues of M. Then, our goal is efficiently compute λ_1.

A random Gaussian vector $x \in \mathbb{R}^n$ is a vector such that each entry is an independently chosen $\mathcal{N}(0,1)$ random variable. Now, here is a simple algorithm (also known as the *Power Method*) that approximates the value of λ_1:

```
Input: A PSD matrix $M$.
Pick a random Gaussian vector $x_0 \in \mathbb{R}^n$.
for $i = 1 \to k$ do
    $x_i \leftarrow Mx_{i-1}$
end for
return $\frac{x_k^T M x_k}{x_k^T x_k}$
```

So the algorithm first picks a random Gaussian vector, then it repeatedly applies the matrix M and it outputs the Rayleigh quotient of the resulting vector.

Note that the runtime of this algorithm is $\mathcal{O}\left(k \cdot \text{nnz}(M)\right)$, where $\text{nnz}(N)$ denotes the number of nonzero entries of M (since each iteration corresponds to a matrix-vector product). So if M is a dense matrix this is $\mathcal{O}\left(kn^2\right)$, but if M is sparse, this is much better. But now the question is: how big does k have to be in order for this to be a good approximation? The following theorem provides the answer:

Theorem 1. Let M be an n by n PSD matrix. Then, for every positive integer k and $\epsilon > 0$, with constant probability

$$\frac{x_k^T M x_k}{x_k^T x_k} \geq \frac{\lambda_1(1-\epsilon)}{1 + \Omega(1)n(1-\epsilon)^{2k}}.$$

So, if we pick $k = \mathcal{O}\left(\frac{\log n}{\epsilon}\right)$, the above is $\lambda_1(1 - \mathcal{O}(\epsilon))$, i.e. an ϵ-approximation of λ_1.

The theorem will follow by combining the two following lemmas.

Lemma 2. Let M be an n by n PSD matrix. Then, for every positive integer k, $\epsilon > 0$, and $x \in \mathbb{R}^n$, if we let $y = M^k x$ we have

$$\frac{y^T M y}{y^T y} \geq \frac{\lambda_1(1-\epsilon)}{1 + \frac{\|x\|^2}{\langle x, v_1\rangle^2}(1-\epsilon)^{2k-1}},$$

where v_1 is the eigenvector of eigenvalue λ_1 of M, i.e. $Mv_1 = \lambda_1 v_1$.

Proof. Our goal will be to prove a lower bound to $y^T M y$ and an upper bound to $y^T y$, the combination of which shall give us the lemma.

Suppose that $v_1, v_2, \ldots, v_n$ are the n eigenvectors of M corresponding respectively to $\lambda_1, \lambda_2, \ldots, \lambda_n$, so $Mv_i = \lambda_i v_i$. Let's write x in the eigenbasis of M

$$x = \sum_i \langle x, v_i\rangle v_i.$$

Our proof strategy will involve dividing the eigenvalues of M into two groups: $\lambda_1, \ldots, \lambda_\ell$ is the group of the ℓ eigenvalues that are at least $\lambda_1(1-\epsilon)$, and $\lambda_{\ell+1}, \ldots \lambda_n$ is the group of eigenvalues that are less than $\lambda_1(1-\epsilon)$.

So let's us first lower bound $y^T M y$. Observe that $y^T M y = x^T (M^k)^T M M^k x = x^T M^{2k+1} x$, since M is symmetric. Let's analyze $x^T M^{2k+1} x$ using the eigenbasis of M:

$$\begin{aligned}
x^T M^{2k+1} x &= \left(\sum_i \langle x, v_i\rangle v_i\right)^T M^{2k+1} \sum_i \langle x, v_i\rangle v_i \\
&= \left(\sum_i \langle x, v_i\rangle v_i\right)^T \sum_i \langle x, v_i\rangle \lambda_i^{2k+1} v_i \\
&= \sum_i \langle x, v_i\rangle^2 \lambda_i^{2k+1} \\
&\geq \sum_i^{\ell} \langle x, v_i\rangle^2 \lambda_i^{2k+1} \\
&\geq \lambda_1(1-\epsilon) \sum_i^{\ell} \langle x, v_i\rangle^2 \lambda_i^{2k}.
\end{aligned}$$

And now we need to upper bound $y^T y$. By a similar observation as above, $y^T y = x^T M^{2k} x$, so:

$$\begin{aligned}
x^T M^{2k} x &= \sum_i \langle x, v_i \rangle^2 \lambda_i^{2k} \\
&= \sum_{i=1}^{\ell} \langle x, v_i \rangle^2 \lambda_i^{2k} + \sum_{i=\ell+1}^{n} \langle x, v_i \rangle^2 \lambda_i^{2k} \\
&\leq \sum_{i=1}^{\ell} \langle x, v_i \rangle^2 \lambda_i^{2k} + (\lambda_1(1-\epsilon))^{2k} \sum_{i=\ell+1}^{n} \langle x, v_i \rangle^2 \\
&\leq \sum_{i=1}^{\ell} \langle x, v_i \rangle^2 \lambda_i^{2k} + \lambda_1^{2k}(1-\epsilon)^{2k} \|x\|^2 .
\end{aligned}$$

The last inequality comes from the fact that $\|x\|^2 = \sum_i \langle x, v_i \rangle^2$, for any basis v. Combining both bounds we obtain:

$$\begin{aligned}
\frac{y^T M y}{y^T y} &\geq \frac{\lambda_1(1-\epsilon) \sum_i^{\ell} \langle x, v_i \rangle^2 \lambda_i^{2k}}{\sum_{i=1}^{\ell} \langle x, v_i \rangle^2 \lambda_i^{2k} + \lambda_1^{2k}(1-\epsilon)^{2k} \|x\|^2} \\
&= \frac{\lambda_1(1-\epsilon)}{1 + \frac{\lambda_1^{2k}(1-\epsilon)^{2k}\|x\|^2}{\sum_{i=1}^{\ell} \langle x, v_i \rangle^2 \lambda_i^{2k}}} \\
&\geq \frac{\lambda_1(1-\epsilon)}{1 + \frac{\lambda_1^{2k}(1-\epsilon)^{2k}\|x\|^2}{\langle x, v_1 \rangle^2 \lambda_1^{2k}}} \\
&= \frac{\lambda_1(1-\epsilon)}{1 + \frac{\|x\|^2}{\langle x, v_1 \rangle^2}(1-\epsilon)^{2k}}.
\end{aligned}$$

This concludes the proof. □

Remark 3. The inequalities we used above are only true because M is PSD. In particular, note that $\sum_i \langle x, v_i \rangle^2 \lambda_i^{2k+1} \geq \sum_i^{\ell} \langle x, v_i \rangle^2 \lambda_i^{2k+1}$ isn't necessarily true if some of the eigenvalues are negative.

It should now be clear that if we lower bound $\frac{\|x\|}{\langle x, v_1 \rangle^2}$ we obtain the theorem we want, so let's do that.

Lemma 4. Let $x \in \mathbb{R}^n$ be a random Gaussian vector and v be any unit vector (so $\|v\| = 1$). Then with constant probability we have:

$$\frac{\|x\|^2}{\langle x, v \rangle^2} \leq \Omega(n)$$

Proof. We will proceed by providing an upper bound to $\|x\|^2$ and a lower bound to $\langle x, v \rangle^2$.

Let's first consider $\|x\|^2$. Observe that $\|x\|^2 = \sum_i g_i^2$, where g_i are independent Gaussian random variables. We claim that the probability that $\|x\|^2$ is greater than $2n$ is exponentially small. We are not going to actually prove this, but this follows by a type of Chernoff bound. Note that the standard Chernoff bound only works for bounded random variables, which isn't the case here.

Finally, we analyze $\langle x, v\rangle^2$. Observe that the inner product is a linear combination of independent standard Gaussians, so it is itself a Gaussian. In fact, a calculation of its mean and variance shows that it is distributed as a standard Gaussian. Thus, with constant probability it's at least any constant (say 2). □

Note that as long as we can show a lower bound on $\frac{\|x\|^2}{\langle x,v\rangle^2}$, then any distribution on x works. For instance, another good distribution to consider is $x \sim \{-1, 1\}^n$, i.e. each coordinate has an equal probability of being -1 or 1.

Remark 5. If we want to find any other eigenvalues/eigenvectors, we can repeat the same method to find an orthogonal vector orthogonal. This works because of the Courant-Fischer theorem.

More formally, say $x^{(1)}$ is the result of applying the power method an adequate number of times. Pick some random Gaussian vector $x \in \mathbb{R}^n$ and make it orthogonal to $x^{(1)}$ by setting $x_0 = x - x^{(1)} \cdot \langle x, x^{(1)} \rangle$. Now, repeat the power method starting from x_0. After an adequate number of iterations the resulting vector would have Rayleigh quotient close to λ_2.

Remark 6. One downside of the method described here is that it's not useful in practice. The main reason is that $M^k x$ grows exponentially fast, and so it's unfeasible to represent the value of x_k for large k. To fix this, we can normalize the value after each step of the iteration, namely we can set $x'_k = Mx_{k-1}$ and then $x_k = x'_k / \|x'_k\|$. We can also show this process converges, but the convergence rate is a bit different, it will depend on the ratio λ_1/λ_2.

Solving Linear Equations through Iterative Methods

We now turn to another fundamental problem on matrices: solving systems of linear equations. So suppose we are given a matrix A and a vector b, and we wish to find x such that $Ax = b$. For simplicity, let's assume A is an n by n PSD matrix, and so both x and b are vectors in $\mathbb{R}^n$ (it will be clear why we are making this assumption soon). We know how to find x in $\mathcal{O}(n^3)$ time using Gaussian elimination, but what if we want a more efficient method? We will describe a simple iterative method known as *First-Order Richardson Iteration* that does so.

Before we describe the algorithm, we will make a simple observation. Consider any parameter α, then $\alpha Ax = \alpha b$ implies $x = (I - \alpha A)x + \alpha b$, where I is the identity matrix. From this, we define the following process:

```
Input: A PSD matrix $M$.
Let $x_0 = 0$, the 0 vector.
for $i = 1 \to k$ do
    $x_i \leftarrow (I - \alpha A)x_{i-1} + \alpha b$
end for
return $x_k$
```

Note that the runtime of this algorithm is $\mathcal{O}(k \cdot (n + \text{nnz}(A)))$. Like in the power method example, we are interested in analyzing the convergence of this method.The following theorem provides the answer:

Theorem 7. Let A be an n by n PSD matrix, and $b \in \mathbb{R}^n$. Let α be such that $\|I - \alpha A\| < 1$. Then, for every positive integer k and $\epsilon > 0$,

$$\frac{\|x - x_k\|}{\|x\|} \le \exp\left(-\frac{2\lambda_n k}{\lambda_1 + \lambda_n}\right),$$

where x is the solution to $Ax = b$, and λ_i are the eigenvalues of A.

Note that if we want an ϵ-approximation of the solution (in terms of the above measure of convergence), we then need to run this process for $k = \left(\frac{\lambda_1}{2\lambda_n} + \frac{1}{2}\right)\ln(1/\epsilon)$. Note that the quantity $\frac{\lambda_1}{\lambda_n}$ is often known as the *condition number*. It is an important quantity associated to a matrix and it is often relevant to the convergence of iterative methods.

Proof. $I - \alpha A$ is symmetric, and so its norm is the maximum absolute value of its eigenvalues. The eigenvalues of $I - \alpha A$ are $1 - \alpha\lambda_i$, and the $\|I - \alpha A\|$ is $\max_i |1 - \alpha\lambda_i| = \max(1 - \alpha\lambda_1, 1 - \alpha\lambda_n)$. This is minimized by taking

$$\alpha = \frac{2}{\lambda_n + \lambda_1},$$

in which case the smallest and largest eigenvalues of $I - \alpha A$ become

$$\pm\frac{\lambda_1 - \lambda_n}{\lambda_n + \lambda_1},$$

and the norm of $I - \alpha A$ becomes

$$1 - \frac{2\lambda_n}{\lambda_n + \lambda_1}.$$

To show that x_k converges to the solution, x, consider the difference $x - x_k$. We have

$$x - x_k = ((I - \alpha A)x + \alpha b) - ((I - \alpha A)x_{k-1} + \alpha b) = (I - \alpha A)(x - x_{k-1}).$$

Thus, by repeatedly applying the above, we get

$$x - x_k = (I - \alpha A)^k (x - x_0) = (I - \alpha A)^k x.$$

and finally we obtain

$$\begin{aligned}
\|x - x_k\| &= \left\|(I - \alpha A)^k x\right\| \\
&\le \left\|(I - \alpha A)^k\right\| \|x\| \\
&\le \left(1 - \frac{2\lambda_n}{\lambda_n + \lambda_1}\right)^k \|x\| \\
&\le \exp\left(-2\lambda_n k/(\lambda_n + \lambda_1)\right) \|x\|.
\end{aligned}$$

□

Remark 8. One interesting way of interpreting the above iterative method is through the lens of gradient descent. Let's write our linear system as $A^{1/2}x = A^{-1/2}b$. Now, consider the objective function $\frac{1}{2}\left\|A^{1/2}x - A^{-1/2}b\right\|^2$ and let's try to minimize it. If we take the gradient of this function, we get $Ax - b$, so one step of gradient descent with learning rate α is $x_{k+1} = x_k - \alpha(Ax_k - b)$, which is exactly our iteration step.