Power Iteration and Deflation

Beier Zhu

1 Motivation

In many problems in scientific computing, machine learning, and data analysis, one needs to compute the eigenvalues and eigenvectors of a matrix. A standard approach is to perform a full eigendecomposition to recover the entire spectrum together with the corresponding eigenvectors. Such methods are general and reliable, but they can be unnecessarily expensive when the matrix is large.

Importantly, in many applications, **we do not need the full spectrum**. Instead, we are often interested only in the **principal eigenvector**, or more generally, the top few eigenvectors associated with the largest eigenvalues. In this note, we focus on the common setting where the leading eigenvalues of interest are **positive**. This covers many practical cases and avoids unnecessary complications such as sign oscillation in the iterates. For example, in (1), the authors estimate the pre-training label prior by solving $\mathbf{q} = P\mathbf{q}$ in Eq. (6), which amounts to finding the principal eigenvector of the probability transition matrix P. Since P is a transition matrix, its largest eigenvalue is 1, which is positive.

Therefore, rather than paying the cost of computing the full spectrum, it is natural to use iterative methods tailored to the leading eigendirections. This is exactly the motivation behind **power iteration**, which computes the principal eigenvector, and **deflation**, which allows us to extract the top few eigenvectors sequentially.

2 Power Iteration

Power iteration is designed to extract the principal eigenvector of a matrix, namely, the eigenvector associated with the largest eigenvalue. For simplicity, we assume throughout this section that the dominant eigenvalue is positive.

To find the principal eigenvector of a matrix A, power iteration starts from an initial vector $\mathbf{x}_0 \neq \mathbf{0}$, repeatedly applies A to the current iterate, and then normalizes the result:

$$\mathbf{x}_{t+1} = \frac{A\mathbf{x}_t}{\|A\mathbf{x}_t\|}. \tag{1}$$

The intuition is that, when A is applied repeatedly, the component of $\mathbf{x}_0$ along the dominant eigenvector is amplified faster than the components along the other eigenvectors. As a result, under mild conditions, the iterates $\mathbf{x}_t$ converge to the principal eigenvector of A.

Formally, suppose that A is diagonalizable with eigenvalues

$$\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n,$$

and corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$. Assume further that $\lambda_1 > 0$. If the initial vector can be written as

$$\mathbf{x}_0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n,$$

with $c_1 \neq 0$, then after t iterations we have

$$A^t\mathbf{x}_0 = c_1\lambda_1^t\mathbf{v}_1 + c_2\lambda_2^t\mathbf{v}_2 + \cdots + c_n\lambda_n^t\mathbf{v}_n.$$

Factoring out λ_1^t gives

$$A^t\mathbf{x}_0 = \lambda_1^t\left(c_1\mathbf{v}_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^t\mathbf{v}_2 + \cdots + c_n\left(\frac{\lambda_n}{\lambda_1}\right)^t\mathbf{v}_n\right).$$

Since $|\lambda_i/\lambda_1| < 1$ for all $i \geq 2$, the contribution of all non-dominant eigenvectors vanishes as $t \to \infty$. Hence, after normalization, the iterate converges to the dominant eigenvector $\mathbf{v}_1$.

Two caveats are worth mentioning. First, the convergence requires that the initialization has a nonzero component along the dominant eigenvector, i.e., $c_1 \neq 0$. If $\mathbf{x}_0$ happens to be orthogonal to $\mathbf{v}_1$, then the dominant eigendirection is absent from the iterate, and power iteration cannot recover $\mathbf{v}_1$. In practice, this issue is usually avoided by random initialization.

Second, the dominant eigenvalue must be unique. If the largest eigenvalue is not simple, e.g., $\lambda_1 = \lambda_2$, then there is no longer a unique dominant eigenvector. In this case, power iteration may fail to converge to a single direction, and may instead converge to some vector in the corresponding eigenspace, depending on the initialization.

3 Deflation

While power iteration is effective for computing the principal eigenvector, in many applications we are interested in the top few eigenvectors. A natural idea is to compute them sequentially: first find the dominant eigenvector, then remove its contribution, and finally apply power iteration again to recover the next one. This strategy is known as **deflation**.

The intuition is to project out the already computed eigendirections during the iteration. For example, after obtaining $\mathbf{v}_1$, one may update

$$\mathbf{x}_{t+1} = \frac{(I - \mathbf{v}_1\mathbf{v}_1^\top)A\mathbf{x}_t}{\|(I - \mathbf{v}_1\mathbf{v}_1^\top)A\mathbf{x}_t\|},$$

which explicitly removes the component along $\mathbf{v}_1$ at each step. More generally, after computing $\mathbf{v}_1, \ldots, \mathbf{v}_k$, one may use the projection

$$P_\perp = I - \sum_{i=1}^{k} \mathbf{v}_i\mathbf{v}_i^\top$$

to eliminate the already recovered eigendirections.

Deflation may accumulate numerical error. Since each stage depends on the accuracy of the previously computed eigenpairs, an inaccurate estimate of an earlier eigenvector may contaminate the later ones. Nevertheless, when implemented carefully, deflation provides a simple and intuitive way to compute the top few eigenvectors without forming the full eigendecomposition.

References

[1] Beier Zhu, Kaihua Tang, Qianru Sun, and Hanwang Zhang. Generalized logit adjustment: Calibrating fine-tuned models by removing label bias in foundation models. In *NeurIPS*, 2024. URL: `https://arxiv.org/pdf/2310.08106v3`.