

# 1 The Power Method

The ratio of elements at two neighboring iterations converges to the largest eigenvalue, and the vector itself converges to the associated eigenvector (scaled by some constant).

Why does it work?

We can write any arbitrary vector as a linear combination of its eigenvector basis, *i.e.*

$$w = \sum_{i=1}^n c_i v_i.$$

Multiplying  $A$  results in

$$\begin{aligned} Aw &= \sum_{i=1}^n c_i A v_i \\ &= \sum_{i=1}^n c_i \lambda_i v_i \end{aligned}$$

After  $k$ th iteration, we obtain

$$\begin{aligned} A^k w &= \sum_{i=1}^n c_i \lambda_i^k v_i \\ &= \lambda_1^k \sum_{i=1}^n c_i \frac{\lambda_i^k}{\lambda_1^k} v_i \\ &\approx c_i \lambda_1^k v_1 \\ A^{k+1} w &\approx c_i \lambda_1^{k+1} v_i \end{aligned}$$

If the arbitrary vector we initialize happens to not contain a  $v_1$  component, then we would not get the first eigenvector.

Can we find the smallest eigenvalue of a positive definite matrix? Yes by using  $A^{-1}$ :

$$\begin{aligned} Ax &= \lambda x \\ \frac{1}{\lambda} x &= A^{-1} x \\ \text{so } \hat{\lambda}_{\max} &= \lambda_{\min} \end{aligned}$$

## 2 Steepest Descent

We can solve the step size that gives us the steepest descent by letting gradient of the descent step to zero. Each step would then be perpendicular to the previous step as we've exhausted descent in that direction, so any direction that is not orthogonal to this direction wouldn't maximize the descent. This method is very robust but the method is very expensive. It's linear convergence.

We can sample three points of  $t$  and solve the minimum of the quadratic formed by the three points.

## 3 Fixed point method for a system

This might accelerate a naive fixed point method.

$$\begin{aligned}x_1^* &= g_1(x_1, x_2, \dots, x_n) \\x_2^* &= g_2(x_1^*, x_2, \dots, x_n) \\x_n^* &= g_n(x_1^*, x_2^*, \dots, x_n)\end{aligned}$$

## 4 Newton's Method for root finding of a system

$$\begin{aligned}f_1(x, y, z) &= 0 \\f_2(x, y, z) &= 0 \\f_3(x, y, z) &= 0\end{aligned}$$

Expand about point  $(x, y, z)_{n-1}$

$$f_i(x, y, z) = f_i(x, y, z)_{n-1} + (x - x_{n-1}) \frac{\partial f_i}{\partial x}(x, y, z)_{n-1} + (y - y_{n-1}) \frac{\partial f_i}{\partial y}(x, y, z)_{n-1} + (z - z_{n-1}) \frac{\partial f_i}{\partial z}(x, y, z)_{n-1}$$

$$\begin{pmatrix} \frac{\partial f_1}{\partial x}(x, y, z)_{n-1} & \frac{\partial f_1}{\partial y}(x, y, z)_{n-1} & \frac{\partial f_1}{\partial z}(x, y, z)_{n-1} \\ \frac{\partial f_2}{\partial x}(x, y, z)_{n-1} & \frac{\partial f_2}{\partial y}(x, y, z)_{n-1} & \frac{\partial f_2}{\partial z}(x, y, z)_{n-1} \\ \frac{\partial f_3}{\partial x}(x, y, z)_{n-1} & \frac{\partial f_3}{\partial y}(x, y, z)_{n-1} & \frac{\partial f_3}{\partial z}(x, y, z)_{n-1} \end{pmatrix} \begin{pmatrix} x - x_{n-1} \\ y - y_{n-1} \\ z - z_{n-1} \end{pmatrix} = \begin{pmatrix} f_1(x, y, z) \\ f_2(x, y, z) \\ f_3(x, y, z) \end{pmatrix}.$$

$$x_n = x_{n-1} - J^{-1}(x_{n-1})F(x_{n-1}).$$