Multidimensional Scaling (MDS)
by HappyZ Mar. 30

- Given d_{ij}, the distances between pairs of N points, MDS can place these points in a low (e.g. 2-D) space such that the Euclidean distance between them is as close as possible (i.e. given $\boldsymbol{D}$, approximate $\boldsymbol{X}$).
- *Squared Euclidean distance* (for points r and s):
  - $d_{rs}^2 = \|\boldsymbol{x}^r - \boldsymbol{x}^s\|_2^2 = \sum_{j=1}^{d}\left(x_j^r - x_j^s\right)^2 = b_{rr} + b_{ss} - 2b_{rs}$ where $b_{rs} = \sum_{j=1}^{d} x_j^r x_j^s$
    - Or $d_{rs}^2 = \|\boldsymbol{x}_r - \boldsymbol{x}_s\|_2^2 = (\boldsymbol{x}_r - \boldsymbol{x}_s)^T(\boldsymbol{x}_r - \boldsymbol{x}_s) = \boldsymbol{x}_r^T\boldsymbol{x}_r - \boldsymbol{x}_s^T\boldsymbol{x}_s - 2\boldsymbol{x}_r^T\boldsymbol{x}_s$
      $\boldsymbol{D} = \boldsymbol{b}\boldsymbol{e}^T + \boldsymbol{e}\boldsymbol{b}^T - \boldsymbol{2XX}^T$ where $\boldsymbol{b}$ is the diagonal elements in $\boldsymbol{B} = \boldsymbol{XX}^T$ and $\boldsymbol{e}$ is $[1 \quad \cdots \quad 1]^T$
  - Define
    - $d_{\bullet\bullet}^2 = \frac{1}{N^2}\sum_r\sum_s d_{rs}^2$ or $\boldsymbol{d}_{\bullet\bullet} = \frac{1}{N^2}\boldsymbol{e}^T\boldsymbol{De}$
    - $d_{r\bullet}^2 = \frac{1}{N}\sum_s d_{rs}^2$ or $\boldsymbol{d}_{r\bullet} = \frac{1}{N}\boldsymbol{De}$
    - $d_{\bullet s}^2 = \frac{1}{N}\sum_r d_{rs}^2$ or $\boldsymbol{d}_{\bullet s} = \boldsymbol{d}_{r\bullet}^T = \frac{1}{N}\boldsymbol{e}^T\boldsymbol{D}$ since $\boldsymbol{D} = \boldsymbol{D}^T$
  - Then $b_{rs} = \frac{1}{2}(d_{r\bullet}^2 + d_{\bullet s}^2 - d_{\bullet\bullet}^2 - d_{rs}^2)$
    - Or $\boldsymbol{B} = \frac{1}{2}(\boldsymbol{d}_{r\bullet}\boldsymbol{e}^T + \boldsymbol{e}\boldsymbol{d}_{\bullet s} - \boldsymbol{d}_{\bullet\bullet}\boldsymbol{ee}^T - \boldsymbol{D}) = \boldsymbol{XX}^T$

    Proof. Assume all data is centered and $\boldsymbol{X}^T\boldsymbol{e} = 0$. Then $\boldsymbol{e}^T\boldsymbol{De} = \boldsymbol{e}^T\boldsymbol{b}\boldsymbol{e}^T\boldsymbol{e} + \boldsymbol{e}^T\boldsymbol{e}\boldsymbol{b}^T\boldsymbol{e} - 2\boldsymbol{e}^T\boldsymbol{XX}^T\boldsymbol{e} = 2\tau N$, where $\tau = \boldsymbol{e}^T\boldsymbol{b}$ is the sum of all square distances, and $\boldsymbol{e}^T\boldsymbol{e} = N$, which is the number of points. Then $\tau = \frac{1}{2N}\boldsymbol{e}^T\boldsymbol{De}$.

    Since $\boldsymbol{De} = \boldsymbol{b}\boldsymbol{e}^T\boldsymbol{e} + \boldsymbol{e}\boldsymbol{b}^T\boldsymbol{e} = N\cdot\boldsymbol{b} + \tau\cdot\boldsymbol{e}$, we get $\boldsymbol{b} = \frac{1}{N}(\boldsymbol{De} - \tau\cdot\boldsymbol{e})$.

    Then $\boldsymbol{D} = \frac{\boldsymbol{De}}{N}\boldsymbol{e}^T - \frac{\tau}{N}\boldsymbol{e}^T\boldsymbol{e} + \boldsymbol{e}\left(\frac{\boldsymbol{De}}{N}\right)^T - \frac{\tau}{N}\boldsymbol{ee}^T - 2\boldsymbol{B} = \frac{\boldsymbol{De}}{N}\boldsymbol{e}^T + \boldsymbol{e}\left(\frac{\boldsymbol{De}}{N}\right)^T - \frac{\boldsymbol{e}^T\boldsymbol{De}}{N^2}\boldsymbol{ee}^T - 2\boldsymbol{B}$.

    So $\boldsymbol{B} = \frac{1}{2}\left(\left(\frac{\boldsymbol{De}}{N}\right)\boldsymbol{e}^T + \boldsymbol{e}\left(\frac{\boldsymbol{De}}{N}\right)^T - \frac{\boldsymbol{e}^T\boldsymbol{De}}{N^2}\boldsymbol{ee}^T - \boldsymbol{D}\right) = \frac{1}{2}(\boldsymbol{d}_{r\bullet}\boldsymbol{e}^T + \boldsymbol{e}\boldsymbol{d}_{\bullet s} - \boldsymbol{d}_{\bullet\bullet}\boldsymbol{ee}^T - \boldsymbol{D})$.
  - So we approximate $\boldsymbol{X} = \boldsymbol{C}\sqrt{\boldsymbol{\Lambda}}$, where $\boldsymbol{C}$ is the matrix whose columns are the eigenvectors of $\boldsymbol{B}$ and $\sqrt{\boldsymbol{\Lambda}}$ is a diagonal matrix with square roots of the eigenvalues on the diagonals.
    - In terms of eigenvalues of $\boldsymbol{B}$, we decide on the dimensionality $k < d$ (and $k < N$) as we did in PCA. So the approximation of $\boldsymbol{X}$ satisfies $z_j^t = \sqrt{\lambda_j}c_j^t, j = 1, 2, \dots, k, t = 1, \dots, N$
- The eigenvalues of $\boldsymbol{XX}^T$ are the same as those of $\boldsymbol{X}^T\boldsymbol{X}$.

Ex. Replica of Figure 6.6 In the book (*Istanbul* and *Athens* are wrong in the book).

```
% HappyZ @ Mar. 30, 2014
clear all; close all; clc
% Initialization
loc = {'Lisbon', 'Madrid', 'Dublin', 'London', 'Paris', 'Zurich', 'Rome', 'Berlin', ...
    'Helsinki', 'Istanbul', 'Moscow', 'Athens'};
% Following data comes from Google Maps (in miles)
D = [0, 388.5, 1734.3, 1352.9, 1078, 1317.2, 1559.5, 1726.7, 2534, 2541.3, 2830.1,
2388.4; ...
    388.5, 0, 1444, 1062.6, 787.7, 1029.9, 1214.5, 1436.4, 2243.7, 2196.3, 2539.8,
2043.5; ...
    1734.3, 1444, 0, 363.4, 656.1, 996.7, 1513.5, 1050.9, 1844.3, 2233.5, 2154.3,
2330.6; ...
    1352.9, 1062.6, 363.4, 0, 283.5, 624.1, 1141, 678.4, 1471.7, 1860.9, 1781.8, 1958.1; ...
    1078, 787.7, 656.1, 283.5, 0, 407.3, 886.1, 651, 1458.3, 1714.3, 1754.4, 1811.4; ...
    1317.2, 1029.9, 996.7, 624.1, 407.3, 0, 533, 525.5, 1504.9, 1366.8, 1628.9, 1464; ...
    1559.5, 1214.5, 1513.5, 1141, 886.1, 533, 0, 936.6, 1773.8, 1085.5, 1897.8, 837.5; ...
    1726.7, 1436.4, 1050.9, 678.4, 651, 525.5, 936.6, 0, 1002.1, 1364.6, 1126.1, 1461.8; ...
    2534, 2243.7, 1844.3, 1471.7, 1458.3, 1504.9, 1773.8, 1002.1, 0, 2021.9, 687.7,
2119.1; ...
    2541.3, 2196.3, 2233.5, 1860.9, 1714.3, 1366.8, 1085.5, 1364.6, 2021.9, 0, 1332.7,
680.4; ...
```

```
    2830.1, 2539.8, 2154.3, 1781.8, 1754.4, 1628.9, 1897.8, 1126.1, 687.7, 1332.7, 0,
1852; ...
    2388.4, 2043.5, 2330.6, 1958.1, 1811.4, 1464, 837.5, 1461.8, 2119.1, 680.4, 1852, 0];
N = length(D);
e = ones(N, 1);
% define d's based on eq. 6.26
d_dotdot = 1/N^2*(e'*D*e);
d_rdot = 1/N*(D*e);
d_dots = 1/N*(e'*D); % since D = D'
B = 1/2*(d_rdot*e' + e*d_dots - d_dotdot*e*e' - D);
[V, EigVals] = eig(B);
[o, i] = sort(diag(EigVals), 'descend');
V = V(:, i); EigVals = EigVals(i, i);
C = V(:, 1:2);
Diag = sqrt([EigVals(1,1); EigVals(2,2)]);
X = [Diag(1)*C(:,1), Diag(2)*C(:,2)];
scatter(-X(:,1), -X(:,2), 'k.');
for i = 1:N
    if (i == 2) text(-X(i, 1) + 0.5, -X(i, 2) + 0.5, loc{i}); % fix overlapping
    else text(-X(i, 1) + 0.5, -X(i, 2) - 0.8, loc{i}); end
end
title({'Replica of Figure 6.6 in book "Introduction to Machine Learning"';
    'Distance data comes from Google Maps, by HappyZ @ Mar. 30'});
xlabel('miles'); ylabel('miles');
xlim([-25, 30]); ylim([-25, 25]);
```