PCA

Suppose we have n data samples $x_1, \dots, x_n \in \mathbb{R}^p$. We denote

1) the sample mean $\mu^* = \frac{1}{n} \sum_{k=1}^{n} x_k$, $\mu^* \in \mathbb{R}^p$

mean done element-wise, i.e., $\mu_i^* = \frac{1}{n} \sum_{k=1}^{n} x_{k,i}$

2) the sample covariance

$$C^* = \frac{1}{n-1} \sum_{k=1}^{n} (x_k - \mu^*)(x_k - \mu^*)^T$$

this is an outer product!

↳ for $a, b \in \mathbb{R}^p$, $(a \cdot b^T)_{ij} = a_i b_j$

$$a \cdot b^T = \begin{pmatrix} a_1 b_1 & a_1 b_2 & \dots & a_1 b_p \\ \vdots & \ddots & & \vdots \\ a_p b_1 & \dots & & a_p b_p \end{pmatrix}$$

First approach

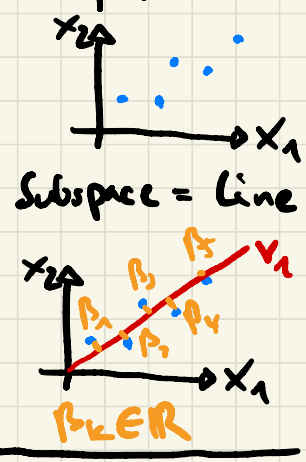
1) We want to find a lower-dim. representation of our data samples.

Idea: Project the samples into a d-dim. subspace with basis vectors $v_i \in \mathbb{R}^p$, $i \in [1, d]$ with $d < p$. For convenience, we choose an orthonormal basis $\rightarrow v_i^T v_j = \delta_{ij}$,

If V has the vectors as columns, $V = (v_1, \dots, v_d)$ then $V^T V = \mathbb{1}$.

(note that $VV^T \neq \mathbb{1}$!!)

Example: 2D $\rightarrow$ 1D



2) A good representation has to be close to the original

$$\mu_k + V\beta_k \approx x_k$$

- μ_k: offset
- V: basis vectors
- β_k: vector components
- x_k: reconstruct the original sample

β_k is our low-dim. representation of x_k!

We phrase this as an optimization problem:

$$\min_{\mu, V, \beta_k, V^T V = \mathbb{1}} \sum_{k=1}^{n} \| x_k - (\mu + V\beta_k) \|_2^2$$

Euclidean norm

Find μ: W.l.o.g. we assume $\sum_{k=1}^{n} \beta_k = 0$.

To find μ, we set the gradient to 0:

$$\nabla_\mu \sum_{k=1}^{n} \| x_k - (\mu + V\beta_k) \|^2 \stackrel{!}{=} 0$$

$$\Leftrightarrow \sum_{k=1}^{n} x_k - (\mu + V\beta_k) = 0$$

$$\Leftrightarrow \mu = \frac{1}{n} \sum_{k=1}^{n} x_k \quad \text{since} \quad \sum_{k=1}^{n} \beta_k = 0$$

$$= \mu^{ML} \quad \text{(for simplicity, we write } \mu\text{)}$$

Find β_k: Same approach! $\nabla_{\beta_i} \sum_{k=1}^{n} \| x_k - (\mu + V\beta_k) \|^2$

note: $\| \cdot \|^2$ terms $\Rightarrow$ decouples in k!

Problem: $\nabla_{\beta_k} \| V\beta_k \|^2 = ?$

($V\beta_k$: matrix-vector product)

For simplicity, lets write this with indices:

$$\frac{\partial}{\partial \beta_{k,m}} \sum_i \Big(\sum_j V_{ij} \beta_{kj} \Big)^2 = 2 \sum_i \Big(\sum_j V_{ij} \beta_{kj} \Big) V_{im}$$

$$= 2 \sum_{ij} V^T_{mi} V_{ij} \beta_{kj}$$

$$\Rightarrow \nabla_{\beta_k} \frac{1}{2} \| V\beta_k \|^2 = V^T V \beta_k$$

Lets apply this: $\nabla_{\beta_k} \| x_k - (\mu + V\beta_k) \|^2 = 0$

$$\Leftrightarrow \quad V^T (x_k - (\mu + V\beta_k)) = 0$$

$$\Leftrightarrow \quad V^T (x_k - \mu) = \underbrace{V^T V}_{= \mathbb{1}} \beta_k = \beta_k$$

Quickly check our assumpt.: $\sum_{k=1}^{n} \beta_k = \sum_{k=1}^{n} V^T (x_k - \mu) = 0$ ✓

Thus, β_k is obtained by projecting x_k to the low.-dim. subspace!

The last step is to find the basis vectors!

↳ insert β_k: $\min_{V,\ V^TV=\mathbb{1}} \| x_k-\mu - \underbrace{VV^T}_{\neq \mathbb{1}!!!}(x_k-\mu)\|^2$

Write this out: $(x_k-\mu-VV^T(x_k-\mu))^2$

$$= (x_k-\mu)^2 + (x_k-\mu)^T V \underbrace{V^TV}_{=\mathbb{1}} V^T(x_k-\mu)$$
$$-2(x_k-\mu)^T VV^T(x_k-\mu)$$
$$= (x_k-\mu)^2 - (x_k-\mu)^T VV^T(x_k-\mu)$$

Note: $(x_k-\mu)^2$ does not depend on V

$\Rightarrow$ min turns to $\max_{V,\ V^TV=\mathbb{1}} \sum_{k=1}^{n} (x_k-\mu)^T VV^T(x_k-\mu)$

due to the minus sign, min becomes max

We can rewrite this to arrive at a nice expression.

Substitute $a_k = (x_k - \mu)$ for simplicity.

$$\rightarrow \sum_k a_k^T V V^T a_k = \sum_k \sum_n \sum_i \sum_j a_{k,n} V_{ni} V_{ij}^T a_{k,j}$$

$$= \sum_{k,j,i,n} V_{in}^T a_{k,n} a_{k,j} V_{ji}$$

$$= \sum_{i,j,n} V_{in}^T \underbrace{\sum_k a_{k,n} a_{k,j}}_{=[a_k a_k^T]_{nj}} V_{ji}$$

$$= \sum_i \left[V^T \left(\sum_k a_k a_k^T \right) V \right]_{ii}$$

$$= \mathrm{Tr}\left(V^T \left(\sum_k a_k a_k^T \right) V \right)$$

Substitute back a_k: $\mathrm{Tr}\left(V^T \underbrace{\left[\sum_k (x_k - \mu)(x_k - \mu)^T\right]}_{= (n-1)\,C^*} V\right)$

C^*: Covariance matrix!

Thus, our problem becomes

$$\max_{V,\ \text{s.t.}\ V^TV=\mathbb{1}} \mathrm{Tr}(V^TC^*V)$$

We can do some more rewriting:

$$\mathrm{Tr}(V^TC^*V) = \sum_{i,n,j} V_{in}^T C^*_{nj} V_{ji} = \sum_{i,n,j} V_{ni} C^*_{nj} V_{ji}$$

product sums over column entries!

$$= \sum_i V_i^T C^* V_i$$

basis vectors

C^* is a symmetric and real-valued matrix.
Thus, it has an eigenvalue decomp.
In the exercises, we showed that $\max_{v_1,\ \text{s.t.}\ \|v_1\|^2=1} v_1^T C^* v_1$ $(d=1)$
is solved by the eigenvector of C^* with the largest eigenvalue. For $d>1$, V is given by the eigenvectors of C^* with the d largest eigenvalues. □

Second approach

Idea: Lets find a projection of the data with maximum variance.

The projection is given by $V \in \mathbb{R}^{1\times d}$, $VV^T = 1$, and the projected points are $y_k = V^T x_k$

Formulate problem:

Sample mean: $\bar{\mu} = \frac{1}{n}\sum_{k=1}^{n} y_k = \frac{1}{n}\sum_{k=1}^{n} V^T x_k = V^T \mu^*$

Sample variance: $\frac{1}{n-1}\sum_{k=1}^{n} \| y_k - \bar{\mu}\|^2$

$$= \frac{1}{n-1}\sum_{k=1}^{n} \| V^T x_k - V^T \mu^*\|^2$$

$$= \frac{1}{n-1}\sum_{k=1}^{n} \| V^T (x_k - \mu^*)\|^2$$

Maximize the sample variance:

$$\max_{V} \sum_{k=1}^{n} \| V^T (x_k - \mu)\|^2 \quad \text{s.t. } VV^T = 1$$

→ same objective as in the prev. approach!

same solution! ⇐

□

Notes on Marčenko-Pastur

$x_k = g + \sqrt{\beta}\, g_0\, u$ → x_k is a sum of scaled Gaussian, and hence also a Gaussian RV.

The moments are:

1) $E(x_k) = \underbrace{E(g)}_{=0} + \sqrt{\beta}\, u\, \underbrace{E(g_0)}_{=0} = 0 \Rightarrow \mu = 0$

2) $\Sigma = E\left((x_k - \mu)(x_k - \mu)^T\right)$

$= E\left(x_k x_k^T\right) = \underbrace{E\left(g g^T\right)}_{=\mathbb{1}} + \underbrace{E(g g_0)}_{=0}\sqrt{\beta}\, u^T$

$+ \underbrace{E(g_0 g^T)}_{=0}\sqrt{\beta}\, u + \underbrace{E(g_0^2)}_{=1}\beta\, u u^T$

$= \mathbb{1} + \beta\, u u^T$

Σ previously

Johnson-Lindenstrauss lemma

Given: $x_1, \dots, x_m \in \mathbb{R}^n$ are our data samples

$y_1, \dots, y_m \in \mathbb{R}^d$. $y_i = \frac{1}{\sqrt{d}} G x_i$ is the proj. data

$G \in \mathbb{R}^{d \times n}$, with $G_{ij} \sim \mathcal{N}(0,1)$

y_i is a rescaled unit Gaussian random variable

↳ $y_{ik} = \frac{1}{\sqrt{d}} \sum_j G_{kj} x_{ij}$ = sum of Gaussian RVs (G_{kj}) scaled by $\frac{1}{\sqrt{d}} x_{ij}$!

$\Rightarrow$ y_{ik} is a Gaussian RV!

Lets calculate the moments:

$$E(y_i) = \frac{1}{\sqrt{d}} E(G) x_i = 0 \quad \text{since } E(G)=0$$

$$\text{Var}(y_{ik}) = \text{Var}\left(\frac{1}{\sqrt{d}} \sum_j G_{kj} x_{ij}\right)$$

$$= \frac{1}{d} \sum_j x_{ij}^2 \underbrace{\text{Var}(G_{kj})}_{=1} = \frac{1}{d} \|x_i\|^2$$

Thus, we can represent y_i as a rescaled unit Gaussian $g_i \in \mathbb{R}^d$, $E(g_i)=0$ and $\text{Var}(g_i)=1$: (1-vector)

$$y_i = \frac{\|x_i\|}{\sqrt{d}} g_i$$

Proof of one side

We will only do the proof for one case, as the technique is the same for proving the whole theorem.

Moreover, we will prove

$$P(\|y_i\|^2 \geq (1+\varepsilon)^2 \|x_i\|^2) \leq \frac{1}{m^3},$$

which is equivalent to what we are looking for:

$$P(\|y_i\| \geq (1+\varepsilon)\|x_i\|) \leq \frac{1}{m^3}$$

↳ the prob. that our projection deviates too much!

We first write out $\|y_i\|^2 = \frac{\|x_i\|^2}{d} \sum_j g_{ij}^2$.

Thus, we want to bound

$$P(\|y_i\|^2 \geq (1+\varepsilon)^2 \|x_i\|^2) = P(\Sigma g_{ij}^2 \geq (1+\varepsilon)^2 d),$$

which can be done via the Chernoff bound!

$$\underbrace{P(\Sigma g_{ij}^2 \geq (1+\varepsilon)^2 d)}_{\text{the probability we would like to bound}} \leq \underbrace{e^{-t(1+\varepsilon)^2 d} E(e^{t\Sigma g_{ij}^2})}_{\text{Chernoff}}$$

↓ moment-gen. function!

We need to calculate the moment gen. fct.

First, we have $E\left(e^{t\sum_j g_{ij}^2}\right) = \prod_j E\left(e^{t g_{ij}^2}\right)$ (independence)

Then, we expand the expectation value: $N(0,1)$

$$E\left(e^{t g_{ij}^2}\right) = \frac{1}{\sqrt{2\pi}} \int e^{t z^2} \cdot e^{-\frac{z^2}{2}} dz$$

(rewrite as z in integral)

$$= \frac{1}{\sqrt{2\pi}} \int e^{-\frac{z^2}{2} + t z^2} dz$$

$$= \frac{1}{\sqrt{2\pi}} \int e^{-\frac{z^2}{2}\cdot(1-2t)} dz$$

Substitute with $z' = z\sqrt{1-2t}$, $dz' = dz\sqrt{1-2t}$

$$= \frac{1}{\sqrt{1-2t}} \cdot \underbrace{\frac{1}{\sqrt{2\pi}} \int e^{-\frac{z'^2}{2}} dz'}_{\equiv 1}$$

$$= \frac{1}{\sqrt{1-2t}}$$

Thus, we have: $E\left(e^{-t\sum g_{ij}^2}\right) = (1-2t)^{-\frac{d}{2}}$

and thus $P\left(\|y_i\| \geq (1+\varepsilon)\|x_i\|\right) \leq e^{-t(1+\varepsilon)d} \cdot (1-2t)^{-\frac{d}{2}}$

We are not yet finished, as we have to choose t! t is arbitrary, so after some meditation (joking, you have to work backwards from the result to find this): $t = \frac{1}{2}\left(\frac{(1+\varepsilon)^2 - 1}{(1+\varepsilon)^2}\right)$

$$\Rightarrow P(\|Y_i\| \geq (1+\varepsilon)\|X_i\|)$$
$$\leq e^{-\frac{d}{2}[(1+\varepsilon)^2 - 1]} \cdot \left(\frac{1}{(1+\varepsilon)^2}\right)^{-\frac{d}{2}}$$
$$= e^{-\frac{d}{2}[(1+\varepsilon)^2 - 1] - \frac{d}{2}\ln\left(\frac{1}{(1+\varepsilon)^2}\right)}$$
$$= e^{-d\left[\varepsilon + \frac{\varepsilon^2}{2} - \ln(1+\varepsilon)\right]}$$

use $\ln(1+x) \leq x - \frac{x^2}{4}$ for $x \in [0,1]$ (just Taylor expansion)

$$\leq e^{-d\left[\varepsilon + \frac{\varepsilon^2}{2} - \varepsilon + \frac{\varepsilon^2}{4}\right]}$$
$$= e^{-d\frac{3\varepsilon^2}{4}}$$

Now set $d = 4\ln(m)/\varepsilon^2$

$$\Rightarrow P(\|Y_i\| \geq (1+\varepsilon)\|X_i\|) \leq \frac{1}{m^3}$$

Proof for the left tail $(1-\varepsilon)\|x_i\|$ is similar!

Note that our result is valid for <u>any</u> vector $v \in \mathbb{R}^n$. Hence, it also holds for, e.g.,

$v = x_i - x_j$

$\hookrightarrow P(L(v) \geq (1+\varepsilon)\|v\|)$

$= P(\|y_i - y_j\| \geq (1+\varepsilon)\|x_i - x_j\|) \leq \frac{1}{n^3}$

since $L(v) = \frac{1}{\sqrt{d}} G(x_i - x_j) = \frac{1}{\sqrt{d}} Gx_i - \frac{1}{\sqrt{d}} Gx_j = y_i - y_j$

The last step

The lemma states that lengths and distances do not deviate too much for <u>all</u> data points in <u>both</u> directions (being too large or too small).

We have

$$P_i = P(\|y_i\| \geq (1+\varepsilon)\|x_i\| \text{ or } \|y_i\| \leq (1-\varepsilon)\|x_i\|) \leq \frac{2}{n^3}$$

and $P_{ij} = P(\|y_i - y_j\| \geq (1+\varepsilon)\|x_i - x_j\|) \text{ or } \|y_i - y_j\| \leq (1-\varepsilon)\|x_i - x_j\|)$

$\leq \frac{2}{n^3}$

What is the probability that at least one of these inequalities is true?

↳ remember Kolmogorov! For event A, B,

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$
$$\leq P(A) + P(B)$$
$$\Rightarrow P(\bigcup_i A_i) \leq \sum P(A_i)$$

$\Rightarrow$ P(at least one deviation)

$$\leq \sum_i \underbrace{p_i}_{\leq \frac{2}{m^3}} + \sum_{i \neq j} \underbrace{p_{ij}}_{\leq \frac{2}{m^3}} \leq \frac{2}{m}$$

in total we have m^2 vectors

$\Rightarrow$ P(no deviation) = 1 − P(at least one deviation)

$$\geq 1 - \frac{2}{m}$$

□

Spectral clustering

Laplacian

We define it as $L = D - W$, $L \in \mathbb{R}^{n \times n}$

with W_{ij} = "weight" between node i and j

and $D_{ii} = \sum_j W_{ij}$, $D_{ij} = 0$ if $i \neq j$

with $W_{ij} \geq 0$

Properties of L:

1) For every $f \in \mathbb{R}^n$, we have $f^T L f = \frac{1}{2} \sum_{ij}^{n} W_{ij}(f_i - f_j)^2$

Proof:

$$f^T L f = f^T D f - f^T W f$$

$$= \sum_i D_{ii} f_i^2 - \sum_{ij} f_i W_{ij} f_j$$

$$= \frac{1}{2}\left(\sum_i D_{ii} f_i^2 - 2 \sum_{ij} f_i f_j W_{ij} + \sum_j D_{jj} f_j^2 \right)$$

$$= \frac{1}{2} \sum_{ij} W_{ij}(f_i^2 - 2 f_i f_j + f_j^2) = \frac{1}{2} \sum_{ij} W_{ij}(f_i - f_j)^2$$

□

2) L is symmetric and pos. def.

↳ We have $D^T = D$ and $W^T = W \Rightarrow L^T = L$

↳ Pos. def. follows from 1)

3) Smallest eigenvalue is 0 with the one-vector $\mathbb{1}$ as eigenvector.

↳ $(1, 1, 1, \ldots, 1)$

Lets look at the product $(L\mathbb{1})_i$:

$$(L\mathbb{1})_i = D_{ii} - \sum_j W_{ij} = \sum W_{ij} - \sum W_{ij} = 0$$

II

4) L has non-negative, real-valued eigenvalues

↳ Follows from symmetry (real-valued) and pos. def. (pos. eigenvalues)

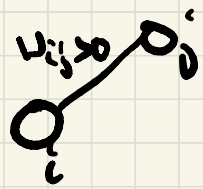
 # Connected components of G = multiplicity of 0-EV of L

1) Lets assume a graph with one connected component. Let f be the eigenvector with eigenvalue 0.

We know that $f^T L f = \frac{1}{2} \sum_{ij} w_{ij} (f_i - f_j)^2 = 0$

since $Lf = 0$

Since $w_{ij} \geq 0$, for all $w_{ij} \neq 0$ we have to demand $f_i = f_j$.



$\Rightarrow f_i = f_j = c$

$\Rightarrow f_i = f_j = f_k = c$

Basically: for all nodes i,k connected by a path, we have $f_i = f_k$.

Since G is connected, we find $f = \mathbb{1} \cdot c$

2) Now assume we have k connected components.

A_k contains all node indices of the k^{th} comp.

$\hookrightarrow V = \bigcup_k A_k$ and $A_i \cap A_j = \{\}$ if $i \neq j$

We can then decompose the sum:

$$f^T L f = \frac{1}{2} \sum_k \sum_{i,j \in A_k} w_{ij} (f_i - f_j)^2 = 0$$

For $k=0$, this is solved by setting $f_i = \begin{cases} 1 & \text{if } i \in A_0 \\ 0 & \text{else} \end{cases}$

For $k=1$, ——— // ——— $f_i = \begin{cases} 1 & \text{if } i \in A_1 \\ 0 & \text{else} \end{cases}$

etc.

Lets call the found EV for k: f^k

From the def. follows: $(f^l)^T f^m = 0$ if $l \neq m$

$\Rightarrow$ We have k orthogonal eigenvectors with eigenvalue 0! $\square$

Graph cut

Lets assume we have again a graph $G=(V,E)$ with edge weights W_{ij}. ← measure of similarity

For clustering, we want to separate the nodes into k sets A_k such that

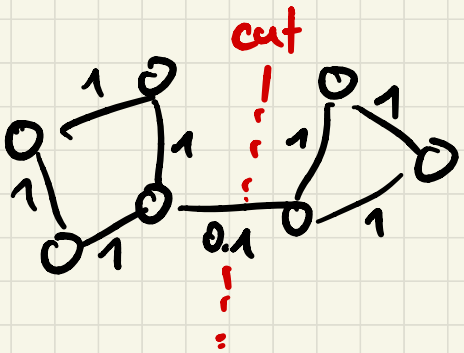
1) Edges between nodes from different sets have low weight

Lets denote by $W(A,B) = \sum_{\substack{i\in A \\ j\in B}} W_{ij}$

We can formalize this as a mincut problem:

The problem of finding k cuts is to minimize $\quad cut(A_1,\dots,A_k) := \frac{1}{2}\sum_{i=1}^{k} W(A_i,\bar{A}_i)$

Example for k=2:



This formulation has one problem:
we can solve it by separating individual nodes from the graph... those are bad clusters (its just outliers!!)

Thus, adjust to also include the set sizes!
We take the so-called Ratiocut.

$$\text{RatioCut}(A_1, \dots, A_k) := \sum_{i=1}^{k} \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|}$$

Note: Look at $f = \frac{1}{\alpha} + \frac{1}{\alpha - 1} \rightsquigarrow \frac{d}{d\alpha} f = -\frac{1}{\alpha^2} - \frac{1}{(\alpha-1)^2} \overset{!}{=} 0$

$$\alpha^2 - 2\alpha + 1 = \alpha^2$$
$$\Rightarrow \alpha = \frac{1}{2}$$

Connecting RatioCut to graph Laplacian

k=2: Our goal is to solve the following

$$\min_{A \subset V} \text{RatioCut}(A, \bar{A})$$

First, we introduce the notation $\bar{A} = \bar{A}$

Then, we define the vector $f \in \mathbb{R}^n$

$$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|} & \text{if } i \in A \\ -\sqrt{|A|/|\bar{A}|} & \text{else} \end{cases}$$

Basically, this is an indicator fct.

↳ if $f_i > 0$, then $i \in A$

$f_i < 0$, then $i \in \bar{A}$

Note that:

1) $$\sum_i f_i = \sum_{i \in A} \sqrt{\frac{|\bar{A}|}{|A|}} - \sum_{i \in \bar{A}} \sqrt{\frac{|A|}{|\bar{A}|}}$$

$$= \sqrt{|A||\bar{A}|} - \sqrt{|\bar{A}||A|} = 0$$

$\Rightarrow f$ is orth. to the 1-vector!

2) $$\sum_i f_i^2 = |\bar{A}| + |A| = \text{number vertices}$$

Lets evaluate $f^T L f$:

$$f^T L f = \frac{1}{2} \sum_{ij} w_{ij} (f_i - f_j)^2$$

$$= \underbrace{\frac{1}{2} \sum_{i \in A, j \in \bar{A}} w_{ij} \left(-\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}} \right)^2}_{= \text{cut}(A, \bar{A})} + \underbrace{\frac{1}{2} \sum_{i \in \bar{A}, j \in A} w_{ij} \left(-\sqrt{\frac{|A|}{|\bar{A}|}} - \sqrt{\frac{|\bar{A}|}{|A|}} \right)^2}_{= \text{cut}(A, \bar{A}) \text{ since } w_{ij} = w_{ji}}$$

$$= \text{cut}(A, \bar{A}) \cdot \left(\frac{|A|}{|\bar{A}|} + \frac{|\bar{A}|}{|A|} + 2 \right)$$

$$= \text{cut}(A, \bar{A}) \cdot \left(\frac{|A|}{|\bar{A}|} + \frac{|\bar{A}|}{|A|} + \frac{|\bar{A}|}{|\bar{A}|} + \frac{|A|}{|A|} \right)$$

$$= \text{cut}(A, \bar{A}) \cdot \left(\frac{|A| + |\bar{A}|}{|\bar{A}|} + \frac{|\bar{A}| + |A|}{|A|} \right) \quad \begin{array}{l} \#\text{vertices} \\ V = |A| + |\bar{A}| \end{array}$$

$$= V \cdot \text{cut}(A, \bar{A}) \cdot \left(\frac{1}{|A|} + \frac{1}{|\bar{A}|} \right) \longrightarrow \text{that's RatioCut!}$$

Thus: Optimising RatioCut is equivalent to

$$\min(f^T L f) \text{ with } f \perp \mathbb{1} \text{ and } \|f\|_2^2 = V$$

with f_i defined as before.

Solving this is NP-hard!

BUT we can relax it by allowing general vectors f!

↳ $\min_f(f^T L f)$ such that $f \perp \mathbb{1}$, $\|f\|_2^2 = V$

This is similar to PCA, but instead of max, we have min!

↳ Instead of the largest eigenvalue, we look at the lowest!

↳ But $f \perp \mathbb{1}$, and $\mathbb{1}$ is the eigenvector of the lowest eigenvalue 0

↳ Thus, this is solved by the rescaled eigenvector corresponding to the second-smallest eigenvalue!

Note: through the relaxation, we lose the simple rule $f_i \geq 0$ to assign clusters. Thus, we have to use k-means!

general k: For $k > 2$, we follow a similar approach. Lets assume a partitioning into k sets A_i, $i \in [1, k]$.

Then define k indicator vectors

$$F_{ij} = \begin{cases} \frac{1}{\sqrt{|A_j|}} & \text{if } i \in A_j \\ 0 & \text{otherwise} \end{cases}$$

$F \in \mathbb{R}^{V \times k}$; $V = \#$ nodes

F is orthonormal! $F^T F = I$

$$\hookrightarrow (F^T F)_{il} = \sum_j F_{ij}^T F_{jl} = \sum_j F_{ji} F_{jl}$$

case 1 ($i = l$): $\sum_j F_{ji} F_{ji} = \sum_{j \in A_i} \frac{1}{|A_i|} = 1$

case 2 ($i \neq l$): $\sum_j F_{ji} F_{jl} = 0$
as both cannot be > 0 at the same time

In the following, lets define h_i as the ith column of F.

Lets look again at $h_i^T L h_i$ ($h_{ik} = F_{ik}$, $h_{jk} = F_{jk}$)

$$h_k^T L h_k = \frac{1}{2} \sum_{ij} w_{ij} (h_{ki} - h_{kj})^2$$

$$= \frac{1}{2} \sum_{\substack{i \in A_k \\ j \in \bar{A}_k}} w_{ij} \left(\frac{1}{\sqrt{|A_k|}} - 0 \right)^2$$

$$+ \frac{1}{2} \sum_{\substack{i \in A_k \\ j \in A_k}} w_{ij} \underbrace{\left(\frac{1}{\sqrt{|A_k|}} - \frac{1}{\sqrt{|A_k|}} \right)^2}_{=0}$$

$$+ \frac{1}{2} \sum_{\substack{i \in \bar{A}_k \\ j \in \bar{A}_k}} w_{ij} \underbrace{(0-0)^2}_{=0}$$

$$+ \frac{1}{2} \sum_{\substack{i \in \bar{A}_k \\ j \in A_k}} w_{ij} \left(0 - \frac{1}{\sqrt{|A_k|}} \right)^2$$

$$= \text{cut}(A_k, \bar{A}_k) \frac{1}{|A_k|} = \text{RatioCut}(A_k, \bar{A}_k)$$

In addition, we have:

$$(F^T L F)_{ii} = \sum_n F_{in}^T \sum_m L_{nm} F_{mi} = \sum_{nm} L_{nm} F_{ni} F_{mi}$$

$$= h_i^T L h_i$$

Thus: $\sum_i h_i^T L h_i = \sum_i (F^T L F)_{ii} = Tr(F^T L F)$

and we arrive at the following statement:

Minimizing RatioCut$(A_1, \ldots, A_k)$

can be rewritten as

$\min_{A_1 \ldots, A_k} Tr(F^T L F)$ such that $F^T F = I$

F defined as previously

which we can again relax to

$$\min_F Tr(F^T L F) \text{ such that } F^T F = I$$

Again, this is like PCA but with min instead of max. So the columns of F are given by the k eigenvectors of L with the smallest eigenvalues (including the 0 eigenvalue)!

Diffusion maps

The first step we have to do is construct a random walk on our data. For this, we have to turn it <u>again</u> into graph form!

Let $x_1, \dots, x_n \in \mathbb{R}^p$ be our n data samples. As for spectral clustering, each data point is represented by a node in the graph, and edges are weighted by the similarity of data points, e.g., $W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$.

<u>$n=3$:</u>

$W_{23} < W_{12} < W_{13}$

Next, we define the transition matrix:

$$M_{ij} = \frac{W_{ij}}{D_{ii}}, \text{ with } D_{ii} = \sum_j W_{ij}$$

and $D_{ij} = 0$ if $i \neq j$

We can write this in a more compact form:

$$M = D^{-1}W$$

This transition matrix represents the probability of moving from node i to node j:

$$P(\text{walk to node } j \mid \text{currently at node } i) = M_{ij}$$

Note: we construct the random walk like this! Here, our random walker prefers walking between data points that are more similar!

There are a few properties:

1) $P(\text{walk to any node} \mid \text{currently at } i) = \sum_j M_{ij} = \frac{\sum_j W_{ij}}{D_{ii}} = 1$

2) The probability to end up at node j when starting at node i and doing t steps is: ← expanded!

$$P(\text{at } j \text{ after } t \text{ steps} \mid \text{start at } i) = (M^t)_{ij}$$

Why? Lets look at $t=2$.

$$(M^2)_{ij} = (M \cdot M)_{ij} = \sum_k \underbrace{M_{ik}}_{\text{jump from } i \text{ to } k} \underbrace{M_{kj}}_{\text{jump from } k \text{ to } j}$$

sum over all k = all possible paths!

Is this still normed?

$$\hookrightarrow \sum_j \sum_k M_{ik} M_{kj} = \sum_k \left(M_{ik} \underbrace{\sum_j M_{kj}}_{=1} \right) = \underbrace{\sum_k M_{ik}}_{=1} = 1$$

Basically, in "every" step we have to end up somewhere $\Rightarrow$ probabilities are conserved.

Starting at node i, we can therefore extract the probabilities of where we end up after t steps:

$$p_{i,t} = \underbrace{e_i^T}_{\text{unit vector}} M^t = \left([M^t]_{i1}, [M^t]_{i2}, \dots, [M^t]_{in} \right)$$

Thus, we assign each data point a vector p_{it} which tells us where a random walk starting at node i typically ends up after t steps.

Insight: If two data points ij are "close", i.e., strongly connected in the graph, then p_{it} and p_{jt} will be similar!

BUT p_{it} and p_{jt} are not low-dimensional! To find such a representation, we will look at the spectrum of M^t.

Spectral decomposition

Note that $M = D^{-1}W$ is, unfortunately, not symmetric. However, the matrix

$$S = D^{1/2} M D^{-1/2}$$

is symmetric, since $S = D^{1/2} M D^{-1/2} = D^{1/2} D^{-1} W D^{-1/2}$

$= D^{-1/2} W D^{-1/2}$

S is symmetric and real-valued, so its spectral decomposition is given by

$$S = V \Lambda V^T \quad \text{with} \quad V^T V = \mathbb{1}$$

$$\Lambda = \text{diag}(\underbrace{\lambda_1, \dots, \lambda_n}_{\text{eigenvalues}})$$

From this, we can recover M:

$$M = D^{-1/2} S D^{1/2} = \underbrace{D^{-1/2} V}_{= \Phi} \Lambda \underbrace{V^T D^{1/2}}_{= \Psi^T}$$

$\Psi = (\Psi_1, \dots, \Psi_n)$

$\Phi = (\Phi_1, \dots, \Phi_n)$

↑ columns = vectors

$$= \Phi \Lambda \Psi^T$$

↑ ↑ rescaled Eigenvectors of S!

Φ and Ψ are bi-orthogonal $\rightsquigarrow \Phi^T \Psi = \mathbb{1}$

or $\Phi_i^T \Psi_j = \delta_{ij}$

Moreover, we have

$$M\phi_k = \phi\Lambda\underbrace{\psi^T\phi_k}_{=e_k} = \phi\Lambda e_k = \phi_k\lambda_k$$

e_k: $(0,\dots,1,\dots,0)$, 1 in k^{th} entry

and similarly: $\psi_k^T M = \underbrace{\psi_k^T\phi}_{e_k^T}\Lambda\psi^T = e_k^T\Lambda\psi^T = \lambda_k\psi_k^T$

$\Rightarrow$ ψ_k^T = left-eigenvector, ϕ_k = right-eigenvector } of M with eigenvalue λ_k

$\Rightarrow$ M can be represented as

$$M = \sum_{k=1}^{n}\lambda_k\underbrace{\phi_k\psi_k^T}_{\text{outer product!}}$$

and $M^t = \sum_{k=1}^{n}\lambda_k^t\,\phi_k\psi_k^T$

Low-dimensional embeddings

We originally aimed for $e_i^T M^t$. We can represent this now using our new eigenbasis:

$$e_i^T M^t = \sum_{k=1}^{n} \lambda_k^t \, e_i^T \phi_k \psi_k^T$$

$$= \sum_{k=1}^{n} \lambda_k^t \, \underbrace{\phi_{k,i}}_{\text{vector components}} \, \underbrace{\psi_k^T}_{\text{eigenbasis}}$$

With this, we define the diffusion map at step t as:

$$Y_{i,t} = \begin{bmatrix} \lambda_1^t \phi_{1,i} \\ \vdots \\ \lambda_n^t \phi_{n,i} \end{bmatrix} \quad \text{for } i = 1, \dots, n$$

We can show that: 1) $\lambda_1 = 1$ and $\phi_1 = \underbrace{1}_{\text{1-vector}}$

2) $|\lambda_k| \leq 1$

$\Rightarrow$ We 1) drop ϕ_1

and for large t we 2) can truncate the vector as $\lambda_k \to 0$!

⇒ **Final embedding**

$$Y_{i,t}^{(d)} = \begin{bmatrix} \lambda_2^t \phi_{2,i} \\ \vdots \\ \lambda_{d+1}^t \phi_{d+1,i} \end{bmatrix}$$

These are our low-dim. embeddings of the data!

We can ground our initial intuition between similarity of random walks and being close in the embedded space some more:

Theorem: For any pair of nodes i, j, we have

$$\|Y_{i,t}^{(n)} - Y_{j,t}^{(n)}\|^2 = \sum_{l=1}^{n} \frac{1}{D_{ll}} \left[P(\text{at } l \text{ after } t \text{ steps} \mid \text{start at } i) - P(\text{at } l \text{ after } t \text{ steps} \mid \text{start at } j) \right]^2$$

$\|Y_{i,t}^{(n)} - Y_{j,t}^{(n)}\|^2$: distance in Eigenspace

$P(\text{at } l \text{ after } t \text{ steps} \mid \text{start at } i) - P(\text{at } l \text{ after } t \text{ steps} \mid \text{start at } j)$: difference of prob. to end up at the same node after t steps.

Proof: write $P(\text{at } l \text{ after } t \text{ steps} \mid \text{start at } i)$

$$= P(Z(t) = l \mid Z(0) = i)$$

First, some rewriting:

$$\sum_{l=1}^{n} \frac{1}{D_{ll}} \left[P(X(t) = l \mid Z(0) = i) - P(Z(t) = l \mid X(0) = j) \right]^2$$

$$= \sum_{l=1}^{n} \frac{1}{D_{ll}} \left[e_i^T M^t e_l - e_j^T M^t e_l \right]^2$$

$$= \sum_{l=1}^{n} \frac{1}{D_{ll}} \left[\sum_{k=1}^{n} \lambda_k^t \left[\phi_{k,i} - \phi_{k,j} \right] \psi_{k,l}^T \right]^2$$

$$= \sum_{l=1}^{n} \left[\sum_{k=1}^{n} \lambda_k^t \left[\phi_{k,i} - \phi_{k,j} \right] \frac{\psi_{k,l}^T}{\sqrt{D_{ll}}} \right]^2$$

$\psi^T = V^T D^{1/2}$

$\Rightarrow \psi_k^T = V_k^T D^{1/2}$

and $\psi_{k,l}^T = V_{k,l}^T \sqrt{D_{ll}}$

$$= \sum_{l=1}^{n} \left[\underbrace{\sum_{k=1}^{n}}_{\text{sum of vectors}} \overbrace{\lambda_k^t \left[\phi_{k,i} - \phi_{k,j} \right] V_{k,l}^T}^{\text{prefac.}} \right]^2$$

ℓ^{th} comp. in sum

$$= \left\| \sum_{k=1}^{n} \lambda_k^t \left[\phi_{k,i} - \phi_{k,j} \right] V_k^T \right\|^2$$

$$= \sum_{k=1}^{n} \left(\lambda_k^t \left[\phi_{k,i} - \phi_{k,j} \right] \right)^2$$

v_k is orthonorm.!

$$\overset{\phi_1 = 1}{=} \sum_{k=2}^{n} \left(\underbrace{\lambda_k^t \phi_{k,i}}_{\gamma_{i,t,k}^{(n)}} - \underbrace{\lambda_k^t \phi_{k,j}}_{\gamma_{j,t,k}^{(n)}} \right)^2$$

$$= \| \gamma_{i,t}^{(n)} - \gamma_{j,t}^{(n)} \|^2 \qquad \square$$

Note on spectral clustering:

Let $L_{rw} = I - D^{-1} W$

and $L_N = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$

If v is an eigenvector of L_{rw} with eigenvalue λ, then $u = D^{1/2} v$ is an eigenvector of L_N with eigenvalue λ.

Proof: $L_N u = \lambda u \iff D^{-1/2} L_N u = \lambda D^{-1/2} u = \lambda v$

and $D^{-1/2} L_N u = D^{-1/2} L_N D^{1/2} v = L_{rw} v$

$\iff L_{rw} v = \lambda v \qquad \square$