Lesson 4

Randomized Low-Rank Factorization [5]

The key idea behind a randomized LRF of a matrix A is that even a contained number of random samples of A already contain enough information to infer the column space and approximated rank of A.

By 'random sampling' we refer to $\underset{n\times m}{A}\,\underset{m\times\tilde{k}}{\Omega}$, where Ω ($m\times m$) is a small random matrix. This produces a smaller $n\times\tilde{k}$ matrix that is easier to manipulate, store and compute. At the same time, given $A=U\Sigma V^T$ the SVD of A, then

$$A\Omega=\sum_{i=1}^{n}\omega_i\cdot\mu_i v_i^*\Omega \quad\Rightarrow\quad (A\Omega)_j=\sum_{i=1}^{n}(\omega_i\cdot v_i^*\Omega_j)\cdot\mu_i$$

if Ω has cont. distr. $\longrightarrow$ and $\mathbb{P}[v_i^*\Omega_j\neq 0]=1$, so with prob. 1 every column of $A\Omega$ contains information about every μ_i that is orth. basis of image(A). Moreover, with high probability $|\omega_i\cdot v_i^*\Omega_j|\sim\omega_i$ and if $\omega_{\tilde{k}}\gg\omega_{\tilde{k}+1}$, then a power method on $A\Omega$ recovers the image.

Using more general $g(A)$ instead of $A\Omega$, 'random sampling' $\rightsquigarrow$ 'processing of small sampled matrix' is the common idea behind most randomized algorithms. Some examples are

- Sparsification or Quantization: Find an approximation of A that is very sparse, or such that all its entries are in a prescribed set. Used to limit storage or to improve matrix-vector mult. time.

- Column Selection: Every A contains a k-column submatrix C s.t.

$$\|A-CC^{+}A\|\leq\sqrt{1+k(n-k)}\,\|A-A_{(k)}\|$$

where $A_{(k)}$ is the k-truncated SVD of A. Finding C is NP-hard, so we need randomized algorithm to find a good C.
A generalization takes rows R, columns C and build a decomp.

$$A\sim CUR \quad \text{where } U \text{ is a small matrix.}$$

We focus on Dimension Reduction Techniques, that are useful also in the case we do not have direct access to A, but we can evaluate $A\cdot x$ and $A^T\cdot x$ for any vector x.

The Randomized Range Finder is described as

Given $A\in\mathbb{R}^{d\times n}$, $\ell>0$ integer, compute the reduced $QR=A\Omega$ with $\Omega\in\mathbb{R}^{n\times\ell}$ random gaussian. Q will be an approx. of a basis of the image of A. Notice that $A\Omega=\mathbb{R}^{d\times\ell}$ is small for $\ell\ll n$.

In this case, if k is the ε-numerical rank of A, we call $p:=\ell-k$ the Oversampling Parameter. Empirically, p should be small since there's no advantage to take $p>5$.

Up to generating cost, the complexity is $O(\ell dn+\ell^2 d)$ where "dn" is actually the cost of matrix-vector mult. Ax.
$\rightsquigarrow$ This is efficient for structured or sparse A or x.

Theorem 8.1 $A\in\mathbb{R}^{d\times n}$, $k\geq 2$, $p\geq 2$, $k+p\leq\min\{d,n\}$.
Then RRF with $\ell=p+n$ on A returns Q such that

$$\mathbb{E}\left[\|(I-QQ^T)A\|\right]\leq\left(1+\frac{4\sqrt{\ell}}{p-1}\sqrt{\min\{d,n\}}\right)\omega_{k+1}$$

Moreover, with prob. $\geq 1-6/p^p$ if $p\leq 4$, $p\log p\leq\min\{d,n\}$

$$\|(I-QQ^T)A\|\leq\left(1+11\sqrt{\ell}\sqrt{\min\{d,n\}}\right)\omega_{k+1}$$

When ω_{k+1} is too large, i.e. A has slowly decaying sv, we can use a Random Power Iteration where we use the 'power' part only $q=4$ times to not fall into inaccuracy building

Given $q=1$ or 2, $A\in\mathbb{R}^{d\times n}$, ℓ so integer, compute the reduced $QR=(AA^*)^q A\Omega$ where $\Omega\in\mathbb{R}^{n\times\ell}$ is random gaussian.
Output Q as approx. basis of the image of A.

The complexity has not changed, but it also loses information of s.v. of A smaller than

$$\omega_i(A)<\mu^{1/2q+1}\|A\|,\quad \mu \text{ machine prec.}$$

so one needs to re-orthonormalize after every multipl. of A, A^T without incrementing the complexity.

Theorem 8.2 Under the same hyp. of Th. 8.1 and Q given by the RPI algorithm, we get

$$\mathbb{E}\left[\|(I-QQ^T)A\|\right]\leq\left(1+\frac{4\sqrt{\ell}}{p-1}\sqrt{\min\{d,n\}}\right)^{1/2q+1}\omega_{k+1}$$

Notice that q drives the error down exponentially fast and if $q\to\infty$, then $\mathbb{E}[\cdot]\sim\omega_{k+1}$.

We said that the algorithms are efficient when it is easy to compute $A\Omega$ and $A^*\Omega$. Since we cannot change A, let's generate a structured Ω instead. In particular

$$\Omega=\sqrt{\frac{n}{\ell}}\;D\;F\;R^T$$

- D: $n\times n$ random diagonal with complex entries unif. distr. on S^1 i.i.d.
- F: Fourier $n\times n$
- R^T: $\in\{0,1\}^{\ell\times n}$ where each row has exactly 1 nonzero and it is full rank

subsampled random Fourier transform (SRFT)

So Ω is a random subset of the columns of F with random phases. The idea is that if $\ell\sim k\log(k)$, then it is improbable that the kernel of Ω contains a fixed k-dim. subsp. Moreover $A\Omega$ is $O(nd\log(\ell))$

Theorem 8.3 For a fixed $V\in\mathbb{R}^{n\times k}$ orth. matrix and Ω SRFT with

$$4\left(\sqrt{k}+\sqrt{8\log(kn)}\right)^2\log(k)\leq\ell\leq n$$

then with probability at least $1-O(1/k)$

$$\sigma_{\min}(V^*\Omega)\geq 0.4 \quad\text{and}\quad \|V^*\Omega\|\leq 1.48.$$

Theorem 8.4 If Q is the output of RRF with Ω SRFT,

$$\|(I-QQ^*)A\|\leq\sqrt{1+\frac{7n}{\ell}}\,\omega_{k+1}$$

with prob. $\geq 1-O(1/k)$ when $\ell\geq k\sqrt{\log k}$.

Notice that the prob. is worse because Ω is a discrete sampling of a deterministic matrix that are worse than continuous ones.
At the same time, one gets a speed up, so you can repeat it multiple times.

There is also a randomized way to see if Q is a good approx. of the image. The deterministic way is $\|(I-QQ^T)A\|<\varepsilon$ but the norm estimation is expensive, mainly because we do not want to form the matrix. A power method could be used, but that would lose accuracy after many multiplications with the same matrix. Instead we can evaluate $B=(I-QQ^T)A$ on random Gaussian vectors since

Lemma 8.5 [Ex] If $B\in\mathbb{R}^{d\times n}$, fix $r>0$ and $\alpha>0$. Then for $w_1,\dots,w_r$ Gaussian vectors i.i.d,

$$\|B\|\leq\alpha\sqrt{\frac{2}{\pi}}\max_{i=1,\dots,r}\|Bw_i\| \quad\text{with prob.}\quad \geq 1-\frac{1}{\alpha^r}$$

and matrix-vector mult. here is very cheap. As a consequence one can double ℓ in the algorithms and test the output without change in the asymptotic complexity.

Once we have obtained Q: $\|(I-QQ^H)A\|<\varepsilon$. We can thus compute a truncated SVD as

Given $A\in\mathbb{R}^{d\times n}$, $Q\in\mathbb{R}^{d\times\ell}$ orth., then let $\underset{\ell\times n}{Q^HA}=\underset{\ell\times\ell}{\tilde{U}}\,\underset{\ell\times n}{\Sigma}\,\underset{n\times n}{V^H}$ be its SVD. The truncated SVD of A is $A\sim(Q\tilde{U})\Sigma V^H=U\Sigma V^H$ ($=QQ^HA$)

We call this Direct SVD and we get $\|A-U\Sigma V^H\|<\varepsilon$ where the cost is dominated by Q^HA. In fact the SVD of $\underset{\ell\times n}{Q^HA}$ is just $O(\ell^2 n)$ while Q^HA is $O(\ell dn)$. This is still less than the full $O(d^2n)$ SVD, but comparable with classic truncated SVD.

With additional techniques (row-extraction) one can get an acceleration of the method as well as algorithms for eigdec. of Hermitian matrices (for example see Nyström method).
In theory one can get down to $O(nd\log(\ell)+\ell^2(m+n))$.
We have seen that the approxim. error is always on the order of $O(n\cdot\omega_{k+1})$ on the worst case, but in general it's way better.
Notice that QR and similar algorithm require $O(ndk)$ to get to the same error, so randomized algorithms are in general faster than that, when $d,n\sim 10^3$, $k\sim 10^2$

K-Means [8]

Given $\{x_1,\dots,x_m\} \subseteq \mathbb{R}^d$ and an integer k, look for $\{c_1,\dots,c_k\}$ centroids in $\mathbb{R}^d$ and a partition of $[n]$ into $S_1,\dots,S_k$ that minimize

$$E_{c,S} := \sum_{i=1}^{k} \sum_{j \in S_i} \|x_j - c_i\|^2$$

(SSE: Sum of Squared Errors)

Given a partition $\{S_1,\dots,S_k\}$ it is easy to see that the minimizing centroids are the average:

$$\operatorname*{argmin}_c \sum_j \|x_j - c\|^2 = \operatorname*{argmin}_c \; n\|c\|^2 - 2c^T \cdot Xe \Rightarrow c = \lambda Xe,\ \lambda > 0$$

$$= \operatorname*{argmin}_\lambda \; n\lambda^2 \|Xe\|^2 - 2\lambda \|Xe\|^2 = \|Xe\|^2 (n\lambda^2 - 2\lambda) \Rightarrow \lambda = 1/n$$

$$\Rightarrow c = Xe/n$$

So the minimum is over a finite (albeit huge) number of partitions. At the same time, if we fix $c_1,\dots,c_k$ it's easy to see that the best partition is to put x_j into S_i if c_i is the closest centroid to x_j.

↝ k-means is the alternating method updating $\{c_i\}$ and $\{S_i\}$ with their optimum values.

K-Means

Given $X \in \mathbb{R}^{d \times n}$ and $0 < k < n$, initialize $\{c_1,\dots,c_k\} \subseteq \mathbb{R}^d$

Repeat until convergence

$$S_j := \{i \mid \operatorname*{argmin}_l \|x_i - c_l\| = j\} \quad \forall j$$

$$c_j = \frac{1}{|S_j|} \sum_{i \in S_j} x_i \quad \forall j$$

Warning: The error $E = \sum_j \sum_{i \in S_j} \|x_i - c_j\|^2$ always decrease, but takes too many it. The configuration $\{S_j\}$ may not stabilize, so it is better to take as stopping criterion the stabilization of E.

It has a complexity of $O(dmk)$ for iteration, but there's no bound on the number of iterations (NP-Hard) and in theory it can go up to n^{kd}, but on practical cases ~ 50 iter.

At the same time, it is likely it gets stuck on local minima and you need several initial guesses for $\{c_i\}$ before finding the good one. Moreover, it cannot deal with non-spherical geometry of the data.

Given the centroids we know how to compute S_i, so

$$E_{c,S} = \sum_j \min_i \|x_j - c_i\|^2$$

$$= \sum_j \|Xe_j - Ce_{F(j)}\|^2$$

$$= \|X - C \cdot H\|_F^2$$

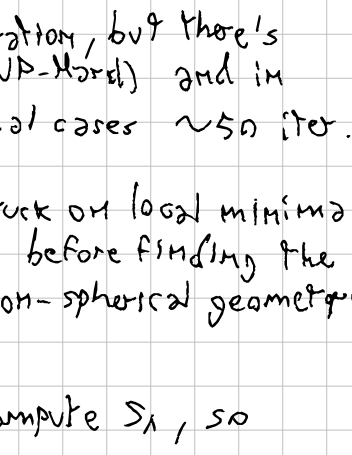
where all columns of H are just canonical basis vectors. As a consequence we are looking for

$$\min_{C,H} \|X - CH\|_F^2 : H \in \{0,1\}^{k \times n},\ C \in \mathbb{R}^{d \times k}$$
$$e^T H = e^T$$

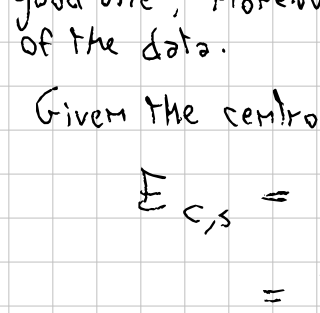
In this case notice that $HH^T = \operatorname{diag}(|S_i|)$.
This is a Constrained Low-Rank Factorization. A common way to relax it is by removing the condition that H must only have 0 and 1 elements. We will see this later..

This can also be seen as an embedding $\mathbb{R}^d \to \mathbb{R}^k$ that sends x_i into h_i so that close x_i, x_j get sent into very close h, and different x_i, x_j get sent into (practically) orthogonal h, or in any case very different ones.

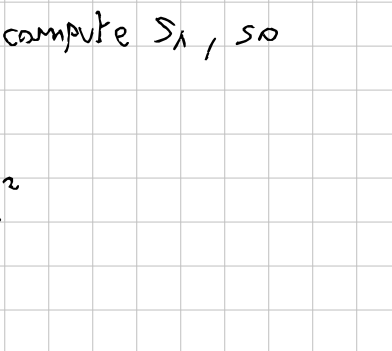
Problems with k-means: It assumes the clusters are spherical around the centroids, and it's also quite dependent on the initialization of the $\{c_i\}$.



Usually one try to adopt a hierarchical system: set a large enough k and then merge the clusters.



And yet, there are conditions where k-means is well behaved, see next theor.



Theorem 9.2 Call $\mathbb{M}$ the set of $n \times k$ matrices with only k distinct rows. Let $\tilde{H}$ be the perturbation of $F \in \mathbb{M}$ whose distinct rows q_i form an orthogonal matrix. Let $n_i = |c_i|$ where c_i are the indices of the rows equal to q_i in F, and suppose that

$$\gamma = \min_{i \neq j} \left\{ \frac{\sqrt{2}}{1/\sqrt{n_i} + 1/\sqrt{n_j}} \right\} \cdot \frac{q}{\min\{\sqrt{k}\|\tilde{H} - F\|, \|\tilde{H} - F\|_F\}} \geq 100.$$

Let $G \in \mathbb{M}$ a 10-approx. of SSE for $\tilde{H}$, i.e.

$$\|\tilde{H} - G\|_F^2 \leq 10 \min_{N \in \mathbb{M}} \|\tilde{H} - N\|$$

and call T_i the clusters associated to the k rows of G. Then

$$\min_\pi \max_i \frac{|T_{\pi(i)} - c_i|}{|c_i|} = O\left(1/\rho^2\right)$$

In a sense, this says that as long as $\frac{\|\tilde{H} - F\|_F}{\min_i \sqrt{n_i}} \to 0$, then the Misclassification Error of k-means goes down faster to 0.

Moreover, this is promising since, even if SSE is NP-Hard, there are algorithms based on k-means that are poly(n) (may be exp in k or d) that gives $(9+\varepsilon)$-approximations.

Lesson 5

Instead of working with X, we fix a distance $d(x)$ on $\mathbb{R}^d$ and work on a

Similarity Matrix: $S_{i,j} = \exp(-c\|x_i - x_j\|^2)$

for some parameter $c > 0$. Since $S_{i,j}$ is inversely proportional to $\|x_i - x_j\|^2$, it thus represent a measure telling us how close are all couples of data. This is $n \times n$ instead of the $d \times n$ matrix X.

What happens is that S is structured like  and it is elementwise positive. This can be read as the adjacency matrix of a weighted undirected graph, where a link has big weight if the two nodes **belong** to the same cluster, but the opposite may not be true.

A way to combinatorically work out what are the clusters is to find, given a fixed k, the MinCut between them.

Ratio Cut: Given a weighted undirected graph with weights W and a partition into k sets $\{C_1, \dots, C_k\}$, the ratio cut of the partition is

$$Rc(\{C_i\}) = \sum_{i=1}^{k} \frac{cut(C_i, C_i^c)}{|C_i|} = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{\substack{j \in C_i \\ l \notin C_i}} w_{j,l}$$

The idea is to minimize the Ratiocut over all partition, since the cut of a single cluster is very low. Notice that without the $|C_i|$ part, it would likely detect single nodes that may be outliers as clusters.

Let h_i be a weighted indicator vector of C_i, i.e. $(h_i)_j = \begin{cases} 1/\sqrt{|C_i|} & j \in C_i \\ 0 & j \notin C_i \end{cases}$ and also let $D = \operatorname{diag}(\underbrace{We}_{d_i})$. One can verify that

$$Rc(\{C_i\}) = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{\substack{j \in C_i \\ l \notin C_i}} w_{j,l} = \sum_{i=1}^{k} \frac{1}{|C_i|} \left(d_i - \sum_{j,l \in C_i} w_{j,l} \right)$$

$$= \sum_{i=1}^{k} h_i^T D h_i - h_i^T W h_i = Tr(H^T (D - W) H)$$

Notice that $H^T H = I$, so the columns are orthogonal and unit. Since optimizing over a quantumized H (i.e. with entries in a set) is NP-Hard, we need to relax the problem, into

$$\min_{H \in \mathbb{R}^{n \times k}} Tr(H^T L H) \quad \text{s.t.} \quad H^T H = I$$

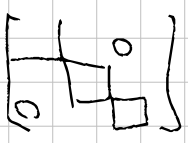
where $L := D - W$ is called the Laplacian matrix of the graph and it has the following properties:

Theorem 9.1 Given a weighted undirected graph with weights W and its Laplacian matrix $L = \operatorname{diag}(We) - W$, then L is spd and

- $\dim \ker(L) = \#$ of connected components $(= k)$
- If $\{C_i\}$ are the connect. comp., then the weighted indicators h_i are an orthogonal basis for $\ker(L)$
- If $\tilde{L}$ is a symmetric perturbation of L with $\{\tilde{h}_i\}$ an orth. basis for the smallest k eigenvalues, then

$$\min_{Q \in O(k)} \|\tilde{H} - HQ\| \leq \frac{\sqrt{2}}{\lambda_{k+1}} \|\tilde{L} - L\|$$

where λ_{k+1} is the $k+1$-smallest eig. of $\tilde{L}$.

The idea is that $L = \left[\begin{array}{c|c} \square & \sim 0 \\ \hline \sim 0 & \square \end{array}\right]$ is a small perturb. of 

where each diagonal block is spd and with sum of rows $= 0$.

(If they have never seen the proof of Th. 9.1, first two points:

$$\sum_{i,j} w_{ij}(x_i - x_j)^2 = \sum_{i,j} w_{ij} x_i^2 + w_{ij} x_j^2 - 2 x_i w_{ij} x_j$$

$$= \sum_i \left(\sum_j w_{ij}\right) x_i^2 + \sum_j \left(\sum_i w_{ji}\right) x_j^2 - 2 x^T W x$$

$$= 2 x^T D x - 2 x^T W x = 2 x^T L x$$

so $x^T L x \geq 0 \ \forall x \Rightarrow L$ spd and $x^T L x = 0 \Rightarrow x_i = x_j \ \forall w_{ij} \neq 0$ i.e. every $x \in \ker(L)$ has entries equal in the same conn. comp.

so $x \in \ker(L) \Rightarrow x \in \operatorname{Span}\{h_1, \dots, h_k\} \subseteq \ker(L)$.

The third point is a modification of Davis-Kahan sinθ theorem we will probably see it later)

Theorem 9.1 says that the last k eigenvectors of L span the same space of H when $\lambda_{k+1} > 0$. This is the so-called **Spectral Gap** of the graph: If L has $\dim \ker L = k$, then (algebraic connectivity) λ_{k+1} is a measure on how well the k components are connected. If it is small, then the graph is close to have more clusters and a bigger k may be correct.

As a consequence, from the EVD of L we get the last k eigenvectors $\tilde{H} \sim HQ = \left[\begin{array}{c|c|c} & & \\ & \square & \\ & & \end{array}\right] \left[\begin{array}{c} q_1^T \\ \vdots \\ q_k^T \end{array}\right] = \left[\begin{array}{c} c_1 q_1^T \\ \hline c_1 q_1^T \\ \hline c_2 q_2^T \\ \vdots \end{array}\right]$ i.e. $\tilde{H}$ has only k different rows repeated and moreover the different rows are orthogonal. This is the best possible embedding we could hope for, since we are mapping $g: x_i \in \mathbb{R}^m \longmapsto q_j \in \mathbb{R}^k$ where x_l, x_m belong to the same clustering iff $g(x_l) = g(x_m)$ and if they don't we get $g(x_l) \perp g(x_m)$. We can thus apply k-means to the rows of $\tilde{H}$

$\longrightarrow$ This is the **Spectral Clustering**

Notice that c_i are just the norm of $c_i q_i^T$, so one can just normalize the rows of $\tilde{H}$ before applying the k-means, and also keep them stored to check afterwards how close $c_i = 1/\sqrt{|C_i|}$ is from the computed $|C_i|$.

Spectral Clustering

Given $X \in \mathbb{R}^{d \times n}$ and $0 < k < n$, $c > 0$
Form $S_{ij} = \exp(-c\|x_i - x_j\|^2)$ simil. matrix
Form the Laplacian $L = \operatorname{diag}(Se) - S$
Compute the last k eigenvectors H of L and normalize the rows
Perform k-means on the rows of H, with starting centroids $\{c_i\} \subseteq \mathbb{R}^k$ forming an orthogonal matrix

Note: It is possible to apply Spectral Clustering to any "similarity" matrix of the data. The other classical ones are

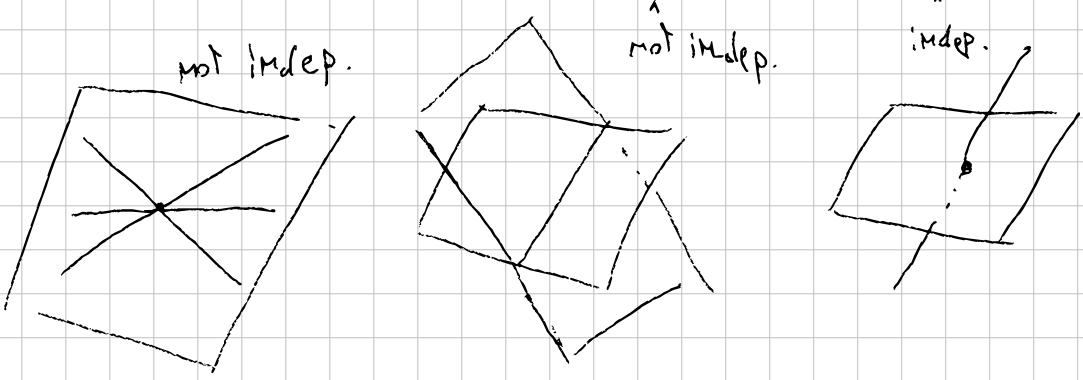
K-NN: Nearest Neighbour, i.e. $S_{ij} = 1$ if j is among the k closest points to i, and $S_{ij} = 0$ otherwise

ε-Neighbourhood: $S_{ij} = 1$ if $\|x_i - x_j\| \leq \varepsilon$, $S_{ij} = 0$ otherwise

An application for Spectral Clustering is the Subspace Clustering that is a particular case of Mixed Models. In particular,

Low Rank Subspace Clustering

Suppose the data matrix $X \in \mathbb{R}^{d \times n}$ has rank K $(\leq d)$ and there exist $\{S_i\}$ independent subspaces such that $x_i \in S_j$ for some j. Here "independent" mean that $\sum_i \dim(S_i) = \dim(\bigoplus_i S_i) = K$,



Theorem 10.1 Let $X = U \Sigma V^T$ the svd of X where $V_1 \in \mathbb{R}^{n \times K}$ ($d \times n$, $d \times d$, $d \times n$, $n \times n$) are the right sin. vec. relative to the k nonzero sv of X. The matrix $C = V_1 V_1^T \in \mathbb{R}^{n \times n}$ has zero entries c_{ij} for x_i, x_j not belonging to the same subspace, and $XC = C$.

Proof $X = U \Sigma V^T = U_1 \Sigma_1 V_1^T = U_1 \Sigma_1 V_1^T V_1 V_1^T = XC$. Up to a perm. Γ, $X = [X_1 \dots X_s]\Gamma$ where X_i are drawn from S_i that has dimension d_i. If $X_i \in \mathbb{R}^{d \times n_i}$, let $A_i \in \mathbb{R}^{n_i \times n_i - d_i}$ an orth. basis for $\ker(X_i)$, i.e. $X_i A_i = 0$ and $A_i^T A_i = 0$. The matrix $A = \Gamma^T(\bigoplus A_i) \in \mathbb{R}^{n \times n-d}$ is thus an orth. basis For $\ker(X) = \ker(\Sigma V^T)$ that is generated by V_2 where $V = [V_1 V_2]$. As a cons., $V_2 = AQ$, Q orth. and $I = VV^T = V_1 V_1^T + V_2 V_2^T = C + AA^T$

$$\Rightarrow \Gamma^T(I - C)\Gamma = I - \Gamma^T C \Gamma = \bigoplus A_i A_i^T \Rightarrow C_{ij} = 0 \text{ if } x_i \in S_p \not\ni x_j \quad \blacksquare$$

As a consequence $C = V_1 V_1^T$ is a block matrix up to permutation and it makes $|C|$ a good similarity matrix to be used in Spectral Clustering to retrieve the subspaces.

- When we want of account for noise, we can model it as $X = XC + E$ with small rank C, so we may solve

$$\min_C \ \text{rank}(C) + \|X - XC\|_F^2 \cdot \tfrac{\lambda}{2} \ ; \ C = C^T$$

or its convex relaxed version

$$\min_C \ \|C\|_* + \|X - XC\|_F^2 \cdot \tfrac{\lambda}{2} \ ; \ C = C^T$$

that has as Unique solution

$$X = U\Sigma V^T \rightsquigarrow C = V\left(I - \tfrac{1}{\lambda}(\Sigma^+)^2\right)V^T$$

- A more accurate model is $X = A + E$ where $AC = C$, $C^T = C$, so

$$\min_{A,C} \ \|C\|_* + \tfrac{\lambda}{2}\|X - A\|_F^2 \ : \ A = AC$$

This is Not convex, but nonetheless it has a close form solution:

If $X = U\Sigma V^T$ and $k = \max\{i : \sigma_i > \sqrt{2/\lambda}\}$, then

$$C = V_k V_k^T, \quad A = U_1 \Sigma_1 V_1^T, \quad E = U_2 \Sigma_2 V_2^T$$

That is just Thresholding of X.

- What about Affine subspaces? The idea is similar: each x_i in subspace S_j is an affine combination of $d_j + 1$ points, i.e.

$$x_i = Xc_i \quad \|c_i\|_0 \leq d_j + 1, \quad c_i^T e = 1$$

and the $d_j + 1$ points can be the same for any x_i in the same S_j so we can find $X = XC$, $\text{rk}(C) = \sum_j (d_j + 1) \ll d$ and $C^T e = e$

$$\rightsquigarrow \min_C \ \|C\|_* + \|X - XC\|_F^2 \cdot \tfrac{\lambda}{2} \ ; \ C^T e = e,$$

with possibly the additional constraint $\text{diag}(C) = 0$ if we want to avoid that x_i is represented by itself.
It can be solved through ADMM methods...

Stochastic Block Model (SBM)

Let G be a random unweighted directed graph (digraph) where each edge may exist or not depending on a Bernoulli r.v. $\mathcal{B}(P_{i,j})$. Suppose that

- The nodes are partitioned in K clusters C_i, $|C_i| = m_i n$
- There is an edge between a node in C_a and C_b with prob. $P_{a,b}$

This is called SBM, but we also need some additional requirement

- $P_{a,b} = f(n) \cdot \theta_{a,b}$ with $f(n) \to 0$, $\max_{a,b} \theta_{a,b} = 1$
- $m = \sum m_i$, $0 < m_{min} \le m_i \le m_{max}$, $rk[\Theta, \Theta^T] = K$
- m_{min}, m_{max}, K, $\Theta = [\theta_{a,b}]$ do not depend on n

If now A_n is its adj matrix (will be random), then $\mathbb{E}[A_n] = [P_{i,j}]$. Up to a permutation, suppose the nodes of A are already clustered, i.e.

$$M_n = \mathbb{E}[A_n] = \begin{bmatrix} P_{1,1} & \cdots & P_{1,K} \\ \vdots & \ddots & \vdots \\ P_{K,1} & \cdots & P_{K,K} \end{bmatrix} = f(n) \begin{bmatrix} \theta_{1,1} & \cdots & \theta_{1,K} \\ \vdots & \ddots & \vdots \\ \theta_{K,1} & \cdots & \theta_{K,K} \end{bmatrix} = f(n) \cdot Z_n \underset{K\times K}{\Theta} Z_n^T$$

where $Z_n = \begin{bmatrix} \vdots & & \vdots \\ \mathbb{1}_{m_1 n} & \cdots & \mathbb{1}_{m_K n} \\ \vdots & & \vdots \end{bmatrix} = \overset{K}{\underset{i=1}{\oplus}} \mathbb{1}_{m_i n} \in \mathbb{R}^{mn \times K}$. If now $\Delta = \mathrm{diag}(\sqrt{m_i n})$

Then $M_n = Z_n \cdot \Delta^{-1} \cdot \left(f(n)\, \Delta \Theta \Delta \right) \Delta^{-1} Z_n^T$ and $Z_n \Delta^{-1}$ has orth. col. so if $U \Sigma V^T = \Delta \Theta \Delta$ is the SVD, we find that

$$M_n = \underset{mn \times K}{(Z_n \Delta^{-1} U)} \cdot \underset{K \times K}{(f(n)\Sigma)} \cdot \underset{K \times mn}{(Z_n \Delta^{-1} V)^T}$$ ~ it's its K-reduced SVD

and

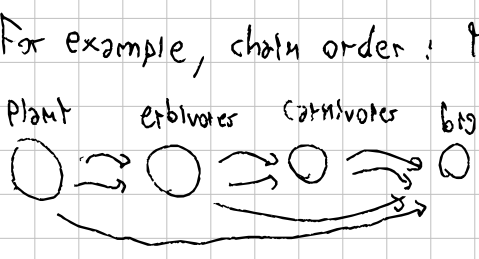
$Z_n \Delta^{-1} U$ has exactly K different rows and they are orthogonal (orthonormal after removing Δ^{-1} practically)

$\rightsquigarrow$ It is enough to apply K-means on $Z_n \Delta^{-1} U$ or $Z_n \Delta^{-1} V$, similar to spec. clust.

Problem: we have A_n, not $M_n = \mathbb{E}[A_n]$. Moreover it would be better to work on a symmetric matrix encompassing both the properties of U, V.

Let's do a step back. We need a 'similarity matrix' on A. Here we try emulate what happens with some real graphs

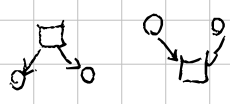
For example, chain order: the edge are the pecking order



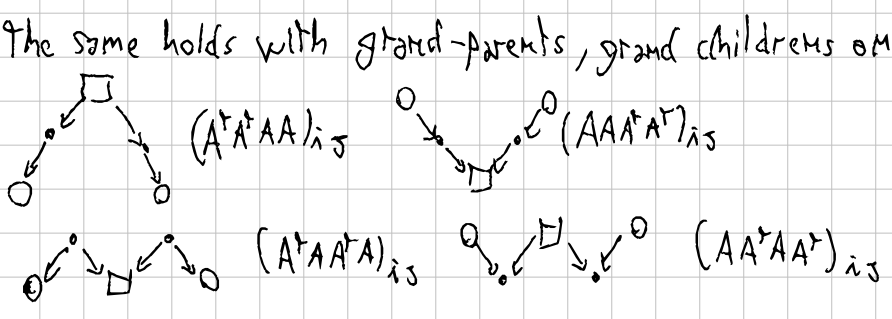
A way to leverage on this directed structure is to design a NPS: Neighbourhood Pattern Similarity Measure

The idea is that

- Two nodes on the same cluster have roughly the same parents and descendents. The number of common parents is $(A^T A)_{ij}$ and common desc. are $(AA^T)_{ij}$



- The same holds with grand-parents, grand childrens and "cousins"



- We can go up a grade and the relations are given by all possible combinations of A and A^T s.t. it is symmetric, etc.

If we now call $L_1 = AA^T + A^T A = [A\ A^T] \begin{bmatrix} A^T \\ A \end{bmatrix}$ we find that

$$L_2 = A^T A^T A A + A A A^T A^T + A A^T A A^T + A^T A A^T A = [A\ A^T] \begin{bmatrix} L_1 & \\ & L_1 \end{bmatrix} \begin{bmatrix} A^T \\ A \end{bmatrix}$$

$$\cdots \quad L_{s+1} = [A\ A^T] \begin{bmatrix} L_s & \\ & L_s \end{bmatrix} \begin{bmatrix} A \\ A^T \end{bmatrix}$$

Given now $\beta^2 \ge 0$, the NPS Measure is the Matrix

$$S_\infty = L_1 + \beta^2 L_2 + \beta^4 L_3 + \beta^6 L_4 + \cdots, \quad S_k = L_1 + \beta^2 L_2 + \cdots + \beta^{2(k-1)} L_k$$

where β^2 scales down the importance of high-degree relatives.
For semplicity let us use just S_1 (i.e. $\beta^2 = 0$), that is also one very well method to symmetrize digraphs: Bibliometric Symmetrization
If we perform the same with $M = \mathbb{E}[A]$ we find

$$T_1 := MM^T + M^T M = [M\ M^T] \begin{bmatrix} M^T \\ M \end{bmatrix} = Z \underbrace{[\Theta\ \Theta^T] \begin{bmatrix} Z^T Z & \\ & Z^T Z \end{bmatrix} \begin{bmatrix} \Theta^T \\ \Theta \end{bmatrix}}_{\hat{T}_1} Z^T f(n)^2$$

$$= (Z\Delta^{-1})(\Delta \hat{T}_1 \Delta)(\Delta^{-1} Z^T) f(n)^2 = (Z\Delta^{-1}U)\, \Lambda\, (Z\Delta^{-1}U)^T f(n)^2$$

($M M^T + M^T M = U \Lambda U^T = U \Lambda U^T$)

and as before, $Z\Delta^{-1}U$ has only r rows orthogonal among them. Moreover $Z^T Z$ is diag. and PD, so $\hat{T}_1$ has rank equal to $rk[\Theta, \Theta^T] = K$.

$\rightsquigarrow$ If we had T_1 we could infer K from its rank and the clustering from an K-mean on its reduced SVD

As before, we need to work with A and $S_4 = AA^T + A^T A$. Luckily, we have probabilistic results telling us it is close to T_4.

NPS

> Given $A \in \mathbb{R}^{n \times n}$ drawn from SBM and $K > 0$
> Form $S_1 = AA^T + A^T A = U \Delta U^T$ K-truncated EVD
> Perform K-means on the rows of U

$\rightsquigarrow$ The real idea behind this is to notice that $Y := A - M$ has $\|Y\| \le 2\sqrt{2}$ Following a Marchenko-Pastur distribution and we can study how it propagates through the computations.

Theorem 11.1 Let $\delta^2 := 4mn\, f(n)$. Then $\|Y\|^2 \le \delta^2$ a.s. and

- $\lambda_K(S_1) \ge \frac{1}{2} \left[\frac{\sigma_K[\Theta \Theta^T]}{4K} \frac{m_{min}}{m_{max}} \right]^2 \delta^4 \qquad 4\delta^2 \ge \lambda_{K+1}(S_1)$

$\rightsquigarrow$ there's a gap that lets us identify the rank K

- $\min_{\pi} \max_{i=1,\dots,K} \frac{|\Upsilon_{\pi(i)} \Delta C_i|}{|C_i|} \le C \cdot \frac{K^5}{\delta^2} \frac{m_{max}^5}{m_{min}^5} \frac{\|[\Theta \Theta^T]\|^2}{\sigma_K([\Theta \Theta^T])^4} = O\left(\frac{1}{\delta^2}\right)$

$\Rightarrow$ It identifies the clusters as long as $\boxed{n \cdot f(n) \to \infty}$. This is actually in line with theoretical bounds for general algorithms applied to SBM. This is useful since in practice one takes $f(n) \to 0$ to regularize the average degree of the graph. The most studied case nowadays is in fact the case $f(n) \sim 1/n$ where there is no theoretical guarantee of perfect recovery [9e]

$\rightsquigarrow$ Notice that for low $\sigma_K[\Theta \Theta^T]$, i.e., when it's almost singular, both the Misclassification Error and the recovery of K are very bad. In fact

$$\Theta \sim \begin{bmatrix} .5 & .5 \\ .5 & .5 \end{bmatrix} \rightsquigarrow A = \text{uniform} \rightsquigarrow 1 \text{ cluster}$$

Spectral Methods are far to be the only methods to apply for SBM or clustering in general. There are countless others based on combinatorial processes (usually expensive), randomized procedures, convex optimization, Hierarchical schemes, etc. You can find a list of references in [10e] (non comprehensive)

Moreover, there are tons of Spectral Methods, starting with regularized to preprocessed graphs ones (for dealing with small hubs, etc.)

- The SBM is a general case of more particular problem like the Planted Partition Problem, where $P_{i,j} = \begin{cases} p & i = j \\ q & i \ne j \end{cases}$ and the Erdős-Rényi model where $P_{i,j} = p\ \forall i,j$.