Swedish Summer School

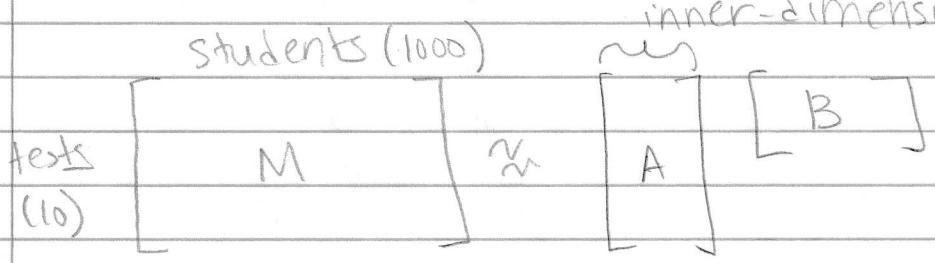
Part I: Tensor Decompositions

Charles Spearman (1904): There are two types of intelligence

(1) eductive : make sense out of complexity

(2) reproductive : store and reproduce info

To test his theory, invented factor analysis



Setup: Given $M = \sum_{i=1}^{r} a_i b_i^T$

$= AB^T$

"correct" factors

However for any rotation R, we have

$M = (AR)(R^{-1}B^{-1})$

alternative factorization

Q: when can we find the factors $\{a_i\}$ and $\{b_i\}$ uniquely?

e.g. up to the trivial rescaling

$$a_i \leftarrow \alpha a_i$$
$$b_i \leftarrow \frac{1}{\alpha} b_i$$

and permutation

Claim: the factors $\{a_i\}$ and $\{b_i\}$ are not uniquely determined, unless we impose additional conditions on them

e.g. if $\{a_i\}$ are orthogonal, same for $\{b_i\}$

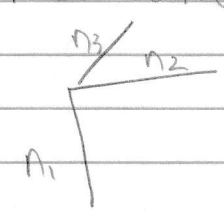
or if rank$(M)=1$

This is called the rotation problem and is a major issue in factor analysis

It motivates the study of tensor methods

Tensor Decompositions: Definitions

A third-order tensor T has three dimensions, sometimes called rows, columns, tubes



The size of T is $n_1 \times n_2 \times n_3$

<u>definition</u>: A <u>rank one</u> third order tensor T is the <u>tensor product</u> of three vectors u, v and w and its entries are

$$T_{i,j,k} = u_i v_j w_k$$

Also written as $T = u \otimes v \otimes w$

The <u>rank</u> of T is the smallest r s.t.

$$T = \sum_{i=1}^{r} u^{(i)} \otimes v^{(i)} \otimes w^{(i)}$$

We can view a tensor T as a stacked collection of matrices

$$T_1 = T_{(:,:,1)}, \; T_2 = T_{(:,:,2)}, \text{ etc}$$

<u>Claim</u>: If $\text{rank}(T) \leq r$ then for all a, $\text{rank}(T_a) \leq r$ too

This follows from the definition of rank, since

$$T = \sum_{i=1}^{r} u^{(i)} \otimes v^{(i)} \otimes w^{(i)} \implies$$

$$T_a = \sum_{i=1}^{r} w_a^{(i)} \left(u^{(i)} \otimes v^{(i)} \right)$$

However a low-rank tensor is not just a collection of arbitrary low rank matrices

Lemma: Consider a rank $\leq r$ tensor T with

$$T = \sum_{i=1}^{r} u^{(i)} \otimes v^{(i)} \otimes w^{(i)}$$

Then for all a we have

$$\text{colspan}(T_a) \subseteq \text{span}(\{u^{(i)}\})$$

$$\text{rowspan}(T_a) \subseteq \text{span}(\{v^{(i)}\})$$

Proof Left as exercise

Intuitively we have

matrix $\triangleq$ one "view" of vectors $\{u^{(i)}\}$ and $\{v^{(i)}\}$

tensor $\triangleq$ multiple "views"

and this is how we will solve the notation problem

The Trouble with Tensors

Many of the properties we know and love about matrices will break for tensors, e.g.

For any matrix M, we have:

$$\text{rank}(M) = \dim(\text{colspan}(M)) = \dim(\text{rowspan}(M))$$

Does this type of relation hold for tensors? No!

Claim: The rank of a tensor depends on the field you are working over.

e.g. Consider

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

It can be shown that

$$\text{rank}_{\mathbb{R}}(T) \geq 3$$

↑ only real values are allowed for the factors

However $\text{rank}_{\mathbb{C}}(T) = 2$; in particular

$$T = \frac{1}{2}\left(\begin{bmatrix} 1 \\ -i \end{bmatrix} \otimes \begin{bmatrix} 1 \\ i \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -i \end{bmatrix} + \begin{bmatrix} 1 \\ i \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -i \end{bmatrix} \otimes \begin{bmatrix} 1 \\ i \end{bmatrix} \right)$$

Even though T is real-valued, can use fewer rank one tensors if we use complex numbers

Claim: There are tensors of rank 3, but which are arbitrarily close to tensors of rank 2

Definition: The border rank of T is the minimum r s.t. $\forall \varepsilon > 0$ $\exists$ a tensor S of rank at most r s.t. T and S are ε-close entry-wise

e.g. consider $T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$

It can be shown that $\text{rank}_{\mathbb{R}}(T) = 3$.

Yet it admits an arbitrarily good rank 2 approximation. Let

$$S_n = \begin{bmatrix} n & 1 \\ 1 & \frac{1}{n} \end{bmatrix}, \begin{bmatrix} 1 & \frac{1}{n} \\ \frac{1}{n} & \frac{1}{n^2} \end{bmatrix}$$

$$R_n = \begin{bmatrix} n & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$\text{rank}_{\mathbb{R}}(S_n - R_n) = 2$ and yet

$S_n - R_n$ and T are $\frac{1}{n}$-close entry-wise

Exercise: Use Eckhart-Young to show that this cannot happen for matrices, i.e. rank and border rank are the same

For matrices, Consider the best rank k approximation to M in Frobenius norm

$$M_k = \text{best rank}_k(M)$$

Then we have

$$\text{best rank}_{k-1}(M) = \text{best rank}_{k-1}(M_k)$$

Claim: For tensors, the best rank k-1 approximation may not share any common factors with the best rank k approximation

For me, the root of those problems is computational

Theorem [Hastad] It is NP-hard to compute the rank of a tensor

So of course it cannot be equal to the dimension of the span of its rows, etc

[Hillar, Lim] showed a laundry list of tensor problems are NP-hard

"Most Tensor Problems are Hard"

We'll explore an important (previously forgotten) positive result and its myriad applications

The Harshman-Jennrich Algorithm

This algorithm has been rediscovered many times, originates in psychometrics

Setup: We are given T, assumed to be of the form

$$T = \sum_{i=1}^{r} u^{(i)} \otimes v^{(i)} \otimes w^{(i)}$$

definition. We say two sets of factors

$$\{(u^{(i)}, v^{(i)}, w^{(i)})\} \text{ and } \{(\hat{u}^{(i)}, \hat{v}^{(i)}, \hat{w}^{(i)})\}$$

are equivalent if there is a permutation π s.t.

$$u^{(i)} \otimes v^{(i)} \otimes w^{(i)} = \hat{u}^{\pi(i)} \otimes \hat{v}^{\pi(i)} \otimes \hat{w}^{\pi(i)}$$

i.e. the produce equivalent low rank decompositions

Main Question when are the factors of T determined up to equivalence?

Theorem [Harshman, Jennrich] suppose the following conditions hold

(1) The vectors $\{u^{(i)}\}$ are linearly indep.

(2) same for $\{v^{(i)}\}$

(3) every pair of vectors in $\{w^{(i)}\}$ are linearly indep.

Then the factors are uniquely determined up to equivalence, and there is a polynomial time algorithm to find them

The algorithm is simple

- Choose $a, b \in S^{n_3}$ uniformly at random, set

$$T_a = \sum_{i=1}^{n_3} a_i \cdot T_{(\cdot,\cdot,i)} \; ; \; T_b = \sum_{i=1}^{n_3} b_i \cdot T_{(\cdot,\cdot,i)}$$

- Compute the eigen decompositions of

$$T_a (T_b)^+ \text{ and } (T_a^+ T_b)^T$$

Let $\hat{u}$ and $\hat{v}$ be eigenvectors with non-zero eigenvalue

Pair up $\hat{u}^{(i)}$ and $\hat{v}^{(j)}$ iff their eigenvalues are reciprocals

- Solve for $\hat{w}^{(i)}$ in the linear system

$$T = \sum_{i=1}^{r} \hat{u}^{(i)} \otimes \hat{v}^{(i)} \otimes \hat{w}^{(i)}$$

The analysis follows by tracking the structure of T through the algorithm, and using standard uniqueness facts about eigen decomp.

Let $D_a = \mathrm{Diag}(a^T w^{(i)})$; $D_b = \mathrm{Diag}(b^T w^{(i)})$

Lemma: we have that

$$T_a = U D_a V^T \text{ and } T_b = U D_b V^T$$

Proof: Since the operation of computing T_a from T is linear, we can do it just for a rank one term:

If $T = u \otimes v \otimes w$ then $T_a = (a^T w)\, u \otimes v$

Thus, in general, we have

$$T_a = \sum_{i=1}^{r} (a^T w^{(i)})\, u^{(i)} \otimes v^{(i)} = U D_a V^T \quad \blacksquare$$

For simplicity, let's assume T_a and T_b are invertible. Then

$$T_a T_b^{-1} \overset{\text{Lemma}}{=} U D_a V^T (V^T)^{-1} D_b^{-1} U^{-1}$$

$$= U D_a D_b^{-1} U^{-1}$$

From property (3), almost surely the diag. entries of

$$D_a D_b^{-1}$$

will be distinct. Thus $T_a T_b^{-1}$ has distinct eigenvalues $\Rightarrow$ its eigendecomposition is unique $\Rightarrow$ we can find U, up to a permutation/rescaling of its columns

Similarly we have

$$(T_a^{-1} T_b)^T = (V^T)^{-1} D_a^{-1} U^{-1} U D_b V^T)^T$$

$$= ((V^T)^{-1} D_a^{-1} D_b V^T)^T$$

$$= V D_a^{-1} D_b V^{-1}$$

Thus we can determine V, up to a permutation/rescaling of its columns, again from uniqueness of the eigendecomp.

Moreover pairing succeeds, again because the diagonal entries of $D_a^{-1} D_b$ are distinct

Finally, we show:

Lemma: The matrices $u^{(i)} \otimes v^{(i)}$ are linearly independent

Proof: Suppose not. Then $\exists \alpha_i$'s s.t.

$$\sum_{i=1}^{r} \alpha_i u^{(i)} \otimes v^{(i)} = 0 \quad \text{Suppose WLOG } \alpha_1 \neq 0$$

By condition (1), we know $\exists x$ s.t.

$$x^T u^{(1)} \neq 0, \; x^T u^{(i)} = 0 \; \forall i \neq 1$$

Now using the identity above, we get

$$(\alpha_i - a^T u^{(i)})\, v^{(i)} = 0 \Rightarrow v^{(i)} = 0$$

which contradicts condition (2). ∎

Why was this algorithm forgotten?

Psychometrics generally cared about uniqueness, and there are better "non-algorithmic" uniqueness theorems known

Now returning to factor analysis

Given: $T = \sum_{i=1}^{r} u^{(i)} \otimes v^{(i)} \otimes w^{(i)}$, when are the factors determined up to equivalence?

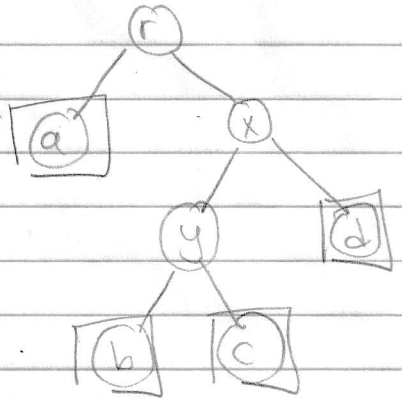
Harshman-Jennrich: when $\{u^{(i)}\}$ and $\{v^{(i)}\}$ are linearly indep, and no pair in $\{w^{(i)}\}$ are scalar multiples of each other

Applications to learning

Next we'll study applications of tensor decomp. to learning latent variable models

Phylogenetic Trees

Setup: There is a rooted binary tree with root r



○ = extinct

□ = extant

An alphabet Σ of states, e.g. $\Sigma = \{A, C, G, T\}$, and a markov model consisting of

(1) An initial distribution $\pi : \Sigma \to \mathbb{R}^{\geq 0}$ for the symbol at the root

(2) Along each edge, a conditional distribution

$$P_{ij}^{uv} = \mathbb{P}[s(u) = j \mid s(u) = i]$$

Can we recover the model from the joint distribution on the extant nodes?

Our algorithm will work in two phases

(1) recover the topology of the tree

Sometimes called "tree of life", identifies speciation events and ancestral relationships

(2) estimate the markov parameters

Evolutionary Distance

Steel introduced a distance function between pairs of nodes on the tree with the properties

(a) it is nonnegative

(b) it can be evaluated for any pair, just given their joint distribution

Definition: Steel's evolutionary distance on an edge (u,v) is

$$\psi_{uv} = -\ln\left|\det\left(P^{uv}\right)\right| - \frac{1}{2}\ln\left(\prod_{i=1}^{k}\pi_u(i)\right) + \frac{1}{2}\ln\left(\prod_{i=1}^{k}\pi_v(i)\right)$$

P^{uv}: conditional distribution $u \to v$

$\pi_u(i)$: 0 if leaf

$\pi_v(i)$: marginal on v

Assuming all P^{uv}'s are full rank,

Lemma: Steel's evolutionary distance satisfies

(1) ψ_{uv} is nonnegative

(2) for any pair (a,b) we have

$$\psi_{ab} \triangleq -\ln|\det(F^{ab})| = \sum_{(u,v)\in P_{ab}} \psi_{uv}$$

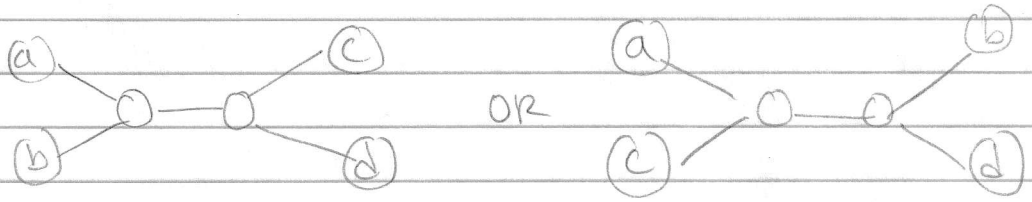
(F^{ab}: joint distribution on (a,b))

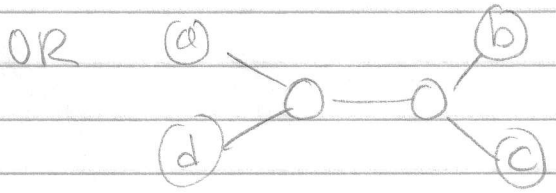
where P_{ab} is the path connecting a and b

[Erdos, Steel, Szekely, Warnow] used Steel's distance and quartet tests to reconstruct the topology

Lemma: If all distances are strictly positive, it is possible to determine the induced topology on any four nodes given an oracle for computing pairwise distances

Proof: there are three possible induced topologies





i.e. if we delete edges not on shortest path, and contract paths to a single edge

Easy to check we're in first case iff.

$$d_{a,b} + d_{c,d} < \min\{d_{a,c} + d_{b,c},\ d_{a,d} + d_{b,d}\}$$

Similar relation holds in other cases ∎

Lemma: If for any quadruple we can determine the induced topology, we can determine the overall topology

Proof: Strategy: determine which pairs of leaves are siblings, and so on

To do this, fix a pair (a,b). They are siblings iff for every other pair (c,d), the quartet test on $\{a,b,c,d\}$ determines we're in the first case

Now to continue, (a,b) are non-siblings but share a grandparent iff for any pair (c,d) neither of which is a sibling of a or b, the quartet test on $\{a,b,c,d\}$ determines we're in first case.

And so on. ∎

Note: we can only estimate $d_{a,b}$ from samples, but [Erdos et al] show how to only use

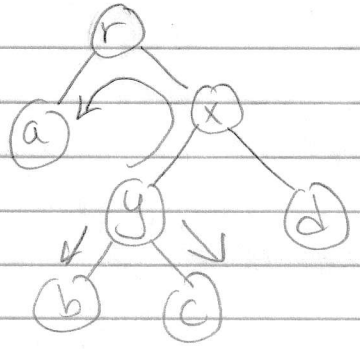
quartet tests when distances are small

Getting the Transition Matrices

If we know the topology, how can we estimate the P^{uv}'s?

Claim: If the P^{uv}'s are full rank, we can reroot the tree arbitrarily

Now consider any triple (a,b,c) of leaves, e.g.



star test

reroot the tree at y and consider the joint distribution on (a,b,c)

$$T^{abc}_{ijk} = \mathbb{P}[s(a)=i, s(b)=j, s(c)=k]$$

Then we have

$$T^{abc} = \sum_{\ell} \mathbb{P}[s(y)=\ell]\,\mathbb{P}[s(a)=\cdot \mid s(y)=\ell]$$
$$\mathbb{P}[s(b)=\cdot \mid s(y)=\ell]\,\mathbb{P}[s(c)=\cdot \mid s(y)=\ell]$$

But notice that

$$\mathbb{P}[s(b) = \cdot \mid s(y) = l] = l^{th} \text{ row of } P^{yb}$$

$$\mathbb{P}[s(c) = \cdot \mid s(y) = l] = l^{th} \text{ row of } P^{yc}$$

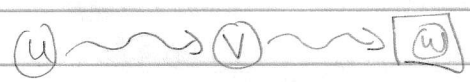
$$\mathbb{P}[s(a) = \cdot \mid s(y) = l] = l^{th} \text{ row of } P^{yx} P^{xr} P^{ra}$$

Thus we have that

(1) T^{abc} is a low rank tensor that can be computed from samples

(2) It's tensor decomposition determines properties of the parameters of the model

We can also determine the internal transitions up to equivalence: E.g. suppose



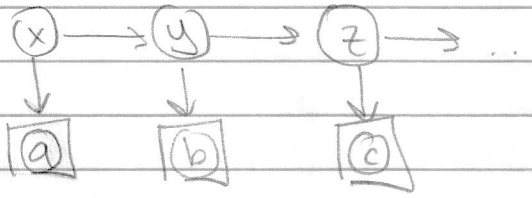
we can find P^{uw} and P^{vw} using star tests, and then solve for

$$P^{uw} = P^{uv} P^{vw} \Rightarrow P^{uv} = P^{uw} (P^{vw})^{-1}$$

[Mossel, Roch] show how to recover transition matrices using only short paths, in Valiant's PAC model

Hidden Markov Models

Setup: A hidden Markov model is given by



where Σ_s and Σ_o are alphabets on hidden and observed states respectively.

Moreover let P^{xy} be the transition matrices and O^{xa} be the observation matrices.

$$P^{xy}_{ij} = \mathbb{P}[s(y)=j \mid s(x)=i]$$
$$O^{xa}_{ij} = \mathbb{P}[s(a)=j \mid s(x)=i]$$

Theorem [Mossel, Roch] There is a polynomial time algorithm for learning HMMs when the observation/transition matrices are full rank and have lower bounded smallest singular value.

What about when the observation matrices are not full rank?

Consider the noisy parity problem

setup: Let $S \subseteq [n]$ and for each sample, choose $X \in_{\text{unif}} \{0,1\}^n$ and set

$$y = \begin{cases} \chi_S(x) \triangleq \sum_{i \in S} x_i \bmod 2 & \text{w/ prob } 2/3 \\ 1 - \chi_S(x) & \text{o.w} \end{cases}$$

Theorem. [Blum, Kalai, Wasserman] there is a $2^{n/\log n}$ time algorithm for finding S, i.e. the hidden noisy parity

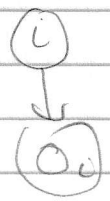
This is the best known algorithm, and noisy parity is widely believed to be hard

Let's embed noisy parity into an HMM

$$(i) = (x_i, p_i)$$

x_i: i^{th} bit; p_i: running parity, i.e. $\sum_{\substack{j \in S \\ j \le i}} x_j \pmod 2$

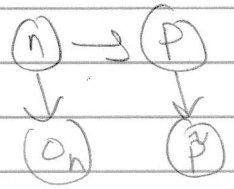
Then the observation



$o_i = x_i$

we get the i^{th} bit. This matrix is 4×2, and hence not full rank

Finally in the last step



we set $P = P_n$, and get $\tilde{P} = \begin{cases} P & \text{w/ prob } 2/3 \\ 1-P & \text{o.w.} \end{cases}$

Theorem [Mossel, Roch]: Learning general HMMs (i.e. without full rank observation matrices) is as hard as noisy parity

their proof uses the simulation above, and also the self-reducibility property of noisy parity, i.e. it's enough to distinguish noisy parity from the case when the label is random

Historical note: In phylogenetics, tensor methods are sometimes called Chang's lemma

Mixtures of Spherical Gaussians

Another powerful application: Mixtures of spherical Gaussians

def: A Gaussian with mean μ and covariance Σ has pdf

$$N(\mu, \Sigma, x) = \frac{e^{\frac{-(x-\mu)^T \Sigma^{-1} (x-\mu)}{2}}}{(2\pi)^{d/2} \det(\Sigma)^{1/2}}$$

In the special case where $\Sigma = I$, i.e. spherical, we get

$$N(\mu, I, x) = \frac{e^{-\frac{\|x-\mu\|^2}{2}}}{(2\pi)^{d/2}}$$

Setup: We get samples from a mixture model

$$M = \sum_{i=1}^{k} w_i \, N(\mu_i, \sigma^2 I, x)$$

(w_i: mixing weight)

Can we learn the parameters/cluster?

Theorem [Hsu, Kakade] There is an algorithm with polynomial running time/sample complexity when the μ_i's have full rank

Note: The running time/sample complexity depend on $1/w_{min}$, $1/\sigma_{min}$, etc.

Lemma: If σ^2 is known, then the tensor

$$T = \sum_{i=1}^{k} w_i \, \mu_i^{\otimes 3}$$

can be expressed through moments of the mixture

Observe that this is the same recipe as before.

(1) There is a low rank tensor that can be estimated from samples

(2) It's tensor decomposition determines the parameters of the model

Proof: Consider T_{abc}

Case #1: a, b, c are distinct

Then $\underset{\mu}{\mathbb{E}}[x_a x_b x_c] = \sum_{i=1}^{k} w_i (\mu_i)_a (\mu_i)_b (\mu_i)_c$

This follows because the noise is independent across coordinates.

Written another way

$$\underset{\mu}{\mathbb{E}}[x_a x_b x_c] = \sum_{i=1}^{k} w_i \left(\mu_i^{\otimes 3}\right)_{a,b,c}$$

Case #2 $a = b \neq c$

Then we have

$$\underset{\mu}{\mathbb{E}}[x_a x_b x_c] = \sum_{i=1}^{k} w_i \left((\mu_i)_a^2 + \sigma^2\right)(\mu_i)_c$$

$$= \sum_{i=1}^{k} w_i \left(\mu_i^{\otimes 3}\right)_{abc} + \sigma^2 \underbrace{\sum_{i=1}^{k} w_i (\mu_i)_c}_{\text{first moment}}$$

Case #3 $a=b=c$

Then we have

$$\underset{\mu}{\mathbb{E}}[x_a x_b x_c] = \sum_{i=1}^{k} w_i (\mu_i^{\otimes 3})_{abc} + 3\sigma^2 \Big(\underbrace{\sum_{i=1}^{k} w_i \mu_i}_{\text{first moment}} \Big)_c$$

This follows because in one-dimension

$$\underset{x \leftarrow N(\mu, \sigma^2)}{\mathbb{E}}[x^3] = \mu^3 + 3\mu\sigma^2$$

Rearranging the inequalities, we get

$$T = \underset{\mu}{\mathbb{E}}[x^{\otimes 3}] - \sigma^2 \sum_{j=1}^{d} M_j$$

where $M_j = (\mathbb{E}[x] \otimes e_j \otimes e_j + e_j \otimes \mathbb{E}[x] \otimes e_j + e_j \otimes e_j \otimes \mathbb{E}[x])$

Now we can use tensor decomp.

Further Applications

In a topic model we have

(1) there are k topics $A_1, \ldots, A_k$ each associated with a distribution on words

(2) For each document, choose a single topic (pure topic model) or a mixture on topics (general topic models) and sample words from the associated distribution

Given many (short) documents, can we reconstruct the topics?

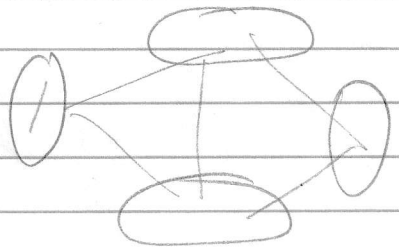
[Anandkumar, Hsu, Kakade] gave a polynomial time algorithm for pure topic models based on the identity

$$T_{ijk} \triangleq \mathbb{P}[\text{first three words are } (i,j,k) \text{ resp}]$$

$$T = \sum_{j=1}^{k} \mathbb{P}[\text{topic} = j]\, A_j \otimes A_j \otimes A_j$$

Again, T can be estimated from samples, and we can use tensor decomp.

In community detection, we have a hidden clustering



with a $k \times k$ matrix R describing the connection probabilities.

Now we can partition the nodes into four sets A, B, C, X and let

$\Pi \in \{0,1\}^{n \times k}$ = assignment of nodes to communities

Then the i^{th} column of ΠR denotes the probability a node in community i has an edge to each of the rest of the nodes

Now let $(\Pi R)_i^A$ denote the restriction of the i^{th} column to nodes in A, and consider

$$T = \sum_{i=1}^{k} p_i \, (\Pi R)_i^A \otimes (\Pi R)_i^B \otimes (\Pi R)_i^C$$

p_i: fraction of nodes of community i in X

Lemma: $T_{abc} = \mathbb{P}[(x,a),(x,b),(x,c) \in E]$ "three star"

for a random $x \in X$, and over the randomness of G

Again, we have the same recipe

(1) Estimate a low rank tensor from counting three stars

(2) Use tensor decomp. to recover the community memberships