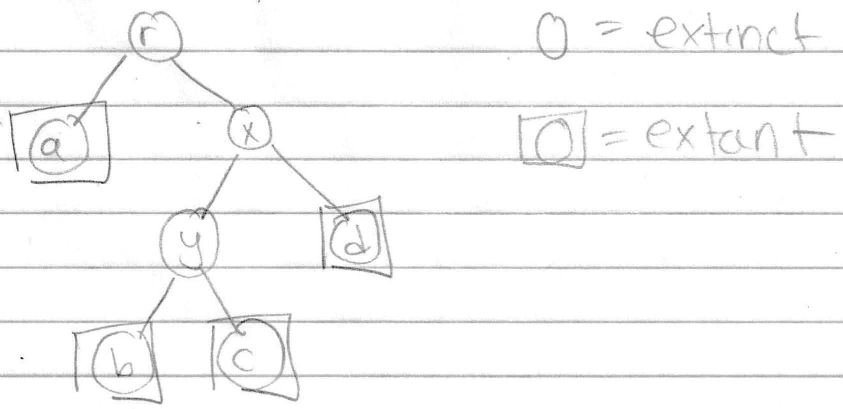
Applications to Learning

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



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

$$\psi_{uv} = -\ln|\det(\underbrace{P^{uv}}_{\text{conditional distribution } u \to v})| - \frac{1}{2}\ln\left(\prod_{i=1}^{k}\pi_u(i)\right) + \frac{1}{2}\ln\left(\prod_{i=1}^{k}\underbrace{\pi_v(i)}_{\text{marginal on } v}\right)$$

0 if leaf

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

Lemma: Steel's evolutionary distance satisfies

(1) $\mathcal{D}_{uv}$ is nonnegative

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

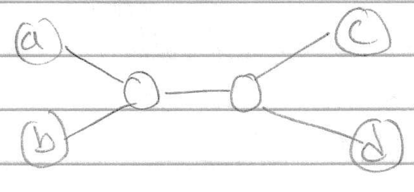
$$\mathcal{D}_{ab} \overset{\Delta}{=} -\ln|\det(\underset{\uparrow \text{ joint distribution on } (a,b)}{F^{ab}})| = \sum_{(u,v)\in \mathcal{P}_{ab}} \Psi_{uv}$$

where $\mathcal{P}_{ab}$ is the path connecting a and b

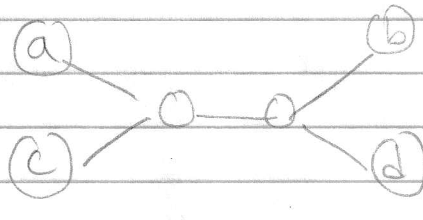
[Erdos, Steel, Szekely, Warnow] used Steel's distance and quartet tests to reconstruct the topology

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

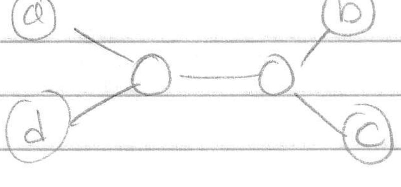
Proof: there are three possible induced topologies



OR



OR



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

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

Similar relation holds in other cases $\boxtimes$

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. $\boxtimes$

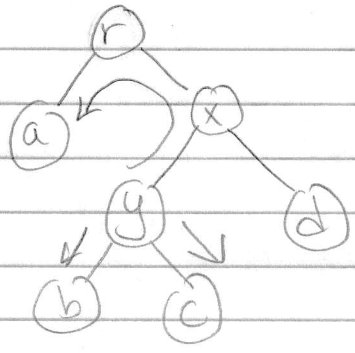
Note: We can only estimate $\nu_{a,b}$ from samples, but [Erdos et al] show how to only use quartet tests when distances are small

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

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

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

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

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

$$\begin{tikzcd} \textcircled{u} \arrow[r, rightsquigarrow] & \textcircled{v} \arrow[r, rightsquigarrow] & \boxed{\textcircled{w}} \end{tikzcd}$$

we can find P^{uw} and P^{vw} using <u>star tests</u>, 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 <u>PAC model</u>

Hidden Markov Models

Setup: A hidden markov model is given by

$$\begin{tikzcd} x \arrow[r] \arrow[d] & y \arrow[r] \arrow[d] & z \arrow[r] \arrow[d] & \cdots \\ \boxed{a} & \boxed{b} & \boxed{c} & \end{tikzcd}$$

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 $X \in_{uar} \{0,1\}^n$ and set

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

<u>Theorem</u>: [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

$$\textcircled{i} = (\underset{\substack{\uparrow \\ i^{th} \text{ bit}}}{x_i}, \underset{\substack{\uparrow \\ \text{running parity, i.e.}}}{p_i})$$

$$\sum_{\substack{j \in S \\ j \le i}} x_j \pmod 2$$

Then the observation

$$\begin{tikzcd} \textcircled{i} \arrow[d] \\ \textcircled{O_i} \end{tikzcd} \qquad O_i = x_i$$

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

Finally in the last step

$$\begin{tikzcd} n \arrow[r] \arrow[d] & p \arrow[d] \\ O_n & \tilde{p} \end{tikzcd}$$

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

$$M = \sum_{i=1}^{k} \underset{\substack{\uparrow \\ \text{mixing weight}}}{w_i} N(\mu_i, \sigma^2 I, x)$$

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

This follows because the noise is independent across coordinates.

Written another way

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

Case #2 $a = b \neq c$

Then we have

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

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

Case #3 $a=b=c$

Then we have

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

this follows because in one-dimension

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

$\boxed{\mathbb{E}}$

Rearranging the inequalities, we get

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

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

$+ e_j \otimes e_j \otimes \mathbb{E}[x]\big)$

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?

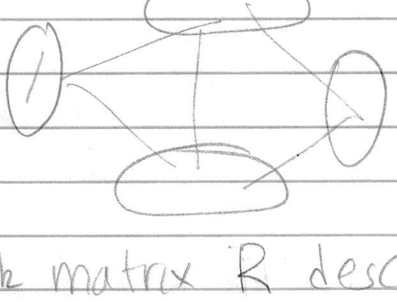
[Anandkumar, Hsu, Kakade] gave a polynomial time algorithm for pure topic models based on the identity

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

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

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

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

Then the ith 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 ith column to nodes in A, and consider

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

fraction of nodes of community i in X

<u>Lemma</u>: $T_{abc} = \mathbb{P}[\underbrace{(x,a),(x,b),(x,c) \in E}_{\text{"three star"}}]$

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