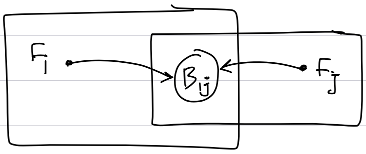


* B_{ij} متغیر تصادفی است که وزن رابطه نود i و j را نشان می دهد
* F_k بردار ویژگی نود k ام (سطر k ام ماتریس F)

Hidden یکی مشاهده شده و یکی

$$\mathcal{L} = p(B \mid f) = \sum_{i,j:i<j} \log p(B_{ij}) = \sum_{i,j:i<j} B_{ij} \log \theta_{ij} - \theta_{ij} - \log(B_{ij})! = \sum_{i,j:i<j} B_{ij} \log F_i^T F_j - F_i^T F_j - \log(B_{ij})!$$

$$\nabla_{F_k} \mathcal{L} = \sum_{j=1}^{k-1} \left(B_{ik} \frac{F_i}{F_i^T F_k} - F_i \right) + \sum_{i=k+1}^{N} \left(B_{ki} \frac{F_i}{F_i^T F_k} - F_i \right) \xRightarrow[B_{ji} = B_{ij}]{} \nabla_{F_k} \mathcal{L} = \sum_{i=1}^{N} \left(B_{ik} \frac{F_i}{F_i^T F_k} - F_i \right)$$

$B'_{ij} = F_i^T F_j$ مقدار hidden تخمینی

* مقادیر از B_{ij} که دیده شده با مقدار A_{ij} و مقادیر دیده نشده با Expectation به دست می آید!

بنابراین فرمول به روز رسانی داریم:

f مربوط به نود k ام، در مرحله t ام EM و m+1 ام گرادیان

آخرین مقدار مرحله t-1 الگوریتم EM

$$f_k^{t,m+1} = f_k^{t,m} + C_k \Bigg\{ \underbrace{\sum_{i \in H}}_{\text{hidden}} \left(f_i^{t-1,\infty\,T} f_k^{t-1,\infty} \frac{f_i^{t,m}}{f_i^{t,m\,T} f_k^{t,m}} - f_i^{t,m} \right)$$

$$\underbrace{\sum_{i \in O}}_{\text{Observe}} \left(A_{ik} \frac{f_i^{t,m}}{f_i^{t,m\,T} f_k^{t,m}} - f_i^{t,m} \right) \Bigg\}$$

* B_{ij} دو حالت دارد، اگر مشاهده شده باشد که همواره مقدار A_{ij} که ورودی مسئله است می گیرد اگر مشاهده نشده باشد با مقدار امید B_{ij} با مشاهده بقیه جایگزین می شود که در هر گام E این مقدار برحسب آخرین F های به دست آمده به صورت $F_i^T F_j$ محاسبه می شود، ولی برای افزایش سرعت همگرایی (با استدلال همگرایی است) در محاسبه امید از آخرین مقادیر F استفاده می کنیم یعنی به جای $f_j^{t-1,\infty}$ مقدار $f_j^{t,m}$ قرار می دهیم که روابط به روز رسانی به صورت زیر می شود:

$$f_k^{t,m+1} = f_k^{t,m} + C_k \Bigg\{ \underbrace{\sum_{i \in H} \left(f_i^{t,m\,T} f_k^{t,m} \frac{f_i^{t,m}}{f_i^{t,m\,T} f_k^{t,m}} - f_i^{t,m} \right)}_{\text{صفر}}$$

$$\sum_{i \in O} \left(A_{ik} \frac{f_i^{t,m}}{f_i^{t,m\,T} f_k^{t,m}} - f_i^{t,m} \right) \Bigg\}$$

$$\Longrightarrow f_k^{t,m+1} = f_k^{t,m} + c_k\left\{\sum_{i\in O}\left(\mathcal{A}_{ik}\frac{f_i^{t,m}}{f_i^{b,m\,T} f_k^{b,m}} - f_i^{t,m}\right)\right\}$$

$$\Longrightarrow f_k^{t,m+1} = f_k^{t,m} + c_k\left\{\underbrace{\sum_{i\in O}\left[\left(\frac{\mathcal{A}_{ik}}{f_i^{b,m\,T} f_k^{b,m}} - 1\right) f_i^{t,m}\right]}_{grad_k}\right\}$$

$$\star\ grad_k = \sum_{i\in O}\left[\left(\frac{\mathcal{A}_{ik}}{f_i^{b,m\,T} f_k^{b,m}} - 1\right) f_i^{t,m}\right]$$

$$= \sum_{i}^{N}\left[\underbrace{w_{ik}\left(\frac{\mathcal{A}_{ik}}{f_i^{b,m\,T} f_k^{b,m}} - 1\right)}_{Q_{ik}} f_i^{t,m}\right] = \left[Q\ \ F\right]_{\text{سطر k ام}}$$

$$\star\ Q \triangleq \left(F^{t,m}\, F^{b,m\,T}\right)^{\overset{\text{Hadamard Inverse}}{\circ -1}} \underset{\text{Hadamard product}}{\circ} \mathcal{A} \circ W - W$$

به فرم ماتریسی داریم:

$$\boxed{\begin{cases} F^{new} = (I + CQ)\,F^{old} \\ Q = \left(F^{old}\, F^{old\,T}\right)^{\circ -1} \circ \mathcal{A} \circ W - W \\ C = \operatorname{diag}(c_1, c_2, \dots, c_N) \end{cases}}$$

که c_i ها برحسب بهبود خطا با هر f_i، α برابر و به ازای افزایش خطا $\frac{1}{2\alpha}$ برابر می‌شود

که α مقدار 4 انتخاب شده! این محاسبات برای افزایش سرعت ماتریسی انجام می‌شود!

پیاده سازی parallel:

انواع مختلف مقداردهی اولیه برای f می‌سازیم که شامل:

* از یک توزیع گاما با α و β تصادفی
* از یک توزیع Normal با سیگما تصادفی
* همه مقادیر یکسان و از توزیع Uniform بین 0 و 0,5

که بهترین نتایج برای حالت آخر است و مقدار حدود 0,3 است!

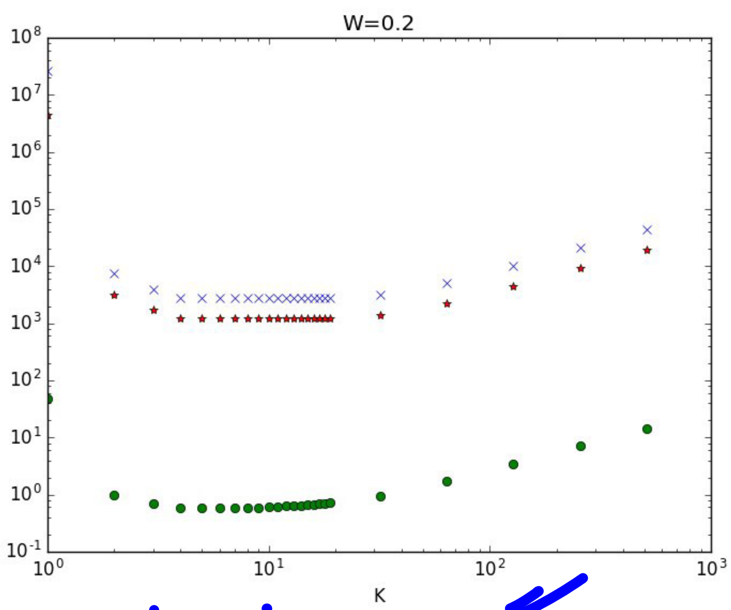
برای پیدا کردن بهترین نقطه شروع f و ایجاد کد موازی به این صورت عمل می‌شود که به تعداد n f تصادفی ایجاد می‌شود و هرکدام حداکثر 20 دور به روز می‌شود پس از بین این f های حامل بهترین k در مرحله بعد وجود دارد به علاوه n مقادیر جدید تصادفی، تعداد f های تصادفی در ابتدا همان n است و در هر مرحله به صورت خطی این تعداد کاهش می‌یابد تا در مرحله‌ی آخر تمام f ها حاصل بهترین f های مرحله قبل باشد که تعداد این مراحل 100 در نظر گرفته شده است

در آخر هم n تا f حاصل بدون محدودیت تعداد به روز رسانی می‌شوند

n برای استفاده‌ی حداکثری از توان پردازشی تعداد core ها می‌گذاریم که برای نتایج حاصل $n=64$ بوده!

یعنی در مجموع 3278 حالت مختلف برای نقطه شروع بررسی شده

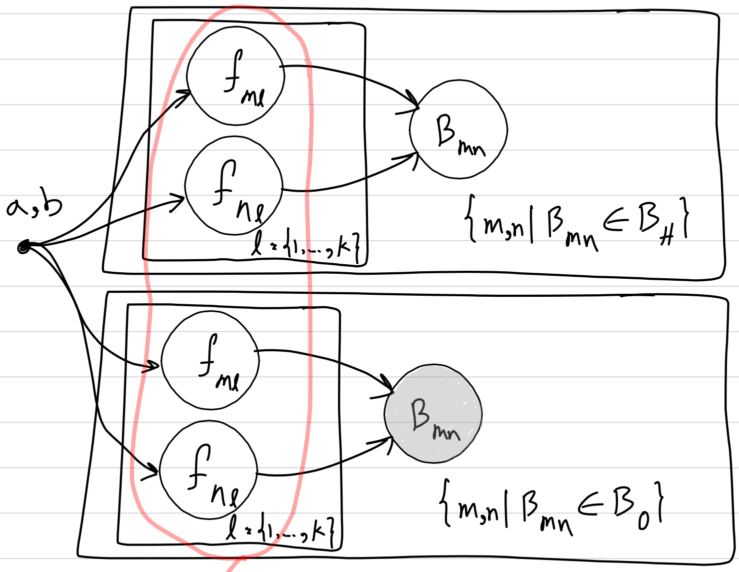
برای پیدا کردن بهترین k روی داده‌های Observe که حکم همان داده‌های Train را دارد برای مقادیری از k الگوریتم اجرا شده که حاصل به صورت زیر است:



* مقدار اولیه f تصادفی بین مقادیر 0 و 1 انتخاب شده و روال انتخاب f اولیه که در بالا گفته شد برای مشخص تر شدن تأثیر k در نظر گرفته نشده
* بهترین مقدار k حدود 12 است که در بقیه مراحل مقدار 12 برای آن انتخاب شده!

نتایج کامل‌تر در قسمت 6 فاز دوم گزارش شده است

مدل اول پیشنهادی :



$$f_{il} \sim Gamma(a,b)$$
$$B_{ij} \sim poisson(f_i^T f_j)$$

توجه شود که در بین دو plate تعداد f مشترک وجود دارد !

$$p(f,B;a,b) = \prod_{n=1}^{N}\prod_{l=1}^{K} p(f_{nl};a,b) \prod_{n,m:\, n<m} p(B_{mn} \mid f_m, f_n)$$

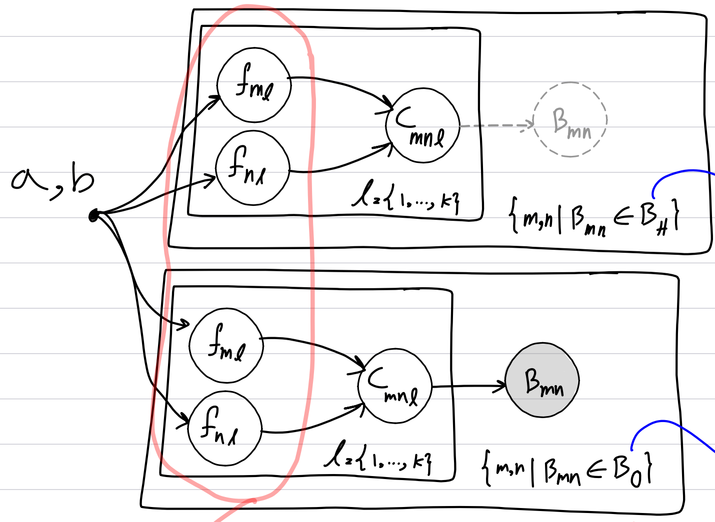
$$\Rightarrow p(f_{n^*l^*} \mid other) = \frac{p(f_{n^*l^*};a,b) \prod_{\substack{m=1 \\ m\neq n^*}}^{N} p(B_{mn^*} \mid f_{n^*}, f_m)}{\sum_{f_{n^*l^*}} \left\{ p(f_{n^*l^*};a,b) \prod_{\substack{m=1 \\ m\neq n^*}}^{N} p(B_{mn^*} \mid f_{n^*}, f_m) \right\}}$$

$$\Rightarrow \ln p(f_{n^*l^*} \mid other) = (a-1)\ln f_{n^*l^*} - b f_{n^*l^*} + \sum_{\substack{m=1 \\ m\neq n^*}}^{N} \left\{ -f_{n^*}^T f_m + B_{mn^*} \ln f_{n^*}^T f_m \right\} + \text{ثابت}$$

که در محاسبه امید مشکل می‌خوریم و مدل بعدی را در نظر می‌گیریم

مدل جدیدی که داریم:

$$\begin{cases} f_i \sim Gamma(a,b) \\ c_{ijl} \sim poisson(f_{il}\, f_{jl}) \\ B_{ij} = \sum_l c_{ijl} \end{cases}$$



مجموعه B های دیده‌نشده

" " " دیده‌شده

توجه شود که در بین دو plate تعدادی f مشترک وجود دارد!

$$\star\ P(f, c, B_O) = \prod_{n=1}^{N}\prod_{l=1}^{k}\{P(f_{nl}; a,b)\} \prod_{m,n: m<n}\prod_{l=1}^{K}\{P(c_{mnl} \mid f_{ml}, f_{nl})\} \prod_{\{m,n \mid B_{m,n}\in B_O\}}\{P(B_{mn} \mid c_{mn})\}$$

★ برای B های latent از آنجا که با مشخص شدن c مربوط به آن مقدار B هم مشخص است، این B ها را از مدل حذف کردن و فقط برای B های مشاهده‌شده متغیری در مدل داریم

★ با توجه به اضافه شدن c ها اثبات می‌کنیم که همان توزیع قبلی را دارد:

$$\left.\begin{matrix} A \sim poisson(\lambda_1) \\ B \sim poisson(\lambda_2) \end{matrix}\right\} \Rightarrow P(\underbrace{A+B}_{C}=\alpha) = \sum_{\beta=0}^{\alpha} \underbrace{P(A+B=\alpha \mid B=\beta)}_{P(A=\alpha-\beta)}\, P(B=\beta)$$

$$\Rightarrow P(C=\alpha) = e^{-\lambda_1-\lambda_2}\sum_{\beta=0}^{\alpha}\frac{\lambda_1^{\alpha-\beta}\lambda_2^{\beta}}{(\alpha-\beta)!\,\beta!} = \frac{e^{-(\lambda_1+\lambda_2)}}{\alpha!}\underbrace{\sum_{\beta=0}^{\alpha}\binom{\alpha}{\beta}\lambda_1^{\alpha-\beta}\lambda_2^{\beta}}_{(\lambda_1+\lambda_2)^{\alpha}} \Rightarrow C \sim poisson(\lambda_1+\lambda_2)$$

$$B_{ij} \sim poisson\Big(\underbrace{\sum_l f_{il}\, f_{jl}}_{f_i^T f_j}\Big) \Rightarrow$$

بنابراین این مدل جدید همان
تناقضی با مدل قبلی ندارد

★ برای تقریب $p(f,c|B_0)$ با روش Varibional تابع $q(f,c)$ را درنظرگرفته و بین $\{f_{n\ell}\}$ و $\{c_{mn}\}$ استقلال فرض کرده و $q_{c_{mn}}$ و $q_{f_{n\ell}}$ را در ادامه محاسبه کرده.

$$q(f,c)=\prod_{n=1}^{N}\prod_{\ell=1}^{K} q_{f_{n\ell}} \prod_{m,n:m<n} q_{c_{mn}}$$

محاسبه $p(f_{m^*n^*}|\text{other})$:

$$p(f_{n^*\ell^*}|\text{other};a,b)=\frac{\prod_{n=1}^{N}\prod_{\ell=1}^{K}\{p(f_{n\ell};a,b)\}\prod_{m,n:m<n}\prod_{\ell=1}^{K}\{p(c_{mn\ell}|f_{m\ell},f_{n\ell})\}\prod_{\{m,n|B_{m,n}\in B_0\}}\cancel{\{p(B_{mn}|c_{mn})\}}}{\sum_{f_{n^*\ell^*}}\prod_{n=1}^{N}\prod_{\ell=1}^{K}\{p(f_{n\ell};a,b)\}\prod_{m,n:m<n}\prod_{\ell=1}^{K}\{p(c_{mn\ell}|f_{m\ell},f_{n\ell})\}\prod_{\{m,n|B_{m,n}\in B_0\}}\cancel{\{p(B_{mn}|c_{mn})\}}}$$

$$=\frac{p(f_{n^*\ell^*};a,b)\prod_{\substack{m=1\\m\neq n^*}}^{N}p(c_{mn^*\ell^*}|f_{m\ell^*},f_{n^*\ell^*})\}}{\sum_{f_{n^*\ell^*}}\{p(f_{n^*\ell^*};a,b)\prod_{\substack{m=1\\m\neq n^*}}^{N}p(c_{mn^*\ell^*}|f_{m\ell^*},f_{n^*\ell^*})\}}$$

$$\propto f_{n^*\ell^*}^{a-1}e^{-bf_{n^*\ell^*}}\prod_{\substack{m=1\\m\neq n^*}}^{N}e^{-f_{m\ell^*}f_{n^*\ell^*}}(f_{m\ell^*}f_{n^*\ell^*})^{c_{mn^*\ell^*}}$$

$$\propto f_{n^*\ell^*}^{a-1+\sum_{\substack{m=1\\m\neq n^*}}^{N}c_{mn^*\ell^*}}\exp\Big\{-\Big(b+\sum_{\substack{m=1\\m\neq n^*}}^{N}f_{m\ell^*}\Big)f_{n^*\ell^*}\Big\}$$

$$\Longrightarrow p(f_{n^*\ell^*}|\text{other};a,b)\sim \text{Gamma}\Big(a+\sum_{\substack{m=1\\m\neq n^*}}^{N}c_{mn^*\ell^*},\ b+\sum_{\substack{m=1\\m\neq n^*}}^{N}f_{m\ell^*}\Big)$$

محاسبه $q_{f_{n^*\ell^*}}$:

$$\Longrightarrow \ln p(f_{n^*\ell^*}|\text{other};a,b)=\text{ثابت} f+\Big(a-1+\sum_{\substack{m=1\\m\neq n^*}}^{N}c_{mn^*\ell^*}\Big)\ln f_{m^*n^*}-\Big(b+\sum_{\substack{m=1\\m\neq n^*}}^{N}f_{m\ell^*}\Big)f_{n^*\ell^*}$$

$$\Longrightarrow E_{-f_{n^*\ell^*}}\big[\ln p(f_{n^*\ell^*}|\text{other};a,b)\big]=$$

$$\text{ثابت} f+\Big(a-1+\sum_{\substack{m=1\\m\neq n^*}}E[c_{mn^*\ell^*}]\Big)\ln f_{m^*n^*}-\Big(b+\sum_{\substack{m=1\\m\neq n^*}}^{N}E[f_{m\ell^*}]\Big)f_{n^*\ell^*}$$

$$\Longrightarrow q^*_{f_{n^*\ell^*}}\propto f_{m^*n^*}^{\left(a-1+\sum_{\substack{m=1\\m\neq n^*}}E[c_{mn^*\ell^*}]\right)}\exp\Big\{-\Big(b+\sum_{\substack{m=1\\m\neq n^*}}^{N}E[f_{m\ell^*}]\Big)f_{n^*\ell^*}\Big\}$$

$$\Longrightarrow q^*_{f_{n^*\ell^*}}\sim \text{Gamma}\Big(a+\sum_{\substack{m=1\\m\neq n^*}}E[c_{mn^*\ell^*}],\ b+\sum_{\substack{m=1\\m\neq n^*}}^{N}E[f_{m\ell^*}]\Big)$$

محاسبه $P(C_{m^*n^*}|\text{other})$:

$$P(C_{m^*n^*}|\text{other};a,b)=\frac{\cancel{\prod_{n=1}^{N}\prod_{l=1}^{k}\{P(f_{nl};a,b)\}}\prod_{m,n:m<n}\prod_{l=1}^{K}\{P(C_{mnl}|f_{ml},f_{nl})\}\prod_{\{m,n|B_{m,n}\in B_0\}}\{P(B_{mn}|C_{mn})\}}{\sum_{C_{m^*n^*}}\cancel{\prod_{n=1}^{N}\prod_{l=1}^{k}\{P(f_{nl};a,b)\}}\prod_{m,n:m<n}\prod_{l=1}^{K}\{P(C_{mnl}|f_{ml},f_{nl})\}\prod_{\{m,n|B_{m,n}\in B_0\}}\{P(B_{mn}|C_{mn})\}}$$

$$=\frac{\prod_{l=1}^{K}\{P(C_{m^*n^*l}|f_{m^*l},f_{n^*l})\}\,P(B_{m^*n^*}|C_{m^*n^*})^{\mathbb{I}(B_{m^*,n^*}\in B_0)}}{\sum_{C_{m^*n^*}}\left\{\prod_{l=1}^{K}\{P(C_{m^*n^*l}|f_{m^*l},f_{n^*l})\}\,P(B_{m^*n^*}|C_{m^*n^*})^{\mathbb{I}(B_{m^*,n^*}\in B_0)}\right\}}$$

$$=\begin{cases}\prod_{l=1}^{K}\text{poisson}(C_{m^*n^*l};f_{m^*l}f_{n^*l}) & B_{m^*,n^*}\in B_H\\[2ex] \text{Multinomial}\left(C_{m^*n^*1},\dots,C_{m^*n^*K};B_{m^*n^*},\frac{f_{m1}f_{n1}}{\sum_q f_{mq}f_{nq}},\dots,\frac{f_{mK}f_{nK}}{\sum_q f_{mq}f_{nq}}\right) & B_{m^*,n^*}\in B_0\end{cases}$$

★ اثبات حالت Multinomial:

$$P(C_{m^*n^*}|\text{other};a,b)=\frac{\dfrac{\cancel{e^{-\sum_l f_{ml}f_{nl}}}\prod_l (f_{ml}f_{nl})^{C_{mnl}}}{\prod_l C_{mnl}!}\,\delta\left(B_{m,n}-\sum_l C_{mnl}\right)}{\dfrac{\cancel{\exp\{\sum_l f_{ml}f_{nl}\}}\left(\sum_l f_{ml}f_{nl}\right)^{B_{mn}}}{B_{mn}!}}$$

تنها حالتی که δ در سیگما غیر صفر است

حالتی است که B=Σc است و چون

ضرب تعدادی پواسن است

عبارت توزیع جمع پواسن است

$$=\frac{B_{mn}!\prod_l (f_{ml}f_{nl})^{C_{mnl}}}{\prod_l C_{mnl}!\left(\sum_l f_{ml}f_{nl}\right)^{B_{mn}}}\,\delta\left(B_{m,n}-\sum_l C_{mnl}\right)$$

$$=\binom{B_{mn}}{C_{mn1},\dots,C_{mnK}}\prod_{l=1}^{K}\left(\frac{f_{ml}f_{nl}}{\sum_q f_{mq}f_{nq}}\right)^{C_{mnl}}\delta\left(B_{m,n}-\sum_l C_{mnl}\right)$$

محاسبه‌ی $q_{c_{m^*n^*}}$ برای $B_{m^*n^*}$ های latent :

$$\Longrightarrow \ln p(c_{m^*n^*} \mid \text{other}\, ; a, b) = \sum_{l=1}^{K} \ln \text{poisson}(c_{m^*n^*l} ; f_{m^*l}\, f_{n^*l})$$

$$\Longrightarrow \ln p(c_{m^*n^*} \mid \text{other}\, ; a, b) = \sum_{l=1}^{K} \left\{ -f_{m^*l}\, f_{n^*l} + c_{m^*n^*l} \ln f_{m^*l}\, f_{n^*l} - \ln(c_{m^*n^*l}!) \right\}$$

$$\Longrightarrow E_{-c_{m^*n^*}} \left\{ \ln p(c_{m^*n^*} \mid \text{other}\, ; a, b) \right\} = cte + \sum_{l=1}^{K} \left\{ c_{m^*n^*l} \left(E[\ln f_{m^*l}] + E[\ln f_{n^*l}] \right) - \ln(c_{m^*n^*l}!) \right\}$$

$$\Longrightarrow q_{c_{m^*n^*}}(c_{m^*n^*}) \propto \frac{\prod_l \exp\left\{ c_{m^*n^*l} \left(E[\ln f_{m^*l}] + E[\ln f_{n^*l}] \right) \right\}}{\left(\prod_l c_{m^*n^*l}! \right)}$$

$$\Longrightarrow q_{c_{m^*n^*}}(c_{m^*n^*}) \propto \prod_{l=1}^{K} \frac{\exp\left\{ \left(E[\ln f_{m^*l}] + E[\ln f_{n^*l}] \right) \right\}^{c_{m^*n^*l}}}{c_{m^*n^*l}!} \Longrightarrow$$ روی تمام l ها شکسته شد

$$\boxed{\Longrightarrow q_{c_{m^*n^*l^*}} \sim \text{poisson}\left(\exp\left\{ \left(E[\ln f_{m^*l}] + E[\ln f_{n^*l}] \right) \right\} \right)}$$

محاسبه‌ی $q_{c_{m^*n^*}}$ برای $B_{m^*n^*}$ های Observe:

$$\Longrightarrow \ln p(c_{m^*n^*}|\text{other};a,b) = \ln\left[\left(\sum_\ell c_{m^*n^*\ell}\right)!\right] - \left(\sum_\ell \lg(c_{m^*n^*\ell}!)\right) + \left(\sum_\ell c_{m^*n^*\ell}\left[\ln f_{m^*\ell} + \ln f_{n^*\ell} - \ln\left(\sum_q f_{m^*q}\, f_{n^*q}\right)\right]\right)$$

$$\Longrightarrow E_{-c_{m^*n^*}}\left\{\ln p(c_{m^*n^*}|\text{other};a,b)\right\} = \ln\left[\left(\sum_\ell c_{m^*n^*\ell}\right)!\right] - \left(\sum_\ell \lg(c_{m^*n^*\ell}!)\right) + \sum_\ell c_{m^*n^*\ell}\left(E[\ln f_{m^*\ell}] + E[\ln f_{n^*\ell}]\right)$$
$$- E[\ln f_{m^*}^T f_{n^*}] \sum_\ell c_{m^*n^*\ell}$$

$$\Longrightarrow q_{c_{m^*n^*}}(c_{m^*n^*}) \propto \frac{\exp\left\{\ln\left[\left(\sum_\ell c_{m^*n^*\ell}\right)!\right]\right\} \prod_\ell \exp\left\{c_{m^*n^*\ell}\left(E[\ln f_{m^*\ell}] + E[\ln f_{n^*\ell}]\right)\right\}}{\left(\prod_\ell \exp\left\{\ln(c_{m^*n^*\ell}!)\right\}\right) \exp\left\{E[\ln f_{m^*}^T f_{n^*}] \sum_\ell c_{m^*n^*\ell}\right\}}$$

$$\Longrightarrow q_{c_{m^*n^*}}(c_{m^*n^*}) \propto \frac{\left(\sum_\ell c_{m^*n^*\ell}\right)!}{\left(\prod_\ell c_{m^*n^*\ell}!\right)} \; \frac{\prod_\ell \exp\left\{c_{m^*n^*\ell}\left(E[\ln f_{m^*\ell}] + E[\ln f_{n^*\ell}]\right)\right\}}{\exp\left\{E[\ln f_{m^*}^T f_{n^*}] \sum_\ell c_{m^*n^*\ell}\right\}}$$

$$\Longrightarrow q_{c_{m^*n^*}}(c_{m^*n^*}) \propto \binom{B_{m^*n^*}}{c_{m^*n^*1}, \dots, c_{m^*n^*K}} \frac{\prod_\ell \exp\left\{\left(E[\ln f_{m^*\ell}] + E[\ln f_{n^*\ell}]\right)\right\}^{c_{m^*n^*\ell}}}{\exp\left\{E[\ln f_{m^*}^T f_{n^*}]\; B_{m^*n^*}\right\}} \delta\left(B_{m^*n^*} - \sum_\ell c_{m^*n^*\ell}\right)$$

$$\Longrightarrow \begin{cases} q_{c_{m^*n^*}}(c_{m^*n^*}) \sim \text{Multinomial}\left(c_{m^*n^*1}, \dots, c_{m^*n^*K}; B_{m^*n^*}, \rho_1, \dots, \rho_K\right) \\[2ex] \rho_\ell = \dfrac{\exp\left\{E[\ln f_{m^*\ell}] + E[\ln f_{n^*\ell}]\right\}}{\sum_i \exp\left\{E[\ln f_{m^*i}] + E[\ln f_{n^*i}]\right\}} \end{cases}$$

نتایج استنتاج وردشی:

$$
\begin{cases}
q_{c_{m^*n^*}}\left(c_{m^*n^*}\right) \sim Multinomial\left(c_{m^*n^*1},\dots,c_{m^*n^*k};\, B_{m^*n^*}, f_1,\dots,f_k\right) \\[2ex]
f_{\ell^*} = \dfrac{\exp\left\{ E[\ln f_{m^*\ell^*}] + E[\ln f_{n^*\ell^*}] \right\}}{\sum_i \exp\left\{ E[\ln f_{m^*i}] + E[\ln f_{n^*i}] \right\}} \\[2ex]
q_{c_{m^*n^*\ell^*}} \sim poisson\Big(\underbrace{\exp\left\{\left(E[\ln f_{m^*\ell^*}]+E[\ln f_{n^*\ell^*}]\right)\right\}}_{\lambda_{m^*n^*\ell^*}}\Big)
\end{cases}
$$

برای $B_{m^*n^*} \in B_0$ ←

برای $B_{m^*n^*} \in B_H$ ←

$$
q^*_{f_{n^*\ell^*}} \sim Gamma\Big(\underbrace{a + \sum_{\substack{m=1\\ m\neq n^*}} E[c_{mn^*\ell^*}]}_{\alpha_{n^*\ell^*}},\ \underbrace{b + \sum_{\substack{m=1\\ m\neq n^*}}^{N} E[f_{m\ell^*}]}_{\beta_{n^*\ell^*}}\Big)
$$

$$
E[\ln f_{n^*\ell^*}] = -\ln \beta_{n^*\ell^*} + \psi(\alpha_{n^*\ell^*}) \rightarrow \text{Digamma function}
$$

$$
E[f_{n^*\ell^*}] = \alpha_{n^*\ell^*} / \beta_{n^*\ell^*}
$$

$$
E[c_{m^*n^*\ell^*}] = \begin{cases} \lambda_{m^*n^*\ell^*} & B_{m^*n^*} \in B_H \\ B_{m^*n^*} \times f_{m^*n^*\ell^*} & B_{m^*n^*} \in B_0 \end{cases}
$$

فرمول های بروز رسانی :

$$
f'_{m^*n^*\ell^*} = \frac{\exp\left\{ E[\ln f_{m^*\ell^*}] + E[\ln f_{n^*\ell^*}] \right\}}{\sum_i \exp\left\{ E[\ln f_{m^*i}] + E[\ln f_{n^*i}] \right\}}
$$

$$
\lambda'_{m^*n^*\ell^*} = \exp\left\{\left(E[\ln f_{m^*\ell^*}]+E[\ln f_{n^*\ell^*}]\right)\right\}
$$

$$
\alpha'_{n^*\ell^*} = a + \sum_{\substack{m=1\\ m\neq n^*}} E[c_{mn^*\ell^*}]
$$

$$
\beta'_{n^*\ell^*} = b + \sum_{\substack{m=1\\ m\neq n^*}}^{N} E[f_{m\ell^*}]
$$

که با به روز رسانی معادلات مجهولات که توزیع ها را مشخص می کنند به دست آورده ایم چون به دنبال مقدار امید B های latent هستیم:

$$\hat{B}_{m^*n^*} = E\left\{\sum_{l=1}^{K} c_{m^*n^*l}\right\} = \sum_{l=1}^{K} \lambda_{m^*n^*l}$$

برای Sampling از روش Gibbs استفاده می‌کنیم که نحوه‌ی نمایش توزیع مثل قسمت قبل است و

مایات لازم برای به دست آوردن توزیع‌ها هم در قسمت قبل انجام شد که به صورت زیر است:

$$p(f_{n^*\ell^*} \mid other; a, b) \sim Gamma\left(a + \sum_{\substack{m=1 \\ m \neq n^*}}^{N} c_{mn^*\ell^*},\ b + \sum_{\substack{m=1 \\ m \neq n^*}}^{N} f_{m\ell^*}\right)$$

$$p(c_{m^*n^*} \mid other) = \begin{cases} \prod_{\ell=1}^{K} poisson(c_{m^*n^*\ell} ; f_{m^*\ell} f_{n^*\ell}) & B_{m^*,n^*} \in B_H \\ Multinomial\left(c_{m^*n^*1}, \dots, c_{m^*n^*K} ; B_{m^*n^*}, \frac{f_{m1} f_{n1}}{\sum_q f_{mq} f_{nq}}, \dots, \frac{f_{mK} f_{nK}}{\sum_q f_{mq} f_{nq}}\right) & B_{m^*,n^*} \in B_O \end{cases}$$

* با ثابت بودن B_O بقیه متغیرها را با نمونه‌گیری با روش Gibbs به دست آورده و بعد از مرحله‌ی Burn-in، با تعدادی نمونه‌گیری مشابه قسمت قبل به دنبال توزیع c های مربوط به B های مشاهده

نشده هستیم که یعنی تخمین λ آنها که با Moment اول به دست می‌آید

$$\lambda_{m^*n^*\ell^*} \approx \frac{1}{r} \sum_{i=1}^{r} c^{(i)}_{m^*n^*\ell^*}$$

و مقادیر امید B های latent را به دست می‌آوریم:

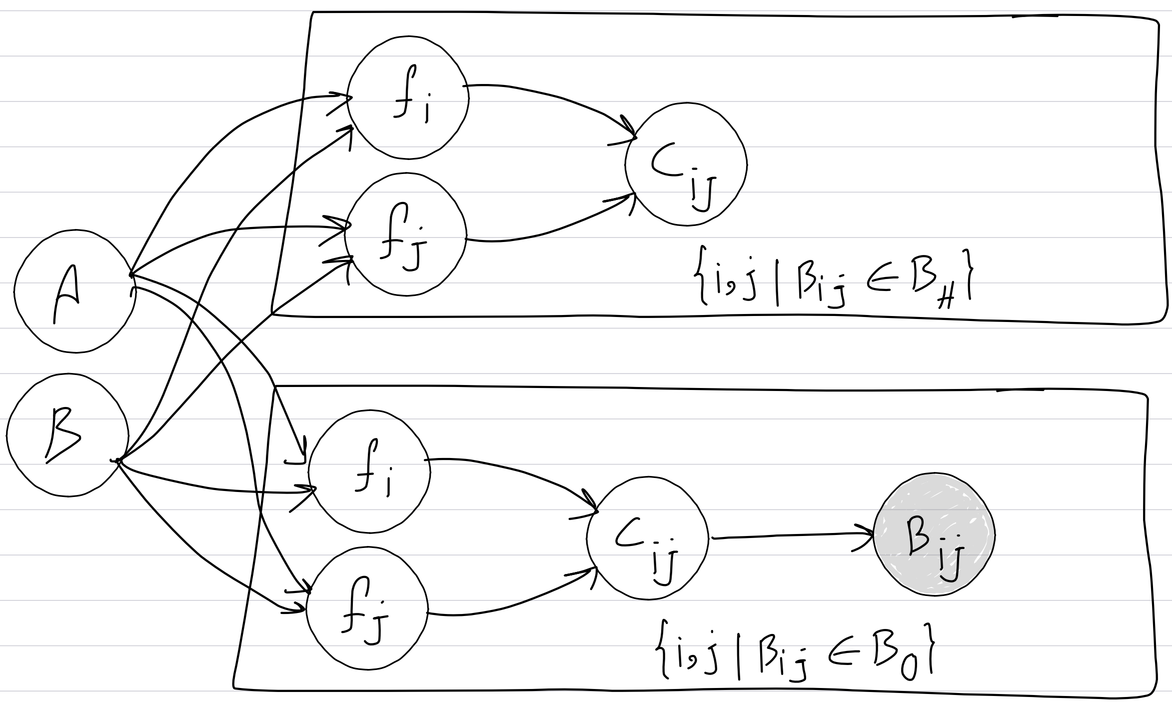
$$\hat{B}_{m^*n^*} = E\left\{\sum_{\ell=1}^{K} c_{m^*n^*\ell}\right\} = \sum_{\ell=1}^{K} \lambda_{m^*n^*\ell}$$

* در پیاده سازی برای یافتن بهترین مقادیر a و b به صورت parallel الگوریتم برای مقادیر تصادفی a و b ، 1000 بار اجرا شده و بهترین مقادیر پیدا شده است .

این مقادیر تصادفی برحسب امید یعنی $\frac{a}{b}$ و واریانس یعنی $\frac{a}{b^2}$ انتخاب شده که از انتخاب Uniform بین 1 تا 20 بوده ، و در آخر هم مقادیر انتخاب شده برای توزیع prior در حالتی که $k=12$ است میانگین حدود 4 دارند و واریانس هم از مقادیر کم به دلیل تاثیر مخرب prior دوری کرده و حدود 8 است .

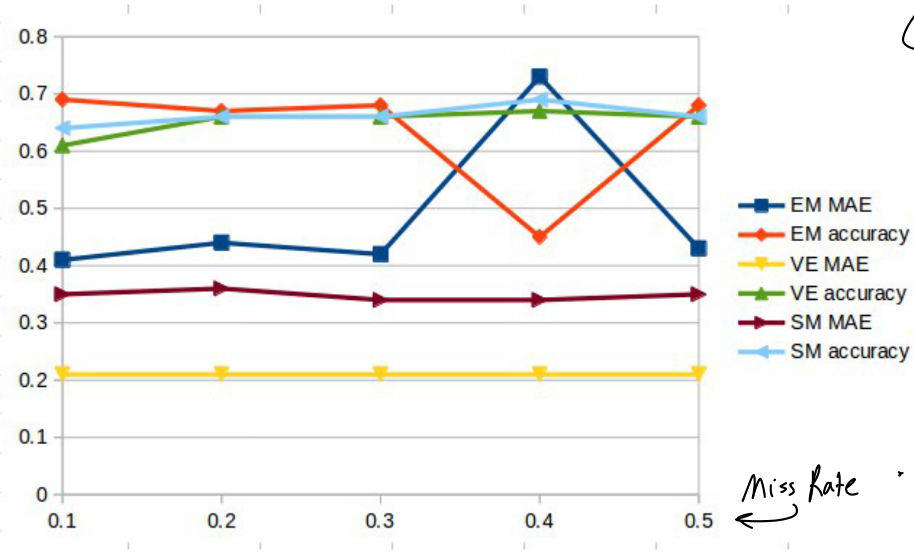
البته می توان برحسب توزیع مقادیر CA این میانگین را تخمین زد! و چون prior است حدودی درست باشد کفایت می کند و در Learning تاثیر آن کم می شود.

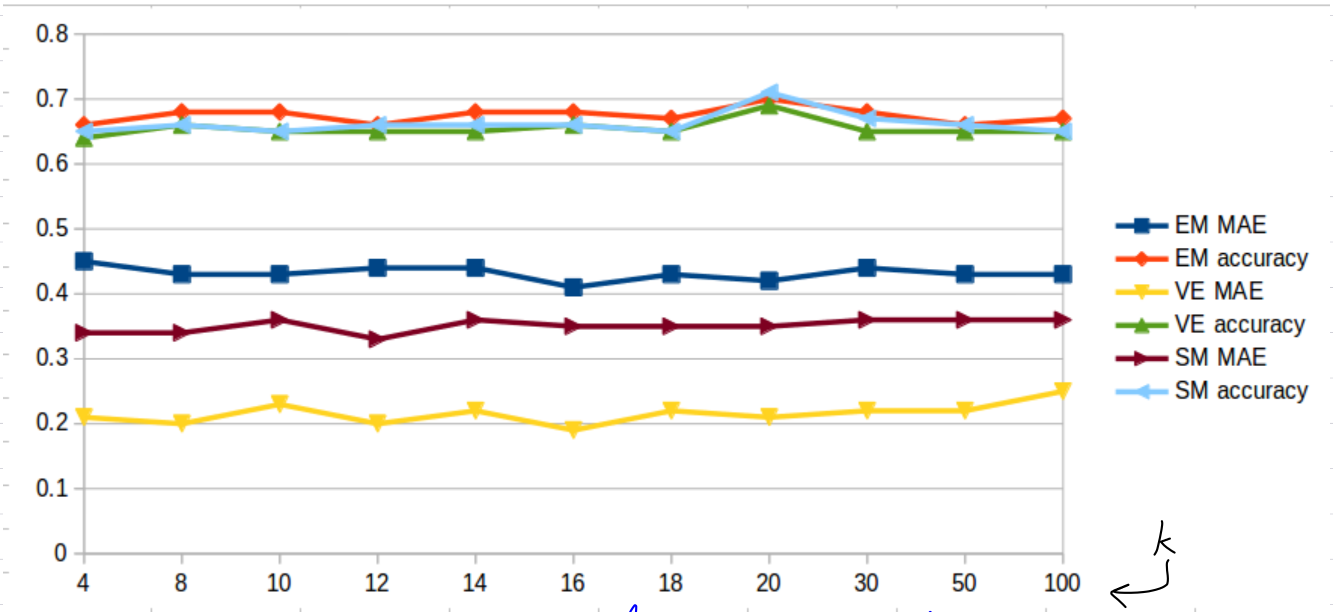
* راه دیگری که برای یافتن a و b به ذهن می آید درنظر گرفتن آنها به عنوان RV و دخیل کردن محاسبه آنها در محاسبات Varitional است که با بررسی انجام شده این مدل تاثیری در روابط دیگر متغیرها ندارد و تنها فایده ی آن کاهش محاسبات برای پیدا کردن a و b است



* دو نمودار زیر حاصل MAE و Accuracy برای حالات k و MissRate های مختلف است

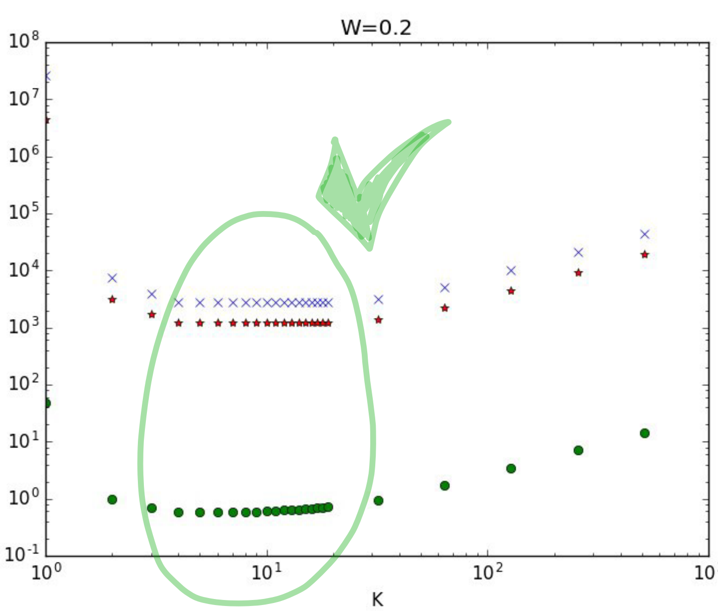
* این نتایج چون برای Result های نهایی انجام شود، تمام Methode های گفته شده برای بررسی ~ 3000 مقدار اولیه در EM، بررسی 1000 مقدار a و b به صورت موازی روی core 64 انجام شده.





* بهترین نتایج مربوط به VE با MAE ~ 0,2 و Accuracy ~ 66% است.

از آنجا که حاصل تقریباً مستقل از k و MissRate شده !!!! احتمالاً به دلیل Search گسترده که در مقادیر f اولیه و به ازای آن k مقادیر مختلف a و b و مقادیر تصادفی مختلف جستجو مفصل انجام شده به ازای تمام k ها حاصل خوب یافت شده.



مثلاً تنها با یک مقدار تصادفی شروع f الگوریتم EM روی دیتاست بزرگ حاصل مقابل را دارد که تاثیر k مشخص است:

* با توجه به این که در Varitional فرض های استقلال اضافه ترتیب به مدل اولیه داریم انتظار می رود که sampling نتایج بهتری دهد اما از آنجا که محاسبات sampling زیاد بوده است بهبود کامل آن فرصت نشد و چیزی که قطعی است حاصل آن نمی تواند از Varitional بهتر شود و این نشان دهنده ی این است که استقلال های اضافه شده در Varitional مناسب بوده.

* از موارد گفته شده و تاثیر مثبت جستجو اولیه می توان فهمید که مدل ارایه شده ضعیف است و جستجوی اولیه تاثیر زیادی روی نتایج می گذارد به حدی که تغییر پارامتر مدل را کم اهمیت کرده است!!

---

* روش EM بر روی دیتاست بزرگ MAE= 0,54 و Accuracy = 95% است.
(بقیه اجرا طولانی شد)

اعداد نمودارهای صفحه ی قبل:

	K (miss=0.2)	4	8	10	12	14
EM	MAE	0.45	0.43	0.43	0.44	0.44
	accuracy	66%	68%	68%	66%	68%
VE	MAE	0.21	0.2	0.23	0.2	0.22
	accuracy	64%	66%	65%	65%	65%
SM	MAE	0.34	0.34	0.36	0.33	0.36
	accuracy	65%	66%	65%	66%	66%

	16	18	20	30	50	100
EM	0.41	0.43	0.42	0.44	0.43	0.43
	68%	67%	70%	68%	66%	67%
VE	0.19	0.22	0.21	0.22	0.22	0.25
	66%	65%	69%	65%	65%	65%
SM	0.35	0.35	0.35	0.36	0.36	0.36
	66%	65%	71%	67%	66%	65%

	miss (K=12)	0.1	0.2	0.3	0.4	0.5
EM	MAE	0.41	0.44	0.42	0.73	0.43
	accuracy	69%	67%	68%	45%	68%
VE	MAE	0.21	0.21	0.21	0.21	0.21
	accuracy	61%	66%	66%	67%	66%
SM	MAE	0.35	0.36	0.34	0.34	0.35
	accuracy	64%	66%	66%	69%	66%

ایده ۴:

* چون در بعضی حالات از نتایج جواب های عجیب مثل همه عناصر 1 دیده می شد یکی از مشکلات می تواند توزیع poisson باشد که حاصل ضرب داخلی به سمت صفر برود و توزیع در حول صفر جمع می شود و برای رفع این موضوع می توان پارامتری در حاصل ضرب داخلی ضرب کرده که با scale کردن از سمت صفر دور شود! (این پارامتر هم در یادگیری دخیل شود)

* ضرب داخلی معیاری مناسب برای شباهت دو مجموعه ویژگی نیست و مثال های دلیل بر ضعف آن وجود دارد صرفاً در حد ایده می توان به جای ضرب داخلی برای تعیین شباهت از مدل پیچیده تری قرار داد مثلاً استفاده از مدل بدون جهت گرافی با فاکتورهای از نوع Sigmoid که هم از یک شبکه عصبی برای پیدا کردن ویژگی های سطح بالاتر است. که برای این کار مدل اصلی هم باید بدون جهت و Moralize شود.

(حدوداً می دانم که می توان یک boltzmann Machin قرار داد که هم از یک شبکه عصبی است و وزن های آن شبکه به عنوان پارامترهای مدل است)

* از آنجا که در شبکه های اجتماعی اکثر وزن ها صفر است می توان توجه کرد که در محاسبات می توان برای ماتریس های sparse بهبود ایجاد کرد و performance اجرا را بهبود داد.

توضیح کد:

* یک فایل dataset.py شامل class به نام DATASET است که load و محاسبه خطا
تهایی را انجام می‌دهد

* فایل Main.py که فایل اصلی است و سه کلاس برای ۳ روش و توابعی برای parallel کردن که وجود دارد، در انتها کد از class ها استفاده شده و مسیر Dataset مورد نظر و مقادیر miss_w و k داده شده است.

* فایل Seed.txt که seed تمام مقادیر تصادفی را مشخص می‌کند!