Active Community Detection with Maximal Expected Model Change

- Supplemental Material -

Dan Kushnir Benjamin Mirabelli

1 Proofs

1.1 Theorem 1

Proof. see [Massoulie(2014), Mossel et al.(2015)] for the details of the proof. □

1.2 Theorem 2

Proof. We start with some definitions:

Definition 1. *For a given adjacency matrix $\boldsymbol{M} = M$, let $e = e(M)$ be the set of edges in M. Also, let $|e|$ be the size of the set e (i.e. the total number of edges).*

Definition 2. *Given the complete labeling assignment $X \in \triangle_r^n$, let $e_{in}(X)$ be the number of edges where both endpoint nodes have the same label according to X. Then, since each X_i is a unit vector,*

$$e_{in}(X) = \frac{1}{2} \sum_{\substack{X_i = X_j \\ (i,j) \in e}} \langle X_i, X_j \rangle. \tag{1}$$

Definition 3. *Given the complete labeling assignment $X \in \triangle_r^n$, let $e_{out}(X)$ be the number of edges where both endpoint nodes have different labels according to X. Then, since each X_i lies on the simplex,*

$$e_{out}(X) = -\frac{r-1}{2} \sum_{\substack{X_i \neq X_j \\ (i,j) \in e}} \langle X_i, X_j \rangle. \tag{2}$$

Definition 4. *Given the complete labeling assignment $X \in \triangle_r^n$, let $g_u(X)$ be the number of nodes assigned to the u^{th} label-vector.*

Remark 1. *It is helpful to notice that $\sum_u \binom{g_u(X)}{2}$ is the total number of within-group pairs of nodes and $\sum_{u<v}[g_u(X)g_v(X)]$ is the total number of between-group pairs of nodes given the labeling X. From this we see that the following equalities hold for any labeling assignment X:*

$$\begin{aligned}
\sum_u \binom{g_u(X)}{2} - e_{in}(X) &= \frac{1}{2} \sum_{\substack{X_i = X_j \\ (i,j) \notin e}} \langle X_i, X_j \rangle, \\
\sum_{u<v} [g_u(X) g_v(X)] - e_{out}(X) &= -\frac{r-1}{2} \sum_{\substack{X_i \neq X_j \\ (i,j) \notin e}} \langle X_i, X_j \rangle, \\
\binom{n}{2} - \sum_{u<v} [g_u(X) g_v(X)] &= \sum_u \binom{g_u(X)}{2}.
\end{aligned} \tag{3}$$

From the definition of the SBM we first notice that, unconditioned on a specific adjacency matrix M, $\mathbb{P}[\mathbf{X} = X] = r^{-n}$ for any $X \in \triangle_r^n$. However, given a specific SBM-generated adjacency matrix M,

$$\begin{aligned}
\mathbb{P}[\mathbf{X} = X|\mathbf{M} = M] =& \frac{\mathbb{P}[\mathbf{M} = M|\mathbf{X} = X]\mathbb{P}[\mathbf{X} = X]}{\mathbb{P}[\mathbf{M} = M]} = C'\mathbb{P}[\mathbf{M} = M|\mathbf{X} = X] \\
=& C'\big(p\big)^{e_{in}(X)}\big(1-p\big)^{\sum_u \binom{g_u(X)}{2} - e_{in}(X)} \\
& \cdot \big(q\big)^{e_{out}(X)}\big(1-q\big)^{\sum_{u<v}[g_u(X)g_v(X)] - e_{out}(X)} \\
=& C'\big(p\big)^{|e| - e_{out}(X)} \cdot \big(1-p\big)^{\binom{n}{2} - |e| - \sum_{u<v}[g_u(X)g_v(X)] + e_{out}(X)} \\
& \cdot \big(q\big)^{e_{out}(X)}\big(1-q\big)^{\sum_{u<v}[g_u(X)g_v(X)] - e_{out}(X)}
\end{aligned}$$

where C' is a constant independent of X, the first equality follows from Bayes' Law and the fourth equality follows from remark 1.

Since, conditioned on M, the values for n, e, p and q are all independent of X we incorporate them into the constant terms C and C'' to get,

$$\begin{aligned}
&\mathbb{P}[\mathbf{X} = X|\mathbf{M} = M] \\
&= C''\left[\Big(\frac{p}{q}\Big)^{-e_{out}(X)} \cdot \Big(\frac{1-p}{1-q}\Big)^{-\left(\sum_{u<v}[g_u(X)g_v(X)] - e_{out}(X)\right)}\right] \\
&= C''\left[\exp\Big(-e_{out}(X)\log\Big(\frac{p}{q}\Big) - \Big(\sum_{u<v}[g_u(X)g_v(X)] - e_{out}(X)\Big)\log\Big(\frac{1-p}{1-q}\Big)\Big)\right] \\
&= C''\left[\exp\Big(-re_{out}(X)\log\Big(\frac{p}{q}\Big) - r\Big(\sum_{u<v}[g_u(X)g_v(X)] - e_{out}(X)\Big)\log\Big(\frac{1-p}{1-q}\Big)\Big)\right]^{\frac{1}{r}} \\
&= C''\Bigg[\exp\Bigg((r-1)(-|e| + e_{in}(X))\log\Big(\frac{p}{q}\Big) - e_{out}(X)\log\Big(\frac{p}{q}\Big) \\
&\qquad + (r-1)\Big(-\binom{n}{2} + |e| + \sum_u \binom{g_u(X)}{2} - e_{in}(X)\Big)\log\Big(\frac{1-p}{1-q}\Big) \\
&\qquad - \Big(\sum_{u<v}[g_u(X)g_v(X)] - e_{out}(X)\Big)\log\Big(\frac{1-p}{1-q}\Big)\Big)\Bigg]^{\frac{1}{r}} \\
&= C\Bigg[\exp\Bigg(e_{in}(X)\Big((r-1)\log\Big(\frac{p}{q}\Big)\Big) - e_{out}(X)\log\Big(\frac{p}{q}\Big) \\
&\qquad + \Big(\sum_u \binom{g_u(X)}{2} - e_{in}(X)\Big)\Big((r-1)\log\Big(\frac{1-p}{1-q}\Big)\Big) \\
&\qquad - \Big(\sum_{u<v}[g_u(X)g_v(X)] - e_{out}(X)\Big)\log\Big(\frac{1-p}{1-q}\Big)\Big)\Bigg]^{\frac{1}{r}}.
\end{aligned}$$

Now, from equations (1) and (2) and remark 1 we get

$$\begin{aligned}
&\mathbb{P}[\mathbf{X} = X|\mathbf{M} = M] \\
&= C\left[\exp\Big(\Big(\sum_{(i,j)\in e}\langle X_i, X_j\rangle\Big)\Big(\frac{r-1}{2}\log\Big(\frac{p}{q}\Big)\Big) + \Big(\sum_{(i,j)\notin e}\langle X_i, X_j\rangle\Big)\Big(\frac{r-1}{2}\log\Big(\frac{1-p}{1-q}\Big)\Big)\Big)\right]^{\frac{1}{r}}. \\
&= C\exp\Big(\frac{r-1}{2r}\sum_{(i,j)} M_{i,j}\langle X_i, X_j\rangle\Big) = Ce^{\frac{r-1}{2r}Tr(X^T M X)}
\end{aligned}$$

as desired. □

1.3 Corollary 1

Proof. Notice that $\mathbb{P}[\mathbf{X}_U = X_U|\mathbf{X}_L = X_L] = r^{-(n-k)}$ and $\mathbb{P}[\mathbf{M} = M|\mathbf{X}_L = X_L]$ are both independent of X_U. Thus, by Bayes' theorem, $\mathbb{P}[\mathbf{X}_U = X_U|\mathbf{M} = M, \mathbf{X}_L = X_L] \propto \mathbb{P}[\mathbf{M} = M|\mathbf{X}_U = X_U, \mathbf{X}_L = X_L]$ and the rest follows from the proof of Theorem 1. □

1.4 Lemma 1

Proof. Define $M_{\neg i}$ to be the matrix M with the i^{th} row and column removed. Then, from Theorem 1, the symmetry of M and the linearity of the trace function we get,

$$\begin{aligned}
&\mathbb{P}[\mathbf{X}_i = X_i|\mathbf{M} = M, \mathbf{X}_L = X_L, \mathbf{X}_{U_{\neg i}} = X_{U_{\neg i}}] \\
&= \frac{\exp\left[\frac{r-1}{2r}\left(Tr(\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right]^T M_{\neg i}\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right]) + 2Tr(X_i^T M_i\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right]) + Tr(X_i^T M_{(i,i)} X_i)\right)\right]}{\sum_{X_j\in\triangle_r}\exp\left[\frac{r-1}{2r}\left(Tr(\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right]^T M_{\neg i}\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right]) + 2Tr(X_j^T M_i\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right]) + Tr(X_j^T M_{(i,i)} X_j)\right)\right]} \\
&= \frac{\exp\frac{r-1}{2r}\left(Tr(\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right]^T M_{\neg i}\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right])\right)\exp\left(2Tr(X_i^T M_i\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right])\right)\exp\left(Tr(X_i^T M_{(i,i)} X_i)\right)}{\exp\frac{r-1}{2r}\left(Tr(\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right]^T M_{\neg i}\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right])\right)\sum_{X_j\in\triangle_r}\exp\left(2Tr(X_j^T M_i\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right])\right)\exp\left(Tr(X_j^T M_{(i,i)} X_j\right)} \quad (4) \\
&= \frac{e^{\frac{r-1}{r}(M_i\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right]X_i^T)} e^{\frac{r-1}{2r}(M_{(i,i)}X_i X_i^T)}}{\sum_{X_j\in\triangle_r} e^{\frac{r-1}{r}(M_i\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right]X_j^T)} e^{\frac{r-1}{2r}(M_{(i,i)}X_j X_j^T)}} \\
&= \frac{e^{\frac{r-1}{r}(M_i\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right]X_i^T)}}{\sum_{X_j\in\triangle_r} e^{\frac{r-1}{r}(M_i\left[\begin{smallmatrix}X_{U_{\neg i}}\\ X_L\end{smallmatrix}\right]X_j^T)}}.
\end{aligned}$$

where the last equality comes from the fact that $X_j X_j^T = 1$ for any $X_j \in \triangle_r$. □

1.5 Theorem 3

Proof. We start with providing three necessary auxiliary results to be used in our proof (their proofs are provided in the following subsections 1.6, 1.7, 1.8): First, we prove in Lemma 2 that when the absolute differential degree is similar across all node types (as in $SNR < 1$), MEMC has a preference for selecting type-1 nodes over type-2 or correctly assigned nodes.

Next, we prove the conditions under which MEMC will have preference to correct type-2 nodes over querying correctly assigned nodes. In particular, Corollary 2 below guarantees that as long as there are type-2 error nodes with absolute differential degree that is lower than other nodes they will be queried by MEMC:

Lastly, Lemma 3 provides the probability of having minority nodes. While majority nodes can be classified correctly by MAP and ML classifiers once most of their neighbors are correctly identified, minority error nodes (i.e. type-2 error nodes) need to be queried directly in order to be corrected. Therefore estimating their proportion in the overall set is crucial for bounding sample complexity analysis for querying type-2 nodes:

To this end we have the necessary ingredients to provide the sample complexity: using Lemma 2 we obtain that all type-1 errors are initially corrected by MEMC using m_1 queries. At the following stage nodes are queried according to their minimal absolute differential degree following Corollary 2. During this stage type-2 minority nodes as well as correctly assigned nodes are queried at a frequency depending on their distribution around zero differential degree as function of a and b. Based on Lemma 3 we can bound the search space around 0-differential degree for type-2 error nodes with the upper bound in Lemma (3)

$$P(c_{out} \geq c_{in}) \leq \exp\Big(-\Big(\sqrt{\frac{b}{2}} - \sqrt{\frac{a}{2}}\Big)^2\Big). \quad (5)$$

and between $[0, -l_c]$ with sampling the mass equal to the summation of the Skellam probability $P(k; a, b)$. The Skellam distribution models the summation of the two racing Poisson processes with means a and b, which forms positive differential degree smaller than $-l_c$:

$$\sum_{k=1}^{-l_c} P(k; a, b),\ where\ P(k; a, b) = e^{-(a+b)}\Big(\frac{a}{b}\Big)^{\frac{k}{2}} I_k(2\sqrt{ab}), \quad (6)$$

and $I_k(2\sqrt{ab})$ is the Bessel function of the first order. The degree value l_c can be computed by using the Skellam distribution. Specifically, by choosing l_c s.t.

$$l_c = inf\{l|P(d_X(v) \leq l) = o(n^{-1})\}. \tag{7}$$

We therefore obtain that the expected sample complexity of MEMC is comprised of sampling first the m_1 type-1 errors and then sampling a.a.s all the nodes of differential degree within $[-l_c, l_c]$.

Since the Random criterion selects nodes uniformly at random it will have to sample order n nodes to discover all m_1 and m_2 nodes. □

1.6 Lemma 2

Proof. Consider the EMC criterion for some node v_q:

$$\text{EMC}(M, X'_U, X_L, \triangle_r)_{X_q} = \sum_{X_q \in \triangle_r} \hat{\mathbb{P}}[\mathbf{X}_q = X_q | \mathbf{M} = M, \mathbf{X}_L = X_L, \mathbf{X}_{U_{\neg q}} = X'_{U_{\neg q}}] \cdot \delta(\Phi, X_q). \tag{8}$$

We first focus on the the model change component $\delta(\Phi, X_q) = \|\Phi(M, [X_L, X_q], \tilde{X}) - \Phi(M, X_L, \tilde{X})\|$. We examine the model change for a candidate q-node v_q that is a v^1-node (type-1 error node) where w.l.g its current label is $\tilde{X}_q = -1$, and its newly assigned label is $+1$. Assume that v_q^1 neighbors are correctly assigned such that $k + \delta$ are $+1$ node, and k are -1 nodes [1].

The model change $\delta(\Phi, X_q)_{I(v_q^1), I(+1)}$, where $I(\cdot)$ maps the input to its corresponding index in the probability model matrix, will have the following value for changing v_q from its current -1 label to $+1$ label (to facilitate notation the denominators in Φ and the constants in the exponents are omitted):

$$\begin{aligned}
\delta(\Phi, \{+1\})_{I(v_q^1), I(+1)} &= \Big\| \exp\big[\underbrace{\sum_{k+\delta} \log \frac{p}{q}}_{\text{+1 neighbors}} + \underbrace{\sum_{k} \log \frac{p}{q}(-1)}_{\text{-1 neighbors}} + \underbrace{\sum_{n-(k+\delta)} \log \frac{1-p}{1-q}}_{\text{+1 non-neighbors}} + \underbrace{\sum_{n-k} \log \frac{1-p}{1-q}(-1)}_{\text{-1 non-neighbors}} \big] \\
&\quad - \exp\big[\sum_{k+\delta} \log \frac{p}{q}(-1) + \sum_{k} \log \frac{p}{q} + \sum_{n-(k+\delta)} \log \frac{1-p}{1-q}(-1) + \sum_{n-k} \log \frac{1-p}{1-q}(-1) \big] \| \\
&= \Big\| \exp\big[\delta \log \frac{p}{q} - \delta \log \frac{1-p}{1-q}\big] - \exp\big[-\delta \log \frac{p}{q} + \delta \log \frac{1-p}{1-q}\big] \Big\| \\
\Rightarrow \text{EMC}(M, X'_U, X_L, \triangle_r)_{I(v_q^1), I(+1)} &= \exp\big[\delta \log \frac{p}{q} - \delta \log \frac{1-p}{1-q}\big] \cdot \Big\| \exp\big[\delta \log \frac{p}{q} - \delta \log \frac{1-p}{1-q}\big] \\
&\quad - \exp\big[-\delta \log \frac{p}{q} + \delta \log \frac{1-p}{1-q}\big] \Big\|
\end{aligned} \tag{9}$$

Examining the model change for each of the neighbors u of v_q^1, and assuming they do not change their label as a result of the new assignment of v_q^1 (and therefore their neighbors do not change their labels either) provides $\text{EMC}(M, X'_U, X_L, \triangle_r)_{X_u} = 0$. Therefore the total model change for a type-1 error node v^1 is

$$\begin{aligned}
\text{EMC}(M, X'_U, [X_L, X_q^1], \triangle_r) = \exp\big[\delta \log \frac{p}{q} - \delta \log \frac{1-p}{1-q}\big] \cdot \Big\| \exp\big[\delta \log \frac{p}{q} - \delta \log \frac{1-p}{1-q}\big] \\
- \exp\big[-\delta \log \frac{p}{q} + \delta \log \frac{1-p}{1-q}\big] \Big\|
\end{aligned} \tag{10}$$

Next, we examine the model change for a candidate q-node v_q that is a v^2-node (type-2 error node) where w.l.g its current label is $\tilde{X}_q = -1$, and its newly assigned label is $+1$. Using similar assumptions on its neighbors we arrive at

$$\begin{aligned}
\text{EMC}(M, X'_U, [X_L, X_q^2], \triangle_r) = \exp\big[-\delta \log \frac{p}{q} + \delta \log \frac{1-p}{1-q}\big] \cdot \Big\| \exp\big[-\delta \log \frac{p}{q} + \delta \log \frac{1-p}{1-q}\big] \\
- \exp\big[\delta \log \frac{p}{q} - \delta \log \frac{1-p}{1-q}\big] \Big\|
\end{aligned} \tag{11}$$

[1] This assumption can be used by using similar argument to [Mossel et al.(2015)]: Let $V_\varepsilon = v : d_X(v) < \varepsilon\sqrt{np \log n}$. According to Proposition 4.7 therein no two nodes in V_ε are adjacent

To this end we can conclude that, under the above assumptions,

$$\mathrm{EMC}(M, X'_U, [X_L, X^1_q], \triangle_r) > \mathrm{EMC}(M, X'_U, [X_L, X^2_q], \triangle_r). \tag{12}$$

Next, we attend the model change introduced by flipping the assignment of a node correctly labeled (w.l.g. to $+1$) to its opposite, resulting in creating a minority node whose model change is similar to that of a v^2 node:

$$\begin{aligned}\mathrm{EMC}(M, X'_U, [X_L, X^3_q], \triangle_r) = \exp\big[-\delta\log\frac{p}{q} + \delta\log\frac{1-p}{1-q}\big] \cdot \Big\| \exp\big[-\delta\log\frac{p}{q} + \delta\log\frac{1-p}{1-q}\big] \\ - \exp\big[\delta\log\frac{p}{q} - \delta\log\frac{1-p}{1-q}\big]\Big\|\end{aligned} \tag{13}$$

We conclude that

$$\mathrm{EMC}(M, X'_U, [X_L, X^3_q], \triangle_r) = \mathrm{EMC}(M, X'_U, [X_L, X^2_q], \triangle_r). \tag{14}$$

□

1.7 Corollary 2

Proof. We use here the result of Lemma 2 where for a given fixed δ for both nodes

$$\begin{aligned}\mathrm{EMC}(M, X'_U, [X_L, X^2_q], \triangle_r) = \mathrm{EMC}(M, X'_U, [X_L, X^3_q], \triangle_r) = \\ \exp\big[-\delta\log\frac{p}{q} + \delta\log\frac{1-p}{1-q}\big] \cdot \Big\| \exp\big[-\delta\log\frac{p}{q} + \delta\log\frac{1-p}{1-q}\big] - \exp\big[\delta\log\frac{p}{q} - \delta\log\frac{1-p}{1-q}\big]\Big\|\end{aligned} \tag{15}$$

However, for different absolute differential degree such that $\delta_2 < \delta_3$ we obtain $\mathrm{EMC}(M, X'_U, [X_L, X^2_q], \triangle_r) > \mathrm{EMC}(M, X'_U, [X_L, X^3_q], \triangle_r)$ □

1.8 Lemma 3

Proof. We consider the generation of the edges as a Poisson process, where c_{out} $Poisson(\frac{b}{2})$ and $c_{in} \sim Poisson(\frac{a}{2})$. Then the difference variable $Z = c_{out} - c_{in}$ Has a Skellam distribution: $Z \sim Skellam(k; b, a)$ such that

$$P(X = k) = e^{(-(a+b))}(\frac{a}{b})^{\frac{k}{2}} I_k(2\sqrt{ab}), \tag{16}$$

where $I_k(z)$ is the Bessel function of first order. Given that $b < a$ we can use the standard Chernoff bound to prove the upper inequality. Further noting that $X + Y \sim Poiss(b + a)$ and $X|X + Y \sim Bin(X + Y, \frac{b}{b+a})$, and $P(X > Y) = P(X > \frac{X+Y}{2})$ and upper bounding it by conditioning on $X + Y = i$ we can show that

$$P(X > Y) > \frac{\exp(-(\sqrt{\frac{b}{2}} - \sqrt{\frac{a}{2}})^2)}{(\frac{a+b}{2})^2} - \frac{\exp(-(\frac{b}{2} + \frac{a}{2}))}{\sqrt{2}ab} - \frac{\exp(-(\frac{b}{2} + \frac{a}{2}))}{2ab} \tag{17}$$

see more details at [Kamath et. al. (2015)] □

1.9 Lemma 4

Proof. We first consider the case of a majority node v_j with neighbor v_i which has changed its label from $\tilde{x}_i$ to $\tilde{x}^{up}_i$. We observe the following probabilities

- $P\{\tilde{x}_j \neq \tilde{x}^{up}_i\} = \frac{1}{2}$ (having different label than the newly revealed neighbor's label),
- $P\{\tilde{x}_j \neq x_j\} = \frac{1}{2}$(having an erroneous assignment) at $SNR < 1$, and
- $P\{x_j = x_i\} = \frac{a-b}{(a+b)}$ (having similar ground truth label as its neighbors flipped label).

Therefore, the probability of v_j flipping its current (erroneous) label to its correct label is $\frac{a-b}{4(a+b)}$. In the same pattern we summarize the different label-flip probabilities for majority and minority nodes, given a label-flip at a neighboring node:

node type	correct flip	incorrect flip
majority	$\frac{a}{4(a+b)}$	$\frac{b}{4(a+b)}$
minority	$\frac{b}{4(a+b)}$	$\frac{a}{4(a+b)}$

To this end we can compute the expected number of nodes to correctly change their label following a query

$$d \cdot \Big(\bar{p}_{maj}\frac{a}{4(a+b)} - \bar{p}_{maj}\frac{b}{4(a+b)}\Big) = d \cdot \bar{p}_{maj}\Big(\frac{a-b}{4(a+b)}\Big) = d \cdot p_{maj}, \tag{18}$$

where $\bar{p}_{maj} = 1 - 2\bar{p}_{min}$, $\bar{p}_{min}$ is defined as the upper bound in Lemma 3:

$$P(c_{out} \geq c_{in}) \leq \exp\Big(-\Big(\sqrt{\frac{b}{2}} - \sqrt{\frac{a}{2}}\Big)^2\Big), \tag{19}$$

and $p_{maj} = \bar{p}_{maj}\Big(\frac{a-b}{4(a+b)}\Big)$.
Next, given the average degree d we consider the cascade of diameter that is $O(log_d(n))$ as the following power series:

$$N_{maj} = dp_{maj} + dp_{maj}dp_{maj} + ... + (dp_{maj})^{log_d n} = \frac{dp_{maj}}{1 - dp_{maj}}(1 - (dp_{maj})^{\log_d(n)}). \tag{20}$$

Similar derivation is applied to minority nodes to obtain N_{min}.

□

1.10 Theorem 4

Proof. The active learning process of MEMC is comprised of 3 stages:
1. **Super-linear cascades phase.** The super-linear phase in which cascades take place poses the highest EMC. This stage concludes once there exists no path of size larger than 2 in which nodes of zero differential degree exist, with respect to the assignment $\tilde{X}$. The expected number of queries to attain this state is obtained by dividing $\frac{n}{2}$ by the number of nodes that have flipped their assignments per query, and as such attained non-zero differential degree. The number of such nodes is derived from Lemma 4 as $N_{maj} + N_{min}$. Therefore, the expected number of queried nodes in this stage is the first component in Eq. (18):

$$\frac{n}{2(N_{maj} + N_{min})}. \tag{21}$$

2. **Local type-1 node queries.** The local model change of type-1 error nodes suggest the next highest EMC. As suggested in Lemma 2 and observed for the $SNR > 1$ case. The local type-1 node queries starts once there exists no path of size larger than 2 in which nodes of zero differential degree exist (which typically gives rise to cascades). We therefore subtract from the existing m_1 errors the type-1 error nodes that have been corrected via the cascades process, and since MEMC will query only type-1 we consider this difference as the set of queries for this stage

$$\Big(m_1 - \frac{nN_{maj}}{2(N_{maj} + N_{min})}\Big) \tag{22}$$

The local type-1 error correction is terminated once all type-1 nodes are corrected.
3. **Type-2 bounded search**. The final active querying stage includes querying both type-2 nodes and already correct nodes with minimal absolute differential degree within the $[l_c, -l_c]$ differential degree segment. The process is also equivalent to the process for the $SNR > 1$ case following Corollary 2 which establishes preference for type-2 nodes with low absolute differential degree. As in Theorem 3, we use the Skellam probability here to represent the mass of nodes with positive differential degree smaller than $-l_c$ and the upper bound in Lemma (3) to cover nodes with negative differential degree down to l_c. This mass is taken from the remaining nodes after subtracting prior m_1 queries and type-1 nodes have been corrected during the super-linear cascades phase:

$$\Big(n - \frac{nN_{min}}{2(N_{maj} + N_{min})} - m_1\Big) \cdot \Big(\sum_{k=1}^{-l_c} P(k; a, b) + \exp\Big(-\Big(\sqrt{\frac{b}{2}} - \sqrt{\frac{a}{2}}\Big)^2\Big)\Big), \tag{23}$$

The Random selection algorithm triggers cascades of correction, similarly to MEMC. However, once all paths of zero differential degree with length $l \geq 2$ have been exhausted, the following process entails uniform unbounded sampling on the remaining mass of nodes, scaling with as n queries. □

1.11 Anchor Nodes

As mentioned in Remark 1, in the algorithm's early stages X_L may not yet contain all existing community labels. In these cases, the active learner instead queries for the label of the node that has the largest probability of being assigned to a community with no current supervised label. We refer to these queried nodes as *Anchor nodes* where

$$\begin{aligned}\text{Anchor}(M, X'_U, X_L, \triangle_r) = \underset{q\in U}{\text{argmax}} \quad \underset{\substack{X_q\in\triangle_r \\ X_q\notin \text{unique}(X_L)}}{\max} \\ \mathbb{P}[\mathbf{X}_q = X_q | \mathbf{M} = M, \mathbf{X}_L = X_L, \mathbf{X}_{U_{\neg q}} = X'_{U_{\neg q}}].\end{aligned} \tag{24}$$

Once every distinct label has at least one corresponding queried node, the best-fit-simplex and the simplex formed by these supervised nodes closely align and the algorithm proceeds in querying according to MEMC criterion.

1.12 Speedup

The fast evaluation of the SDP is manageable due to the fast growing field of low-rank SDP solvers [Bandeira et al.(2016)]. There are two additional modifications designated to further accelerate the active learning cycles:

- Initialize each SDP$(M, [X_L, X_q])$ with the previous output of SDP(M, X_L).
- In each iteration greedily query a 'batch' of MEMC nodes per full iteration.

2 Best-Fit Simplex

We present the following algorithm for finding the best-fit simplex for a given set of unit-vectors.

bestFitSimplex(X, r)
Input: X: set of unit vectors, r: , r: number of vectors in simplex
Output: $\triangle_r$: best-fit simplex
1. V = K-Means(X, r)
2. $\triangle_r$ = bestFitSDP(V)

Figure 1: Pseudo-code for *bestFitSimplex*.

We provide pseudo-code in Figure 1. In this algorithm K-Means is the well-known algorithm and outputs a set of r vectors. For the algorithm bestFitSDP we define the $(2r \times 2r)$-matrix A where,

$$\langle A_{i,j}\rangle = \begin{cases} 1 & \text{if } i = j + r \\ 1 & \text{if } i = j - r \\ 0 & \textit{otherwise.} \end{cases}$$

Then, bestFitSDP finds the best-fit simplex $\triangle_r$ by factoring the solution $\mathbb{X} = \left[\begin{smallmatrix}\triangle_r \\ V\end{smallmatrix}\right]\left[\begin{smallmatrix}\triangle_r \\ V\end{smallmatrix}\right]^T$ of the following SDP

$$\begin{aligned}\text{bestFitSDP}(V, r)\text{:} \max_{\mathbb{X}} \ & \text{Tr}(A\mathbb{X}) \\ \text{s.t. } & \mathbb{X}_{ii} = 1, \text{ for } 1 \le i \le 2r \\ & \mathbb{X}_{ij} = -\frac{1}{r-1} \text{ for } 1 \le i, j \le r \\ & \mathbb{X}_{ij} = \langle V_i, V_j\rangle \text{ for } r+1 \le i, j \le 2r \\ & \mathbb{X} \succeq 0.\end{aligned} \tag{25}$$

We define the output of bestFitSDP(V, r) to be $\triangle_r$ rotated so that the vectors V in our output $\left[\begin{smallmatrix}\triangle_r \\ V\end{smallmatrix}\right]$ line up with the original input vectors V. This completes the algorithm.

3 Increased SNR error behaviour

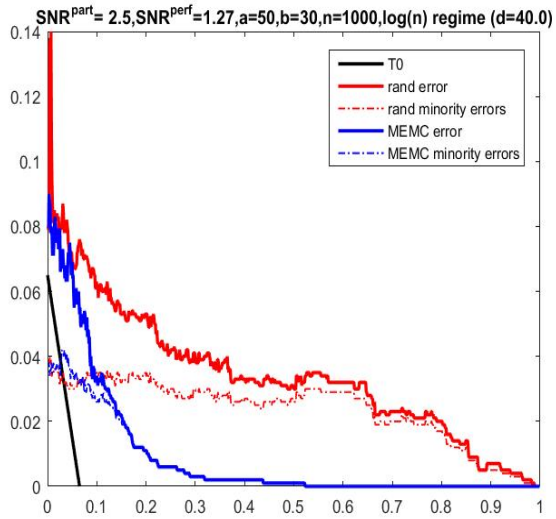

Figure 2: High SNR comparison of MEMC error with Random error and the optimal active learner error

References

A. E. Allahverdyan, G. Ver Steeg, and A. Galstyan. Community detection with and without prior information. *EPL (Europhysics Letters)*, 90(1):18002, 2010.

A. S. Bandeira, N. Boumal, and V. Voroninski. On the low-rank approach for semidefinite programs arising in synchronization and community detection. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 361–382, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

J. Cheng, M. Leng, L. Li, H. Zhou, and X. Chen. Active semi-supervised community detection based on must-link and cannot-link constraints. *PLoS One*, 9(10):18002, 2014.

A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E : Statistical, Nonlinear, and Soft Matter Physics*, 84:066106, 2011. 25 pages, 9 figures.

E. Eaton and R. Mansbach. A spin-glass model for semi-supervised community detection. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 900–906. AAAI Press, 2012.

S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

A. Freytag, E. Rodner, and J. Denzler. Selecting influential examples: Active learning with expected model output changes. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 562–577, Cham, 2014. Springer International Publishing.

A. Gadde, E. En Gad, S. Avestimehr, and A. Ortega. Active learning for community detection in stochastic block models. *CoRR*, 1605 02372, 2016.

M. X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

P. Holland, K. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks - SOC NETWORKS*, 5:109–137, 06 1983.

E. Mossel and J. Xu. Local algorithms for block models with side information *7th Innovations in Theoretical Computer Science (ITCS)*, Jan. 2016.

H. Saad, and A. Nosratinia. Community Detection with Side Information: Exact Recovery under the Stochastic Block Model. *IEEE Journal of Selected Topics in Signal Processing*, May. 2018

V. Kanade, E. Mossel, and T. Schramm. Global and local information in clustering labeled block models. *IEEE Trans. Information Theory*, 62(10):5906–5917, 2016.

Erdős,Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17-61, 1960.

D. Kushnir. Active-transductive learning with label-adapted kernels. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 462–471, New York, NY, USA, 2014. ACM.

M. Leng, Y. Yao, J. Cheng, W. Lv, and X. Chen. Active semi-supervised community detection algorithm with label propagation. In Weiyi Meng, Ling Feng, Stéphane Bressan, Werner Winiwarter, and Wei Song, editors, *Database Systems for Advanced Applications*, pages 324–338, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

S. Liu. Maximum likelihood estimation and inference: With examples in r, sas, and admb by russell b. millar. *International Statistical Review*, 80(2):346–346, 2012.

L. Massoulié. Community detection thresholds and the weak ramanujan property. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 694–703, New York, NY, USA, 2014. ACM.

C. Moore, X. Yan, Y. Zhu, J.-Baptiste Rouquier, and Terran Lane. Active learning for node classification in assortative and disassortative networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 841–849, New York, NY, USA, 2011. ACM.

E. Mossel, J. Neeman, and A. Sly. Consistency thresholds for the planted bisection model. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 69–75, New York, NY, USA, 2015. ACM.

E. Mossel, J. Neeman, and A. Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162(3):431–461, Aug 2015.

R. R. Nadakuditi, and M. EJ Newman. Graph spectral and detectability of community structure in networks. Physical Review Letters 108(18): 188701, 2012.

S.-Y. Yun and A. Proutiere. Optimal Cluster Recovery in the Labeled Stochastic Block Model. ArXiv e-prints, October 2015.

E. Abbe. Community detection and stochastic block models: recent developments *Journal of Machine Learning Research* 18, pages 1–177, 2017

B. Settles. Active learning literature survey. Technical report, 2010.

L. Tang and H. Liu. *Community Detection and Mining in Social Media*, volume 2. 01 2010.

A. L. Traud, P. J. Mucha, and M. A. Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

A. Vezhnevets, J. Buhmann, and V. Ferrari. Active learning for semantic segmentation with expected change. 06 2012.

L. Yang, D. Jin, X. Wang, and X. Cao. Active link selection for efficient semi-supervised community detection. *Scientific Reports*, 5, 2015.

P. Zhang, C. Moore, and L. Zdeborova. Phase transitions in semisupervised clustering of sparse networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 90, 04 2014.

J. Leskovec, and A. Krevl. Stanford Large Network Dataset Collection. *http://snap.stanford.edu/data*. 06 2014.

G. M. Kamath and E. Sasoglu and D. N. C. Tse. Optimal Haplotype Assembly from High-Throughput Mate-Pair Reads. In *http://arxiv.org/abs/1502.01975*, 2015.