Bounding the Error of Markov Network Inference Algorithms

Guy Rom, Amir Globerson

July 11th 2017

Abstract

We study the expected errors in the setting suggested by Globerson et al. in [Globerson, Roughgarden, Sontag, Yildirim, 2015] of exact and approximate inference algorithms on Markov Networks. We give a lower bound for the exact algorithm on a general graph, and show that it is tight in several cases, using the results of Globerson et al. and [Foster, Reichman, Sridharan, 2017], including chains, trees and 2D grids. We present an upper bound on a poly time approximate inference for general graphs and discuss the meaning of the margin between the recovery performances.

1 Introduction

Markov Networks (Markov Random Fields- MRF) is a family of multivariate probability distributions that are represented as undirected graphical models and satisfy several conditional independence properties. These models are widely used in several fields of Machine Learning, namely as a general setting for structured prediction. Three tasks then come to mind when using MRFs- graph architecture, learning parameters and performing inference (usually calculating Marginals or Maximum A-Posteriori assignment). Graph Architecture usually comes from some prior knowledge of the problem and is the place where the specific modelling is performed, where learning and inference are generic and have textbook algorithms that perform each task. Algorithms for both might be intractable for a general graph, but simple graphs such as trees have a closed form solution and are tractable. For the intractable cases, approximation algorithms were devised which show empirical good results but a theoretical understanding of how good they approximate the exact inference task is yet to be found in the general setting for any such approximation algorithm that runs in polynomial time w.r.t. graph size.

Approximate Inference Algorithms Common approximate inference algorithms for MRFs including Linear Programming based algorithms, which take the original inference problem (Let's focus on the MAP case for simplicity) and describes it as a discrete LP problem. in this setting, likelihood is a direction in space and the possible assignments correspond to the vertices of a convex polygon. this is a convex polygon of exponential complexity for general graphs hence is still intractable, but one can create an enclosing convex shape with lower complexity and solve there, finding an approximate solution to the MAP problem. Other marginal approximation algorithms include Variational Approaches (Bethe optimization problem) and The Mean Fields algorithm (best Unimodel Approximation) which are iterative approaches to find different types of marginal distribution functions that approximate the true marginals in the kullback-liebler divergence sense. One of the most popular iterative algorithms for approximate inference is belief propagation, which is exact for trees but doesn't necessarily converge for 'loopy' graphs; when it does converge, it converges to local optima of the Bethe optimization problem, relating it to the Variational Approaches [J. Yedidia, W. T. Freeman and Y. Weiss 2001]

1.1 Preliminaries

The Statistical Recovery Setting The inference setting proposed by Globerson et al. , which is typically called "Statistical Recovery", is composed of an MRF with **identical** pair and singleton functions, that can be expressed as

$$\psi_{ij}\left(y_i, y_j | x_{ij}\right) = \begin{cases} (1-p) & y_i \cdot y_j = x_{ij} \\ p & otherwise \end{cases} \quad \psi_i\left(y_i | x_i\right) = \begin{cases} (1-q) & y_i = x_i \\ q & otherwise \end{cases} \tag{1}$$

where $y, x \in \{\pm 1\}$. notice that they are **symmetric** in the sense that no explicit values are used to define them, but rather agreement with other observable variables is used. The variables p and q are hence named *edge noise/error rate* and *vertex noise/error rate* correspondingly, because they can be seen as describing a process of obtaining a sample from the model- given the y values (that are associated to the vertices), calculate the x values as $x_{ij} = y_i \cdot y_j$ and $x_i = y_j$, then flip each x_i with probability q, and each x_{ij} value with probability p. In this setting [Globerson, Roughgarden, Sontag, Yildirim, 2015] show that the statistical recovery algorithm that inputs only the $x_i$and x_{ij} values and recovers y_i values that errors the least, in the minimax hamming distance sense, is the marginal maximum likelihood algorithm (MML). This task is as hard as calculating marginals of the the MRF hence it is not known to be efficient on general graph; They then define the expected error of any statical recovery algorithm $\mathcal{A}$, given a specifi graph, as the max expected error over all possible Y values, with the expectation over the X generation process, namely-

$$e_Y\left(\mathcal{A}\right) = \max_Y \mathbb{E}_{X|Y}\left[HammingDistance\left(\mathcal{A}\left(X\right), Y\right)\right] \tag{2}$$

Several questions then rise-
for a given graph, what is $e_Y\left(MML\right)$, namely, the expected error of exact recovery?
what is the best polynomial time algorithm for that graph? i.e. $\underset{\mathcal{A} \in PolyTime}{\arg\min}\left(e_Y\left(\mathcal{A}\right)\right)$

Bounds on e_Y [Globerson, Roughgarden, Sontag, Yildirim, 2015] introduce tight upper and lower bounds for e_Y of a concrete algorithm they suggest for statistical recovery of a 2D grid graph which gives

$$\lim_{p \to 0} e_Y\left(\mathcal{A}\right) = \Theta\left(p^2 N\right) \tag{3}$$

This algorithm is relevant for other planar graphs that share some 'weak expansion' properties and a bounded dual graph degree. The lower bound is given by revealing the ground of half of the grid vertices (the ones that correspond to the black vertices of a chess board) and performing MML on the resulting graph; In addition they provide an analysis of e_Y for the same algorithm when applied to d-regular expanders, though it is not efficient for non planar ones. In [Foster, Reichman, Sridharan, 2017] they introduce a message passing algorithm for statistical recovery of tree graphs that solves a linear program whose result has a small hamming distance from the ground truth $\left(O\left(pN\right)\right)$; They then go on to explain how to leverage this algorithm to general graphs by finding a tree decomposition in the sense described by [Neil Robertson and Paul D. Seymour], performing local maximum likelihood recovery by enumeration for each tree node up to a sign, and then choosing the sign according to a global recovery of the tree, by reducing this problem to the original tree recovery but with variable edge error rates; They go on to show that this algorithm produces a result that with high probability has a hamming error which has a term for each tree node that is exponentially decreasing with that subgraph's MinCut. They show that for grids, that have a super logarithmic tree width, using the algorithm on a modified graph where some edges were removed to achieve a logarithmic tree width (and hence, a polynomial time algorithm), result in asymptotically optimal recovery. They present a lower bound for e_Y to any algorithm using an estimator that gets access to all other ground truth while recovering a specific vertex - we will retrace the steps for this lower bound.

1.2 Related Work

LDPC Codes A similar setting arises in the context of stochastic error correcting codes. Low Density Parity Check Codes (LDPC Codes) is a type of code that can be expressed as a graph containing variable nodes and constraint nodes, usually randomly generated under sparsity constraints, that was shown to have assymptotic optimal rate on the Binary Symmetric Channel (BSC) in [Robert G. Gallager, 1963] for a typical graph and a range of crossover probabilities. The LDPC decoding problem coincides with statistic recovery and typically utilizes iterative message passing algorithms and Loopy Belief Propagation specifically. In the context of error correcting codes, there is great interest studying the properties of a graph that result in decrease in channel capacity, which relates to causes for high expected error in exact recovery. A thorough review of the current understanding of graph constructs that are thought to account for a large portion of the errors and are named *trappping sets* for the BSC and *stopping sets* for the *Binary Erasure Channel* can be found in [Aiden Price and Joanne Hall, 2017]. Another work concerning a unification scheme of several decoding algorithm due to [Jon Feldman, Martin J. Wainwright, 2005] is called Linear Programming (LP) Decoding and formalizes terms in LP Relaxation of ML decoding such as pseudo-codewords, fractional distance of an LP relaxation, ML certificate and more.

Loopy Belief Propagation Some work was done by directly confronting the Loopy Belief Propagation (LBP) algorithm. As already mentioned, LBP doesn't converge in the general settings, if it converges, the steady state may depend on the initial conditions of the iterative algorithm, but [J. Yedidia, W. T. Freeman and Y. Weiss 2001] showed that when LBP converges, it converges to local optima of the Bethe optimization problem. A correctness proof MAP assignments calculated using LBP in graphs with a single loop and other insights on convergence rate using spectral methods can be found in [Yair Weiss, 2000]. Some work was done on finding sufficient conditions for convergence that is independent of initial conditions, such as [Joris M. Mooij, Hilbert J. Kappen, 2007], but these conditions are not necessary.

2 Lower Bound for Expected Recovery Error

Min Weight Cover Lower Bound

The method used to lower bound the expected error of recovery of a 2d grid in Globerson et al. can be extended to graphs other than the original 2D grid- if one reveal the ground truth labels of enough vertices so that all the unrevealed vertices are now disjoint and then perform MML recovery on the unrevealed vertices one is left with a generic scheme of lower bounding. Notice that all unrevealed vertices are surrounded by 'd' revealed neighbors, which is the only parameter that governs the expected error of their recovery, hence one can denote the expected 'singleton' error of such a 'star graph' of degree d (unrevealed center with all 'rays' composed of revealed vertices) as

$$e_s\left(d;p,q\right) \equiv e_Y\left(\text{MML d degree star ; Ground Truth for outer vertices}\right) \tag{4}$$

Now for any revelation scheme $S \subset V$ the bound that results can be written as

$$e_Y\left(\mathcal{A}\right) \geq \sum_{v \in V \setminus S} e_s\left(d\left(v\right);p,q\right) = \sum_{d=0}^{\infty} \#_d\left(G;S\right) \cdot e_s\left(d;p,q\right) \tag{5}$$

where $d\left(v\right)$ corresponds to the degree of the vertex v, and we introduced the operator notation $\#_d$ that counts how many unrevealed vertices of degree d are in G given S. Finally, we can see that by defining $\vec{\#}$ as an infinite vector of operators (where $\left(\vec{\#}\right)_i = \#_i$) and a

Table 1: e_s values and bounds for low degrees

d	e_s	$LowerBound\,(e_s)$	$UpperBound\,(e_s)$
0	q	q	q
1	p	p	p
2	$2qp + (1-2\cdot q)\,p^2$	$2qp$	$2qp + p^2$
3	$p^2 + 2\left(p^2 - p^3\right)$	p^2	$3p^2$
4	$p^2 \cdot \left(p^2 + 6q\left(1-p^2\right)\right)$	$\frac{9}{2}p^2q$	$p^2 \cdot \left(p^2 + 6q\right)$
5	$\left(10 - 15\cdot p + 6\cdot p^2\right)\cdot p^3$	$4\cdot p^3$	$10p^3$

corresponding vector $\left(\vec{e_s}\,(p,q)\right)_i = e_s\,(i\,;p,q)$ one can rewrite the bound as the linear product $\left\langle \vec{\#}\,(G\,;S)\,,\,\vec{e_s}\,(p,q)\right\rangle$. Lastly, $\vec{\#}$ can be expressed as a linear product, if G is expressed by a matrix of dimension $|V| \times (|V|-1)$ where each row corresponding to a vertex and it is a binary one-hot representation of that vertex's degree in G, and S represented as a row vector of length $|V|$with a bit set in every unrevealed vertex, leading to the representation $\vec{\#} = S \cdot G$. The bound can then be written as $S \cdot G \cdot \vec{e_s}\,(p,q) = S \cdot \vec{w}\,(G)$ where we are regarding only the first $|V|-1$ elements of $\vec{e_s}\,(p,q)$, and denote $\vec{w}\,(G) \equiv G \cdot \vec{e_s}\,(p,q)$. The tightest lower bound from all the possible schemes of revelation is the one that will reveal the least vertices or the 'most erroneous vertices' in singleton expected error sense, while satisfying the disjointness property. Formally, given G, this is given by

$$\underset{\substack{S \\ s.t.\ S\ results\ in\ a\ disjoint\ partition}}{\arg\max}\ S \cdot \vec{w}\,(G) \tag{6}$$

On can now notice that the definition for disjoint partition coincides with that of a vertex cover- every edge in the graph must contribute at least one of its incident vertices to the revealed vertices of the partition, which are the unset bits of S. This insight makes the tightest bound optimization problem equivalent to a min weight vertex cover with degree dependent weights- under the constraint that S is a valid vertex cover,$\arg\max_S S \cdot \vec{w}\,(G) = \arg\min_S\left(-S \cdot \vec{w}\,(G)\right) = \arg\min_S\left(\vec{1} - S \cdot \vec{w}\,(G)\right) = \arg\min_S\left(\bar{S} \cdot \vec{w}\,(G)\right)$ which is exactly the min weight vertex problem with the weights given by $\vec{w}\,(G)$. For a d-regular graph, the weight is identical hence we are left with a regular vertex cover problem.

Analyzing e_s Without loss of generality we'll assume that both p and q are smaller than$\frac{1}{2}$. If this doesn't hold, invert the inputs of the algorithm. For simplicity, we'll assume $p < q$. Now we can calculate $e_s\,(d\,;p,q)$ explicitly w

$$e_s\,(d\,;p,q) = \mathbb{P}\left[E > \frac{d}{2}\right] + q \cdot \mathbb{P}\left[E = \frac{d}{2}\right] \tag{7}$$

Where $E \sim Bin\,(d,p)$ and represents edge errors in the incident edges, and $\mathbb{P}$ is the probability for this event. See Appendix for clarification and proof. The second term is the tie breaker in the even degrees. Table 1 illustrates some e_s values for low degree, or small d values where it is assumed that $p < q$. In general, if $p < q$ than for odd degrees the tie breaker is an edge, hence the result is independent of q, and for even degrees the tie breaker is the vertex hence, e_s has a linear factor of q.

Asymptotic Behavior of e_s for Low Edge Noise for constant d,q and $p \to 0$ we want the smallest amount of edge errors that lead to an error, which are given around $i = \frac{d}{2}$, so central binomial coefficient approximations ($\frac{4^n}{\sqrt{4n}} \leq \binom{2n}{n} \leq \frac{4^n}{\sqrt{3n+1}}$) combined with inspecting only the largest element from each sum ($\binom{d}{i}$ and $p^i\,(1-p)^{d-i}$ decrease

exponentially with i after $d/2$) give

$$e_s\left(d\,;p,q\right) \geq \begin{cases} \frac{(2p)^{\frac{d+1}{2}}}{\sqrt{2\cdot(d+1)}} & odd\,d \\ q\frac{(2p)^{\frac{d}{2}}}{\sqrt{2d}} & even\,d \end{cases} \tag{8}$$

see Appendix and [Foster, Reichman, Sridharan, 2017] for a derivation.
An Upper bound can be achieved in the same scheme, but instead of using only the largest element, we'll use the largest element as an upper bound for each of the last $d/2$ elements.

$$e_s\left(d\,;p,q\right) \leq \begin{cases} \frac{(d-1)(4p)^{\frac{d+1}{2}}}{\sqrt{24d+40}} & odd\,d \\ \left(\frac{q}{\sqrt{\frac{3d+2}{2}}}+\frac{(d-2)(4p)}{\sqrt{24d+64}}\right)(4p)^{\frac{d}{2}} & even\,d \end{cases} \tag{9}$$

Deriving Chain and Loop Lower Bounds This method can be applied trivially to a chain, or a single loop of length n- reveal every 2nd vertex (starting for either odd or even, depending on the length), and using the calculated e_s values for 1 and 2, one can reach the conclusion that, up to the error of one or two edges,

$$e_Y\left(MML\,on\,chain\,or\,loop\right) \geq 2\left(\frac{n-1}{2}\right)pq = \left(n-1\right)pq \tag{10}$$

Better Bounds with Temporary Revelations

In the min weight vertex cover approach we committed to a single set of ground truth revelations and then performed inference using this information. If we use a 'reveal and forget' scheme we can change the original single revelation scheme to a tighter and more explicit one. If, when trying to recover each vertex of G, we are given the ground truth of all of the immediate neighbors, but use these values only for the neighbors reconstruction, rather than forfeiting recovery of revealed vertices, we are left with a better bound that is given by

$$\sum_{v\in V} e_s\left(d\left(v\right);p,q\right) = \left\langle \vec{\#}\left(G\,;\emptyset\right),\,\vec{e_s}\left(p,q\right)\right\rangle \tag{11}$$

i.e. this is the same bound and derivation as before but the revealed set can be treated as an empty set, but the bound is still valid because of the scheme differences. This gives a factor of 2 to our chain or loop recovery, because now we try to recover twice as many vertices. This was earlier suggested by [Foster, Reichman, Sridharan, 2017].

Revisiting the Chain and Loop Bound The new approach follows the same path, but twice as many vertices contribute to the bound, giving

$$e_Y\left(MML\,on\,chain\,or\,loop\right) \geq 2\left(n-1\right)pq \tag{12}$$

Deriving a Tree Lower Bound Using this bound, one can go on to inspect trees. If there are L leaves in a tree, then there are exactly $L-2$ vertices of degree 3 (each inner vertex reducing the amount of connected components from L up until 2) and then one root vertex with degree 2. To reach total number of vertices, we must count all the inner unary vertices (which have a degree of 2), which consist of $n-2L+1$ nodes. hence we get that $\vec{\#}\left(G\,;\emptyset\right) = \langle 0, L, n-2L+2, L-2\rangle$ and assuming $p<q$ and using the lower bounds for e_s, one can use $\left\langle q, p, 2qp, p^2\right\rangle$ for $\vec{e_s}$, which gives

$$e_Y\left(MML\,Tree\,Recovery\right) \geq pL + 2pq\left(n-2L+2\right) + p^2\left(L-2\right) \tag{13}$$

If we write $l = \frac{L}{n}$ (where $l \in (0, 0.5)$) then we get

$$e_Y\left(MML\,Tree\,Recovery\right) \geq np\left(l + 2q\left(1-2l\right) + pl\right) + 2pq \tag{14}$$

for $l \to 0.5$ (few chains) we get $np\left(\frac{1}{2}\left(1+p\right)\right) + 2pq > \frac{np}{2}$. For $l \to 0$ (a chain) we get $2nqp$ as expected, which agrees with the upper bounds result for trees of [Foster, Reichman, Sridharan, 2017].

d Dimensional Grids Lower Bound When the side length n is large enough, the majority of vertices are of degree 2d, specifically, $(n-2)^d$ of them, while the rest are of smaller degree. Hence the lower bound can be given by

$$e_Y\left(MML\,on\,d\,dimensional\,grid\right) \geq e_s\left(2d\,;p,q\right)\cdot\left(n-2\right)^d \tag{15}$$

which for a 2D grid gives $\frac{9}{2}qp^2\left(n-2\right)^2$ which is or the same order of the results in Globerson et al.

3 Upper bound

The best upper bounds to complement the lower bound would be those who are also a linear combination of the degree counts, because the difference between them will give an explicit bound for the gap. Hence, We should use a naive algorithm that ignores information from all vertices that are not immediate neighbors- algorithm $\mathcal{R}$ iterates all the vertices in G and performs MML using the information apparent in the immediate neighborhood of a vertex, i.e. The subgraph that is generated by the vertex, its incident edges and immediate neighbors. Hence, the expected error of such an algorithm will also depend only on the degree of the vertex. It can be shown (Appendix 6.2) that

$$e_Y\left(\mathcal{R}\right) = \left\langle \vec{\#}\left(G\,;\emptyset\right), \vec{e_s}\left(p' = \left(p+q-2pq\right), q\right)\right\rangle \tag{16}$$

which is the same as exact recovery with the edge error rate has increased to $p' = p+q-2pq$ to account for the missing ground truth neighbor vertices. This results with an upper bound on error that is decreasing with p' in the power of d, and if $4p' < 1$ then it is exponentially decreasing with d.

4 The Recovery Margin

The upper and lower bounds introduce a margin or band in which all graphs must have an efficient recovery algorithm, and no algorithm can perform better than the bottom of the band. To make the gap more visible, let's use the e_s bounds, and get that

$$\frac{e_Y\left(\mathcal{R}\right)}{e_Y\left(MML\right)} \leq \left\langle \vec{\#}\left(G\,;\emptyset\right), \frac{\vec{e_s}\left(p' = \left(p+q-2pq\right), q\right)}{\vec{e_s}\left(p,q\right)}\right\rangle \tag{17}$$

Where the division is element-wise. It can be shown (see Appendix) that the e_s ratio is upper bounded by

$$\left(\frac{\vec{e_s}\left(p' = \left(p+q-2pq\right), q\right)}{\vec{e_s}\left(p,q\right)}\right)_d \leq \begin{cases} \frac{d}{2}\left(2\left(p'/p\right)\right)^{\frac{d+1}{2}} & odd\,d \\ \frac{4}{3}\left(1+d\left(p'/q\right)\right)\left(2\left(p'/p\right)\right)^{\frac{d}{2}} & even\,d \end{cases} \tag{18}$$

Though it is tighter, but less explicit. If we write the explicit division result $2\cdot p'/p = 2\cdot\left(1-2q+q/p\right) \geq 1$, assuming again $p < q$, we can see that $\mathcal{R}$ suffers from impaired performance as d grows. Hence, for bounded degree graphs we see that when n increases, the performance of poly time algorithms is always within a constant ratio of the optimal.

5 Discussion

We have shown lower bounds of expected recovery error for exact recovery, and a constructive upper bound for approximate recovery using the algorithm $\mathcal{R}$. We analyzed the error margin of recovery in which exact recovery and all the best polynomial algorithms, including naive $\mathcal{R}$ reside in. We have shown that the lower bound is tight in several cases, such as chains, loops, trees and 2D grids using others' results. The analysis of the star graph recovery with and without known neighboring vertices may be used to further analyze LBP and create propagation schemes that may result in greedy process that always improves the expected error of that vertex's reconstruction, by extending the analysis to stars with nonidentical probabilities, and performing BP only on improving vertices.

6 Appendix

6.1 Analysis of e_s

Explicit expression The main formula is reached by an explicit enumeration of all the possible vertex and edge values, using $I[\cdot]$ as the indicator function, and denoting the center vertex with index 0

$$e_s(d;p,q) = P(X_0=-1)\cdot\sum_{i=0}^{d} P(i\,bad\,edges)\cdot I[i\,bad\,edges\,and\,X_0=-1\,result\,in\,false\,recovery] + \tag{19}$$

$$P(X_0=1)\cdot\sum_{i=0}^{d} P(i\,bad\,edges)\cdot I[i\,bad\,edges\,and\,X_0=1\,result\,in\,false\,recovery] = \tag{20}$$

$$q\cdot\sum_{i=0}^{d}\binom{d}{i}\cdot p^i(1-p)^{d-i}\cdot I\left[p^i\cdot q\cdot(1-p)^{d-i} < p^{d-i}(1-p)^i(1-q)\right] + \tag{21}$$

$$(1-q)\cdot\sum_{i=0}^{d}\binom{d}{i}\cdot p^i(1-p)^{d-i}\cdot I\left[(1-q)\,p^i(1-p)^{d-i} < q\cdot p^{d-i}(1-p)^i\right] = \tag{22}$$

$$\mathbb{E}_{i\sim Bin(d,p)}\left[q\cdot I\left[\frac{q}{1-q} < \left(\frac{p}{1-p}\right)^{d-2i}\right] + (1-q)\cdot I\left[\left(\frac{p}{1-p}\right)^{-(d-2i)} < \frac{q}{1-q}\right]\right] = \tag{23}$$

$$\mathbb{E}_{i\sim Bin(d,p)}\left[q\cdot I\left[\left(\frac{p}{1-p}\right)^{2i} < \left(\frac{p}{1-p}\right)^{d}\left(\frac{q}{1-q}\right)^{-1}\right] + (1-q)\cdot I\left[\left(\frac{p}{1-p}\right)^{2i} < \left(\frac{p}{1-p}\right)^{d}\frac{q}{1-q}\right]\right] = \tag{24}$$

$$\mathbb{E}_{i\sim Bin(d,p)}\left[q\cdot I[2i\cdot\gamma_p < d\cdot\gamma_p - \gamma_q] + (1-q)\cdot I[2i\cdot\gamma_p < d\cdot\gamma_p + \gamma_q]\right] \tag{25}$$

$$\mathbb{E}_{i\sim Bin(d,p)}\left[q\cdot I\left[2i > d - \frac{\gamma_q}{\gamma_p}\right] + (1-q)\cdot I\left[2i > d + \frac{\gamma_q}{\gamma_p}\right]\right] \tag{26}$$

where we applied to log function on both sides of the inequality and defined $\gamma_p = \log\left(\frac{p}{1-p}\right)$ and γ_q similarly, and used the fact that for the domains $p,q\in(0,0.5)$ $\gamma_p,\gamma_q<0$. Because$\gamma_{p,q}$ are strictly monotonous in $p,q\in(0,1)$, hence, $p<q\to\gamma_p<\gamma_q$ but because they are both negative, the result is in $(0,1)$ hence, they only effect the value in equality

$$=\mathbb{E}_{i\sim Bin(d,p)}\left[q\cdot I\left[2i > d - \frac{\gamma_q}{\gamma_p}\right] + (1-q)\cdot I\left[2i > d + \frac{\gamma_q}{\gamma_p}\right]\right] \tag{27}$$

$$\mathbb{E}_{i\sim Bin(d,p)}\left[q\cdot I[2i\geq d] + (1-q)\cdot I[2i>d]\right] = \tag{28}$$

$$\mathbb{E}_{i\sim Bin(d,p)}\left[q\cdot(I[2i>d] + I[2i=d]) + (1-q)\cdot I[2i>d]\right] = \tag{29}$$

$$\mathbb{E}_{i\sim Bin(d,p)}\left[I[2i>d] + q\cdot I[2i=d]\right] = \tag{30}$$

$$\mathbb{E}_{i\sim Bin(d,p)}\left[I[2i>d]\right] + \mathbb{E}_{i\sim Bin(d,p)}\left[q\cdot I[2i=d]\right] = \tag{31}$$

$$\mathbb{P}\left[i>\frac{d}{2}; i\sim Bin(d,p)\right] + q\cdot\mathbb{P}\left[i=\frac{d}{2}; i\sim Bin(d,p)\right] \tag{32}$$

Values of Low Degree e_s The calculation follows a similar path for odd and one for even degrees, as follows

$$e_s(d=1;p,q) = \mathbb{P}[BadEdges=1] = p \tag{33}$$

$$e_s\left(d=2;p,q\right)=\mathbb{P}\left[BadEdges=2\right]+q\mathbb{P}\left[BadEdges=1\right]=p^2+q\cdot 2p\left(1-p\right)=2pq+\left(1-2q\right)p^2 \tag{34}$$

$$e_s\left(d=3;p,q\right)=\mathbb{P}\left[BadEdges=3\right]+\mathbb{P}\left[BadEdges=2\right]=p^3+\binom{3}{2}p^2\left(1-p\right)= \tag{35}$$

$$p^3+3p^2-2p^3=p^2+2\left(p^2-p^3\right)>p^2 \tag{36}$$

$$e_s\left(d=4;p,q\right)=\mathbb{P}\left[BadEdges=4\right]+\mathbb{P}\left[BadEdges=3\right]+q\mathbb{P}\left[BadEdges=2\right]= \tag{37}$$

$$p^4+\binom{4}{3}p^3\left(1-p\right)+q\cdot\binom{4}{2}p^2\left(1-p\right)^2=p^4+4p^3\left(1-p\right)+6q\cdot p^2\left(1-p\right)^2= \tag{38}$$

$$p^4+4\cdot p^3-4\cdot p^4+\left(6\cdot p^2-12\cdot p^3+6\cdot p^4\right)\cdot q= \tag{39}$$

$$p^4\left(6q-3\right)+p^3\left(4-12q\right)+p^2\cdot 6\cdot q=p^2\cdot\left(p^2\left(6q-3\right)+p\left(4-12q\right)+6\cdot q\right)> \tag{40}$$

$$p^2\cdot\left(p^2\left(6q-3\right)+p^2\left(4-12q\right)+6\cdot q\right)=p^2\cdot\left(p^2\left(1-6q\right)+6\cdot q\right)= \tag{41}$$

$$p^2\cdot\left(p^2+6q\left(1-p^2\right)\right)>p^2\cdot\left(p^2+6q\left(1-\left(\frac{1}{2}\right)^2\right)\right)=p^2\cdot\left(p^2+q\frac{9}{2}\right)>\frac{9}{2}p^2q \tag{42}$$

$$e_s\left(d=5;p,q\right)=\mathbb{P}\left[BadEdges=5\right]+\mathbb{P}\left[BadEdges=4\right]+\mathbb{P}\left[BadEdges=3\right]= \tag{43}$$

$$p^5+\binom{5}{4}p^4\left(1-p\right)+\binom{5}{3}p^3\left(1-p\right)^2=p^5+5p^4\left(1-p\right)+10p^3\left(1-p\right)^2= \tag{44}$$

$$\left(10-15\cdot p+6\cdot p^2\right)\cdot p^3>4\cdot p^3 \tag{45}$$

Asymptotic Analysis The lower bound is given by using only the central elements, and central binomial coefficient approximations. We'll calculate odd and even bounds separately. For odd

$$e_s\left(d\,;p,q\right)\geq\underset{EdgeErrors\sim Bin(d,p)}{\mathbb{P}}\left[EdgeErrors=\lceil\frac{d}{2}\rceil\right]= \tag{46}$$

$$\binom{d}{\frac{d+1}{2}}p^{\frac{d+1}{2}}\left(1-p\right)^{\frac{d-1}{2}}>\binom{d}{\frac{d+1}{2}}p^{\frac{d+1}{2}}\left(\frac{1}{2}\right)^{\frac{d-1}{2}}= \tag{47}$$

$$2\binom{d}{\frac{d+1}{2}}\left(\frac{p}{2}\right)^{\frac{d+1}{2}}=\binom{d+1}{\frac{d+1}{2}}\left(\frac{p}{2}\right)^{\frac{d+1}{2}}\geq\frac{4^{\frac{d+1}{2}}}{\sqrt{4\cdot\frac{d+1}{2}}}\left(\frac{p}{2}\right)^{\frac{d+1}{2}}= \tag{48}$$

$$\frac{\left(2p\right)^{\frac{d+1}{2}}}{\sqrt{2\cdot\left(d+1\right)}} \tag{49}$$

For even

$$e_s\left(d\,;p,q\right)\geq q\cdot\underset{EdgeErrors\sim Bin(d,p)}{\mathbb{P}}\left[EdgeErrors=\frac{d}{2}\right]=q\binom{d}{\frac{d}{2}}p^{\frac{d}{2}}\left(1-p\right)^{\frac{d}{2}}\geq \tag{50}$$

$$q\binom{d}{\frac{d}{2}}\left(\frac{p}{2}\right)^{\frac{d}{2}}\geq q\frac{4^{\frac{d}{2}}}{\sqrt{4\cdot\frac{d}{2}}}\left(\frac{p}{2}\right)^{\frac{d}{2}}=q\frac{\left(2p\right)^{\frac{d}{2}}}{\sqrt{2d}} \tag{51}$$

hence, we can create a unified expression

$$e_s\left(d\,;p,q\right) \geq \frac{q}{\sqrt{2\cdot\left(d+1\right)}}\left(2p\right)^{\frac{d+1}{2}} \tag{52}$$

while the upper bound, for odd

$$e_s\left(d\,;p,q\right) \leq \left(d-\lceil\frac{d}{2}\rceil\right)\cdot \underset{EdgeErrors\sim Bin(d,p)}{\mathbb{P}}\left[EdgeErrors=\lceil\frac{d}{2}\rceil\right] = \tag{53}$$

$$\left(\frac{d-1}{2}\right)\binom{d}{\frac{d+1}{2}}p^{\frac{d+1}{2}}\left(1-p\right)^{\frac{d-1}{2}} < \left(\frac{d-1}{2}\right)\binom{d}{\frac{d+1}{2}}p^{\frac{d+1}{2}} = \tag{54}$$

$$\left(\frac{d-1}{4}\right)\binom{d+1}{\frac{d+1}{2}}p^{\frac{d+1}{2}} \leq \left(\frac{d-1}{4}\right)p^{\frac{d+1}{2}}\frac{4^{\frac{d+1}{2}}}{\sqrt{3\frac{d+1}{2}+1}} = \tag{55}$$

$$\frac{\left(d-1\right)\left(4p\right)^{\frac{d+1}{2}}}{\sqrt{24d+40}} \tag{56}$$

and for even

$$e_s\left(d\,;p,q\right) \leq q\cdot \underset{EdgeErrors\sim Bin(d,p)}{\mathbb{P}}\left[EdgeErrors=\frac{d}{2}\right] + \tag{57}$$

$$\left(\frac{d-2}{2}\right)\underset{EdgeErrors\sim Bin(d,p)}{\mathbb{P}}\left[EdgeErrors=\lceil\frac{d+1}{2}\rceil\right] \leq \tag{58}$$

$$q\binom{d}{\frac{d}{2}}p^{\frac{d}{2}}\left(1-p\right)^{\frac{d}{2}} + \frac{\left(d-2\right)\left(4p\right)^{\frac{d+2}{2}}}{\sqrt{24\left(d+1\right)+40}} \leq q\binom{d}{\frac{d}{2}}p^{\frac{d}{2}} + \frac{\left(d-2\right)\left(4p\right)^{\frac{d+2}{2}}}{\sqrt{24d+64}} \leq \tag{59}$$

$$q\frac{4^{\frac{d}{2}}}{\sqrt{3\frac{d}{2}+1}}p^{\frac{d}{2}} + \frac{\left(d-2\right)\left(4p\right)^{\frac{d+2}{2}}}{\sqrt{24d+64}} = \left(\frac{q}{\sqrt{\frac{3d+2}{2}}} + \frac{\left(d-2\right)\left(4p\right)}{\sqrt{24d+64}}\right)\left(4p\right)^{\frac{d}{2}} \tag{60}$$

unifying the expressions result in

$$e_s\left(d\,;p,q\right) \leq \frac{\sqrt{d}\left(4p\right)^{\frac{d+1}{2}}}{\sqrt{24}} \tag{61}$$

6.2 Algorithm $\mathcal{R}$ analysis

The recovery of a single vertex resembles the algorithm with revelation that was used for the lower bound, but doesn't have any revelation. Due to the symmetry of flipping, let's use the all 1 assignment to examine the error. Let's first define $p' \equiv q+p-2pq$. p' corresponds to the probability a ray (edge + vertex) will change it's message to the center from the original message due to noise

$$p' = \left(1-q\right)p + \left(1-p\right)q = p-qp+q-qp = p+q-2pq \tag{62}$$

Because while performing recovery the algorithm cannot distinguish between an edge error or a vertex error (both result in a different 'message' to the center) and cannot distinguish between no error and edge+vertex error (due to the same concept), the contribution to marginal likelihood of a specific assignment to Y_0 (the center vertex) of a single ray is $(1-p')$ if the message agrees with the assignment and p' if the message is different.
Hence, the expected error **of a single vertex recovery** where the recovered vertex is

denoted by index 0, is given by

$$e_Y\left(single\,\mathcal{R}\right)= \tag{63}$$

$$P\left(X_0=-1\right)\cdot\sum_{i=0}^{d}P(i\,rays\,claim\,Y_0=-1)\cdot I\left[\hat{Y_0}\neq 0\,;i\,rays\,claim\,Y_0=-1\,and\,X_0=-1\right]+ \tag{64}$$

$$P\left(X_0=1\right)\cdot\sum_{i=0}^{d}P(i\,rays\,claim\,Y_0=-1)\cdot I\left[\hat{Y_0}\neq 0\,;i\,rays\,claim\,Y_0=-1\,and\,X_0=1\right]= \tag{65}$$

$$q\cdot\sum_{i=0}^{d}\binom{d}{i}\cdot p'^{i}\left(1-p'\right)^{d-i}\cdot I\left[p'^{i}\cdot q\cdot\left(1-p'\right)^{d-i}<p'^{d-i}\left(1-p'\right)^{i}\left(1-q\right)\right]+ \tag{66}$$

$$\left(1-q\right)\cdot\sum_{i=0}^{d}\binom{d}{i}\cdot p'^{i}\left(1-p'\right)^{d-i}\cdot I\left[\left(1-q\right)p'^{i}\left(1-p'\right)^{d-i}<q\cdot p'^{d-i}\left(1-p'\right)^{i}\right] \tag{67}$$

This is equivalent to a reconstruction with revelations where all the ground truths are the same as the middle vertex, and the edge noise is changed from p to p'.

6.3 Gap analysis

The ratio between the expected error of $\mathcal{R}$ and MML can be bounded using the upper bound for $e_s\left(d\,;p',q\right)$ and the lower bound for $e_s\left(d\,;p,q\right)-$

$$\frac{e_s\left(d\,;p',q\right)}{e_s\left(d\,;p,q\right)}\leq\begin{cases}\frac{(d-1)\left(4p'\right)^{\frac{d+1}{2}}}{\sqrt{24d+40}}/\frac{(2p)^{\frac{d+1}{2}}}{\sqrt{2\cdot(d+1)}} & odd\,d\\ \left(\frac{q}{\sqrt{\frac{3d+2}{2}}}+\frac{(d-2)\left(4p'\right)}{\sqrt{24d+64}}\right)\left(4p'\right)^{\frac{d}{2}}/q\frac{(2p)^{\frac{d}{2}}}{\sqrt{2d}} & even\,d\end{cases}= \tag{68}$$

$$\begin{cases}\frac{\sqrt{2\cdot(d+1)}(d-1)\left(2\left(p'/p\right)\right)^{\frac{d+1}{2}}}{\sqrt{24d+40}} & odd\,d\\ \sqrt{2d}\left(\frac{1}{\sqrt{\frac{3d+2}{2}}}+\frac{(d-2)\left(4\left(p'/q\right)\right)}{\sqrt{24d+64}}\right)\left(2\left(p'/p\right)\right)^{\frac{d}{2}} & even\,d\end{cases}= \tag{69}$$

$$\begin{cases}\frac{\sqrt{2\cdot(d+1)}(d-1)\left(2\left(p'/p\right)\right)^{\frac{d+1}{2}}}{\sqrt{24d+40}} & odd\,d\\ \left(\frac{1}{\sqrt{\frac{3+2/d}{4}}}+\frac{(d-2)\left(4\left(p'/q\right)\right)}{\sqrt{\frac{24+64/d}{2}}}\right)\left(2\left(p'/p\right)\right)^{\frac{d}{2}} & even\,d\end{cases}\leq \tag{70}$$

$$\begin{cases}\frac{d\left(2\left(p'/p\right)\right)^{\frac{d+1}{2}}}{2} & odd\,d\\ \frac{4}{3}\left(1+d\left(p'/q\right)\right)\left(2\left(p'/p\right)\right)^{\frac{d}{2}} & even\,d\end{cases} \tag{71}$$

References

[Globerson, Roughgarden, Sontag, Yildirim, 2015] How Hard is Inference for Structured Prediction?Amir Globerson, Tim Roughgarden, David Sontag and Cafer Yildirim International Conference on Machine Learning (ICML) 2015

[Foster, Reichman, Sridharan, 2017] Inference in Sparse Graphs with Pairwise Measurements and Side InformationarXiv preprint arXiv:1703.02728

[J. Yedidia, W. T. Freeman and Y. Weiss 2001] Understanding belief propagation and its generalizations International Joint Conference on Artificial Intelligence (IJCAI 2001), Distinguished Papers Track

[Neil Robertson and Paul D. Seymour] Graph minors. ii. algorithmic aspects of tree-width. Journal of algorithms, 7(3):309-322, 1986

[Robert G. Gallager, 1963] Low Density Parity Check Codes. Monograph, M.I.T. Press. Retrieved August 7, 2013

[Aiden Price and Joanne Hall, 2017] A Survey on Trapping Sets and Stopping Sets, arXiv:1705.05996

[Jon Feldman, Martin J. Wainwright, 2005] Using Linear Programming to Decode Binary Linear Codes, Jon Feldman, Martin J. Wainwright, Member, IEEE, and David R. Karger, Associate Member, IEEE, IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 51, NO. 3, MARCH 2005

[Yair Weiss, 2000] Correctness of Local Probability Propagation in Graphical Models with Loops, Y Weiss, MIT Press, Neural computation, 2000

[Joris M. Mooij, Hilbert J. Kappen, 2007] Sufficient conditions for convergence of the Sum-Product Algorithm, IEEE Transactions on Information Theory, 53(12):4422-4437 Dec. 2007