On consistent vertex nomination schemes

Vince Lyzinski†, Keith Levin‡, and Carey E. Priebe∗

†Department of Mathematics and Statistics, University of Massachusetts Amherst
‡Department of Statistics, University of Michigan
∗Department of Applied Mathematics and Statistics, Johns Hopkins University

May 30, 2018

Abstract

Given a vertex of interest in a network \( G_1 \), the vertex nomination problem seeks to find the corresponding vertex of interest (if it exists) in a second network \( G_2 \). A vertex nomination scheme produces a rank list of the vertices in \( G_2 \), where the vertices are ranked by how likely they are judged to be the corresponding vertex of interest in \( G_2 \). The vertex nomination problem and related information retrieval tasks have attracted much attention in the machine learning literature, with numerous applications in social and biological networks. However, the current framework has often been confined to a comparatively small class of network models, and the concept of statistically consistent vertex nomination schemes has been only shallowly explored. In this paper, we extend the vertex nomination problem to a very general statistical model of graphs. Further, drawing inspiration from the long-established classification framework in the pattern recognition literature, we provide definitions for the key notions of Bayes optimality and consistency in our extended vertex nomination framework, including a derivation of the Bayes optimal vertex nomination scheme. In addition, we prove that no universally consistent vertex nomination schemes exist. Illustrative examples are provided throughout.

1 Introduction

Statistical inference on graphs is an important branch of modern statistics and machine learning. In recent years, there have been numerous papers in the literature developing graph analogues of statistical inference tasks such as hypothesis testing \([5, 62]\), classification \([63, 10]\), and clustering \([34, 52, 59, 46]\), to name a few. Moreover, growth in the size and complexity of network data sets have necessitated techniques for network-specific data mining tasks such as link prediction \([29, 31]\); entity resolution and network alignment \([13, 35]\); and vertex nomination \([15, 14, 60, 21, 37]\). Akin to the development of classical statistics, algorithmic advancement has, in many ways, outpaced theoretical developments in these emerging graph-driven domains. This development has been necessitated by the dizzying pace of data generation, but there is nevertheless the need for a firm theoretical context in which to frame algorithmic progress. Toward this end, in this paper, drawing inspiration from the long-established classification framework in the pattern recognition literature \([16]\), we provide a rigorous theoretical framework for understanding statistical consistency in the vertex nomination (VN) inference task.

The inference task in vertex nomination, which can be viewed as the graph analogue of the more classical recommender system task \([50]\), has traditionally been stated as follows: given a community of interest in a network and some examples of vertices that are or are not part of a community of interest, vertex nomination seeks to rank the remaining vertices in the network into a nomination list, with those vertices from the community of interest (ideally) concentrating at the top of the nomination
Figure 1: A visual representation of the classical Vertex Nomination framework: Given a community of interest in a network (here the red community) and some examples of vertices that are/are not part of the community of interest (colored red and green, respectively), rank the remaining vertices in the network into a nomination list, with those vertices from the community of interest concentrating at the top of the nomination list.

list. See Figure 1 for a visual representation of this classical Vertex Nomination framework. In limited resource settings, vertex nomination tools have proven to be effective in efficiently searching/querying large networks, with applications of the methodology including detecting fraudsters in the Enron email network [15, 42, 60], uncovering web advertisements that have association with human trafficking [21], and identifying latent structure in connectome data [21, 65], among others.

While related to the community detection problem [45, 34, 8, 46], this traditional formulation of the VN problem is a semi-supervised inference task whose output is not an assignment of vertices to communities, but rather ranked estimates of which vertices belong to a particular community of interest. That is, in contrast to community detection, the VN problem does not attempt to recover the community memberships of any vertices not in the community of interest. Clearly, any method that can recover the community memberships of all vertices in a graph can recover the interesting community, and hence any community detection algorithm can be repurposed for the VN problem just described with minor adaptation (e.g., by ranking vertices according to their probability of membership in the community of interest); see, for example, the spectral vertex nomination scheme of [21]. The specific performance of such an adaptation is highly dependent on the fidelity of the base clustering procedure, and the performance is often below that of the semi-supervised VN specific analogues; see [65].

The above formulation of the VN task assumes the presence of strong community structure among the vertices of interest in the graph. In practice, this is often a reasonable assumption, particularly if it is expected that interesting vertices will behave similarly to one another in the network. However, the particular features that define a vertex as interesting are entirely task-dependent. To paraphrase the common proverb, interestingness is in the eye of the practitioner. Interesting vertices may be, for example, those with large network centrality [25, 44], those with a particular role in the network [33], or those corresponding to a given user across social networks [47]. In these applications, interesting vertices need not correspond precisely to the community structure captured by a generative network model, and hence such cases are ill-described by the community-based VN problem described above. To accommodate this task-dependency and broader notion of interesting vertices, we consider the following generalization and extension of the previously-presented VN problem: Given a vertex of interest $v^*$ in a graph $G_1 = (V_1, E_1)$, find the corresponding vertex of interest $u^*$ (if it exists) in a second graph $G_2 = (V_2, E_2)$ by ranking the vertices of $G_2$ according to our confidence that they correspond to $v^*$ in graph $G_1$; see Figure 2 for a visual representation of this VN framework. In this formulation, which is an (potentially) unsupervised inference task, what defines $v^*$ as interesting is entirely model-dependent,
Figure 2: A visual representation of the generalized Vertex Nomination framework: Given a vertex of interest $v^*$ (colored red) in a graph $G_1 = (V_1, E_1)$, find the corresponding vertex of interest $u^*$ (if it exists) in a second graph $G_2 = (V_2, E_2)$, ranking the vertices of $G_2$ into a nomination list so that $u^*$ ideally appears at the top of the nomination list.

and different network models can highlight different characteristics of interest in the graph. Potential application domains for this VN generalization abound, including identifying users of interest across social network platforms, see for example [47]; identifying structural signal across connectomes, see for example [58]; and identifying topics of interest across graphical knowledge bases, see for example [57]; among others.

In [21] and [37], the notion of a consistent vertex nomination scheme (i.e., an asymptotically optimal solution to the VN problem) was proposed for the original formulation of the VN problem, in which community membership entirely determines whether or not a given vertex is interesting. This definition of consistency was based on the mean average precision (MAP) of a nomination scheme operating on a graph model with explicit community structure (the Stochastic Block Model, or SBM, of [24]). Under this restricted notion of consistency, [21] derived the analogue of universal Bayes optimality in the VN setting, namely a scheme that achieves the optimal mean average precision for all parameterizations of the underlying SBM. While this derivation of the Bayes optimal scheme somewhat parallels the derivation of the Bayes optimal classifier in the classical pattern recognition literature, the SBM model assumption and MAP formulation greatly narrow the set of models and sets of interesting vertices we can consider. In this paper, we revamp and generalize the concept of VN consistency—and of VN Bayes optimality—in the two-graph VN framework. This framework is quite general, and further allows us to highlight the similarities and differences between our new VN consistency formulation and the classification consistency analogue defined in, for example, [16].

This paper is laid out as follows. In the remainder of this section, we provide brief overviews of information retrieval as it relates to vertex nomination (Section 1.1) and the Bayes optimal classifier in the classical setting (Section 1.2), and conclude the introduction by establishing notation for the remainder of the paper (Section 1.3). In Section 2, we define the VN problem framework that is the focus of this paper, and in Section 3 we derive the VN analogue of a Bayes optimal scheme. In Section 4 we define a new notion of VN consistency, and we prove that no universally consistent VN scheme exists, providing an interesting contrast to the standard classification setting. We conclude in Section 5 with a short summary comparing and contrasting VN with classical classification and a discussion of
implications and future directions.

1.1 Connections to information retrieval

The vertex nomination task is, in some ways, similar to the task faced by recommender systems [49, 50], in which the aim is to retrieve objects (e.g., documents or images) likely to be of interest to a user based on his or her previous behavior. For example, the celebrated PageRank algorithm [9] recommends webpages based on random walks on the world wide web graph, in which websites are nodes and (directed) edges reflect hyperlinks between pages. The information retrieval (IR) literature includes many such graph-based approaches. We refer the reader to [50] and [43] for the state of the art circa 2010, and concentrate here on recapping more recent graph-based information retrieval techniques.

Many graph-based IR techniques rely on the assumption that similar objects (i.e., documents, webpages, etc) will lie near one another in a suitably-constructed graph, an intuition underlying many graph-based approaches throughout machine learning and related disciplines; see, for example, [6, 67]. Techniques along these lines have been applied toward many tasks in natural language processing, typically inspired by PageRank [53]. Along similar lines, [40] applies diffusion on the world wide web graph [12] to yield an approach to ranking for query completion and recommendation. These information retrieval techniques can be naturally adapted to the vertex nomination problem by treating the vertex or vertices of interest as the object or objects to be retrieved.

The vertex nomination problem also bears similarities to the task of learning to rank [17, 30, 28], in which the goal is to learn an ordering on a set of objects (i.e., documents, images, videos, etc.) according to (estimated) similarity or relevance to a given query object. In the learning to rank literature, graphs usually appear as training instances, with nodes corresponding to objects and edges encoding preferences or similarities among them elicited from users (e.g., an undirected weighted edge may join two documents judged to be similar, or a directed edge may point from one image to another, more preferable image). The work in [1] is among the first to consider the problem of ranking objects in a network. The authors modified the PageRank algorithm to take preference information into account, rather than working solely with the hyperlink graph. In [2], the authors use a data graph encoding object similarities to obtain a regularizer similar to [7] on the empirical ranking error, with the target ranking encoded in a preference graph. More recent efforts along these lines have focused on the problem of incorporating network structure present between entities of different types, for example, between users and events in a social network [32, 48]. Here again, any learning to rank algorithm has a natural adaptation to the VN problem by using the first graph, in which some vertices are labeled, as training data to learn a ranking on the vertices of the second graph.

1.2 Bayes error in classical pattern recognition

In this section, we recall the analogous concepts of consistency and Bayes error in the context of statistical classification. We do not aim to give an exhaustive overview of the subject, but only to provide a rough outline as to the structures that we would like to replicate in the context of vertex nomination. For a more thorough treatment of statistical classification, we refer the interested reader to [16], whose presentation we follow below.

We begin by recalling the classical definition of Bayes error. Note that we will restrict our attention to the two-class problem to maximally bring forth the similarity (and differences) between statistical classification and VN, as in VN vertices are either of interest or not.

**Definition 1.** Consider a set of potential observations $\mathcal{X}$ and a set of unknown class labels $\{0, 1\}$ for objects in $\mathcal{X}$. A classifier is a function $h : \mathcal{X} \rightarrow \{0, 1\}$, which aims to predict the class label of a given observation in $\mathcal{X}$. Given a distribution $F$ supported on $\mathcal{X} \times \{0, 1\}$, the error for the classifier $h$ is given by $L(h) = P(h(X) \neq Y)$ where $(X, Y) \sim F$. 

4
The best possible classifier is the one that achieves the lowest possible error and is denoted as the Bayes optimal classifier. We write \( h^* \) for any such optimal classifier, which by definition satisfies
\[
L^* \equiv L(h^*) = \min_{h: \mathcal{X} \to \{0,1\}} L(h).
\]

It is easily seen in this two-class framework that the Bayes optimal classifier is given by
\[
h^*(x) = \begin{cases} 1 & \text{if } \mathbb{E}(Y | X = x) = \mathbb{P}(Y = 1 | X = x) > 1/2; \\ 0 & \text{else}. \end{cases}
\]

Practically speaking, the Bayes optimal scheme chooses the label which maximizes the class-conditional probability of the observed data. The corresponding error, \( L^* = L(h^*) \), is called the Bayes error. Of course, \( h^* \) depends on the distribution \( F \) of \((X,Y)\), and, when appropriate, we will make this dependence explicit by writing \( L^*_F \).

In practice, a classifier is often constructed based on training data \((X_1,Y_1),(X_2,Y_2), \ldots, (X_n,Y_n)\), where the data \((X_i,Y_i)\) are drawn i.i.d. according to \( F \). This supervised classification framework is defined as follows.

**Definition 2.** Consider a set of potential observations \( \mathcal{X} \) and a set of unknown class labels \( \{0,1\} \) for objects in \( \mathcal{X} \). A (supervised) classifier is a function
\[
h_n : \mathcal{X} \times \{ \mathcal{X} \times \{0,1\} \}^n \rightarrow \{0,1\},
\]
which aims to predict the class label of a given observation in \( \mathcal{X} \) based on \( n \) training observations \((x_1,y_1),(x_2,y_2), \ldots, (x_n,y_n) \) drawn i.i.d. from \( X \times \{0,1\} \). Given a distribution \( F \) supported on \( \mathcal{X} \times \{0,1\} \), the error for the classifier \( h_n \) is given by
\[
L_F(h_n) = \mathbb{P}[h_n(X,(X_i,Y_i)_{i=1}^n) \neq Y \mid (X_i,Y_i)_{i=1}^n]
\]
where \((X,Y), (X_1,Y_1), (X_2,Y_2), \ldots, (X_n,Y_n) \) are drawn i.i.d. from \( F \).

Note that \( L_F(h_n) \) is a random variable in which \( \{(X_i,Y_i)\}_{i=1}^n \) are drawn i.i.d. from \( F \), but then held fixed as we average over \((X,Y) \sim F\).

A sequence of classifiers \( \mathbf{h} = (h_n)_{n=1}^{\infty} \) is called a classification rule. Informally, a good classification rule is one for which the probability of error becomes arbitrarily close to Bayes optimal as \( n \to \infty \). The precise nature of what we mean by close is codified in the concept of statistical consistency.

**Definition 3.** A classification rule \( \mathbf{h} = (h_n)_{n=1}^{\infty} \) is consistent with respect to \( F \) if
\[
\mathbb{E}_F(L(h_n)) \to L^*_F.
\]
The rule \( \mathbf{h} \) is strongly consistent if
\[
L_F(h_n) \xrightarrow{a.s.} L^*_F.
\]
A rule that is (strongly) consistent for all distributions \( F \) on \( \mathcal{X} \times \{0,1\} \) is called (strongly) universally consistent.

Perhaps surprisingly, given that \( F \) can have arbitrary structure on \( \mathcal{X} \times \{0,1\} \), universally consistent classification rules exist; see [56] for the first proof of this phenomena.

In [21], a notion of consistency for vertex nomination was presented, roughly analogous to Definition 3. In contrast to the classification task presented above, vertex nomination requires a ranking of the vertices, rather than merely the classification of a single vertex. As such, a vertex nomination scheme is evaluated with respect to a provably optimal canonical nomination scheme. This canonical scheme plays an analogous role of Bayes optimal classifiers in this restricted model framework (see Section 3 below). The goal of this paper is to explore and further develop a broader notion of VN consistency that encompasses a more expressive class of models than the SBM.
1.3 Notation and background

We conclude this section by establishing notation and reviewing a few of the more popular statistical network models that we will make use of as examples in the sequel.

1.3.1 Notation

For a set \( S \), we let \(|S|\) denote its cardinality and \( \binom{S}{2} \) denote the set of all unordered pairs of distinct elements from \( S \). Throughout, we will denote graphs via the ordered pair \( G = (V, E) \), with vertices \( V \) and edges \( E \subseteq \binom{V}{2} \). All graphs considered herein will be labeled, hollow (i.e., containing no self-edges), and undirected. Given a graph \( G \), we will let \( V(G) \) denote the vertices of \( G \) and \( E(G) \) denote its edges. We note that when \( G \) is random, this latter set is a random subset of \( \binom{V}{2} \). For a set of vertices \( S \subseteq V(G) \), we let \( G[S] \) denote the subgraph of \( G \) induced by \( S \), i.e., the graph \( G' = (S, E') \) with \( \{u, v\} \in E \) if and only if \( \{u, v\} \in E(G) \). For a positive integer \( n \in \mathbb{Z} \), we will define \( [n] = \{1, 2, \ldots, n\} \), and \( G_n \) to be the set of labeled graphs on \( n \) vertices. Throughout this paper, we will often, in order to simplify notation, suppress dependence of parameters on \( n \). Throughout, the reader should assume that, unless specified otherwise, all parameters depend on the number of vertices \( n \).

1.3.2 Stochastic block models

The stochastic block model (SBM) is a widely studied model for edge-independent random graphs with latent community structure [24, 23, 26].

**Definition 4.** We say that a random graph \( G = (V, E) \in G_n \) is an instantiation of a stochastic block model with parameters \((K, B, b)\), written \( G \sim \text{SBM}(K, B, b) \), if

i. \( V \) is partitioned into \( K \) classes (usually called communities or blocks), \( V = V_1 \cup V_2 \cup \cdots \cup V_K \).

ii. The block membership vector \( b \in [K]^{|V|} \) is such that for all \( k \in [K] \), \( b_v = k \) if and only if \( v \in V_k \).

iii. The symmetric matrix \( B \in [0, 1]^{K \times K} \) denotes the edge probabilities between and within blocks, with

\[
1\{\{u, v\} \in E(G)\} \overset{\text{ind.}}{\sim} \text{Bernoulli}(B_{b_u, b_v}).
\]

We note that when \( K = 1 \), we recover the Erdős-Rényi random graph [18], in which the edges of \( G \) are present or absent independently with probability \( p \). In this special case, we write \( G \sim \text{ER}(n, p) \). By a slight abuse of notation, for a symmetric matrix \( P \in [0, 1]^{n \times n} \), we will write \( G \sim \text{ER}(P) \) if, identifying the vertices of \( G \) with \( [n] \), we have \( \{i, j\} \in E(G) \) with probability \( P_{i,j} \) independent of the other edges. With no restrictions on \( P \), \( \text{ER}(P) \) random graphs can be viewed as \( n \)-block SBMs and are the most general edge-independent random graph model.

The latent community structure inherent to SBMs makes them a natural model for use in the traditional vertex nomination framework. Recall the traditional VN task: given a community of interest in a network and some examples of vertices that are/are not part of the community of interest, vertex nomination seeks to rank the remaining vertices in the network into a nomination list, with those vertices from the community of interest (ideally) concentrating at the top of the nomination list. As a result, previous work on VN consistency [21] has been posed within the SBM framework, with the optimal scheme only obtaining its optimality for SBMs. We note that we consider herein the SBM setting where communities are disjoint and each vertex can only belong to a single community. However, the results contained herein translate immediately to the mixed membership SBM setting [3]; details are omitted for brevity.
1.3.3 Random dot product graphs

In stochastic block models, the block assignment vector can be viewed as a latent feature vector for the vertices in the network, with these features (i.e., block memberships) defining the connectivity structure in the network. The random dot product graph (RDPG) model [66] allows for more nuanced vertex features to be incorporated into the model and has been used as the setting for a VN formulation similar to the one proposed here; see [47] for details.

**Definition 5.** We say that a random graph $G = (V, E) \in \mathcal{G}_n$ is an instantiation of a $d$-dimensional random dot product graph with parameters $X$, written $G \sim \text{RDPG}(X)$, if

1. The matrix $X \in \mathbb{R}^{n \times d}$ is such that $0 \leq (XX^T)_{i,j} \leq 1$ for all $i, j \in [n]$. The rows of $X$ provide the latent features for the vertices in $V$.

2. The edges of $G$ are present or absent independently, with $\{i, j\} \in E(G)$ with probability $(XX^T)_{i,j}$. Written succinctly, $G \sim \text{ER}(XX^T)$.

We can view the RDPG model as an example of the more general latent position random graph model [23], in which edge probabilities are determined by hidden vertex-level geometry. Estimating the latent position structure in RDPGs is particularly amenable to spectral methods, and this model has played a prominent role in recent theoretical developments of spectral graph methods, see for example [52, 59, 61]. Note that the RDPG can be extended to a broader class of models, in which edge probabilities are given by evaluating a positive definite link function at vertices’ latent positions as in, for example, [63]. While incorporating this more general family of latent position graphs into the present VN framework would be straightforward, we restrict our focus to the RDPG model of Definition 5 for ease of exposition.

1.3.4 Correlation across networks

The vertex nomination problem we consider in this paper presupposes the existence of a vertex of interest in a network $G_1$ and, ideally, a corresponding vertex of interest in a second network $G_2$. Often, such correspondences across networks are encoded into random graph models via edge-wise graph correlation, see for example [20]. Arguably the simplest such structure is seen in the $\rho$-correlated Erdős-Rényi model, see for example [36].

**Definition 6.** We say that bivariate random graphs $(G_1, G_2) \in \mathcal{G}_n \times \mathcal{G}_n$ are an instantiation of a $\rho$-correlated ER$(P)$ model, written $(G_1, G_2) \sim \rho$-ER$(P)$, if

1. Marginally, $G_1 \sim \text{ER}(P)$ and $G_2 \sim \text{ER}(P)$.

2. Edges are independent across $G_1$ and $G_2$ except that the indicators of the events $\{u, v\} \in E(G_1)$ and $\{u, v\} \in E(G_2)$ are jointly distributed as a pair of Bernoulli random variables with success probability $P_{u,v}$ and correlation $\rho$. If the correlation is allowed to vary across edges, so that these two events have correlation $\rho_{u,v}$, then collecting these correlations in a symmetric matrix $R = [\rho_{i,j}]_{i,j=1}^n$, we write $(G_1, G_2) \sim R$-ER$(P)$, see [78].

Ranging the values in $R$ from 0 to 1 allows for the consideration of graphs that range from independent ($R = 0$) to isomorphic ($R = 1$). Intermediate values of $R$ allow for the encoding of a correspondence across networks between these two extremes. We will also consider $R < 0$, in which case edges across networks are anti-correlated. This is particularly useful for modeling situations in which corresponding vertices stochastically behave differently across networks.
2 Vertex Nomination

Loosely stated, the vertex nomination problem we consider in this paper can be summarized as follows: Given a vertex of interest \( v^* \) in a graph \( G_1 = (V_1, E_1) \), find the corresponding vertex of interest \( v^* \) (if it exists) in a second graph \( G_2 = (V_2, E_2) \) by ranking the vertices of \( G_2 \) according to our confidence that they correspond to \( v^* \) in graph \( G_1 \). To formally define this version of vertex nomination, we will need to consider distributions on \( G \) of labeled graphs on \( n \) vertices, with vertex labels given by \( \{v_1, v_2, \ldots, v_n\} \), and \( G \) is the set of labeled graphs on \( m \) vertices, with vertex labels given by \( \{u_1, u_2, \ldots, u_m\} \). Note that for \( i \in [n] \) \( \cap \) \( [m] \), \( v_i \) and \( u_i \) are merely vertex labels, and it is not necessarily the case that \( v_i = u_i \). We follow this labeling convention in order to emphasize the reality that the vertex sets of \( G_1 \) and \( G_2 \) may only partially overlap.

**Definition 7** (Nominatable Distributions). We define the family of Nominatable Distributions, which we denote \( \mathcal{N} \), to be the family of distributions

\[
\mathcal{N} = \{ F_{c,n,m,\theta} \text{ s.t. } n, m \in \mathbb{Z}^+, \theta \in \Theta, \}
\]

where \( F_{c,n,m,\theta} \) is a distribution on \( \mathcal{G}_n \times \mathcal{G}_m \) parameterized by \( \theta \in \Theta \) and satisfying:

i. The vertex sets \( V_1 = \{v_1, v_2, \ldots, v_n\} \) and \( V_2 = \{u_1, u_2, \ldots, u_m\} \) satisfy \( v_i = u_i \) for \( 0 < i \leq c \). We refer to \( C = \{v_1, v_2, \ldots, v_c\} = \{u_1, u_2, \ldots, u_c\} \) as the core vertices. These are the vertices that are shared across the two graphs and imbue the model with a natural notion of corresponding vertices.

ii. Vertices in \( J_1 = V_1 \setminus C \) and \( J_2 = V_2 \setminus C \), satisfy \( J_1 \cap J_2 = \emptyset \). We refer to \( J_1 \) and \( J_2 \) as junk vertices. These are the vertices in each graph that have no corresponding vertex in the other graph.

iii. The induced subgraphs \( G_1[J_1] \) and \( G_2[J_2] \) are conditionally independent given \( \theta \).

A few examples will serve to illustrate this definition. We will return to the three example settings below several times throughout the rest of the paper in order to highlight and illustrate phenomena of interest.

**Example 8** (\( R \)-ER(\( P \))). Let \( (G_1, G_2) \sim R \)-ER(\( P \)) with \( P, R \in \mathbb{R}^{n \times n} \) and \( R > 0 \) entrywise, so that \( G_1 \) and \( G_2 \) have correlated edges as described in Section 1.3.4. In this example, the model parameter is \( \theta = (P, R) \), and the vertex sets of the two graphs can be thought of as fully overlapping, i.e., \( V_1 = V_2 = C = \{n\} \) with \( J_1 = J_2 = \emptyset \), since the correlation structure conveyed in the entries of \( R \) encodes an explicit correspondence between the edges of \( G_1 \) and the edges of \( G_2 \) (and hence also a correspondence between \( V_1 \) and \( V_2 \)). Note that if we consider \( C = [k] \) with \( k < n \), then we would require (after suitably ordering the vertices) \( R_{u,v} = 0 \) for \( u, v > k \). This highlights the way in which \( \theta \) (and hence the distribution \( F_{c,n,m,\theta} \)) can vary with \( c \), and vice-versa.

**Example 9** (RDPG). Let \( m > n \) and suppose that \( Y \in \mathbb{R}^{m \times d} \) has distinct rows and satisfies \( (YY^T)_{i,j} \in [0,1] \) for all \( i, j \in [m] \). Let \( X \in \mathbb{R}^{n \times d} \) be a submatrix of \( Y \), and consider \( G_1 \sim \text{RDPG}(X) \) and \( G_2 \sim \text{RDPG}(Y) \), where \( G_1 \) and \( G_2 \) are conditionally independent given \( Y \). In this example, we can consider \( \theta = Y \), \( V_1 = \{n\} = C \), \( J_1 = \emptyset \), \( V_2 = \{n\} \cup J_2 \), and \( J_2 = \{u_{n+1}, u_{n+2}, \ldots, u_m\} \). Note that as \( G_1 \) and \( G_2 \) are conditionally independent given \( \theta \), we could also consider \( 0 < c < n \) here as well. This illustrates that \( \theta \) need not necessarily vary with \( c \), and hence \( F_{c,n,m,\theta} \) need not vary with \( c \), either.

**Example 10** (Independent Erdős-Rényi graphs). Let \( (G_1, G_2) \) be independent ER(\( n, p \)) random graphs. In this example, we can consider any \( c \in [n] \). Note that if \( c = 0 \) here, then there is no corresponding vertex of interest in \( G_2 \), and this example serves as a natural boundary case between models in which
nomination is possible and those in which it is not. As we will see below in Theorem 26, \( c > 0 \) may still yield chance performance for any nomination scheme, and the existence of a vertex correspondence does not necessarily imply any performance guarantees.

**Remark 11.** In addition to the edge-independent and conditionally edge-independent network models considered above, the class of nominatable distributions contains a host of other popular random graph models, including the Exponential Random Graph Model [23, 54, 51], the preferential attachment model [4], and the Watts-Strogatz small world model [64], among others. Indeed, if we consider the case where \( c = n = m \), then any parametric distribution on \( \mathcal{G}_n \times \mathcal{G}_n \) is a nominatable distribution.

**Remark 12.** The core vertices \( C \) in a nominatable distribution correspond to the vertices that can be sensibly identified across graphs. Note that this set does not require any further structure, aside from the conditional independence of \( G_1[J_1] \) and \( G_2[J_2] \) given the parameter \( \theta \). Thus, we are largely free to specify any notion of correspondence we please. Depending on the application, this correspondence may be that of vertices playing similar structural roles, belonging to the same community, or some more complicated application-specific notion of correspondence. That is, the notion of cross-graph correspondence, and hence the notion of vertex similarity, is largely left to the practitioner to specify when she or he specifies an appropriate random graph model.

Given a pair of graphs \((G_1, G_2) \sim F_{c,n,m,\theta} \in \mathcal{N}\), if the vertices in \( C \) are known across graphs then identifying the corresponding vertex to \( v^* \in C \) is immediate from the vertex labels. In practice, this information is unknown and the correspondences across graphs are only partially observed or even unobserved entirely. To model this added uncertainty, we consider passing the vertex labels of \( G \) through an obfuscating function.

**Definition 13.** Let \((G_1, G_2) \sim F_{c,n,m,\theta} \in \mathcal{N}\). An obfuscating function \( \sigma : V_2 \to W \) is a bijection from \( V_2 \) to \( W \) with \( W \cap V_i = \emptyset \) for \( i = 1, 2 \).

Here, \( \sigma \) models the practical reality that the correspondence of labels across graph is unknown a priori. Note that to ease notation, we shall write \( \sigma(G_2) \) (resp., \( \sigma(g_2) \) and \( \sigma(G_m) \)) to denote the graph \( G_2 \) (respectively, \( g_2 \) and \( G_m \)) whose labels have been obfuscated via \( \sigma \).

Before defining a VN scheme, we must make one additional definition: for a graph \( g \in G_m \) and \( u \in V(g) \), define

\[
\mathcal{I}_g(u) = \{w \in V(g) \text{ s.t. } \exists \text{ an automorphism } \sigma \text{ of } g, \text{ s.t. } \sigma(u) = w\}.
\]

Note that by taking \( \sigma \) to be the identity, we have \( u \in \mathcal{I}_g(u) \). The vertices in \( \mathcal{I}_g(u) \) are those that are, in a sense, topologically equivalent to the vertex \( u \) in \( g \), and hence, in the absence of labels, indistinguishable from one another. As such, any sensibly-defined vertex nomination scheme should view all vertices in \( \mathcal{I}_g(u) \) as being equally good matches to a vertex of interest \( v^* \). Thus, a well-defined VN scheme should be “label-independent” in the following sense: The set of ranks of each set of equivalent vertices (i.e., each \( \mathcal{I}_{g_2}(u) \)) needs to be invariant to the particular choice of obfuscating function; see Figure 3 for an illustration of this consistency criterion. Formally, we have the following.

**Definition 14** (Vertex Nomination (VN) Scheme). Let \( \mathcal{O} \) be the set of all obfuscating functions \( \sigma : V_2 \to W \) for a fixed \( W \). For a set \( A \), let \( \mathcal{T}_A \) denote the set of all total orderings of the elements of \( A \). For \( n, m > 0 \) fixed, a vertex nomination scheme is a function \( \Phi : \mathcal{G}_n \times \mathcal{G}_m \times \mathcal{O} \times V_1 \to \mathcal{T}_W \) satisfying the following consistency property: If for each \( u \in V_2 \), we define \( \text{rank}_{\Phi(g_1, \sigma(g_2), v^*)}(\sigma(u)) \) to be the position of \( \sigma(u) \) in the total ordering provided by \( \Phi(g_1, \sigma(g_2), v^*) \), and we define \( \mathcal{R}_\Phi : \mathcal{G}_n \times \mathcal{G}_m \times \mathcal{O} \times V_1 \times 2^{V_2} \to 2^{[m]} \) via

\[
\mathcal{R}_\Phi(g_1, g_2, \sigma, v^*, S) = \{\text{rank}_{\Phi(g_1, \sigma(g_2), v^*)}(\sigma(u)) \text{ s.t. } u \in S\},
\]

9
Figure 3: An illustration of the “label-independence” property of VN schemes. If the blue vertex in \( o_1(g_2) \) (resp., \( o_2(g_2) \)) is \( o_1(u) \) (resp., \( o_2(u) \)) for \( u \in V_2 \), then we require the ranks of \( I_{o_1(g_2)}(o_1(u)) \) (outlined in red in the network \( o_1(g_2) \) and colored red/blue in the ordering provided by \( \Phi \)) to be equal to the ranks of \( I_{o_2(g_2)}(o_2(u)) \) (outlined in grey in the network \( o_2(g_2) \) and colored grey/blue in the ordering provided by \( \Phi \)). Indeed, the set of ranks of \( o(I_{g_2}(u)) \) via \( \Phi \) is independent of the choice of obfuscation function \( o \).

then we require that for any \( g_1 \in G_n, g_2 \in G_m, v^* \in V_1 \), obfuscating functions \( o_1, o_2 \in \mathcal{O} \) and any \( u \in V(g_2) \),

\[
\tau_{\Phi}(g_1, g_2, o_1, v^*, I_{g_2}(u)) = \tau_{\Phi}(g_1, g_2, o_2, v^*, I_{g_2}(u))
\]

\[
\Leftrightarrow o_2 \circ o_1^{-1}(I_{o_1(g_2)}(\Phi(g_1, o_1(g_2), v^*)[k])) = I_{o_2(g_2)}(\Phi(g_1, o_2(g_2), v^*)[k]) \quad \text{for all } k \in [m],
\]

where \( \Phi(g_1, o(g_2), v^*)[k] \) denotes the \( k \)-th element (i.e., the rank-\( k \) vertex) in the ordering \( \Phi(g_1, o(g_2), v^*) \).

We let \( V_{n,m} \) denote the set of all such VN schemes.

Given \( (G_1, G_2) \sim F_{c,n,m,\theta} \in \mathcal{N} \) realized as \( G_1 = g_1 \) and \( G_2 = g_2 \) with \( v^* \in V_1 \) the vertex of interest in \( G_1 \), a VN scheme \( \Phi(\cdot, \cdot, \cdot) \) produces a ranked list \( \Phi(g_1, o(g_2), v^*) \) of the vertices of \( o(g_2) \) (i.e., the set \( W \)), ordered according to how likely each vertex in \( V(o(g_2)) \) is judged to correspond to \( v^* \), with optimal performance corresponding to

\[
\Phi(g_1, o(g_2), v^*)[1] = \begin{cases} 
  o(v^*) & \text{if } v^* \in C \\
  \text{arbitrary } v \in W & \text{if } v^* \notin C.
\end{cases}
\]

Less formally, one can think of a VN scheme as ranking the vertices of \( G_2 \) according to how well they resemble the vertex of interest \( v^* \) according to some task-dependent measure.

**Remark 15.** Note that if \( u \in V_2 \) is such that \( I_{g_2}(u) = \{u\} \) (i.e., \( u \) is topologically distinct within \( g_2 \)), then Equation 2 implies that

\[
\text{rank}_{\Phi(g_1, o_1(g_2), v^*)}(o_1(u)) = \text{rank}_{\Phi(g_1, o_2(g_2), v^*)}(o_2(u))
\]
a. Well-defined scheme $\Phi$

b. Not well-defined scheme $\Phi$

Figure 4: An example of the consistency criterion, Equation (2), in action. The left panel (a) shows a well-defined nomination scheme while the right panel (b) shows an ill-defined scheme. The key in this example is that, as $I_{g_2}(1) = \{1, 4\}$ and $I_{g_2}(2) = \{2, 3\}$, any scheme satisfying Equation (2) must have $r_{\Phi}(g_1, g_2, o_1, v^*, \{1, 4\}) = r_{\Phi}(g_1, g_2, o_2, v^*, \{1, 4\})$ and $r_{\Phi}(g_1, g_2, o_1, v^*, \{2, 3\}) = r_{\Phi}(g_1, g_2, o_2, v^*, \{2, 3\})$

for any $o_1, o_2$ in $D$. If $I_{g_2}(u)$ contains vertices in addition to $u$, then Equation (2) implies that the set of vertices topologically equivalent to $u$ (namely, those in $I_{g_2}(u)$) must achieve the same ranks via $\Phi$ under any two obfuscating functions; see Figure 4 for a simple example of this consistency criterion in action.

Remark 16 (Relation to [21, 37]). Recall the one-graph vertex nomination task considered in earlier works [14, 21, 37] and described in Section 1, in which vertices are considered interesting precisely when they belong to one of $K$ communities in a stochastic block model. While the two-graph VN formulation we consider in the present work involves a single vertex of interest across graphs, the framework is easily extended to the setting where one may have multiple vertices of interest (and not of interest), and in particular it can encode instances of the one-graph version VN problem. To see this, consider an instance of the single-graph VN problem on graph $G = (V, E)$ where $V$ is partitioned into $K$ communities as $V = V_1 \cup V_2 \cup \cdots \cup V_K$ and each of the communities is comprised of labeled (i.e., seed vertices, whose community memberships are observed) and unlabeled (i.e., nonseed, whose community memberships are unobserved) vertices $V_k = S_k \cup U_k$ where $S_k \subseteq V_k$ is the set of seeds from the $k$-th block and $U_k \subseteq V_k$ is the set of nonseed vertices. We can encode this one-graph VN instance as an instance of the two-graph problem by encoding additional information in the graph $G_1$. Construct a vertex set $V' = V \cup \{\ell_1, \ell_2, \ldots, \ell_K\}$. The $K$ new vertices $\{\ell_k\}_{k=1}^K$ will encode the label information present in the graph $G$. Let $E' = E \cup L$, where $L = \{\ell_k, s) : s \in S_k, k \in [K]\}$, so that edges connect from seed vertices in $S = S_1 \cup S_2 \cup \cdots \cup S_K$ to their corresponding label vertices. Take $G_1 = (V', E')$, and let the interesting vertices (and possible uninteresting vertices) be given by the elements of $S \subseteq V'$. The second graph $G_2$ is then the subgraph of $G$ induced by the unlabeled vertices $U \subseteq V$ passed through an appropriate obfuscating function. This pair $(G_1, G_2)$, with any $s \in S_1$ chosen to be the interesting vertex, encodes the label information present in the one-graph VN problem as well as the graph structure of $G$, as required.

3 Bayes error and Bayes optimality in Vertex Nomination

Viewing a VN scheme as an information retrieval system suggests that a scheme that puts $o(v^*)$ close to the top of the nomination list is potentially of great practical value, even if it fails to obtain perfect
performance. Motivated by this, we adapt the recall-at-$k$ metric from classical information retrieval as a measure of performance. To wit, we define the level-$k$ loss function and error for VN as follows.

**Definition 17** (VN loss function, level-$k$ error). Let $\Phi \in \mathcal{V}_{n,m}$ be a vertex nomination scheme and $o$ an obfuscating function. For $(g_1, g_2)$ realized from $(G_1, G_2) \sim F_{c,n,m,0}$ with vertex of interest $v^* \in C$, and for $k \in [m - 1]$, we define the level-$k$ nomination loss via

$$
\ell_k(\Phi, g_1, g_2, v^*) = \mathbb{I}\{\text{rank}_{\Phi,g_2}(v^*) \geq k + 1\} = 1 - \mathbb{I}\{\text{rank}_{\Phi,g_2}(v^*) \leq k\}.
$$

(3)

The level-$k$ error of $\Phi$ at $v^*$ is then defined to be

$$
L_k(\Phi, v^*) = \mathbb{E}_{(G_1,G_2)\sim F_{c,n,m,0}}[\ell_k(\Phi, G_1, G_2, v^*)] = \mathbb{P}_{(G_1,G_2)\sim F_{c,n,m,0}}[\text{rank}_{\Phi,g_2}(v^*) \geq k + 1].
$$

(4)

From the definition of the level-$k$ error in Eq. (4), it is immediate that

$$
L_1(\Phi, v^*) = 1 - \mathbb{P}_{(G_1,G_2)\sim F_{c,n,m,0}}[\text{rank}_{\Phi,g_2}(v^*) = 1] \\
\geq L_2(\Phi, v^*) = 1 - \mathbb{P}_{(G_1,G_2)\sim F_{c,n,m,0}}[\text{rank}_{\Phi,g_2}(v^*) \in \{1, 2\}] \\
\geq L_3(\Phi, v^*) = 1 - \mathbb{P}_{(G_1,G_2)\sim F_{c,n,m,0}}[\text{rank}_{\Phi,g_2}(v^*) \in \{1, 2, 3\}] \\
\vdots \\
\geq L_{m-1}(\Phi, v^*) = \mathbb{P}_{(G_1,G_2)\sim F_{c,n,m,0}}[\text{rank}_{\Phi,g_2}(v^*) = m],
$$

The level-1 loss function is analogous to the classical 0/1 loss function in classification, as $L_1(\Phi, v^*)$ is simply the probability that $\Phi$ fails to “classify” $o(v^*)$ as the vertex corresponding to $v^*$ in $o(G_2)$ (i.e., fails to rank it first). Considering $1 < k \ll m$ enables us to model the practical loss of utilizing a VN scheme to systematically search $o(V_2)$ for $o(v^*)$ given limited resources.

**Remark 18.** Unlike in the classification setting described in Section 1.2—where $L_F(h_n)$ is a random variable indexed by $n$—the nomination errors defined in Definition 17 are constants indexed by $n$ and $m$. In the classical setting, $L_F(h_n)$ denotes the error rate of a classifier that classifies a single observation $X$ based on $n$ training instances $\{(X_i, y_i)\}_{i=1}^n$. In the case of VN, the notion of labeled training instances is, at best, more hazy. Indeed, in the present setting, the training data and test data are inseparable. The graphs (or, more specifically, their edges) are the training data, and in the present work, the graph orders $n, m$ are better thought of as measuring problem dimension rather than as training set size.

Analogous to the classification literature, we are now able to define the concept of Bayes optimality in the VN framework.

**Definition 19** (Bayes error of a VN scheme). Let $(G_1, G_2) \sim F_{c,n,m,0}$ with vertex of interest $v^* \in C$, and let $o : V_2 \to W$ be an obfuscating function. We define the level-$k$ Bayes optimal VN scheme to be any element $\Psi \in \arg\min_{\Phi \in \mathcal{V}_{n,m}} L_k(\Phi, v^*)$, and define the level-$k$ Bayes error to be $L_k^*(v^*) = L_k(\Psi, v^*)$ for level-$k$ Bayes optimal $\Psi$.

Now that we have a notion of Bayes error for VN, it is natural to ask whether an optimal VN scheme exists analogous to the Bayes optimal classifier of Equation (1). Toward this end, let $(g_1, g_2)$ be realized from $(G_1, G_2) \sim F_{c,n,m,0} \in \mathcal{N}$, and consider a vertex of interest $v^* \in C$ and obfuscating function $o : V_2 \to W$. Letting $\simeq$ denote graph isomorphism, define the set

$$
(g_1, o(g_2)) = \left\{ (g_1, \tau(o(g_2))) \in \mathcal{G}_n \times \mathcal{G}_m \text{ s.t. } \tau : W \to V_2 \text{ is a bijection} \right\}
$$

$$
= \left\{ (g_1, \hat{g}_2) \in \mathcal{G}_n \times \mathcal{G}_m \text{ s.t. } o(\hat{g}_2) \simeq o(g_2) \right\}
$$

$$
= \left\{ (g_1, \hat{g}_2) \in \mathcal{G}_n \times \mathcal{G}_m \text{ s.t. } \hat{g}_2 \simeq g_2 \right\}
$$

(6)
We refer to the functions \( r \) in Eq. (6) as recovery functions, since they are meant to undo the obfuscation of the vertex correspondence by \( \sigma \). In particular, each \( r \) considered in \((g_1, \sigma(g_2))\) identifies a set of \( c \) potential core vertices in the graph \( \sigma(g_2) \). These are precisely the \( c \) vertices believed to be in the set \( C \), i.e., the vertices that correspond across the two graphs. In order to define the Bayes optimal scheme, we will also need the following restriction of \((g_1, \sigma(g_2))\): for each \( w \in W \), and \( u \in V_2 \) we define

\[
(g_1, \sigma(g_2))_{u=\sigma(u)} = \begin{cases} 
(g_1, r(\sigma(g_2))) \in G_n \times G_m \text{ s.t. } r : W \rightarrow V_2 \text{ is a bijection, } r(w) = u \\
\end{cases}
\]

\[
= \begin{cases} 
(g_1, \tilde{g}_2) \in G_n \times G_m \text{ s.t. } \sigma(\tilde{g}_2) = \sigma(\sigma(g_2)), \sigma \text{ an isomorphism, } \sigma(w) = \sigma(u) \\
\end{cases}
\]

\[
= \begin{cases} 
(g_1, \tilde{g}_2) \in G_n \times G_m \text{ s.t. } \tilde{g}_2 = \sigma(g_2), \sigma \text{ an isomorphism, } \sigma(\sigma^{-1}(w)) = u \\
\end{cases}
\]

We are now ready to define a Bayes optimal VN scheme.

For ease of notation, in the sequel we will write \( P_{F,c,n,m,\theta} \) or even simply \( P \) in place of \( P_{(G_1,G_2) \sim F_{c,n,m,\theta}} \) where there is no risk of ambiguity. Note that the sets \{ \((g_1, \sigma(g_2))\) \} partition \( G_n \times G_m \). We will call this partition \( P_{n,m} \), where we suppress dependence on \( g_1, g_2 \) and \( \sigma \) for ease of notation. We will define a Bayes optimal scheme \( \Phi^* \) piecewise on each element of this partition. For a given \((\tilde{g}_1, \sigma(\tilde{g}_2))\) let \((g_1, g_2)\) be a fixed element in \((\tilde{g}_1, \sigma(\tilde{g}_2))\). We will now define a ranking function, which Theorem 20 will show to be level-\( k \) Bayes optimal for all \( k \in [m-1] \):

\[
\Phi^*(g_1, \sigma(g_2), v^*)[1] \in \arg\max_{u \in W} P\left[(g_1, \sigma(g_2))_{u=\sigma(v^*)} \mid (g_1, \sigma(g_2))\right]
\]

\[
\Phi^*(g_1, \sigma(g_2), v^*)[2] \in \arg\max_{u \in W \setminus \{\Phi^*[1]\}} P\left[(g_1, \sigma(g_2))_{u=\sigma(v^*)} \mid (g_1, \sigma(g_2))\right]
\]

\[
\vdots
\]

\[
\Phi^*(g_1, \sigma(g_2), v^*)[m] \in \arg\max_{u \in W \setminus \bigcup_{i \in [m-1]} \Phi^*[i]} P\left[(g_1, \sigma(g_2))_{u=\sigma(v^*)} \mid (g_1, \sigma(g_2))\right],
\]

with ties broken arbitrarily but deterministically. We refer the interested reader to Appendix A for discussion of the the case where ties are allowed in the ranking function. For each element \((g_1, g_2)\) \((\tilde{g}_1, \sigma(\tilde{g}_2))\) \(\{ (g_1, g_2) \} \), choose a permutation \( \sigma \) such that \( \sigma(g_2) = \sigma(\sigma(g_2)) \), and define \( \Phi^*(g_1, \sigma(g_2), v^*) = \sigma(\Phi^*(g_1, \sigma(g_2), v^*)) \); note that if there are multiple \( \sigma \) such that \( \sigma(g_2) = \sigma(\sigma(g_2)) \), then we could define multiple Bayes optimal schemes and that the optimality proved in Theorem 20 is independent of the choices of \( \sigma \).

The following theorem shows that this scheme is indeed Bayes optimal. A proof can be found in Appendix A.1

**Theorem 20.** Let \( \Phi^* \) be as defined in Equation (8). Suppose that \((G_1, G_2) \sim F_{c,n,m,\theta} \in \mathcal{N} \), and consider a vertex of interest \( v^* \in C \). We have that \( L_k(\Phi^*, v^*) = L^*_k(v^*) \) for all \( k \in [m-1] \) and all obfuscating functions \( \sigma \).

**Example 8** continued. Let \((G_1, G_2) \sim R\text{-ER}(P) \) for \( P, R \in \mathbb{R}^{n \times n} \). Under mild model assumptions, we have that \( \lim_{n \to \infty} L_k^*(v^*) = 0 \) for any fixed \( k \). This is due to the fact that the optimal graph matching of \( G_1 \) to \( \sigma(G_2) \) will almost surely recover the true vertex labels of \( \sigma(G_2) \) for \( n \) suitably large; i.e.,

\[
\arg\min_{Q \in \Pi_n} \| AQ - QB \|_F = \{ I_n \} \text{ with probability } \to 1,
\]

where \( \Pi_n \) is the set of \( n \times n \) permutation matrices, \( A \) is the adjacency matrix for \( G_1 \) and \( B \) the adjacency matrix for \( G_2 \). More concretely, we have the following theorem adapted to our present setting from [38]. A proof sketch can be found in Appendix C.
Theorem 21. Let \((A, B) \sim R\text{-}ER(P)\), and for any fixed permutation matrix \(Q\) define the random variable 
\[\delta(Q) := \|AQ - QB\|_F.\]
Define \(0 < \epsilon := \min_{i,j;i\neq j} 2R_{i,j}P_{i,j}(1 - P_{i,j})\). There exist positive constants \(c_1, c_2\) such that if \(\epsilon^2 > c_1 \log(n)\), then for sufficiently large \(n\),

\[P(\exists Q \in \Pi_n \setminus \{I_n\} \text{ s.t. } \delta(Q) \leq \delta(I_n)) \leq 2 \exp\{-c_2\epsilon^2 n\}.\]

Similarly to the Bayes optimal scheme, we define the graph matching VN scheme, denoted \(\Phi_M\), separately on each element of \(\mathcal{P}_{n,n}\). For a given \((\tilde{g}_1, [\sigma(\tilde{g}_2)]) \in \mathcal{P}_{n,n}\), let \((g_1, g_2)\) be an an fixed element in \((\tilde{g}_1, [\sigma(\tilde{g}_2)])\). Define \(\Phi_M(g_1, \sigma(g_2), v^*)[1]\) to be a fixed but arbitrary element from

\[\{r^{-1}(v^*) \text{ s.t. } Q_r \in \arg\min_{Q \in \Pi_n} \|AQ - QB\|_F\},\]

where each \(r: W \to V_2\) appearing above is a recovery function and \(Q_r\) its associated permutation matrix (having identified both \(W\) and \(V_2\) with the set \([n]\)). Define

\[\mathcal{R}_1 = \left\{ r \text{ s.t. } r \text{ is a recovery function with } r^{-1}(v^*) = \Phi_M(g_1, \sigma(g_2), v^*)[1] \right\}.
\]

If \(i > 1\), define \(\Phi_M(g_1, \sigma(g_2), v^*)[i]\) to be any element of

\[\left\{r^{-1}(v^*) \text{ s.t. } Q_r \in \arg\max_{Q \in \Pi(n) \setminus \{Q_{r_i} : Q_r \in \cup_{j=0}^{i-1} \mathcal{R}_j\}} \|AQ - QB\|_F\right\}
\]

where \(\mathcal{R}_j\) is defined analogously to \(\mathcal{R}_1\). For each element \((g_1', g_2') \in (\tilde{g}_1, [\sigma(\tilde{g}_2)]) \setminus \{(g_1, g_2)\}\), choose a permutation \(\sigma\) such that \(\sigma(g_2') = \sigma(g_2)\), and we then define \(\Phi_M(g_1, \sigma(g_2'), v^*) = \sigma(\Phi_M(g_1, \sigma(g_2), v^*))\).

Theorem 21 above states that under mild model assumptions, we have that \(\Phi_M(G_1, \sigma(G_2), v^*)[1] = \sigma(v^*)\) asymptotically almost surely, and thus \(\lim_{n \to \infty} L_1(\Phi_M, v^*) = 0\). Indeed, in this setting, for any fixed \(k \geq 1\), \(\lim_{n \to \infty} L_k(\Phi_M, v^*) = 0\). It is then immediate that \(\lim_{n \to \infty} L_k^*(v^*) = 0\) in this model for any fixed \(k\) as desired.

The next two examples serve to illustrate how the level-\(k\) Bayes error behaves in the presence of stochastically indistinguishable vertices. In essence, we cannot hope to perform better than randomly ordering stochastically equivalent vertices.

Example 10, continued. Let \(G_1\) and \(G_2\) be independent \(ER(n, p)\) graphs. Since the vertices are stochastically indistinguishable within each of the two graphs, no nomination scheme can do better than random chance in this model. Thus, with \(c = n\), we have that \(L_k^*(v^*) = 1 - k/n\) for all \(k \in [n - 1]\) and all \(v^* \in [n]\).

Example 22. Let \(p_1, p_2, q \in [0, 1]\) with \(1 \geq p_1 > p_2 \geq 0\) and \(q \neq p_1, p_2\). Define the matrix

\[B = \begin{pmatrix} p_1 & q \\ q & p_2 \end{pmatrix},\]

and let \(G_1\) and \(G_2\) be independent \(SBM(2, B, b_n)\) graphs where \(b_n(i) = 1\) if \(i \leq n\) and \(b_n(i) = 2\) if \(i > n\). With \(c_n = n\) and the correspondence equal to the identity function, let \((k_n)_{n=2}^\infty\) be a nondecreasing divergent sequence satisfying \(k_n \leq n\) for all \(n > 1\), then \(\lim_{n \to \infty} L_{k_n}^*(v^*) = \lim_{n \to \infty} [(1 - 2k_n/n) \vee 0] \) for all \(v^*\). Indeed, \(L_{k_n}^*\) is asymptotically equivalent to randomly ordering the \(n/2\) vertices in \(G_2\) that are stochastically equivalent to \(v^*\).
4 VN consistency

With the definition of Bayes optimality and the Bayes optimal scheme in hand, it is now possible to define a notion of consistent vertex nomination analogous to consistent classification in the pattern recognition literature. Before defining a consistent VN rule (i.e., a sequence of VN schemes), we must first define the notion of sequences of distributions in $\mathcal{N}$ with nested cores. Such sequences of distributions make it sensible to speak of a sequence of vertex nomination problem instances.

**Definition 23.** Let $\mathbf{F} = \{ F_c(n), n, m(n), \theta(n) \}_{n=n_0}^\infty$ be a sequence of distributions in $\mathcal{N}$. We say that $\mathbf{F}$ has nested cores if there exists an $n_0$ such that for all $n_0 \leq n < \tilde{n}$, if $(G_1, G_2) \sim F_c(n), n, m(n), \theta(n)$ and $(\tilde{G}_1, \tilde{G}_2) \sim F_c(\tilde{n}), \tilde{n}, m(\tilde{n}), \theta(\tilde{n})$, we have, letting $C$ and $\tilde{C}$ be the core vertices associated with $F_c(n), n, m(n), \theta(n)$ and $F_c(\tilde{n}), \tilde{n}, m(\tilde{n}), \theta(\tilde{n})$ respectively, and denoting the junk vertices $J_1, \tilde{J}_1, J_2, \tilde{J}_2$ analogously,

(i) $V(G_1) = C \cup J_1 \subset V(\tilde{G}_1) = \tilde{C} \cup \tilde{J}_1$;

(ii) $V(G_2) = C \cup J_2 \subset V(\tilde{G}_2) = \tilde{C} \cup \tilde{J}_2$;

(iii) $C \subset \tilde{C}$.

We are now ready to define a consistent VN rule.

**Definition 24.** Let $\mathbf{F} = \{ F_c(n), n, m(n), \theta(n) \}_{n=n_0}^\infty$ be a sequence of nominatable distributions in $\mathcal{N}$ with nested cores satisfying $\lim_{n \to \infty} m(n) = \infty$. For a given non-decreasing sequence $(k_n)$, we say that a VN rule $\Phi = (\Phi_{n,m(n)})_{n=n_0}^\infty$ is level-$(k_n)$ consistent at $v^*$ with respect to $\mathbf{F}$ if

$$\lim_{n \to \infty} L_{k_n}(\Phi_{n,m(n)}, v^*) - L_{k_n}^*(v^*) = 0,$$

for any sequence of obfuscating functions of $V_2$ with $|V_2| = m(n)$. If a scheme $\Phi$ is level-$(k_n)$ consistent at $v^*$ for a constant sequence $k_n = k$, $n = 1, 2, \ldots$, then we say simply that $\Phi$ is level-$k$ consistent.

**Remark 25.** Equation (5) has an interesting implication on VN consistency in the setting where $L_{k_n}^*(v^*) \to 0$. In this case, level-$(k_n)$ consistency of a VN rule $\Phi$ implies that $\Phi$ is $(k'_n)$-consistent for all $(k'_n)$ such that $\liminf \frac{k'_n}{k_n} \geq 1$. We conjecture that this implication holds true for the case where $L_{k_n}^*(v^*) \to c > 0$, but this problem remains open at present.

**Example 8 continued.** Let $\mathbf{F} = \{ F_{n,n,n,\theta_n=(P_n,R_n)} \}$ be a sequence of $R_n$-ER$(P_n)$ random graph models in $\mathcal{N}$ for some sequence of probability matrices $(P_n)_{n=n_0}^\infty$ and correlation matrices $(R_n)_{n=n_0}^\infty$. Under mild model assumptions (see Theorem 21), the graph matching vertex nomination rule $\Phi_{n,m}$ defined in Equation (9) above is level-1 consistent (and hence level-$(k_n)$ consistent for all $(k_n)$ sequences).

**Example 10 continued.** Let $\mathbf{F} = \{ F_{n,n,n,\theta_n=p} \}$ be a sequence of independent ER$(n, p)$ random graph models in $\mathcal{N}$. All vertex nomination rules are level-$(k_n)$ consistent for all $(k_n)$ sequences. This holds for all possible values of $c \in [n]$ in the nested sequence of ER$(n, p)$ distributions, as all VN rules have effectively chance performance independent of core size.

We define the consistency of a VN rule with respect to a broad class of graph sequences, and it is perhaps no surprise that there cannot be any level-$(k_n)$ universally consistent VN rules, not even for constant sequences $k_n := k$ (i.e., those that are level-$(k_n)$ consistent for all sequences of nominatable distributions $\mathbf{F}$ with nested cores). To prove this result, we will first establish an analogue to the “arbitrary poor performance” theorems for classifiers, see e.g., Theorem 7.1 in [16], which state that for a fixed $n$ and $m$, any VN scheme can be shown to have arbitrarily poor performance with respect to a well-chosen adversarial distribution $F_{c,n,m,\theta}$. Our theorem mirrors the classical classification literature, as for a given classification rule, there exists “a sufficiently complex distribution for which the sample size $n$ is hopelessly
The practical implications of Corollary 27 are multitudinous. Below, we will briefly outline two such implications. Unlike in the classification setting where universally consistent rules (e.g., the set of models for which the VN rule is consistent) before applying it in real big-data settings. Indeed, nested-core nominatable sequences do not exist for any sequence \((k_n)_{n=0}^{\infty}\) satisfying \(k_n = o(m(n))\) that does not grow as fast as \(m(n) = |V(G_2)|\).

**Corollary 27.** Let \(\epsilon \in (0, 1)\) be arbitrary, and consider a VN rule \(\Phi = (\Phi_{n,m(n)})\). For any nondecreasing sequence \((k_n)_{n=0}^{\infty}\) satisfying \(k_n = o(m(n))\), there exists a sequence of distributions \((F_{c,n,m(n),\theta})\) in \(\mathcal{N}\) with nested cores such that

\[
\lim_{n \to \infty} \sup_{v^*} L^*_n(v^*) = \epsilon < 1 = \lim_{n \to \infty} L_{k_n}(\Phi_{n,m(n)}, v^*).
\]

The practical implications of Corollary 27 are multitudinous. Below, we will briefly outline two such implications. Unlike in the classification setting where universally consistent rules (e.g., \(k\)-nearest neighbors) are (at least in theory) guaranteed to perform well in big-data settings, in VN the practitioner enjoys no such surety. Indeed, in VN the practitioner first needs to identify the consistency class of a VN rule (i.e., the set of models for which the VN rule is consistent) before applying it in real big-data settings. Unfortunately, identifying and enumerating these consistency classes is theoretically and practically non-trivial, and we are investigating theory and heuristics for this at present. In a streaming data setting, the performance of a universally consistent classifier will approach Bayes optimality for the distribution governing the data, and the classifier will be guaranteed to successfully adapt itself to any changes in the underlying data distribution. A lack of universal consistency in the VN setting implies that this is not the case, as the performance of a consistent VN scheme in the streaming setting could precipitously decline in the presence of distributional shifts in the data. Recognizing these shifts and the potential impact on VN performance is paramount and is the subject of current research.

### 4.1 Global consistency

We have just seen that no universally consistent VN schemes exist. This is a consequence of the complexity of the models available when choosing a sequence of nominatable distributions. Indeed, nested-core nominatable sequences \(F\) allow for (nearly) arbitrary dependence structure and model complexity as \(n\) increases to \(\infty\): corresponding vertex behavior may be correlated (see Example 8), independent, or negatively correlated (see Example 30) across networks. This model flexibility is in service of modeling
the complexity of real world networks, although (as we will demonstrate below) restricting our model
class to simpler dependency structures often still does not result in the existence of universally consistent
schemes.

It is natural to explore a weaker notion of consistency, namely consistency for a sufficiently large
family of nominatable sequences rather than for all nominatable sequences.

**Definition 28.** Let \( \mathcal{F} = \{ F_\alpha = (F_{\varphi(n),\varphi(m),\varphi(\theta)})_{\alpha \in \mathcal{A}} : \alpha \in \mathcal{A} \} \) be a family of nominatable sequences,
indexed by some index set \( \mathcal{A} \). We say that VN scheme \( \Phi \) is level-(\( k_n \)) \( \mathcal{F} \)-globally consistent if \( \Phi \) is
level-(\( k_n \)) consistent for every \( F \in \mathcal{F} \). We call such a family level-(\( k_n \)) globally consistent.

The question of the maximal family \( \mathcal{F} \) for which a level-(\( k_n \)) \( \mathcal{F} \)-globally consistent rule exists is of prime
interest. While we cannot offer a satisfactory complete answer to that question in the present work, we
do offer some examples of jointly consistent families.

**Example 8 continued:** In settings where corresponding vertices have correlated neighborhood
structures across networks, there is hope for finding globally consistent rules. In the ongoing Example 8
we have seen a simple example of this in the R-ER(\( P \)) model, in which the matrix of correlations \( R \) encodes
a correspondence across the two graphs. As mentioned previously, Theorem 21 asserts that under some
mild model assumptions on \( R \) and \( P \) in the R-ER(\( P \)) model, level-1 globally consistent VN rules exist
(namely the graph matching VN scheme). If \( \mathcal{F} \) denotes the set of distributions obeying these model as-
sumptions, then we have that level-(\( k_n \)) \( \mathcal{F} \)-globally consistent rules exist for all sequences \( (k_n) \). While we
do not expect the conditions of Theorem 21 to produce a maximal level-(\( k_n \)) globally consistent family for
any given \( (k_n) \), this example nonetheless provides an important intuition for the properties such maximal
families might possess.

**Remark 29.** An attempt at systematically constructing a maximal globally consistent family might begin
by putting model restrictions onto elements of \( \mathcal{N} \). A natural restriction to consider would be to demand
that the models in \( \mathcal{F} \) be nested in the following sense: For \( F \in \mathcal{F} \), if \( (G_1, G_2) \sim F_{c(n_2),n_2,m(n_2),\varphi(n_2)} \) with
\( n_1 < n_2 \), then \( (G_1[[n_1]], G_2[[m(n_1)]]) \equiv (G'_1, G'_2) \) where \( (G'_1, G'_2) \sim F_{c(n_1),n_1,m(n_1),\varphi(n_1)} \). This property
would allow us to consider “streaming” network models \( F \), where for \( n_1 < n_2 \), if \( (g_1, g_2) \) is realized from
\( (G_1, G_2) \sim F_{c(n_2),n_2,m(n_2),\varphi(n_2)} \), and \( (g'_1, g'_2) \) is realized from \( (G'_1, G'_2) \sim F_{c(n_1),n_1,m(n_1),\varphi(n_1)} \) then \( (g_1, g_2) \)
can be constructed by appropriately adding \( n_2 - n_1 \) vertices to \( (g'_1, g'_2) \). Additionally, this would serve to mimic
the nested nature of the data in the classification consistency literature. However, as we will see in
Example 30 global consistency depends both on the dependency structure within each graph (as seen
in Theorem 26 and the vertex correspondence (i.e., the potential dependency structure across graphs)
posited by the model.

### 4.2 Behavioral (in)consistency and global (in)consistency

We suspect that if the vertices of interest have a common distinguishing probabilistic and/or topological
characteristic (correlated neighborhoods, common SBM block structure, high network centrality, etc.)
then a globally consistent rule may exist. Indeed, this is the case (under mild model assumptions) in
the R-ER(\( P \)) of Example 8 in the i.i.d. SBM of Example 22 where the correspondence is the identity
function \[39\], and in the i.i.d. ER of Example 10 to name a few. In each of these examples, there
is a stochastic/topological similarity (or in the ER case, uniformity) between corresponding vertices
across networks; in each, corresponding vertices behave similarly across networks. While we suspect
that this behavioral similarity is not sufficient for global consistency, Example 30 demonstrates that
behavioral inconsistencies within a family of nominatable distributions can preclude the existence of
globally consistent rules.
Example 30. For each $n$, consider $n$-vertex random graphs $G_1 \sim \text{SBM}(2, B_1, b_n^{(1)})$ independent of $G_2 \sim \text{SBM}(2, B_2, b_n^{(2)})$.

Case 1. In this case, corresponding vertices behave similarly across networks. To wit, let $\mathbf{F} = (F_n)_{n=n_0}$ be the sequence of models where

$$B_1 = B_2 = \begin{pmatrix} p_1 & q \\ q & p_2 \end{pmatrix}, \quad b_n^{(1)}(i) = b_n^{(2)}(i) = \begin{cases} 1 & \text{if } i \leq n/2; \\ 2 & \text{if } i > n/2, \end{cases}$$

where $p_1 \neq p_2$, $c = n$, and the correspondence is the identity function. As stated before, in this model $L_n^{*}(v^*) \to 0$ for all $v^* \in C$. Without loss of generality, consider $v^* = v_1 = u_1$. If $\Phi$ is consistent with respect to $\mathbf{F}$ then

$$P_{F_n}(\text{rank}_{\Phi(G_1, \sigma(G_2), v_1)}(\sigma(u_1)) \geq n/2 + 1) \to 0.$$

By the distributional equivalence of vertices within the same block, and the consistency property in the definition of a VN scheme, for any $u, v \in b_n^{-1}(1)$, $k \in [n]$ we have that

$$P_{F_n}(\text{rank}_{\Phi(G_1, \sigma(G_2), v_1)}(\sigma(u)) = k) = P_{F_n}(\text{rank}_{\Phi(G_1, \sigma(G_2), v_1)}(\sigma(v)) = k).$$

Letting this common value be set to $\alpha_{k,n}$ (with $\beta_{k,n}$ defined similarly as the common value of $P_{F_n}(\text{rank}_{\Phi(G_1, \sigma(G_2), v_1)}(\sigma(u)) = k)$ for $u$ in block 2), we have that

$$\sum_{i=1}^{n} P_{F_n}(\text{rank}_{\Phi(G_1, \sigma(G_2), v_1)}(\sigma(u_i)) = k) = 1 = \frac{n(\alpha_{k,n} + \beta_{k,n})}{2}$$

giving us that $\alpha_{k,n} = 2/n - \beta_{k,n}$. Consistency implies that

$$\sum_{k=1}^{n/2} \alpha_{k,n} \to 1,$$

which implies that

$$\sum_{k=1}^{n/2} \alpha_{k,n} = \sum_{k=1}^{n/2} (2/n - \beta_{k,n}) = 1 - \sum_{k=1}^{n/2} \beta_{k,n} \to 1,$$

implying $\sum_{k=1}^{n/2} \beta_{k,n} \to 0$. Therefore, for any $u$ in block 2,

$$P_{F_n}(\text{rank}_{\Phi(G_1, \sigma(G_2), v_1)}(\sigma(u)) \geq n/2 + 1) \to 1.$$

Case 2. In this case, corresponding vertices behave differently across networks. To wit, let $\mathbf{\tilde{F}} = (\tilde{F}_n)_{n=n_0}$ be the sequence of models where

$$B_1 = \begin{pmatrix} p_1 & q \\ q & p_2 \end{pmatrix}, \quad B_2 = \begin{pmatrix} p_2 & q \\ q & p_1 \end{pmatrix}, \quad b_n^{(1)}(i) = b_n^{(2)}(i) = \begin{cases} 1 & \text{if } i \leq n/2; \\ 2 & \text{if } i > n/2, \end{cases}$$

$c = n$, and the correspondence is the identity function. As in Case 1 considered above, in this model $L_n^{*}(v^*) \to 0$ for all $v^* \in C$, and, as above, consider $v^* = v_1 = u_1$. If $\Phi$ is consistent with respect to $\mathbf{\tilde{F}}$ then

$$P_{\tilde{F}_n}(\text{rank}_{\Phi(G_1, \sigma(G_2), v_1)}(\sigma(u_1)) \geq n/2 + 1) \to 0.$$
Note that if \( \sigma \) is the permutation such that
\[
\sigma(i) = \begin{cases} 
  i + n/2 & \text{if } i \leq n/2; \\
  i - n/2 & \text{if } i > n/2,
\end{cases}
\]
then \( \mathbb{P}_{F_n}(g_1, g_2) = \mathbb{P}_{F_n}(g_1, \sigma(g_2)) \). Define
\[
E_n = \{(g_1, g_2) \text{ s.t. } \text{rank}_{\sigma(g_1, \sigma(g_2), v_1)}(\sigma(u_1)) \geq n/2 + 1\}
\]
i.e., and \( \tilde{E}_n = \{(g_1, g_2) \text{ s.t. } (g_1, \sigma(g_2)) \in E_n\} \), i.e.,
\[
\tilde{E}_n = \{(g_1, g_2) \text{ s.t. } \text{rank}_{\sigma(g_1, \sigma(g_2), v_1)}(\sigma(u_{n/2+1})) \geq n/2 + 1\}.
\]
If \( \Phi \) is consistent with respect to \( F \) we have that \( \mathbb{P}_{F_n}(E_n) \to 0 \) which implies (as \( (g_1, g_2) \in E_n \Leftrightarrow (g_1, \sigma(g_2)) \in \tilde{E}_n \)) \( \mathbb{P}_{F_n}(\tilde{E}_n) \to 0 \). If \( \Phi \) is consistent with respect to \( \tilde{F} \) then \( \mathbb{P}_{F_n}(E_n) \to 0 \) and \( \mathbb{P}_{F_n}(\tilde{E}_n) \to 1 \). We arrive at a contradiction, and \( \Phi \) cannot be \((n/2)\)-consistent with respect to both \( F \) and \( \tilde{F} \).

Although Example 30 may seem artificial, it is a simple representation of a common phenomenon observed in network data. Often the same entity can behave quite differently across networks (see [47] for an example of this in social networks and [11] for an example of this in connectomics). In such a setting, an example of this in social networks and [11] for an example of this in connectomics). In such a setting, a universal scheme that works in both behavioral settings intuitively should not exist. Indeed, at least in the simple block model setting considered above, we see that no such scheme does exist. Example 30 also highlights an important difference between the VN setting and the more standard classification framework. We already noted that classification’s universal consistency relies on the distribution not changing in \( n \), whereas in VN the distributions must vary with \( n \) (indeed, the graph sizes grow in \( n \)). Further, this example shows that the nonexistence of a universally consistent scheme is not simply a consequence of changing the underlying distributional parameters with \( n \), as these two SBM distributions are (essentially) fixed, in that the matrix \( B \) does not change with \( n \). In this example the “training data” provided by \( G_1 \) cannot be uniformly beneficial for a single VN scheme across the two differing model settings we consider. In contrast, in the classification setting of [10], the training data uniformly provides progressively better estimates of the class delineations, whereas here it does not. Indeed, the training data helps delineate potentially interesting versus non-interesting vertices in \( G_2 \) in Case (1) for one VN scheme, and in Case (2) for another VN scheme, but there does not exist a VN scheme that achieves this desired class separation across both cases.

Remark 31. In the cases considered in Example 30, if we introduce edge-wise correlation of
\[
\rho \leq \sqrt{\min \left( \frac{p_1(1-p_2)}{p_2(1-p_1)}, \frac{p_2(1-p_1)}{p_1(1-p_2)} \right)},
\]
into both Case 1 and Case 2, then under mild assumptions on \( p_1 \) and \( p_2 \), joint consistency can be recovered via a USVT centered graph matching nomination scheme; for detail see [38]. This example demonstrates that it is sometimes possible to toggle a family of models to allow for global consistency. A note of caution is needed however, as in this particular example the correlation \( \rho \) is introducing a behavioral consistency across networks that addresses the precise issue brought forth by the behavioral inconsistency in Example 30. In other, more nuanced model families, we do not expect the global-consistency modification (if it indeed exists) to be as straightforward as adding additional edge-correlation into the model.
5 Discussion

In this work, we have introduced a notion of consistency for the vertex nomination task that better reflects the broad range of models under which VN may be deployed. Rather than being restricted to the stochastic block model structure required in previous formulations of the problem, our framework allows for arbitrary dependence structure both within and between graph pairs, while encompassing the original SBM formulation of the problem. Additionally, we have demonstrated how this framework relates to the well-studied notion of Bayes optimal classifiers in the pattern recognition literature. Unlike in the classification setting, we have seen that while Bayes optimal VN schemes always exist, no universally consistent scheme exists. This fact is due essentially to the additional leeway provided by the graph model, in which observing more vertices does not necessarily correspond to receiving more information about the underlying distribution as in the classification setting studied in [56] and others [16]. For this reason, one especially interesting line of investigation concerns the nominatable distributions for which larger $n$ does indeed correspond to better performance. A simple example of this is the initial formulation of the vertex nomination problem, in which observing more vertices allows one to better estimate the model parameters, including the block membership, and thus more accurately identify the vertices from the interesting block. We suspect that the essential property at play here is that each vertex is analogous to a sample from a single distribution, though this may not be in and of itself a sufficient condition for consistency. For example, in the case of $(G_1, G_2)$ being i.i.d. draws from a random dot product graph model with the identity correspondence, each vertex (along with its incident edges) is, in a sense, a noisy sample from the underlying distribution $F$. Hence, for large $n$, one can estimate $F$ or the latent positions themselves to arbitrary accuracy, and provided that the latent positions of the interesting vertices are suitably separated from the rest of the graph, one should have VN consistency for the set of these latent position models.

More broadly, it would be good to better understand whether there exist families of nominatable distributions $\mathcal{F}$ for which certain VN schemes are consistent, and precisely how large these families can be made to be. In a similar vein, it would be of interest to explore how the dependence structure allowed both within and between graphs influences vertex nomination. In particular, if one rules out certain pathologically hard dependence structures as considered in Example 30, can one obtain global consistency with respect to this restricted set of distributions? We hope to explore these questions in future work.

We are also exploring alternative formulations of the VN problem and alternate formulations of the VN loss function. While the extension to multiple vertices of interest in each network $G_1$ and $G_2$ is straightforward, we are considering several generalizations of the VN problem considered herein. One formulation of prime interest in applications (especially in connectomics and social networks) is as follows: given a collection of vertices of interest in one graph, find those that play a similar structural (based on the topology of the underlying network) or functional (based on vertex or edge covariates) role in the other graph. In addition, the impact on VN consistency (and the potential existence of universally consistent schemes) when incorporating edge and vertex covariates into the VN framework is of prime interest, and formalizing the VN inference task (and the notion of consistency) in this framework is the subject of our current and future work.

The loss function considered in the present work is an analogue of the 0/1 recall-at-$k$ loss function in the information retrieval literature. Under this loss function, we have shown that no universally consistent VN rule exists. Presently, we are exploring whether alternative loss functions can be considered under which universal consistency is achievable. While we conjecture that Example 30 will nearly always provide a counterexample to universal consistency, this question remains open and is the subject of current research.
A Proofs of main results

Herein, we collect the proofs of the two main theorems in this manuscript, Theorems 20 and 26.

A.1 Proof of Theorem 20

Before establishing the proof of Theorem 20 we must first establish the following lemma.

Lemma 32. With $\Phi^*$ defined as above, for all $(g_1, g_2) \in G_n \times G_m$, we have that

$$\mathbb{P}_F \left[ (g_1, [o(g_2)])_{\Phi^*(g_1, o(g_2), v^*)} = \sigma(v^*) \mid (g_1, [o(g_2)]) \right]$$

(10)

is non-increasing as $k$ increases, where $\mathbb{P}_F$ is shorthand for $\mathbb{P}_{F_{c.n.m,0}}$.

Proof. By construction, there is a $(g_1, \tilde{g}_2) \in (g_1, [o(g_2)])$ such that

$$\mathbb{P}_F \left[ (g_1, [o(\tilde{g}_2)])_{\Phi^*(g_1, o(\tilde{g}_2), v^*)} = \sigma(v^*) \mid (g_1, [o(\tilde{g}_2)]) \right]$$

(11)

is non-increasing as $k$ increases, and there exists a vertex permutation $\sigma$ such that $o(g_2) = \sigma(o(\tilde{g}_2))$ and $\sigma(\Phi^*(g_1, o(\tilde{g}_2), v^*)) = \Phi^*(g_1, o(g_2), v^*)$; such a $(g_1, \tilde{g}_2)$ and $\sigma$ is guaranteed to exist by the construction of $\Phi^*$. Note that $(g_1, [o(\tilde{g}_2)]) = (g_1, [o(g_2)])$, and

$$(g_1, [o(g_2)])_{\Phi^*(g_1, o(g_2), v^*)} \mid \sigma(v^*) = \{ (g_1, g'_2) \in G_n \times G_m \text{ s.t. } o(g'_2) = \tau(o(g_2)), \tau \text{ an isomorphism,}$$

and $\tau(\Phi^*(g_1, o(g_2), v^*)[k]) = o(v^*) \}$$

$$= \{ (g_1, g'_2) \in G_n \times G_m \text{ s.t. } o(g'_2) = \tau(\sigma(o(\tilde{g}_2))), \tau \text{ an isomorphism,}$$

and $\tau(\sigma(\Phi^*(g_1, o(\tilde{g}_2), v^*)[k])) = o(v^*) \}$$

$$= \{ (g_1, g'_2) \in G_n \times G_m \text{ s.t. } o(g'_2) = \phi(o(\tilde{g}_2)), \phi \text{ an isomorphism,}$$

and $\phi(\Phi^*(g_1, o(\tilde{g}_2), v^*)[k]) = o(v^*) \}$$

$$= (g_1, [o(\tilde{g}_2)])_{\Phi^*(g_1, o(\tilde{g}_2), v^*)} \mid o(v^*)$$

It follows that for each $k \in \{m\}$, Eq. (11) and Eq. (10) are equal and the lemma follows immediately.

Proof of Theorem 20. For $\Phi \in \mathcal{V}_{n,m}$, obfuscating function $o$, $i \in \{m\}$, and $g_1 \in G_n$, $g_2 \in G_m$, let $w = \text{rank}_{\Phi^*(g_1, o(g_2), v^*)}^{-1}(i) \in W$. Note that for any $(g_1, g_2) \in G_n \times G_m$, we have that

$$\left\{ (\tilde{g}_1, \tilde{g}_2) \in (g_1, [o(g_2)]) \text{ s.t. } \text{rank}_{\Phi^*(g_1, o(g_2), v^*)}(o(v^*)) = i \right\}$$

$$= \left\{ (g_1, o(\tilde{g}_2)) \text{ s.t. } o(\tilde{g}_2) = \sigma(o(g_2)), \sigma \text{ an isomorphism, } \sigma(w) = o(v^*) \right\}$$

$$= (g_1, [o(g_2)])_{o(o(v^*))}$$

$$= (g_1, [o(g_2)])_{\Phi^*(g_1, o(g_2), v^*)[i]=o(v^*)}$$

Next note that if $g'_2$ is isomorphic to $g_2$, then $(g_1, [o(g_2)]) = (g_1, [o(g'_2)])$ and

$$(g_1, [o(g_2)])_{\Phi^*(g_1, o(g_2), v^*)[i]=o(v^*)} = (g_1, [o(g'_2)])_{\Phi^*(g_1, o(g'_2), v^*)[i]=o(v^*)}$$

For $(g_1, g_2) \in G_n \times G_m$, define $p_{\Phi} \in [0,1]^m$ via

$$p_{\Phi}[g_1, [o(g_2)], i] = p_{\Phi}[i] := \mathbb{P}_{F_{c.n.m,0}} \left[ (g_1, [o(g_2)])_{\Phi^*(g_1, o(g_2), v^*)[i]=o(v^*)} \right]$$.
and observe that $p_{Φ^*}$ majorizes $p_{Φ}$. To see this, note that for any fixed $k$, letting $q_{Φ,k}$ be $(p_{Φ}[i])_{i=1}^k$ with entries sorted in descending order, we have $p_{Φ^*}[i] ≥ q_{Φ,k}[i]$ for all $i ∈ [k]$, and majorization follows immediately.

The sets $\{(g_1, [0(g_2)])\}$ partition $G_n × G_m$, and denote the partition by $P_{n,m}$. For ease of notation, write $P$ for $P_{F_{c,n,m,θ}}$, and let

$$U_{i,g_1,g_2} = \{\text{rank}_{Φ(G_1, o(G_2), v^*)}(o(v^*)) = i\} \cap (g_1, [0(g_2)]).$$

We have

$$L_k(Φ, v^*) = 1 - \sum_{i=1}^{k} P(\text{rank}_{Φ(G_1, o(G_2), v^*)}(o(v^*)) = i)$$

$$= \sum_{P_{n,m}} \left(1 - \sum_{i=1}^{k} P[U_{i,g_1,g_2}] P[(g_1, [0(g_2)])]\right)$$

$$= \sum_{P_{n,m}} \left(1 - \sum_{i=1}^{k} p_{Φ}[g_1, o(g_2), i] P[(g_1, [0(g_2)])]\right)$$

$$≥ \sum_{P_{n,m}} \left(1 - \sum_{i=1}^{k} p_{Φ^*}[g_1, o(g_2), i] P[(g_1, [0(g_2)])]\right)$$

$$= L_k^*(v^*),$$

where the sum over this partition is well-defined by noting that $p_{Φ}[, o(·), i]$ is constant within each element of $P_{n,m}$. As $Φ$ and $o$ were arbitrary, the proof follows. \hfill \Box

## A.2 Proof of Theorem 26

**Proof.** For $i ∈ [m - 1]$, let $ξ[i] = ϵ_i - ϵ_{i-1}$ (where $ϵ_0 := 0$), and let $ξ[m] = 1 - ϵ_{m-1}$. Consider asymmetric graphs $(g_1, g_2) ∈ G_n × G_m$. Construct a distribution $F_{c,n,m,θ} ∈ N$ as follows.

i. $c = n ∧ m$;

ii. The support of $F_{c,n,m,θ}$ is $(g_1, [0(g_2)])$;

iii. For each $k ∈ [m]$ define

$$R_{Φ,k} = \{(g_1, g_2) ∈ (g_1, [0(g_2)]) \text{ s.t. } Φ(g_1, o(g_2), v^*)[k] = o(v^*)\}.$$

Then we define $P_{F_{c,n,m,θ}}(R_{Φ,k}) := ξ[k]$ with all elements of $R_{Φ,k}$ being assigned equal mass under $F_{c,n,m,θ}$.

It is clear then that $L_k(Φ, v^*) = 1 - ϵ_k > 1 - \frac{k}{m}$. It is also clear that $L_k^*(v^*) ≤ ϵ_{m-k}$. Indeed, consider $Φ'$ which is defined by reversing the order provided by $Φ$; then $L_k(Φ', v^*) = ϵ_{m-k}$; which completes the proof. \hfill \Box

## B Breaking Ties in VN Rankings

Allowing ties in the VN framework is straightforward except for the difficulty it creates in handling vertex rankings. In order to define Bayes optimality in this tied vertex nomination framework, we require a slight modification of the concept of rank in the nomination list. Rather than producing an ordering of $V_2$, we
now allow a VN scheme $\Phi$ to produce a partial ordering, which we represent by allowing $\Phi(g_1, o(g_2), v^\star)[k]$ to be a subset of $V_2$ rather than a specific element of $V_2$. Such a subset corresponds to a set of vertices that are tied in the ranking, i.e., all vertices in the set $\Phi(g_1, o(g_2), v^\star)[k]$ are tied for rank $k$ in the nomination list.

**Definition 33** (Vertex Nomination (VN) Scheme with ties). Let $o : V_2 \mapsto W$ be an obfuscating function. For a set $A$, let $\mathcal{W}_A$ denote the set of all weak orderings of the elements of $A$; i.e., total orderings of $A$ that allow for ties. A vertex nomination scheme is a function $\Phi : G_n \times o(G_m) \times V_1 \mapsto \mathcal{W}_W$, satisfying the following self-consistency properties.

i. If $\sigma$ is a permutation of $W$, then for all $k \in [m]$ we have that $\sigma(\Phi(g_1, o(g_2), v^\star)[k]) = \Phi(g_1, \sigma \circ o(g_2), v^\star)[k]$.

ii. If $u \in \Phi(g_1, o(g_2), v^\star)[k]$, then $w \in \Phi(g_1, o(g_2), v^\star)[k]$ for all $w \in \mathcal{T}_o(g_2)(u)$.

We let $\mathcal{V}_{n,m}$ denote the set of all such VN schemes.

A VN scheme effectively produces a partial ordering of the elements of $W$: if $w \in \Phi(g_1, o(g_2), v^\star)[j]$ and $w' \in \Phi(g_1, o(g_2), v^\star)[j']$ for $j < j'$ then $w < w'$. This partial ordering provides a natural ranking of the vertices in distinct $\Phi(g_1, o(g_2), v^\star)[j]$ sets, but does not provide a ranking of vertices within a single $\Phi(g_1, o(g_2), v^\star)[j]$. We can think of these vertices as being tied in the nomination scheme’s weak ordering, and it is tempting to adopt a standard tie-breaking system (i.e., averaging the rank over the tied elements). However, as we will see below in Remark 38, this has some unwanted consequences in the VN framework.

Practically speaking, a VN scheme provides an ordered list of vertices to check against the vertex of interest $v^\star$. As such, it is sensible to model the order in which ties are checked (i.e., the rank of tied vertices) to be uniform at random. This motivates the following definition of nomination rank.

**Definition 34.** Consider $(g_1, g_2) \in G_n \times G_m$, $\Phi \in \mathcal{V}_{n,m}$, obfuscation function $o$, and $v^\star \in V(g_2)$. For a set $A$, let $\mathcal{T}_A$ be the set of total orderings of $A$; i.e., rankings of $A$ that do not allow ties. For each $j \in [m]$ define $r_j := |\Phi(g_1, o(g_2), v^\star)[j]|$. We define the random vector of nomination ranks of elements of $\Phi(g_1, o(g_2), v^\star)[j]$ as follows. Let $\tilde{X}_j \sim \text{Unif}(\mathcal{T}_\Phi(g_1, o(g_2), v^\star)[j])$, and for $u \in \Phi(g_1, o(g_2), v^\star)[j]$, let $x_u := \{h \mid \tilde{X}_j(h) = u\}$. Then

$$\text{rank}_{\Phi(g_1, o(g_2), v^\star)}(u) := \sum_{i<j} r_i + x_u.$$
where \( m' \) is such that \( \cup_{i \in [m']} \Phi^*(g_1, o(g_2), v^*)[i] = W \), and \( \cup_{i \in [m'-1]} \Phi^*(g_1, o(g_2), v^*)[i] \neq W \). As in the untied setting, we have the following theorem.

**Theorem 35.** Let \((G_1, G_2) \sim F_{c,n,m,\theta} \in \mathcal{N}\), and consider a vertex of interest \( v^* \in C \). We have that \( L_k(\Phi^*, v^*) = L^*_k(v^*) \) for all \( k \in [m-1] \) and all obfuscating functions \( o \).

The proof of Theorem 35 closely follows that of Theorem 20 and is the main reason behind the slightly less standard randomized tie-breaking scheme adopted in Definition 34.

In addition, an analogue of the universal consistency theorem, Theorem 26 holds as well. Again, the proof of Theorem 36 closely mirrors that of Theorem 26 and so is omitted.

**Theorem 36.** Let \( n \) and \( m \) be large enough to guarantee the existence of asymmetric graphs \( g_1 \in \mathcal{G}_n \) and \( g_2 \in \mathcal{G}_m \). Consider a VN scheme \( \Phi \in \mathcal{V}_{n,m} \), obfuscating function \( o \), and strictly increasing sequence \((\epsilon_m)_{i=1}^m\) satisfying \( \epsilon_i \in (0, \frac{1}{m}) \). Then there exists a distribution \( F_{c,n,m,\theta} \) over \( \mathcal{G}_n \times \mathcal{G}_m \) such that for each \( v^* \in C \) and each \( k \in [c-1] \),

\[
L_k^*(v^*) \leq \epsilon_{m-k} < 1 - \frac{k}{m} \leq L_k(\Phi^*, v^*),
\]

where \( 1 - \frac{k}{m} \) represents the error probability of chance performance; i.e., the error for the scheme \( \Phi \) that sets \( \Phi(g_1, o(g_2), v^*)[1] = W \) for all \( (g_1, g_2) \in \mathcal{G}_n \times \mathcal{G}_m \) and \( v^* \in C \).

If we consider sequences \((\epsilon_{m(n,i)})_{i=1}^{m(n)}\) satisfying the assumptions of Theorem 36 and \( \lim_n \epsilon_{m(n,m(n)-k} = \epsilon \in (0, 1) \) for any given \( k \), we arrive at the following Corollary, namely that level-\((k_n)\) universally consistent VN schemes do not exist for any sequence \((k_n)\) that does not grow as fast as \( m(n) = |V(G_2)| \).

**Corollary 37.** Let \( \epsilon \in (0, 1) \) be arbitrary, and consider a VN rule \( \Phi = (\Phi_{n,m(n)}) \). For any nondecreasing sequence \((k_n)\) satisfying \( k_n = o(m(n)) \) for all \( n > 0 \), there exists a sequence of distributions \((F_{c,n,m(n),\theta})\) in \( \mathcal{N} \) with nested cores such that

\[
\lim_{n \to \infty} L_{k_n}(v^*) \leq \epsilon < 1 = \lim_{n \to \infty} L_{k_n}(\Phi_{n,m(n)}, v^*).
\]

**Remark 38.** If we choose to break ties in ranks in a more traditional manner by averaging the rank over the tied elements, then the level-\(k\) Bayes optimal scheme for a given \( F_{c,n,m,\theta} \in \mathcal{N} \) varies in \( k \). This can be easily seen as follows. Define \( \Phi^* \) as in the setting without ties. A level-\(k\) Bayes optimal scheme can then easily be seen to equal

\[
\begin{align*}
\Phi^*_T(g_1, o(g_2), v^*)[1] &= \cup_{i=1}^{2k-1} \Phi^*(g_1, o(g_2), v^*)[i] \\
\Phi^*_T(g_1, o(g_2), v^*)[2] &= \Phi^*(g_1, o(g_2), v^*)[2k] \\
&\vdots \\
\Phi^*_T(g_1, o(g_2), v^*)[m-2k+2] &= \Phi^*(g_1, o(g_2), v^*)[m],
\end{align*}
\]

with Bayes error equal to

\[
1 - \sum_{i=1}^{2k-1} \Phi(g_1, o(g_2), v^*)[i];
\]

note the dependence on \( k \) here prohibits the same scheme from being Bayes optimal for all \( k \). We find the seemingly arbitrary nature of this Bayes-optimal scheme, not to mention the fact that the Bayes optimal scheme comes to depend on \( k \), to be unacceptable. A ranking scheme that needs be dependent on \( k \) makes little sense, in our view, in light of the fact that a practitioner may not know, a priori, whether she or he is interested in the top-ranked vertex or the top 50 vertices. This dependence on \( k \) contrasts sharply with Theorem 20 and is the main reason behind the slightly less standard randomized tie-breaking scheme adopted in Definition 34.

24
C Proof of Theorem 21

Herein we will provide a sketch of the proof of Theorem 21 for completeness. Let $Q$ be a permutation matrix in $\Pi_n$ that permutes precisely $k$ labels (i.e., $\sum_i Q_{i,i} = n - k$), and let $T$ denote the number of transpositions induced by $Q$. By exploiting the cyclic structure of $Q$ acting on $\text{vec}(B)$, we have that

$$
E\delta(Q) - E\delta(I_n) = E\|AQ - QB\|_F^2 - E\|A - B\|_F^2 \geq \epsilon \left( (n - k)k + \frac{k^2}{2} - T \right).
$$

Combining this expectation bound with the following McDiarmid-like concentration result will yield the proof of Theorem 21.

**Proposition 39** (Proposition 3.2 from [27]). Let $X_1, \ldots, X_m$ be a sequence of independent Bernoulli random variables where $E[X_i] = p_i$. Let $f : \{0, 1\}^m \mapsto \mathbb{R}$ be such that changing any $X_i$ to $1 - X_i$ changes $f$ by at most $M = \sup_i \sup_{X_1, \ldots, X_n} |f(X_1, \ldots, X_i, \ldots, X_n) - f(X_1, \ldots, 1 - X_i, \ldots, X_n)|$. Let $\sigma^2 = M^2 \sum_i p_i(1 - p_i)$ and let $Y = f(X_1, \ldots, X_n)$.

Then

$$
\Pr[|Y - E[Y]| \geq t\sigma] \leq 2e^{-t^2/4}
$$

for all $0 < t < 2\sigma/M$.

Indeed, we see that $X_Q := \delta(Q) - \delta(I_n)$ is a function of $N_Q$ independent Bernoulli random variables, where $N_Q = 3 \left( \binom{k}{2} + k(n - k) \right) \leq 3kn$. Let $S_P$ be the sum of these $N_P$ Bernoulli random variables, and it follows that $\text{Var}(S_P) \leq N_P/4$. By setting $t = C^{\frac{2kn}{\sigma}}$ for an appropriate constant $C > 0$ in Proposition 39 we have

$$
\Pr(X_Q \leq 0) \leq \Pr(|X_Q - E(X_Q)| \geq \epsilon(X_Q)) \leq 2\exp\left\{ -\Theta(\epsilon^2 kn) \right\}.
$$

A union bound over all such $Q$ (of which there are $\leq n^k$) and over $k$ yields

$$
\Pr(\exists Q \in \Pi_n \setminus \{I_n\} \text{ s.t. } \delta(Q) \leq \delta(I_n)) \leq \sum_{k=2}^{n} 2\exp\left\{ k \log(n) - \Theta(\epsilon^2 kn) \right\} = 2\exp\left\{ -\Theta(\epsilon^2 n) \right\},
$$

as desired.

References

[1] A. Agarwal, S. Chakrabarti, and S. Aggarwal. Learning to rank networked entities. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pages 14–23, 2006.

[2] S. Agarwal. Learning to rank on graphs. *Machine Learning*, 10(3):333–357, 2010.

[3] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9:1981–2014, 2008.

[4] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[5] D. Asta and C. R. Shalizi. Geometric network comparison. In M. Meila and T. Haskes, editors, *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*, pages 102–110, 2015.
[6] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[7] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold Regularization: A Geometric Framework for Learning from Examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[8] P. J. Bickel and A. Chen. A nonparametric view of network models and Newman-Girvan and other modularities. *Proc. National Academy of Sciences, USA*, 106:21068–21073, 2009.

[9] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web conference*, 1998.

[10] L. Chen, C. Shen, J. T. Vogelstein, and C. E. Priebe. Robust vertex classification. *IEEE transactions on pattern analysis and machine intelligence*, 38(3):578–590, 2016.

[11] L. Chen, J. T. Vogelstein, V. Lyzinski, and C. E. Priebe. A joint graph inference case study: The C. elegans chemical and electrical connectomes. *Worm*, 5(2), 2016.

[12] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21:5–30, 2006.

[13] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004.

[14] G. Coppersmith. Vertex nomination. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(2):144–153, 2014.

[15] G. A. Coppersmith and C. E. Priebe. Vertex nomination via content and context. *arXiv preprint arXiv:1201.4118*, 2012.

[16] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1997.

[17] K. K. Duh. *Learning to Rank with Partially-Labeled Data*. PhD thesis, University of Washington, 2009.

[18] P. Erdős and A. Rényi. On random graphs, I. *Publicationes Mathematicae*, 6:290–297, 1959.

[19] P. Erdős and A. Rényi. Asymmetric graphs. *Acta Mathematica Academiae Scientiarum Hungarica*, 14(3–4):295–315, 1963.

[20] D. E. Fishkind, S. Adali, H. G. Patsolic, L. Meng, V. Lyzinski, and C. E. Priebe. Seeded graph matching. *arXiv:1209.0367*, 2017.

[21] D. E. Fishkind, V. Lyzinski, H. Pao, L. Chen, and C. E. Priebe. Vertex nomination schemes for membership prediction. *The Annals of Applied Statistics*, 9(3):1510–1532, 2015.

[22] O. Frank and D. Strauss. Markov graphs. *Journal of the american Statistical association*, 81(395):832–842, 1986.

[23] P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

[24] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.
[25] H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. Nature, 411(6833):41–42, 2001.

[26] B. Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. Physical Review E, 83, 2011.

[27] J. H. Kim, B. Sudakov, and V. H. Vu. On the asymmetry of random regular graphs and random graphs. Random Structures and Algorithms, 21:216–224, 2002.

[28] H. Li. Learning to rank for information retrieval and natural language processing. Synthesis Lectures on Human Language Technologies, 4(1), 2011.

[29] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. journal of the Association for Information Science and Technology, 58(7):1019–1031, 2007.

[30] T.-Y. Liu. Learning to rank for information retrieval. Foundations and Trends in Information Retrieval, 3(3):225–331, 2009.

[31] L. Lü and T. Zhou. Link prediction in complex networks: A survey. Physica A: statistical mechanics and its applications, 390(6):1150–1170, 2011.

[32] C. Luo, W. Pang, Z. Wang, and C. Lin. Hete-CF: Social-based collaborative filtering recommendation using heterogeneous relations. In Proceedings of the IEEE International Conference on Data Mining, pages 917–922, 2014.

[33] D. Lusseau and M. E. J. Newman. Identifying the role that animals play in their social networks. Proceedings of the Royal Society of London B: Biological Sciences, 271(Suppl 6):S477–S481, 2004.

[34] U. Von Luxburg. A tutorial on spectral clustering. Statistics and computing, 17(4):395–416, 2007.

[35] V. Lyzinski. Information recovery in shuffled graphs via graph matching. IEEE Transactions on Information Theory, 64(5):3254–3273, 2018.

[36] V. Lyzinski, D. Fishkind, M. Fiori, J.T. Vogelstein, C.E. Priebe, and G. Sapiro. Graph matching: Relax at your own risk. Pattern Analysis and Machine Intelligence, IEEE Transactions on, In press, 2015.

[37] V. Lyzinski, K. Levin, D. E. Fishkind, and C. E. Priebe. On the consistency of the likelihood maximization vertex nomination scheme: Bridging the gap between maximum likelihood estimation and graph matching. Journal of Machine Learning Research, 17(179):1–34, 2016.

[38] V. Lyzinski and D. L. Sussman. Matchability of heterogeneous networks pairs. arXiv preprint arXiv:1705.02294, 2018.

[39] V. Lyzinski, D. L. Sussman, M. Tang, A. Athreya, and C. E. Priebe. Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding. Electronic Journal of Statistics, 8:2905–2922, 2014.

[40] H. Ma, I. King, and M. R. Lyu. Mining web graphs for recommendations. IEEE Transactions on Knowledge and Data Engineering, 24(6):1051–1064, 2012.

[41] C. Manning, P. Raghavan, and H. Schütze. Introduction to Information Retrieval. Cambridge University Press, 2008.
[42] D. Marchette, C. E. Priebe, and G. Coppersmith. Vertex nomination via attributed random dot product graphs. In Proceedings of the 57th ISI World Statistics Congress, volume 6, page 16, 2011.

[43] R. Mihalcea and D. Radev. Graph-based Natural Language Processing and Information Retrieval. Cambridge University Press, 2011.

[44] M. E. J. Newman. A measure of betweenness centrality based on random walks. Social networks, 27(1):39–54, 2005.

[45] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E, 74(3):036104, 2006.

[46] M. E. J. Newman and A. Clauset. Structure and inference in annotated networks. Nature Communications, 7(11863), 2016.

[47] H. G. Patsolic, Y. Park, V. Lyzinski, and C. E. Priebe. Vertex nomination via local neighborhood matching. arXiv preprint arXiv:1705.00674, 2017.

[48] T. A. N. Pham, X. Li, G. Cong, and Z. Zhang. A general recommendation model for heterogeneous networks. IEEE Transactions on Knowledge and Data Engineering, 28(12):3140–3153, 2016.

[49] P. Resnick and H. R. Varian. Recommender systems. Communications of the ACM, 40(3):56–58, 1997.

[50] F. Ricci, L. Rokach, and B. Shapira. Introduction to Recommender Systems Handbook. Springer, 2011.

[51] G. Robins, P. Pattison, Y. Kalish, and D. Lusher. An introduction to exponential random graph (p*) models for social networks. Social networks, 29(2):173–191, 2007.

[52] K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. Annals of Statistics, 39:1878–1915, 2011.

[53] S. Rothe and H. Schütze. CoSimRank: A flexible & efficient graph-theoretic similarity measure. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 1392–1402, 2014.

[54] T. Snijders, P. Pattison, G. Robins, and M. Handcock. New specifications for exponential random graph models. Sociological methodology, 36(1):99–153, 2006.

[55] I. Steinwart. Support vector machines are universally consistent. Journal of Complexity, 18(3):768–791, 2002.

[56] C. Stone. Consistent nonparametric regression. Annals of Statistics, 5:595–645, 1977.

[57] M. Sun and C. E. Priebe. Efficiency investigation of manifold matching for text document classification. Pattern Recognition Letters, 34(11):1263–1269, 2013.

[58] D. L. Sussman, V. Lyzinski, Y. Park, and C. E. Priebe. Matched filters for noisy induced subgraph detection. arXiv preprint arXiv:1803.02423, 2018.

[59] D. L. Sussman, M. Tang, D. E. Fishkind, and C. E. Priebe. A consistent adjacency spectral embedding for stochastic blockmodel graphs. Journal of the American Statistical Association, 107(499):1119–1128, 2012.
[60] S. Suwan, D. S. Lee, and C. E. Priebe. Bayesian vertex nomination using content and context. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(6):400–416, 2015.

[61] M. Tang, A. Athreya, D. L. Sussman, V. Lyzinski, Y. Park, and C. E. Priebe. A semiparametric two-sample hypothesis testing problem for random graphs. *Journal of Computational and Graphical Statistics*, 26(2):344–354, 2017.

[62] M. Tang, A. Athreya, D. L. Sussman, V. Lyzinski, and C. E. Priebe. A nonparametric two-sample hypothesis testing problem for random dot product graphs. *Bernoulli*, 23(3):1599–1630, 2017.

[63] M. Tang, D. L. Sussman, and C. E. Priebe. Universally consistent vertex classification for latent positions graphs. *The Annals of Statistics*, 41(3):1406–1430, 2013.

[64] D. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *nature*, 393(6684):440, 1998.

[65] J. Yoder, L. Chen, H. Pao, E. Bridgeford, K. Levin, D. E. Fishkind, C. E. Priebe, and V. Lyzinski. Vertex nomination: The canonical sampling and the extended spectral nomination schemes. *arXiv preprint arXiv:1802.04960*, 2018.

[66] S. Young and E. Scheinerman. Random dot product graph models for social networks. In *Proceedings of the 5th international conference on algorithms and models for the web-graph*, pages 138–149, 2007.

[67] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In S. Thrun, L. K. Saul, and P. B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 321–328, 2004.