Network Science **12** (1): 21–40, 2024
doi:10.1017/nws.2023.20

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Graph-based methods for discrete choice

Kiran Tomlinson and Austin R. Benson

Cornell University, Ithaca, NY, USA
Corresponding author: Kiran Tomlinson; Email: kt@cs.cornell.edu

Action Editor: Matteo Magnani

Abstract

Choices made by individuals have widespread impacts—for instance, people choose between political candidates to vote for, between social media posts to share, and between brands to purchase—moreover, data on these choices are increasingly abundant. *Discrete choice models* are a key tool for learning individual preferences from such data. Additionally, social factors like conformity and contagion influence individual choice. Traditional methods for incorporating these factors into choice models do not account for the entire social network and require hand-crafted features. To overcome these limitations, we use graph learning to study choice in networked contexts. We identify three ways in which graph learning techniques can be used for discrete choice: learning chooser representations, regularizing choice model parameters, and directly constructing predictions from a network. We design methods in each category and test them on real-world choice datasets, including county-level 2016 US election results and Android app installation and usage data. We show that incorporating social network structure can improve the predictions of the standard econometric choice model, the multinomial logit. We provide evidence that app installations are influenced by social context, but we find no such effect on app usage among the same participants, which instead is habit-driven. In the election data, we highlight the additional insights a discrete choice framework provides over classification or regression, the typical approaches. On synthetic data, we demonstrate the sample complexity benefit of using social information in choice models.

Keywords: Discrete choice; social networks; graph learning

1. Introduction

Predicting and understanding the decisions individuals make has a host of applications, including modeling online shopping preferences (Ruiz et al., 2020), forecasting the demand for renewable energy (Axsen & Kurani, 2012; Michelsen & Madlener, 2012), and analyzing elections (Dreher et al., 2014; Glasgow, 2001). These and other scenarios are studied under the umbrella of *discrete choice* (Train, 2009), which describes any setting where people select items from a set of available alternatives. While discrete choice has its roots in econometrics, machine learning approaches have recently found great success in discrete choice applications (Seshadri et al., 2019; Rosenfeld et al., 2020; Tomlinson & Benson, 2021; Bower & Balzano, 2020). This recent interest is driven by the increasing importance of Web-based choices (e.g., purchases on Amazon or bookings on Expedia), which provide both motivating applications and benchmark datasets. These new methods extend existing econometric models—most notably the classic *conditional* or *multinomial logit* (CL/MNL) (McFadden, 1973)—by learning more complex effects, such as context-dependent and nonlinear preferences.

One of the crucial aspects of human decision making is that, as fundamentally social creatures, our preferences are strongly influenced by our social context. Viral trends, conformity,

word-of-mouth, and signaling all play roles in behavior, including choices (Feinberg et al., 2020; Axsen & Kurani, 2012). Additionally, people with similar preferences, beliefs, and identities are more likely to be friends in the first place, a phenomenon known as *homophily* (McPherson et al., 2001). Together, these factors indicate that social network structure could be very informative in predicting choices. In economics and sociology, there has been growing interest in incorporating social factors into discrete choice models (McFadden, 2010; Maness et al., 2015; Feinberg et al., 2020). However, the methods used so far in these fields have largely been limited to simple feature-based summaries of social influence [e.g., what fraction of someone's friends have selected an item (Goetzke & Rave, 2011)].

On the other hand, the machine learning community has developed a rich assortment of graph learning techniques that can incorporate entire social networks into predictive models (Kipf & Welling, 2017; Jia & Benson, 2021; Wu et al., 2020), such as graph neural networks (GNNs) and graph-based regularization. These approaches can handle longer-range interactions and are less reliant on hand-crafted features. Because of the large gulf between the discrete choice and machine learning communities, there has been almost no study of the application of graph learning methods to discrete choice, where they have the potential for major impact. Perhaps one factor hindering the use of graph learning in discrete choice is that machine learning methods are typically designed for either regression or classification. Discrete choice has several features distinguishing it from multiclass classification (its closest analog)—for instance, each observation can have a different set of available items. As a concrete example, any image could be labeled as a cat in a classification setting, but people choosing between doctors may have their options dictated by their insurance policy.

Motivated by this need, we adapt graph learning techniques to incorporate social network structure into discrete choice modeling. By taking advantage of phenomena like homophily and social contagion, these approaches improve the performance of choice prediction in a social context. In particular, we demonstrate how GNNs can be applied to discrete choice, derive Laplacian regularization for the multinomial logit model, and adapt label propagation for discrete choice. We show in synthetic data that Laplacian regularization can improve sample complexity by orders of magnitude in an idealized scenario.

To evaluate our methods, we perform experiments on real-world election data and Android app installations, with networks derived from Facebook friendships, geographic adjacency, and Bluetooth pings between phones. We find that such network structures can improve the predictions of discrete choice models in a semi-supervised learning task. For instance, Laplacian regularization improves the mean relative rank (MRR)[1] of predictions by up to a 6.8% in the Android app installation data and up to 2.6% in the 2016 US election data. In contrast with our results on app installations, we find no evidence of social influence in app usage among the same participants: social factors appear to influence the apps people get, but less so the apps they actually use. Instead, we find that app usage is dominated by personal habit. Another interesting insight provided by our discrete choice models in the app installation data is the discovery of two separate groups of participants, one in which Facebook is popular, while the other prefers Myspace.[2] We further showcase the power provided by a discrete choice approach by making counterfactual predictions in the 2016 US election data with different third-party candidates on the ballot. While a common narrative is that Clinton's loss was due to spoiler effects by third-party candidates (Chalabi, 2016; Rothenberg, 2019), our results do not support this theory, although we emphasize the likelihood of confounding factors. Our tools enable us to rigorously analyze these types of questions.

2. Related work

There is a long line of work in sociology and network science on social behavior, including effects like contagion and herding (Centola & Macy, 2007; Easley & Kleinberg, 2010; Banerjee, 1992).

More recently, there has been interest in the use of discrete choice in conjunction with network-based analysis (Feinberg et al., 2020) enabled by rich data with both social and choice components (Aharony et al., 2011). The traditional econometric approach to discrete choice modeling with social effects is to add terms to an individual's utility that depend on the actions or preferences of others (Brock & Durlauf, 2001; McFadden, 2010; Maness et al., 2015). For instance, this approach can account for an individual's desire for conformity (Bernheim, 1994). This is done by treating the choices made by a chooser's community as a feature of the chooser and applying a standard multinomial logit (Páez et al., 2008; Kim et al., 2014; Goetzke & Rave, 2011; Walker et al., 2011; Kim et al., 2018). In contrast, we focus on methods that employ the entire graph rather than derived features. This enables methods to account for longer-range interactions and phenomena such as network clustering without hand-crafting features. We are aware of one econometric paper that uses preference correlations over a full network in a choice model (Leung, 2019), but inference under this method requires Monte Carlo simulation. Laplacian regularization, on the other hand, allows us to find our model's maximum likelihood estimator with straightforward convex optimization. Mixture models are another way of incorporating structured preference heterogeneity into discrete choice, such as the mixed logit (McFadden & Train, 2000) and hierarchical Bayes models with mixture priors (Allenby & Rossi, 2006; Burda et al., 2008). Again, these approaches present significant challenges for inference, requiring Monte Carlo methods, variational approximations, or expectation maximization. Additionally, in positing unknown latent populations, mixture models ignore the key information provided by the structure of the network. Another large area of research in discrete choice concerns models that allow deviations from the axiom of *independence of irrelevant alternatives* (IIA) (Luce, 1959). Many of these models, such as the multinomial probit (Hausman & Wise, 1978), are very challenging to estimate. To keep our focus on incorporating network effects, we use tractable logit models obeying IIA. However, there are recent non-IIA models admitting efficient inference to which we could apply our methods (Seshadri et al., 2019; Bower & Balzano, 2020; Tomlinson & Benson, 2021); this is beyond the scope of the present work, but we expand further on this idea in the discussion.

In another direction, there are many machine learning methods that use network structure in predictive tasks; GNNs (Kipf & Welling, 2017; Xu et al., 2019; Wu et al., 2020) are a popular example. Discrete choice is related to classification tasks, but the set of available items (i.e., labels) is specific to each observation—additionally, discrete choice models are heavily informed by economic notions of preference and rationality (McFadden, 1973; Train, 2009). A more traditional machine learning method of exploiting network structure for classification is label propagation (Zhu & Ghahramani, 2002), which we extend to the discrete choice setting. Recent work has shown how to combine label propagation with GNNs for improved performance (Jia & Benson, 2020) and presented a unified generative model framework for label propagation, GNNs, and Laplacian regularization (Jia & Benson, 2021). The present work can be seen as an adaptation and empirical study of the methods from (Jia & Benson, 2021) for discrete choice rather than regression.

The idea of applying Laplacian regularization to discrete choice models appeared several years ago in an unpublished draft (Zhang et al., 2017). However, the draft did not provide experiments beyond binary choices [which reduces to standard semi-supervised node classification (Kipf & Welling, 2017)]. In contrast, we compare Laplacian regularization with other methods of incorporating social network structure (GNNs and propagation) on real-world multialternative choice datasets.

There is a large body of existing research on predicting app usage and installation, including using social network structure (Baeza-Yates et al., 2015; Pan et al., 2011; Xu et al., 2013), but our use of network-based discrete choice models for this problem is novel. Our approach has the advantage of being applicable to both usage and installation with minimal differences, allowing us to compare the relative importance of social structure in these settings. Another line of related work applies discrete choice models to networks in order to model edge formation (Overgoor et al., 2019; Tomlinson & Benson, 2021; Gupta & Porter, 2020; Overgoor et al., 2020).

3. Preliminaries

In a discrete choice setting, we have a universe of *items* $\mathcal{U}$ and a set of *choosers* $\mathcal{A}$. In each choice instance, a chooser $a \in \mathcal{A}$ observes a *choice set* $C \subset \mathcal{U}$ and chooses one item $i \in C$. Each item $i \in \mathcal{U}$ may be described by a vector of features $\boldsymbol{y}_i \in \mathbb{R}^{d_y}$. Similarly, a chooser a may have a vector of features $\boldsymbol{x}_a \in \mathbb{R}^{d_x}$. In the most general form, a choice model assigns choice probabilities for a to each item $i \in C$:

$$\Pr(i \mid a, C) = \frac{\exp(u_{\boldsymbol{\theta}}(i, C, a))}{\sum_{j \in C} \exp(u_{\boldsymbol{\theta}}(i, C, a))}, \tag{1}$$

where $u_{\boldsymbol{\theta}}(i, C, a)$ is the utility of item i to chooser a when seen in choice set C, a function with parameters $\boldsymbol{\theta}$. Note that since the utilities in Equation (1) can depend on the choice set, this general form can express choice probabilities that vary arbitrarily across choice sets (this is sometimes called the *universal logit*). When constructing more useful parsimonious models, the utilities $u_{\boldsymbol{\theta}}(i, C, a)$ can depend on $\boldsymbol{x}_a$, $\boldsymbol{y}_i$, both, or neither. In the simplest case—the traditional logit model—$u_{\boldsymbol{\theta}}(i, C, a) = u_i$ is constant over choosers and sets. This formulation is attractive from an econometric perspective, since it corresponds to a rationality assumption: if we suppose a chooser has underlying utilities $u_1, \ldots, u_k$ and observes a perturbation of their utilities $u_i + \varepsilon_i$ (where ε_i follows a Gumbel distribution) before selecting the maximum observed utility item, then their resulting choice probabilities take the form of a logit (McFadden, 1973).

When we add a linear term in chooser features to the logit model, the result is the *multinomial logit* (MNL) (Hoffman & Duncan, 1988; McFadden, 1973), with utilities $u_\theta(i, C, a) = u_i + \boldsymbol{\gamma_i}^T \boldsymbol{x_a}$, where u_i are item-specific utilities and $\boldsymbol{\gamma_i}$ is a vector of item-specific coefficients capturing interactions with the chooser features $\boldsymbol{x_a}$. Similarly, when we add a linear term in item features, the result is a *conditional logit* (CL), with utilities $u_i + \boldsymbol{\varphi}^T \boldsymbol{y_i}$. The *conditional multinomial logit* (CML) has both the chooser and item feature terms: $u_i + \boldsymbol{\varphi}^T \boldsymbol{y_i} + \boldsymbol{\gamma_i}^T \boldsymbol{x_a}$. In order to capture heterogeneous preferences among a group of choosers, one natural approach is to allow each chooser a to have different logit utilities. We call this a *per-chooser logit*, which is specified by per-chooser utilities $u_{\boldsymbol{\theta}}(i, C, a) = u_{ia}$. Similarly, a *per-chooser* CL has varying item feature coefficients $\boldsymbol{\varphi}_a$ for each chooser a, with $u_{\boldsymbol{\theta}}(i, C, a) = u_{ia} + \boldsymbol{\varphi}_a^T \boldsymbol{y_i}$. More generally, we call any choice model parameter that varies across choosers a *per-chooser parameter*.

In addition to this standard discrete choice setup, our settings also have a network describing the relationships between choosers. Choosers are nodes in an undirected graph $G = (\mathcal{A}, E)$ where the presence of an edge $(a, b) \in E$ indicates a connection between a and b (e.g., a friendship). We assume G is connected. The *Laplacian* of G is $L = D - A$, where D is the diagonal degree matrix of G and A is the adjacency matrix. The Laplacian has a number of useful applications, including graph clustering (Hagen & Kahng, 1991) and counting spanning trees (Merris, 1994). For our purposes, the key property of the Laplacian is that quadratic forms of L measure how much a node-wise vector differs across edges of the graph (we elaborate on this property below). We use $n = |\mathcal{A}|$, $m = |E|$, and $k = |U|$. Finally, I denotes the identity matrix.

4. Graph-based methods for discrete choice

We identify three phases in choice prediction where networks can be incorporated: networks can be used (1) to inform model parameters, (2) to learn chooser representations, or (3) to directly produce predictions. In this section, we develop representative methods in each category. We briefly describe each method before diving into more detail.

First, networks can inform inference for a model that already accounts for chooser heterogeneity. This is done by incorporating the correlations in utilities (or other choice model parameters) of individuals who are close to each other in the network; we refer to these as *preference correlations* for simplicity. Our Laplacian regularization approach (described in Section 4.1) does

exactly this, and we show that it corresponds to a Bayesian prior on network-based preference correlations. Second, networks can be used to learn latent representations of choosers that are then used as features in a choice model like the MNL. GNNs have been extensively studied as representation-learning tools—in Section 4.2, we focus on how to incorporate them into choice models, using graph convolutional networks (GCNs) (Kipf & Welling, 2017) as our canonical example. Third, direct network-based methods (such as label propagation (Zhu & Ghahramani, 2002), which repeatedly averages a node's neighboring labels) can also be used as a simple baseline for choice predictions. While this approach is simple and efficient, it lacks the proper handling of choice sets of the previous probabilistic approaches. Nonetheless, we find it a useful and effective baseline, and we adapt label propagation for discrete choice in Section 4.3.

4.1. Laplacian regularization

We begin by describing how to incorporate network information in a choice model like MNL through Laplacian regularization (Ando & Zhang, 2007). Laplacian regularization encourages parameters corresponding to connected nodes to be similar through a loss term of the form $\lambda \boldsymbol{\alpha}^T L \boldsymbol{\alpha}$, where L is the graph Laplacian (as defined in Section 3), $\boldsymbol{\alpha}$ is the vector of parameter values for each node, and λ is the scalar regularization strength. A famous identity is that $\boldsymbol{\alpha}^T L \boldsymbol{\alpha} = \sum_{(i,j)\in E} (\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j)^2$, which more clearly shows the regularization of connected nodes' parameters towards each other. This also shows that the Laplacian is positive semi-definite, since $\boldsymbol{\alpha}^T L \boldsymbol{\alpha} \geq 0$, which will be useful to preserve the convexity of the multinomial logit's (negative) log-likelihood.

The idea of using Laplacian regularization for discrete choice was proposed in (Zhang et al., 2017) (although they focused on regularizing intercept terms in binary logistic regression). We generalize the idea to be applicable to any logit-based choice model and show that it corresponds to Bayesian inference with a network correlation prior. We then specialize to the models we use in our experiments. Laplacian regularization is simple to implement, can be added to any logit-based choice model with per-chooser parameters, and only requires training one extra hyperparameter. Laplacian regularization also carries a number of advantages over another approach to accounting for structured preference heterogeneity, mixture modeling.

4.1.1. Theory of Laplacian-regularized choice models

Consider a general choice model, as in Equation (1). We split the parameters $\boldsymbol{\theta}$ into two sets $\theta_{\mathcal{A}}$ and θ_G, where parameters $\boldsymbol{\alpha} \in \theta_{\mathcal{A}}, \boldsymbol{\alpha} \in \mathbb{R}^n$ vary over choosers and parameters $\beta \in \theta_G, \beta \in \mathbb{R}$ are constant over choosers. The log-likelihood of a general choice model is

$$\ell(\boldsymbol{\theta}; \mathcal{D}) = \sum_{(i,a,C)\in\mathcal{D}} \Big[\log\left(u_{\boldsymbol{\theta}}(i, C, a)\right) - \log \sum_{j\in C} \exp(u_{\boldsymbol{\theta}}(j, C, a)) \Big]. \tag{2}$$

The Laplacian- and L_2-regularized log-likelihood (with L_2 regularization strength γ) is then

$$\ell_L(\boldsymbol{\theta}; \mathcal{D}) = \ell(\boldsymbol{\theta}; \mathcal{D}) - \frac{\lambda}{2} \sum_{\boldsymbol{\alpha}\in\theta_{\mathcal{A}}} \boldsymbol{\alpha}^T L \boldsymbol{\alpha} - \frac{\gamma}{2} \sum_{\boldsymbol{\alpha}\in\theta_{\mathcal{A}}} ||\boldsymbol{\alpha}||_2^2. \tag{3}$$

We show that regularized maximum likelihood estimation of $\boldsymbol{\theta}$ corresponds to Bayesian inference with a prior on per-chooser parameters that encourages smoothness over the network. In contrast, existing results on priors for semi-supervised regression (Xu et al., 2010; Chin et al., 2019) typically split the nodes into observed and unobserved, fixing the observed values and only considering randomness over unobserved nodes. In choice modeling, observing choices at a node only updates our beliefs about their preferences, leaving some uncertainty. Our result also allows some parameters of the choice model to be chooser dependent and others to be constant across

choosers, allowing it to be fully general over choice models. Finally, we note that L_2 regularization can also be applied to the global parameters β, which as usual corresponds to a Gaussian prior on these parameters—however, we state the result with uniform priors to emphasize the Laplacian regularization on the per-chooser parameters $\boldsymbol{\alpha}$.

Theorem. *The maximizer $\boldsymbol{\theta}^*_{\text{MLE}}$ of the Laplacian- and L_2-regularized log-likelihood $\ell_L(\boldsymbol{\theta}; \mathcal{D})$ is the maximum a posteriori estimate $\boldsymbol{\theta}^*_{\text{MAP}}$ after observing $\mathcal{D}$ under the i.i.d. priors $\boldsymbol{\alpha} \sim \mathcal{N}(0, [\lambda L + \gamma I]^{-1})$ for each $\boldsymbol{\alpha} \in \theta_{\mathcal{A}}$ and i.i.d. uniform priors for each $\beta \in \theta_G$.*

Proof. First, recall that L is positive semi-definite, so $\lambda L + \gamma I$ (with $\gamma, \lambda > 0$) is positive definite and invertible. Now, using Bayes' Theorem,

$$\begin{aligned}\boldsymbol{\theta}^*_{\text{MAP}} &= \text{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D}) \\ &= \text{argmax}_{\boldsymbol{\theta}} \frac{\Pr(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\Pr(\mathcal{D})}.\end{aligned}$$

Since $\Pr(\mathcal{D})$ is independent of the parameters and log is monotonic and increasing,

$$\boldsymbol{\theta}^*_{\text{MAP}} = \text{argmax}_{\boldsymbol{\theta}} \left[\log \Pr(\mathcal{D} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\right].$$

Notice that the first term is exactly the log-likelihood $\ell(\boldsymbol{\theta}; \mathcal{D})$. Additionally, the priors of each parameter are independent, so

$$\log p(\boldsymbol{\theta}) = \sum_{\boldsymbol{\alpha} \in \theta_{\mathcal{A}}} \log p(\boldsymbol{\alpha}) + \sum_{\beta \in \theta_G} \log p(\beta).$$

Since the priors $p(\beta)$ are uniform, they do not affect the maximizer:

$$\boldsymbol{\theta}^*_{\text{MAP}} = \text{argmax}_{\boldsymbol{\theta}} \left[\ell(\boldsymbol{\theta}; \mathcal{D}) + \sum_{\boldsymbol{\alpha} \in \theta_{\mathcal{A}}} \log p(\boldsymbol{\alpha}) \right].$$

Now consider the Gaussian priors $p(\boldsymbol{\alpha})$:

$$p(\boldsymbol{\alpha}) = (2\pi)^{n/2} \det[(\lambda L + \gamma I)^{-1}]^{-1/2} \exp\left(-\frac{1}{2}\boldsymbol{\alpha}^T [(\lambda L + \gamma I)^{-1}]^{-1} \boldsymbol{\alpha}\right).$$

Simplifying the term in the exp reveals the two regularization terms:

$$\begin{aligned}-\frac{1}{2}\boldsymbol{\alpha}^T [(\lambda L + \gamma I)^{-1}]^{-1} \boldsymbol{\alpha} &= -\frac{1}{2}\boldsymbol{\alpha}^T (\lambda L + \gamma I)\boldsymbol{\alpha} \\ &= -\frac{\lambda}{2}\boldsymbol{\alpha}^T L \boldsymbol{\alpha} - \frac{\gamma}{2}\boldsymbol{\alpha}^T \boldsymbol{\alpha} \\ &= -\frac{\lambda}{2}\boldsymbol{\alpha}^T L \boldsymbol{\alpha} - \frac{\gamma}{2}||\boldsymbol{\alpha}||_2^2.\end{aligned}$$

We thus have for a constant c independent of $\boldsymbol{\alpha}$,

$$\begin{aligned}\log p(\boldsymbol{\alpha}) &= \log\left((2\pi)^{n/2} \det[(\lambda L + \gamma I)^{-1}]^{-1/2} \exp\left(-\frac{\lambda}{2}\boldsymbol{\alpha}^T L \boldsymbol{\alpha} - \frac{\gamma}{2}||\boldsymbol{\alpha}||_2^2\right)\right) \\ &= -\frac{\lambda}{2}\boldsymbol{\alpha}^T L \boldsymbol{\alpha} - \frac{\gamma}{2}||\boldsymbol{\alpha}||_2^2 + c.\end{aligned}$$

Plugging this is in and dropping the constants not affecting the maximizer yields

$$\begin{aligned}\boldsymbol{\theta}^*_{\text{MAP}} &= \text{argmax}_{\boldsymbol{\theta}} \left[\ell(\boldsymbol{\theta}; \mathcal{D}) - \frac{\lambda}{2} \sum_{\boldsymbol{\alpha} \in \theta_{\mathcal{A}}} \boldsymbol{\alpha}^T L \boldsymbol{\alpha} - \frac{\gamma}{2} \sum_{\boldsymbol{\alpha} \in \theta_{\mathcal{A}}} ||\boldsymbol{\alpha}||_2^2 \right] \\ &= \text{argmax}_{\boldsymbol{\theta}} \ell_L(\boldsymbol{\theta}; \mathcal{D}) \\ &= \boldsymbol{\theta}^*_{\text{MLE}}.\end{aligned}$$

□

Notice that the Gaussian in the theorem above has precision (i.e., inverse covariance) matrix $\lambda L + \gamma I$. The partial correlation between the per-chooser parameters $\boldsymbol{\alpha}_i$ and $\boldsymbol{\alpha}_j$, $i \neq j$ (controlling for all other nodes) is therefore

$$-\frac{\lambda L_{ij}}{\sqrt{(\lambda L_{ii} + \gamma)(\lambda L_{jj} + \gamma)}} = \frac{\lambda A_{ij}}{\sqrt{(\lambda d_i + \gamma)(\lambda d_j + \gamma)}} \tag{4}$$

using the standard Gaussian identity relating precision and partial correlation (Liang et al., 2015) (where d_i is the degree of i). If both $d_i, d_j > 0$ and γ is small, then we can approximate

$$\frac{\lambda A_{ij}}{\sqrt{(\lambda d_i + \gamma)(\lambda d_j + \gamma)}} \approx \frac{\lambda A_{ij}}{\sqrt{(\lambda d_i)(\lambda d_j)}} = \frac{A_{ij}}{\sqrt{d_i d_j}}. \tag{5}$$

This is easy to interpret: $\boldsymbol{\alpha}_i$ and $\boldsymbol{\alpha}_j$ have partial correlation 0 when i and j are unconnected ($A_{ij} = 0$) and positive partial correlation when they are connected (larger when they have fewer other neighbors). That is, the Gaussian prior in the theorem assumes neighboring nodes have correlated preferences.

4.1.2. Laplacian-regularized logit models

To incorporate Laplacian regularization in our four logit models (logit, MNL, CL, and CML), we add per-chooser utilities v_{ia} for each item i and chooser a to the utility formulations. For instance, this results in the following utility function for a per-chooser MNL: $u_\theta(i, C, a) = u_i + \boldsymbol{x}_{\boldsymbol{a}}^T \boldsymbol{\gamma}_{\boldsymbol{i}} + v_{ia}$. While we could get rid of the global utilities u_i, L_2 regularization enables us to learn a parsimonious model where u_i is the global baseline utility and v_{ia} represents per-chooser deviations. The per-chooser parameters of a Laplacian-regularized logit are $\theta_A = \{\boldsymbol{v_i}\}_{i \in U}$, where the vector $\boldsymbol{v_i}$ stacks the values of v_{ia} for each chooser $a \in A$. All other parameters are global. The Laplacian- and L_2-regularized log-likelihood can then be written down by combining Equations (2) and (3). Crucially, since the Laplacian is positive semi-definite, the terms $-\frac{\lambda}{2} \boldsymbol{v_i}^T L \boldsymbol{v_i}$ are concave—and since all four logit log-likelihoods are concave (as is the L_2 regularization term), their regularized negative log likelihoods (NLLs) are convex. This enables us to easily learn maximum-likelihood models with standard convex optimization methods.

4.2. Graph neural networks

GNNs (Wu et al., 2020) use a graph to structure the aggregations performed by a neural network, allowing parameters for neighboring nodes to influence each other. We test the canonical GNN, a graph convolutional network (GCN) (Kipf & Welling, 2017), where node embeddings are averaged across neighbors before each neural network layer. There are many other types of GNNs (see Wu et al. (2020) for a survey)—we emphasize that we do not claim this particular GCN approach to be optimal for discrete choice. Rather, we illustrate how GNNs can be applied to choice data and encourage further exploration.

In a depth-d GCN, each layer performs the following operation, producing a sequence of embeddings $H^{(0)}, \ldots, H^{(d)}$:

$$H^{(i+1)} = \sigma(A' H^{(i)} W^{(i)}) \tag{6}$$

where $H^{(0)}$ is initialized using node features (if they are available—if not, $H^{(0)}$ is learned), σ is an activation function, $W^{(i)}$ are parameters, and $A' = (D + 2I)^{-1/2}(A + I)(D + 2I)^{-1/2}$ is the degree-normalized adjacency matrix (with self-loops). Self-loops are added to G to allow a node's current embedding to influence its embedding in the next layer. We can either use $H^{(d)}$ as the final embeddings or concatenate each layer's embedding into a final embedding H. In our experiments, we use a two-layer GCN (both with output dimension 16) and concatenate the layer embeddings. For simplicity, we fix the dropout rate at 0.5.

To apply GCNs to discrete choice, we can treat the final node embeddings as chooser features and apply an MNL, modeling utilities as $u_\theta(i, C, a) = u_i + H_a^T \boldsymbol{\gamma_i}$, where u_i and $\boldsymbol{\gamma_i}$ are per-item parameters (the intercept and embedding coefficients, respectively). If item features are also available, we add the CL term $\boldsymbol{\theta}^T \boldsymbol{y_i}$. Thanks to automatic differentiation software such as PyTorch (Paszke et al., 2019), we can train both the GCN and MNL/CML weights end-to-end. Again, any node representation learning method could be used for the embeddings H—we use a GCN for simplicity.

In general, GNNs have the advantage of being highly flexible, able to capture complex interactions between the features of neighboring nodes. However, some recent research has indicated that nonlinearity is less helpful for classification in GNNs than in traditional neural networks tasks (Wu et al., 2019). With the additional modeling power comes significant additional difficulty in training and hyperparameter selection (for embedding dimensions, depth, dropout rate, and activation function).

4.3. Choice fraction propagation

We also consider a baseline method that uses the graph to directly derive choice predictions, without a probabilistic model of choice. We extend label propagation (Zhou et al., 2004; Jia & Benson, 2021) to multialternative discrete choice. The three features distinguishing the choice setting from standard label propagation are that we can observe multiple "labels" (i.e., choices) per chooser, each observation may have had different available labels, and that not all labels are available at inference time. Given training data of observed choices of the form (i, C, a), where chooser $a \in \mathcal{A}$ chose item $i \in C \subseteq \mathcal{U}$, we assign each chooser a a vector $\boldsymbol{z_a}$ of size $k = |U|$ with each item's *choice fraction*. That is, the ith entry of $\boldsymbol{z_a}$ stores the fraction of times a chose i in the observed data out of all opportunities they had to do so (i.e., the number of times i appeared in their choice set). We use choice fraction rather than counts to normalize by the number of observations for a chooser and not to count against an item instances when it was not available.

We then apply label propagation to the vectors $\boldsymbol{z_a}$ over G. Let $Z^{(0)}$ be the matrix whose rows are $\boldsymbol{z_a}$. As in standard label propagation, we iterate $Z^{(i+1)} \leftarrow (1 - \rho)Z^{(0)} + \rho D^{-1/2} A D^{-1/2} Z^{(i)}$ until convergence, where $\rho \in [0, 1]$ is a hyperparameter that controls the strength of the smoothing. Let $Z^{(\infty)}$ denote the stationary point of the iterated map. For inference, we can use the ath row of $Z^{(\infty)}$ (in practice, we will have a matrix arbitrarily close to $Z^{(\infty)}$), denoted $\boldsymbol{z}_a^{(\infty)}$, to make predictions for chooser a. Given a choice set C, we predict a will choose the argmax of $\boldsymbol{z}_a^{(\infty)}$ restricted to items appearing in C. Note that in a semi-supervised setting, we do not observe any choices from the test choosers, so their entries of $Z^{(0)}$ will be zero. The term $(1 - \rho)Z^{(0)}$ then acts as a uniform prior, regularizing the test chooser entries of $Z^{(\infty)}$ toward 0. Since choice fraction propagation does not use chooser or item features, it is best suited to scenarios where neither are available.

Table 1. Dataset summary. $|A|$: number of choosers (aggregated at the county/precinct for elections), $|U|$: number of items, $|C|$: choice set sizes, N: number of observed choices, d_x: number of chooser features

Dataset	$\|A\|$	$\|U\|$	$\|C\|$	N	d_x	d_y
APP-INSTALL	139	127	51–127	4,039	—	—
APP-USAGE	104	121	2–55	20,564	—	1
US-ELECTION-2016	3,112	32	3–22	135,382,576	19	—
CA-ELECTION-2016	21,495	182	2	261,278,336*	17	—
CA-ELECTION-2020	17,282	170	2	225,606,176*	17	—

*Voters had more than one election (i.e., choice) on their ballots.

5. Networked discrete choice data

We now describe several datasets in which we can leverage social network structure for improved choice prediction using the methods we develop. Table 1 shows a summary of our datasets, which are available at https://osf.io/egj4q/.

5.1. Friends and Family app data

The Friends and Family dataset (Aharony et al., 2011) follows over 400 residents of a young-family community in North America during 2010-2011. The dataset is remarkably rich, capturing many aspects of the participants' lives. For instance, they were given Android phones with purpose-made logging software that captured app installation and usage as well as Bluetooth pings between participants' phones. We use the installation and usage data to construct two separate choice datasets (APP-INSTALL and APP-USAGE) and use a network built from Bluetooth pings, as in (Aharony et al., 2011). We ignore uncommon and built-in apps (for instance, we ignore apps whose package names begin with `com.android`, `com.motorola`, `com.htc`, `com.sec`, and `com.google`), leaving a universe $\mathcal{U}$ of 127 apps in APP-INSTALL and 121 in APP-USAGE (e.g., Twitter, Facebook, and Myspace).

To construct APP-INSTALL, we use scans that checked which apps were installed on each participant's phone every 10 minutes—3 hours. Each time a new app i appears in a scan for a participant a, we consider that a choice from the set of apps C that were not installed at the time of the last scan. We use a plain logit as the baseline model in APP-INSTALL, since no item features are readily available. To construct APP-USAGE, we use 30-second resolution scans of running apps. To separate usage into sessions, we select instances where a participant ran an app for the first time in the last hour. We consider such app runs to be a choice i from the set of all apps C installed on participant a's phone at that time. Our discrete choice approach enables us to account for these differences in app availability. In APP-USAGE, we use a CL with a single instance-specific item feature: *recency*, defined as $\log^{-1}$ (seconds since last use) or 0 if the user has not used the app. While it would be possible to construct more complex sets of features with additional effort (for instance categorizing different types of apps or tracking down their Android store ratings), a simple baseline suffices to demonstrate how social network structure can benefit choice modeling even in the absence of item and user features.

To form the social network G over participants for both datasets, we use Bluetooth proximity hits—like the original study (Aharony et al., 2011), we only consider hits in the month of April between 7 am and midnight (to avoid coincidental hits from neighbors at night). For each participant a, we form the link (a, b) to each of their 10 most common interaction partners b (we also tested thresholds 2–9, but our methods all performed very similarly). We perform this

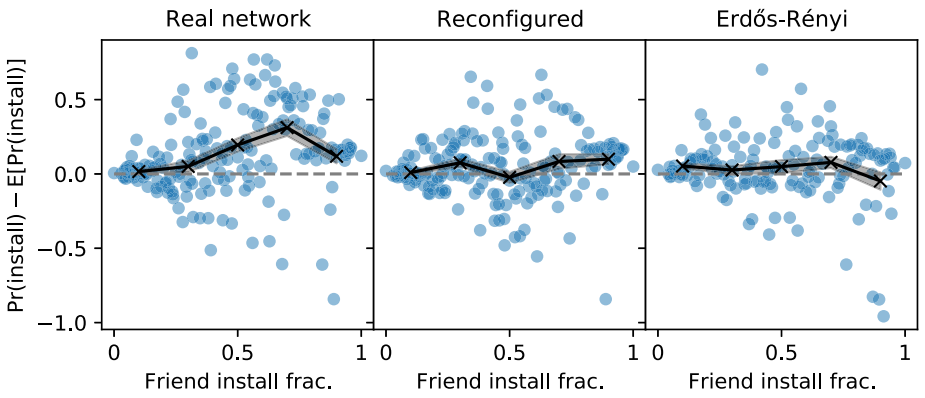

Figure 1. Difference between actual and expected install rates (if friendships were irrelevant). The left subplot is with the real network, while the right two are two null models. Each dot is a (participant, app) pair. The black line marks the mean over five bins, with the shaded region showing the standard error of the mean.

thresholding because the Bluetooth ping network is extremely dense and contains many edges that are likely not socially meaningful (for instance, nearby phones may ping each other when two strangers shop in the same store). Prior research on this data found that social contacts were useful in predicting app installations, but did not employ a discrete choice approach (Aharony et al., 2011; Pan et al., 2011). Our discrete choice approach allows us to account for multi-hop social connections and the context of each installation (i.e., what apps were already installed).

As a warm-up data analysis, we show that people are more likely to install an app the more of their friends have it (but not if we randomize friendships). Let n the total number of people, n_i be the number of people who installed application i, f_a the number of friends of person a, and f_{ai} the number of friends of person a who have app i. Suppose app installations are independent of friendships. If we sample some person a uniformly at random and check which of their friends have app i, then the probability that a also has app i is $(n_i - f_{ai})/(n - f_a)$ (simply the remaining fraction of people who have the app, after observing the friends of a). However, if app installations correlate across friendships, the observed probability would be higher when f_{ai}/f_a is larger. We measure the empirical probability that a person has an app at different friend-installation fractions. Specifically, we measure

$$\frac{1}{nk} \sum_{i \in U, a \in \mathcal{A}} \left(\mathbf{1}_{ai} - \frac{n_i - f_{ai}}{n - f_a} \right), \tag{7}$$

where $\mathbf{1}_{ai}$ is an indicator for whether person a has app i. Notice that if friendships are uncorrelated with app installations, the expectation of the summand is 0. Instead of taking the mean over all app pairs, we take the mean at each unique friend-installation fraction to see if having more friends with an app results in stronger deviations from uniform installations. This is exactly what we observe: when people have more friends with an app, they are more likely to install it (Figure 1). In contrast with two null models (a configuration model with the same degree distribution and an Erdős–Rényi graph with the same density), we see an increase in peoples' installation probabilities as a larger fraction of their friends have an app. This is in line with findings that the probability an individuals joins a social network community increases with the number of their friends in the community (Backstrom et al., 2006). However, it is worth emphasizing that this finding is purely correlational—we have no way of knowing whether increased installation rates are due to homophily in the social network, word-of-mouth contagion, or other confounding factors.

Figure 2. 2016 US presidential election vote shares for conservative independent Evan McMullin. Notice his regional popularity and the spillover from Utah to southeast Idaho. McMullin was not on the ballot in filled-in states. The lack of spillover into Colorado may be due to its crowded field (22 candidates) or because it is less conservative than Idaho.

5.2. County-level US presidential election data

US presidential election data is a common testbed for graph learning methods using a county-level adjacency network, but the typical approaches are to treat elections as binary classification or regression problems (predicting the vote shares of one party) (Jia & Benson, 2020; Zhou et al., 2020; Huang et al., 2020). However, this ignores the fact that voters have more than two options—moreover, different candidates can be on the ballot in different states. The universe of items $\mathcal{U}$ in our 2016 election data contains no fewer than 31 different candidates (and a "none of these candidates" option in Nevada, which received nearly 4% of the votes in one county). While third-party candidates are unlikely to win in the US, they often receive a nontrivial (and quite possibly consequential) fraction of votes. For instance, in the 2016 election, independent candidate Evan McMullin received 21.5% of the vote in Utah, while Libertarian candidate Gary Johnson and Green Party candidate Jill Stein received 3.3% and 1.1% nationally (the gap between Clinton and Trump was only 2%). A discrete choice approach enables us to include third-party candidates and account for different ballots in different states. As a visual example, in Figure 2, we show the states in which McMullin appeared on the ballot as well as his per-county vote share. By accounting for ballot variation, we can make counterfactual predictions about what would happen if different candidates had been on the ballot, which is difficult without a discrete choice framework. For example, given McMullin's regional support in Utah, it is possible that he would have fared better in Nevada (Utah's western neighbor) than in an East Coast state like New York. Using the entire ballots also allows us to account for one possible reason why McMullin's vote share appears not to spilled over into Colorado, while it did into Idaho: Colorado had fully 22 candidates on the ballot, while Idaho only had 8. A discrete choice approach handles this issue cleanly, while regression on vote shares does not. We note that, due to inherent limitations of observational data, we cannot be sure of the causes of the effects we observe (Tomlinson et al., 2021)—nonetheless, a discrete choice approach enables more flexible modeling and can improve prediction performance regardless of the cause of preference correlations.

We gathered county-level 2016 presidential voting data from (Kearney, 2018) and county data from (Jia & Benson, 2020),[3] which includes a county adjacency network, county-level demographic data (e.g., education, income, birth rates, USDA economic typology,[4] and unemployment rates), and the Social Connectedness Index (SCI) (Bailey et al., 2018) measuring the relative frequency of Facebook friendships between each pair of counties. We aggregate all votes at the county level, treating each county as a chooser a and using county features as $\boldsymbol{x}_a$ [modeling voting choices in aggregate is standard practice (Alvarez & Nagler, 1998)]. For the graph G, we tested using both the geographic adjacency network and a network formed by connecting each county to the 10 others with which it has the highest SCI. We found almost identical results with both networks, so we only discuss the results using the SCI network. We refer to the resulting dataset as US-ELECTION-2016.

5.3. California precinct-level election data

The presidential election data is particularly interesting because different ballots have different candidates, all running in the same election. For instance, this is analogous to having different regional availability of goods within a category in an online shopping service. In our next two datasets, CA-ELECTION-2016 and CA-ELECTION-2020, we highlight a different scenario: when ballots in different locations may have different *elections*. Extending the online shopping analogy, this emulates the case where different users view different recommended categories of items. Although it is beyond the scope of the present work, a discrete choice approach would enable measuring cross-election effects, such as coattail effects (Hogan, 2005; Ferejohn & Calvert, 1984) where higher-office elections increase excitement for down-ballot races.

To construct these datasets, we used data from the 2016 and 2020 California general elections from the Statewide Database.[5] This includes per-precinct registration and voting data as well as shapefiles describing the geographic boundaries of each precinct (California has over 20,000 voting precincts). The registration data contain precinct-level demographics (counts for party affiliation, sex, ethnicity, and age ranges), although such data were not available for all precincts. We restrict the data to the precincts for which all three data types were available: voting, registration, and shapefile (99.8% of votes cast are included in our processed 2016 data and 99.0% in our 2020 data). Again, we treat each precinct as a chooser a with demographic features $\boldsymbol{x}_a$.

Our processed California data include elections for the US Senate, US House of Representatives, California State Senate, and California ballot propositions. We set aside presidential votes due to overlap with the previous dataset and state assembly votes to keep the data size manageable. Due to California's nonpartisan top-two primary system,[6] there are two candidates running for each office—however, each voter has a different set of elections on their ballot due to differences in US congress and California state senate districts (the state has 53 congressional districts and 40 state senate districts). A discrete choice approach enables us to train a single model accounting for preferences over all types of candidates. We use the precinct adjacency network G (since SCI is not available at the finer-grained precinct level), which we constructed from the Statewide Database shapefiles using QGIS (https://qgis.org).

6. Empirical results

We begin by demonstrating the sample complexity benefit of using network structure through Laplacian regularization on synthetic data. We then apply all three approaches to our datasets, compare their performance, and demonstrate the insights provided by a networked discrete choice approach. See Table 1 in Section 5 for a dataset overview. Our code and instructions for reproducing results are available at https://github.com/tomlinsonk/graph-based-discrete-choice/.

6.1. Improved sample complexity with Laplacian regularization

By leveraging correlations between node preferences through Laplacian regularization, we need fewer samples per node in order to achieve the same inference quality. When preferences are smooth over the network, an observation of a choice by one node gives us information about the preferences of its neighbors (and its neighbors' neighbors, etc.), effectively increasing the usefulness of each observation. In Figure 3, we show the sample complexity benefit of Laplacian regularization in synthetic data with 100-node Erdős–Rényi graphs ($p = 0.1$) and preferences over 20 items generated according to the prior from Section 4.1.1. In each of 8 trials, we generate the graph, sample utilities, and then simulate a varying number of choices by each chooser. We repeat this for different homophily strengths λ. For each simulated choice, we first draw a choice set size uniformly between 2 and 20, then pick a uniformly random choice set of that size. We then measure the mean-squared error in inferred utilities of observed items (fixing the utility of the

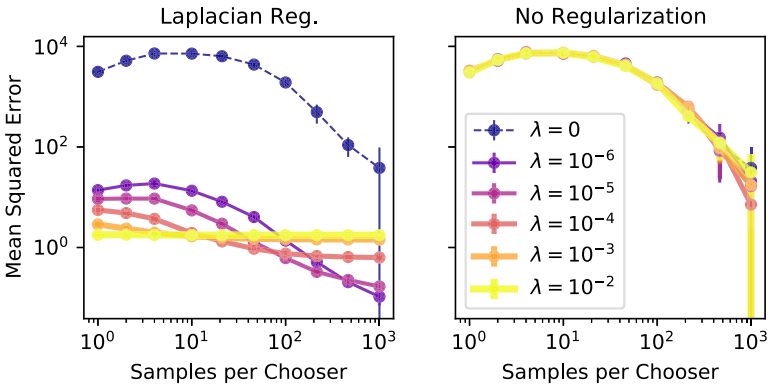

Figure 3. Estimation error of item utilities with (left) and without (right) Laplacian regularization on synthetic data generated according to the priors in Section 4.1.1, with varying homophily strength λ. Error bars (most are tiny) show standard error over 8 trials. Using Laplacian regularization can improve sample complexity by orders of magnitude.

first item to 0 for identification). When applying Laplacian regularization, we use the corresponding value of λ used to generate the data (in real-world data, this needs to be selected through cross-validation). We train the models for 100 epochs.

In this best-case scenario, we need orders of magnitude fewer samples per chooser if we take advantage of preference correlations: with Laplacian regularization, estimation error with only one sample per chooser is lower than the estimation error with no regularization and 1000 samples per chooser. The stronger the homophily, the fewer observations are needed to achieve optimal performance, since a node's neighbor's choices are more informative.

6.2. Prediction performance comparison

We now evaluate our approaches on real-world choice data. In the style of semi-supervised learning, we use a subset of choosers for training and held-out choosers for validation and testing. This emulates a scenario where it is too expensive to gather data from everyone in the network or existing data is not available for all nodes (e.g., perhaps not all individuals have consented to choice data collection). We vary the fraction of training choosers from 0.1 to 0.8 in increments of 0.1, using half of the remaining choosers for validation and half for testing. We perform 8 independent sampling trials at each fraction in the election datasets and 64 in the smaller Friends and Family datasets.

As a baseline, we use standard logit models with no network information. For the election datasets, we use an MNL that uses county/precinct features to predict votes. This approach to modeling elections is common in political science (Dow & Endersby, 2004). For APP-INSTALL, we use a simple logit. For APP-USAGE, we use a CL with recency (as defined in Section 5.1). We then compare the three graph-based methods we propose to the baseline choice model: a GCN-augmented MNL (or CML), a Laplacian-regularized logit (or CL/MNL) with per-chooser utilities, and choice fraction propagation. Aside from propagation, we train the other methods with batch Rprop (Riedmiller & Braun, 1993), as implemented in PyTorch (Paszke et al., 2019). For each dataset–model pair, we select the hyperparameters that result in the lowest validation loss in a grid search; we tested learning rates $10^{-3}, 10^{-2}, 10^{-1}$ and L_2 regularization strengths 10^{-5}, $10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$ (we also tested no L_2 regularization in the two app datasets). We similarly select Laplacian λ using validation data from $10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}$ in the election datasets (in addition to these, we also test $10^{0}, 10^{-1}, 10^{-6}, 10^{-7}$ in the app datasets) and propagation ρ from 0.1, 0.25, 0.5, 0.75, 1. The smaller hyperparameter ranges in the election datasets were used due to

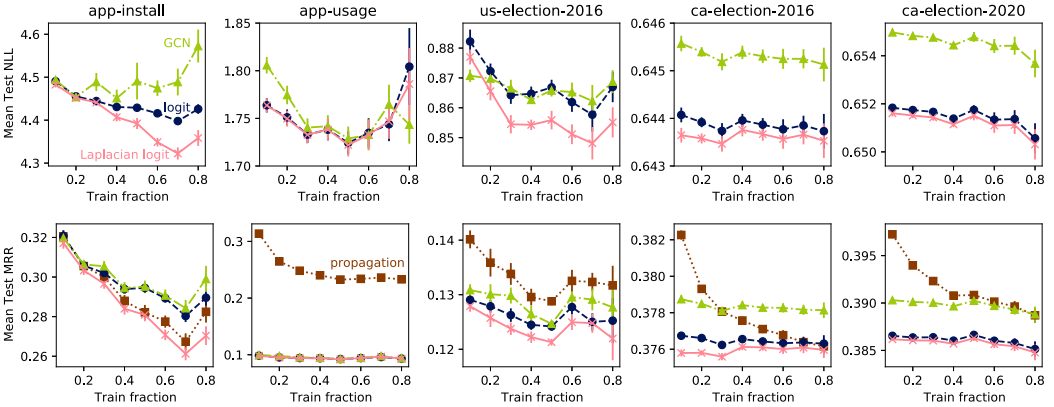

Figure 4. Test negative log likelihoods (NLL; top row; lower is better) and mean relative ranks (MRR; bottom row; lower is better) on the two Friends and Family datasets and three election datasets (error bars show standard error over chooser sampling). "Logit" signifies plain logit in APP-INSTALL, CL in APP-USAGE, and MNL in the election datasets. Laplacian regularization improves performance in APP-INSTALL, while no method improves on CL in APP-USAGE. In the election data, Laplacian MNL, but not GCN, outperforms MNL across train fractions. Propagation performs well on APP-INSTALL, but very poorly on APP-USAGE, as it does not utilize recency. Despite not using county/precinct features, propagation can be competitive in the election data.

runtime constraints. We train the likelihood-based models for 100 epochs, or until the squared gradient magnitude falls below 10^{-8}. For propagation, we perform 256 iterations, breaking if the sum of squared differences between consecutive iterates falls below 10^{-8}. We note that we did not aggressively fine-tune the GCN beyond learning rate and L_2 regularization strength, since it has many more hyperparameters than our other approaches and is more expensive to train. Our GCN results should therefore be interpreted as the performance a discrete choice practitioner should expect to achieve in a reasonable amount of time using the model, which we believe is an important metric.

In Figure 4, we show results of all four approaches on all five datasets. We evaluate the three likelihood-based methods using their test set NLL and use MRR (Tomlinson & Benson, 2021) to evaluate propagation. For one sample, MRR is defined as the relative position of the actual choice in the list of predictions in decreasing confidence order (where 0 is the beginning of the list and and 1 is the end). We then report the mean MRR over the test set. In APP-INSTALL, both Laplacian regularization and propagation improve prediction performance over the baseline logit model, and the advantage increases with the fraction of participants used for training (up to 6.8% better MRR). However, the GCN performs worse than logit in terms of likelihood and the same or worse in terms of MRR. In contrast, graph-based methods do not outperform a CL in APP-INSTALL. In the three election datasets, Laplacian-regularized MNL consistently outperforms MNL (with up to 2.6% better MRR in US-ELECTION-2016; the margin in the California data is small but outside errorbars), while the GCN performs on par with MNL in US-ELECTION-2016 and worse in the California datasets.

These results yield insight into the role networks play in different choice behaviors. In APP-USAGE, we find no benefit from using social network structure using any method. Instead, the recency feature appears to dominate, with propagation (which has no access to item features) performing much worse than the three models that do incorporate recency. This indicates that app usage is driven by individual habit rather than by external social factors. On the other hand, our results show that app *installation* has a strong social component: even simple Bluetooth proximity between friends provides a signal that they will install (but not necessarily use) similar apps. This finding highlights how combining a discrete choice approach with network data

Table 2. Runtime in seconds to train and test each model, with standard err over 4 trials

Dataset	CL/MNL	Laplacian	GCN	Propagation
APP-INSTALL	2.5 ± 0.0	2.2 ± 0.0	9.0 ± 0.2	0.2 ± 0.0
APP-USAGE	12 ± 0	13 ± 0	41 ± 0	0.9 ± 0.0
US-ELECTION-2016	18 ± 0	19 ± 0	20 ± 0.0	1.0 ± 0.0
CA-ELECTION-2016	605 ± 6	647 ± 3	758 ± 4	63 ± 0
CA-ELECTION-2020	450 ± 79	397 ± 2	485 ± 51	38 ± 0

Table 3. Edge densities within/between the groups preferring Facebook ($|F| = 70$) and Myspace ($|M| = 27$) in APP-INSTALL. Left: including the 3 choosers in $F \cap M$. Right: excluding $F \cap M$

	F	M		F	M
F	11.2%	4.9%	F	11.3%	5.8%
M	4.9%	11.7%	M	5.8%	12.0%

can illuminate the role social networks play in different choice behaviors. In the election data, especially CA-ELECTION-2016, even simple choice propagation performs remarkably well, despite *entirely ignoring demographic features*. This reveals that many of the important predictive demographic features (such as party affiliation, age, and ethnicity) are so strongly correlated over the adjacency network that we do not need to know information about you to predict your vote: it suffices to know about your neighbors or your neighbors' neighbors.

We also compare the runtime of each method. To measure runtime, each model was run on a 50-25-25 train-validation-test split of each dataset four times. Since the hyperparameters are not crucial for runtime measurements (especially because Rprop is not sensitive to initial learning rate as an adaptive method), we fixed the learning rate at 0.01, L_2 regularization strength at 0.001, Laplace regularization strength at 0.0001, and propagation ρ at 0.5. For each trial, we trained and tested each model once, shuffling the order of models to avoid systematic bias due to caching. Laplacian regularization has very low overhead over CL/MNL, while GCN is up to $4\times$ slower in the smaller datasets (see Table 2). In the larger datasets, PyTorch's built-in parallelism reduces this relative gap. Propagation is more than $10\times$ faster than the choice models in every dataset.

6.3. Facebook and Myspace communities in APP-INSTALL

Given that we observed significant improvement in prediction performance in APP-INSTALL, we take a closer look at the patterns learned by the Laplacian-regularized logit compared to the plain logit. In particular, the Facebook and Myspace apps were in the top 20 most-preferred apps under both models. Given that these were competitor apps at the time,[7] we hypothesized that they might be popular among different groups of participants. This is exactly what we observe in the learned parameters of the Laplacian-regularized logit. Facebook and Myspace are in the top 10 highest-utility apps for 70 and 27 participants, respectively (out of 139 total; we refer to these sets as F and M). Intriguingly, the overlap between F and M is only 3. Moreover, looking at the Bluetooth interaction network, we find the edge densities are more than twice as high within each of F and M than between them (Table 3), indicating they are true communities in the social network. In short, the Laplacian-regularized logit learns about two separate subcommunities, one in which Facebook is popular and one in which Myspace is popular.

Table 4. Maximum likelihood 2016 election outcomes under our model under the three scenarios in Section 6.4. We show mean vote shares (with 95% confidence interval over trials) for the top three predicted candidates and differences in state outcomes between the counterfactual prediction and reality. C: Clinton, T: Trump, Outcome: Electoral College votes. T → C denotes that a state won by Trump goes for Clinton under the model. States abbreviated by postal code

	Scenario 1	Scenario 2	Scenario 3
C %	47.7 ± 0.1	50.7 ± 0.1	37.5 ± 2.2
T %	46.3 ± 0.1	49.3 ± 0.1	37.0 ± 1.9
T → C	—	—	PA
C → T	ME*, MN, NV, NH	ME*, MN, NV, NH	MN, NV, NH
Other	—	—	RI ("None")
Outcome	T 326, C 205	T 326, C 205	T 304, C 223

*Maine allocates Electoral College votes proportionally—we assume a 3-1 split.

6.4. Counterfactuals in the 2016 US election

One of the powerful uses of discrete choice models is applying them to counterfactual scenarios to predict what might happen under different choice sets [e.g., in assortment optimization (Rusmevichientong et al., 2010)]. For instance, we can use our models to make predictions about election outcomes if different candidates had been on ballots in 2016. However, we begin this exploration with a warning: making predictions from observational data is subject to *confounders*, unobserved factors that affected both who was on which ballot and how the states voted. For example, only Nevadans had the option to vote for "None of these options," and Nevada is an outlier in a number of ways that are likely to impact voting, including its reliance on tourism, high level of diversity, and lack of income tax. This makes it less likely that the preferences of Nevadans for "None of these options" will neatly generalize to voters in other states. There are causal inference methods of managing confounding in discrete choice models; for instance, our county covariates act as regression controls (Tomlinson et al., 2021). If those covariates fully described variation in county voting preferences, then the resulting choice models would be unbiased, even with confounding (Tomlinson et al., 2021). However, we do not believe the covariates fully describe voting, since we can improve prediction by using regional or social correlations not captured by the county features. Nonetheless, examining our model's counterfactual predictions is still instructive, demonstrating an application of choice models, providing insight into the model's behavior, and motivating randomized experiments to test predictions about the effect of ballot changes. We note that the MNL we use obeys IIA, preventing relative preferences for candidates changing within a particular county when choice sets change. However, since states contain many counties, they are mixtures of MNLs (which can violate IIA), so their outcomes can change under the model.

A widespread narrative of the 2016 election is that third-party candidates cost Clinton the election by disproportionately taking votes from her (Chalabi, 2016; Rothenberg, 2019). To test this hypothesis, we examine three counterfactual scenarios: (Scenario 1) all ballots have five options: Clinton, Trump, Johnson, Stein, and McMullin; (Scenario 2) ballots only list Clinton and Trump, and (Scenario 3) ballots are as they were in 2016, but "None of these candidates" is added to every ballot. For each scenario, we take the best (validation-selected) Laplacian-regularized MNL trained on 80% of counties from each of the 8 county sampling trials and average their vote count predictions. Maximum-likelihood outcomes under the model are shown in Table 4. We find no evidence to support the claim that third-party candidates hurt Clinton more than Trump. None

of the scenarios changed the two major measures of outcome: Clinton maintained the popular vote advantage, while Trump carried the Electoral College. A few swing states change hands in the predictions. The model places more weight on "None of these candidates" than seems realistic (for instance, predicting it to be the plurality winner in Rhode Island), likely because training data is only available for this option in a single state, leading to confounding. We also note that under the true choice sets, the model's maximum likelihood state outcomes are the same as in Scenarios 1 and 2. A more complete analysis would examine the full distribution of Electoral College outcomes rather than just the maximum likelihood outcome, but we leave such analysis for future work as it is not our main focus.

7. Discussion

As we have seen, social and geographic network structure can be very useful in modeling the choices of a group of connected individuals, since people tend to have more similar preferences to their network neighborhood than to distant strangers. Several possible explanations are possible for this phenomenon: people may be more likely to become friends with similarly-minded individuals (homophily) or trends may spread across existing friendships (contagion). Unfortunately, determining whether homophily or contagion is responsible for similar behavior among friends is notoriously difficult [and often impossible (Shalizi & Thomas, 2011)].

We saw poor performance from the GCN relative to the logit models—as we noted, there are many hyperparameters that could be fine-tuned to possibly improve this performance, although this might not be practical for nonexperts. Additionally, there are a host of other GNNs that could outperform GCNs in a choice task. Our contributions in this area are to demonstrate how GNN models can be adapted for networked choice problems and to encourage further exploration of such problems. However, our findings are consistent with several lines of recent work that show simple propagation-based methods outperforming GNNs (Huang et al., 2020; Wu et al., 2019; He et al., 2020).

There are several interesting avenues for future work in graph-based methods for discrete choice. As we noted, much of the recent machine learning interest in discrete choice (Seshadri et al., 2019; Bower & Balzano, 2020; Rosenfeld et al., 2020; Tomlinson & Benson, 2021) has revolved around incorporating context effects (violations of IIA). Combining our methods with such approaches could answer questions that are to our knowledge entirely unaddressed in the literature (and possibly even unasked): Do context effects have a social component? If so, what kinds of context effects? Can we improve contextual choice prediction with social structure (in terms of accuracy or sample complexity)? Another natural extension of our work is to use a weighted Laplacian when we have a weighted social network. In another direction, choice data could be studied as an extra signal for community detection in networks, building on our identification of the Facebook and Myspace communities in the Friends and Family data.

Acknowledgments. This research was supported by ARO MURI, ARO Award W911NF19-1-0057, NSF CAREER Award IIS-2045555, and NSF DMS-EPSRC Award 2146079. We thank Marios Papachristou for helpful discussions.

Competing interests. None.

Notes

1 MRR measures the relative position of the true choice in the predicted ranking (Tomlinson & Benson, 2021).
2 The dataset is from 2010, when both were popular options.
3 One county—Oglala Lakota County, South Dakota (FIPS 46102)—was named Shannon County (FIPS 46113) until 2015, which resulted in some missing data. We manually renamed it in the data and extrapolated missing data from previous years.
4 https://www.ers.usda.gov/data-products/county-typology-codes/

5 https://statewidedatabase.org; 2016 and 2020 data accessed 8/20/20 and 3/22/21, resp.
6 https://www.sos.ca.gov/elections/primary-elections-california
7 The dataset is from 2010; Facebook surpassed Myspace's popularity in the US in 2009.

References

Aharony, N., Pan, W., Ip, C., Khayal, I., & Pentland, A. (2011). Social fMRI: Investigating and shaping social mechanisms in the real world. *Pervasive and Mobile Computing*, *7*(6), 643–659.
Allenby, G. M., & Rossi, P. E. (2006). Hierarchical Bayes models. In *The handbook of marketing research: Uses, misuses, and future advances* (pp. 418–440). Thousand Oaks: SAGE.
Alvarez, R. M., & Nagler, J. (1998). When politics and models collide: Estimating models of multiparty elections. *American Journal of Political Science*, *42*(1), 55–96.
Ando, R. K., & Zhang, T. (2007). Learning on graph with laplacian regularization. In *Advances in Neural Information Processing Systems* (pp. 25–32).
Axsen, J., & Kurani, K. S. (2012). Social influence, consumer behavior, and low-carbon energy transitions. *Annual Review of Environment and Resources*, *37*(1), 311–340.
Backstrom, L., Huttenlocher, D., Kleinberg, J., & Lan, X. (2006). Group formation in large social networks: Membership, growth, and evolution. In *KDD* (pp. 44–54).
Baeza-Yates, R., Jiang, D., Silvestri, F., & Harrison, B. (2015). Predicting the next app that you are going to use. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining* (pp. 285–294).
Bailey, M., Cao, R., Kuchler, T., Stroebel, J., & Wong, A. (2018). Social connectedness: Measurement, determinants, and effects. *Journal of Economic Perspectives*, *32*(3), 259–280.
Banerjee, A. V. (1992). A simple model of herd behavior. *The Quarterly Journal of Economics*, *107*(3), 797–817.
Bernheim, B. D. (1994). A theory of conformity. *Journal of Political Economy*, *102*(5), 841–877.
Bower, A., & Balzano, L. (2020). Preference modeling with context-dependent salient features. In *International Conference on Machine Learning* (pp. 1067–1077). PMLR.
Brock, W. A., & Durlauf, S. N. (2001). Discrete choice with social interactions. *The Review of Economic Studies*, *68*(2), 235–260.
Burda, M., Harding, M., & Hausman, J. (2008). A Bayesian mixed logit–probit model for multinomial choice. *Journal of Econometrics*, *147*(2), 232–246.
Centola, D., & Macy, M. (2007). Complex contagions and the weakness of long ties. *American Journal of Sociology*, *113*(3), 702–734.
Chalabi, M. (2016). Did third-party candidates Jill Stein and Gary Johnson lose Clinton the election? *The Guardian*. https://www.theguardian.com/us-news/2016/nov/10/third-party-candidate-gary-johnson-jill-stein-clinton-loss
Chin, A., Chen, Y., Altenburger, K. M., & Ugander, J. (2019). Decoupled smoothing on graphs. In *The World Wide Web Conference*, pp. 263–272.
Dow, J. K., & Endersby, J. W. (2004). Multinomial probit and multinomial logit: A comparison of choice models for voting research. *Electoral Studies*, *23*(1), 107–122.
Dreher, A., Gould, M., Rablen, M. D., & Vreeland, J. R. (2014). The determinants of election to the united nations security council. *Public Choice*, *158*(1), 51–83.
Easley, D., & Kleinberg, J. (2010). *Networks, crowds, and markets* (Vol. *8*). Cambridge: Cambridge University Press.
Feinberg, F., Bruch, E., Braun, M., Falk, B. H., Fefferman, N., Feit, E. M., . . .Small, M. L. (2020). Choices in networks: a research framework. *Marketing Letters*, *31*(4), 349–359.
Ferejohn, J. A., & Calvert, R. L. (1984). Presidential coattails in historical perspective. *American Journal of Political Science*, *28*(1), 127–146.
Glasgow, G. (2001). Mixed logit models for multiparty elections. *Political Analysis*, *9*(2), 116–136.
Goetzke, F., & Rave, T. (2011). Bicycle use in germany: Explaining differences between municipalities with social network effects. *Urban Studies*, *48*(2), 427–437.
Gupta, H., & Porter, M. A. (2020). Mixed logit models and network formation. *arXiv preprint arXiv*:2006.16516.
Hagen, L., & Kahng, A. (1991). Fast spectral methods for ratio cut partitioning and clustering. In *IEEE International Conference on Computer-Aided Design* (pp. 10–11). IEEE Computer Society.
Hausman, J. A., & Wise, D. A. (1978). A conditional probit model for qualitative choice: Discrete decisions recognizing interdependence and heterogeneous preferences. *Econometrica*, *46*(2), 403–426.
He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. L. (2020). Simplifying and powering graph convolution network for recommendation. In *SIGIR* (pp. 639–648).
Hoffman, S. D., & Duncan, G. J. (1988). Multinomial and conditional logit discrete-choice models in demography. *Demography*, *25*(3), 415–427.
Hogan, R. E. (2005). Gubernatorial coattail effects in state legislative elections. *Political Research Quarterly*, *58*(4), 587–597.
Huang, Q., He, H., Singh, A., Lim, S.-N., & Benson, A. (2020). Combining label propagation and simple models out-performs graph neural networks. In *International Conference on Learning Representations*.

Jia, J., & Benson, A. R. (2020). Residual correlation in graph neural network regression. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 588–598).
Jia, J., & Benson, A. R. (2021). A unifying generative model for graph learning algorithms: Label propagation, graph convolutions, and combinations. *arXiv preprint arXiv:*2101.07730.
Kearney, M. W. (2018). Presidential election county results 2016. https://github.com/mkearney/presidential_election_county_results_2016
Kim, J., Rasouli, S., & Timmermans, H. (2014). Expanding scope of hybrid choice models allowing for mixture of social influences and latent attitudes: Application to intended purchase of electric cars. *Transportation Research Part A: Policy and Practice*, *69*, 71–85.
Kim, J., Rasouli, S., & Timmermans, H. J. (2018). Social networks, social influence and activity-travel behaviour: A review of models and empirical evidence. *Transport Reviews*, *38*(4), 499–523.
Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *ICLR*.
Leung, M. P. (2019). Inference in models of discrete choice with social interactions using network data. *Available at SSRN 3446926*.
Liang, F., Song, Q., & Qiu, P. (2015). An equivalent measure of partial correlation coefficients for high-dimensional gaussian graphical models. *Journal of the American Statistical Association*, *110*(511), 1248–1265.
Luce, R. D. (1959). *Individual choice behavior*. New York: John Wiley.
Maness, M., Cirillo, C., & Dugundji, E. R. (2015). Generalized behavioral framework for choice models of social influence: Behavioral and data concerns in travel behavior. *Journal of Transport Geography*, *46*, 137–150.
McFadden, D. (1973). Conditional logit analysis of qualitative choice behavior. In P. Zarembka (Ed.), *Frontiers in econometrics* (pp. 105–142). New York: Academic Press.
McFadden, D. (2010). Sociality, rationality, and the ecology of choice. In *Choice modelling: The state-of-the-art and the state-of-practice*. Bingley: Emerald Group Publishing Limited.
McFadden, D., & Train, K. (2000). Mixed MNL models for discrete response. *Journal of applied Econometrics*, *15*(5), 447–470.
McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, *27*(1), 415–444.
Merris, R. (1994). Laplacian matrices of graphs: A survey. *Linear Algebra and Its Applications*, *197*, 143–176.
Michelsen, C. C., & Madlener, R. (2012). Homeowners' preferences for adopting innovative residential heating systems: A discrete choice analysis for germany. *Energy Economics*, *34*(5), 1271–1283.
Overgoor, J., Benson, A., & Ugander, J. (2019). Choosing to grow a graph: Modeling network formation as discrete choice. In *The World Wide Web Conference* (pp. 1409–1420).
Overgoor, J., Pakapol Supaniratisai, G., & Ugander, J. (2020). Scaling choice models of relational social data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1990–1998).
Páez, A., Scott, D. M., & Volz, E. (2008). A discrete-choice approach to modeling social influence on individual decision making. *Environment and Planning B: Planning and Design*, *35*(6), 1055–1069.
Pan, W., Aharony, N., & Pentland, A. (2011). Composite social network for predicting mobile apps installation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. *25*).
Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., . . .Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*, 8026–8037.
Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *IEEE International Conference on Neural Networks* (pp. 586–591). IEEE.
Rosenfeld, N., Oshiba, K., & Singer, Y. (2020). Predicting choice with set-dependent aggregation. In *International Conference on Machine Learning* (pp. 8220–8229). PMLR.
Rothenberg, S. (2019). How third-party votes sunk clinton, what they mean for trump. *Roll Call*. https://rollcall.com/2019/07/29/how-third-party-votes-sunk-clinton-what-they-mean-for-trump/
Ruiz, F. J., Athey, S., & Blei, D. M. (2020). Shopper: A probabilistic model of consumer choice with substitutes and complements. *The Annals of Applied Statistics*, *14*(1), 1–27.
Rusmevichientong, P., Shen, Z.-J. M., & Shmoys, D. B. (2010). Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research*, *58*(6), 1666–1680.
Seshadri, A., Peysakhovich, A., & Ugander, J. (2019). Discovering context effects from raw choice data. In *International Conference on Machine Learning* (pp. 5660–5669). PMLR.
Shalizi, C. R., & Thomas, A. C. (2011). Homophily and contagion are generically confounded in observational social network studies. *Sociological Methods & Research*, *40*(2), 211–239.
Tomlinson, K., & Benson, A. R. (2021). Learning interpretable feature context effects in discrete choice. In *KDD* (pp. 1582–1592).
Tomlinson, K., Ugander, J., & Benson, A. R. (2021). Choice set confounding in discrete choice. In *KDD* (pp. 1571–1581).
Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge: Cambridge University Press.
Walker, J. L., Ehlers, E., Banerjee, I., & Dugundji, E. R. (2011). Correcting for endogeneity in behavioral choice models with social influence variables. *Transportation Research Part A: Policy and Practice*, *45*(4), 362–374.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. (2019). Simplifying graph convolutional networks. In *International Conference on Machine Learning* (pp. 6861–6871). PMLR.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How powerful are graph neural networks? In *International Conference on Learning Representations*.

Xu, Y., Dyer, J. S., & Owen, A. B. (2010). Empirical stationary correlations for semi-supervised learning on graphs. *The Annals of Applied Statistics*, *4*(2), 589–614.

Xu, Y., Lin, M., Lu, H., Cardone, G., Lane, N., Chen, Z., . . .Choudhury, T. (2013). Preference, context and communities: A multi-faceted approach to predicting smartphone app usage patterns. In *Proceedings of the 2013 International Symposium on Wearable Computers* (pp. 69–76).

Zhang, D., Fountoulakis, K., Cao, J., Mahoney, M., & Pozdnoukhov, A. (2017). Social discrete choice models. *arXiv preprint arXiv:*1703.07520.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. In *Advances in Neural Information Processing Systems* (pp. 321–328).

Zhou, K., Dong, Y., Wang, K., Lee, W. S., Hooi, B., Xu, H., & Feng, J. (2020). Understanding and resolving performance degradation in graph convolutional networks. *arXiv preprint arXiv:*2006.07107.

Zhu, X., & Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University.

Cite this article: Tomlinson K. and Benson A. R. (2024). Graph-based methods for discrete choice. *Network Science* **12**, 21–40. https://doi.org/10.1017/nws.2023.20