# EECS 545 Final Project Report: Tractable Online Inference of Infectious Disease Dynamics on Social Networks

Patrick Harrington, Yasin Senbabaoglu, James Sweeney

December 14, 2007

## 1 Introduction

Exact inference on networks is often intractable due to the exponential complexity of updating the posterior probabilities. Dynamic Bayesian Networks (DBNs) allow for a compact representation of discrete time, finite state, Markov processes. However, DBNs are subject to the inherent complexity of exact inference. Particle filtering (PF) [1] is a Sequential Monte-Carlo based method for approximate online inference and has the desirable property that it tends to the true distribution in the limit of infinite number of samples. While particle filtering itself is a powerful method for online approximate inference, the variance associated with approximating the posterior distributions is large for high-dimensional models. Factored particle filtering (FPF) [2] aims at mitigating such high variance by imposing a factorization of the joint distribution. The factorizations should coincide with the natural decomposition of the physical system by exploiting weak interactions between different random variables in the network.

In our study, this inference engine will perform monitoring against a predictive disease surveillance application; a spatio-temporal process that makes active predictions in a population of individuals by tracking a particular trajectory of infectious disease diffusion across the social network encoded by the interactions in this population. We will refer to this predictive disease surveillance application as predictive health and disease (PHD). Because it may be unrealistic to continuously monitor each individuals' hidden states we will be presented with partial observations on only a subset of individuals and must rely on understanding the physics of disease transmission in order to make accurate predictions under this additional uncertainty. We propose using FPF as a means to exploit the highly observed community structure in social networks for tractable online inference. Various structurally inspired factorizations will be explored and the accuracy of the approximate inference will be evaluated based upon the root mean squared deviation metric. The purpose of this investigation is three fold: 1 - extending PF and FPF to monitoring a particular path of disease diffusion on a social network and 2 -exploring the effect of imposing physical

system decomposition constraints on the Bayesian Updating (Boyen-Koller factorizations), and 3-fuse the Factored Frontier (FF) and the Boyen-Koller (BK) algorithms [2, 3] so that the marginal posterior distributions are propagated forward in time instead of directly updating the joint distribution.

# 2 Preliminaries

## 2.1 Dynamic Bayesian Networks (DBN)

In this report, we are interested in making online predictions of individuals' hidden disease state in a loosely monitored social network. Here, we will assume a state-space representation of this discrete time, finite state, Markov Process where $\mathbf{Z}_{t_k} = \{Z^1_{t_k}, \ldots, Z^N_{t_k}\}$ and $\mathbf{Y}_{t_k} = \{Y^1_{t_k}, \ldots, Y^N_{t_k}\}$ represent the joint hidden and observed states, respectively, at discrete time $t_k$ and $N$ is the number of individuals (we will reserve the index $t$ for continuous time processes). Here, $Z^i_{t_k} \in \{S, I, R\}$, where $S$ represents *susceptible*, $I$ represents *infected*, and $R$ represents *recovered*. This description of the hidden disease state $\mathbf{Z}_{t_k}$ is the commonly used $SIR$ compartmental model from epidemiology. There have been many proposed variations on this low dimensional model of disease progression to model the appropriate resolution of a particular disease state. Here we will allow the *recovered* individual to transition back to a *susceptible* state, thus yielding the standard $SIRS$ model of infectious disease state progression. The observation variables in our state-space representation, $Y^i_{t_k} \rightarrow \mathbf{Y}^i_{t_k}$, can be vectors of various observations of an individuals' health state, with elements in either $\mathbb{R}^d$ and/or $\mathbb{Z}$. An example of an observation vector could be some realization of some $d$ dimensional continuous valued gene expression signature vector or some clinical response categorical variable represented as an indicator function.

In this project, we will use a DBN representation of our state-space model. A DBN can be fully specified by its two slice temporal Bayesian network ($2TBN$) representation. The structure of the $2TBN$ includes directed edges between the nodes that may be *within* and *between* the two time slices. A prior distribution must be assigned to the nodes in the first slice. The priors for the observations variables are conditioned on their corresponding hidden state, $P(Y^i_{t_k}|Z^i_{t_k})$, while the hidden states in slice one can either have conditional or marginal prior distributions, $\pi(Z^i_{t_0})$, pending on any within time slice dependencies (edges) between the hidden variables or not, respectively. The Markovian dynamic are represented by specifying the incoming directed edges from slice one to slice two. This topology of "between time slice" communication encodes the causal structure of the physical process modeled by the DBN. By specifying these cross time dependencies, we must explicitly assign a stationary transition distribution to each of the hidden nodes, $P(Z^i_{t_k}|\mathbf{Pa}(Z^i_{t_k}))$. Here, the *parent set* of $Z^i_{t_k}$ is $\mathbf{Pa}(Z^i_{t_k}) = \{\boldsymbol{\eta}(Z^i_{t_k}) \cup Z^i_{t_{k-1}}\}$, where $\boldsymbol{\eta}(Z^i_{t_k})$ are the hidden states, at time $t_{k-1}$, of the "neighbor" (adjacent) nodes of $i$ in the social network. We are now in a position to use our DBN to perform online inference. The joint transition

distribution may be written compactly as:

$$P(\mathbf{Z}_{t_k}|\mathbf{Z}_{t_{k-1}}) = \prod_{i=1}^{N} P(Z^i_{t_k}|\mathbf{Pa}(Z^i_{t_k})). \tag{1}$$

The goal will be to use this distribution for online Bayesian updating via the Chapman-Kolmogorov equations (predicting forward from $t_{k-1}$ and conditioning on current evidence at $t_k$)

$$P(\mathbf{Z}_{t_k}|\mathbf{Y}^{t_k}_{t_0}) = \frac{P(\mathbf{Y}_{t_k}|\mathbf{Z}_{t_k})}{P(\mathbf{Y}_{t_k}|\mathbf{Y}^{t_k}_{t_0})} \int P(\mathbf{Z}_{t_k}|\mathbf{Z}_{t_{k-1}})\rho_{t_{k-1}}(\mathbf{Z}_{t_{k-1}})d\mathbf{Z}_{t_{k-1}} \tag{2}$$

and find sufficient approximations to computing the high-dimensional integrals in eq (2) to allow for tractable online inference. We will refer to $\rho_{t_k}(\mathbf{Z}_{t_k}) = P(\mathbf{Z}_{t_k}|\mathbf{Y}^{t_k}_{t_0})$ as the joint belief state or filtered joint distribution. It is worth noting that given the belief state $\rho_{t_k}(\mathbf{Z}_{t_k})$ and the transition distribution $P(\mathbf{Z}_{t+\delta}|\mathbf{Z}_{t_k})$ ($\delta > 0$), we can make predictions into the future regarding the hidden infectious disease states of the individuals within our social network via the Chapman-Kolmogorov equations.

## 2.2 The Boyen-Koller Algorithm

One well known parametric approximation to representing the filtered posterior is known as the Boyen-Koller (BK) algorithm [4, 5]. This approximation is motivated by the idea that some variables/sets of variables in the physical system interact weakly with each other. The BK algorithm factorizes the joint distribution by assuming the variables form various *clusters*, or subsets of variables. Denote the set of clusters, $C = \{c_1, \ldots, c_K\}$, where each $c_j$ corresponds to a particular cluster. Each node $i$ is assigned to the $j^{th}$ cluster via the mapping $c(i) \rightarrow c_j$. Therefore, the joint filtered distribution is represented as a product of the marginals over the clusters via

$$\hat{P}(\mathbf{Z}_{t_k}|\mathbf{Y}^{t_k}_{t_0}) \approx \prod_{c \in C} \hat{P}(\mathbf{Z}^c_{t_k}|\mathbf{Y}^{t_k}_{t_0}). \tag{3}$$

By inducing factorizations indexed by $c_j$, the BK algorithm is indeed a parametric approach to inference and is sensitive to the employed clustering assignment.

At each time iteration, the filtered posterior is obtained by performing one step of Bayesian updating, prediction and conditioning on current evidence. The BK approximation occurs by using the current filtered posterior to perform the next round of Bayesian updating. The prediction step is computed using an exact inference method, such as the junction tree algorithm, and utilizes the complete structure of the DBN for updating the posterior. Because exact inference is a subroutine of the BK algorithm, this step is subject to the same complexity issues as any other exact inference method and therefore, becomes intractable complex, interconnected models. Because exact inference is performed over the entire structure, the resulting distribution will need to be projected down onto the clusters. Boyen-Koller have shown that the expected error induced by continuously applying this projection is in fact bound.

## 2.3 Particle Filtering

Particle filtering [1] is a sequential monte carlo based method for estimating the posterior distribution by generating sets of *particles*. Like any monte carlo method, the approximation converges to the true distribution in the limit of an infinite number of samples. So problems that have slower rate constants can accommodate more time for approximating the true distribution.

At each time $t_k$, a set of $M$ particles $\{\mathbf{z}_{t_k}^{(1)}, \ldots, \mathbf{z}_{t_k}^{(M)}\}$, where each $\mathbf{z}_{t_k}^{(j)}$ is some complete realization of the hidden variables $\mathbf{Z}_{t_k}$. The filtered posterior distribution is approximated by an empirical weighted sum of these particles

$$\hat{P}(\mathbf{Z}_t|\ \mathbf{Y}_{t_0}^{t_k}) = \frac{1}{M}\sum_{i=1}^{M}\delta(\mathbf{Z}_{t_k} - \mathbf{z}_{t_k}^{(i)}) \tag{4}$$

Here, $\delta(\cdot)$ is the Dirac mass function. The algorithm for employing PF on DBNs is as follows:

1. *Initialization* at $t_k = t_0$:
   For $m = 1, \ldots, M$, traverse the first slice of the *2TBN* in topological order and sample each node $Z_{t_k}^i$ according to it's prior distribution, $\pi(Z_{t_0}^i)$ (possibly conditional if $i$ has parents in same time slice).

2. *For* $t_k > t_0$:

   (a) *Importance Sampling Step*:
   For $m = 1, \ldots, M$, initialize $w_{t_k}^{(m)} = 1$, traverse the *2TBN* in topological order. For each node $Z_{t_k}^i$,

   i. If $Z_{t_k}^i$ is unobserved, generate a sample $(Z_{t_k}^i)^{(m)}$ according to $P(Z_{t_k}^i|\mathbf{pa}(Z_{t_k}^i)^{(m)})$.

   ii. If $Z_{t_k}^i$ is observed, set $(z_{t_k}^i)^{(m)}$equal to $z_{t_k}^i$. $w_{t_k}^{(m)} \leftarrow w_{t_k}^{(m)} \cdot P(z_{t_k}^i|\mathbf{pa}(Z_{t_k}^i)^{(m)})$.

   (b) *Resampling Step*: Resample with replacement $M$ particles, $\{\mathbf{z}_{t_k}^{(1)}, \ldots, \mathbf{z}_{t_k}^{(M)}\}$ with probability proportional to their weights $w_{t_k}^{(m)}$.

## 2.4 Factored Particles

As mentioned above, one drawback of employing PF for inference is the resulting high variance associated with sampling a finite number of particles in high dimensional space. Ng *et al* [2] have proposed factored particles (FP) as a means of addressing this issue. FP works on the basis of factorizing the joint distribution as in the BK algorithm so that samples of particles are generated from lower dimensional individual clusters and therefore, have a lower variance than representing the entire joint distribution as one cluster (as in normal PF). The bias induced in FP is similar to that induced in the BK method. By introducing

the factorization, the resulting distribution has a difficult time capturing inter-cluster dependencies. One important note is that in the limit of an infinite number of particles, FP converges to the BK algorithm.

Let the set of $K$ clusters be denoted by $C = \{c_1, \ldots, c_K\}$ where the set of nodes associated with the $j^{th}$ cluster with $N_{c_j}$ members is $\mathbf{Z}_{t_k}^{c_j} = \{Z_{t_k}^{c_j(1)}, \ldots, Z_{t_k}^{c_j(N_{c_j})}\}$. For each cluster $c_j$, the corresponding set of $M_{c_j}$ number of particles is $\{\mathbf{z}_{c_j,t_k}^{(1)}, \ldots, \mathbf{z}_{c_j,t_k}^{(M_{c_j})}\}$. The filtered joint posterior distribution can now be written in factored form via:

$$\hat{P}(\mathbf{Z}_{t_k}|\mathbf{Y}_{t_0}^{t_k}) \approx \prod_{c \in C} \frac{1}{M_c} \sum_{i=1}^{M_c} \delta(\mathbf{Z}_{c,t_k} - \mathbf{z}_{c,t_k}^{(i)}) \tag{5}$$

Ng *et al.* have proposed three different implementations of the FP algorithm, FP1, FP2, and FP3. FP1 and FP2 aggregate the particles via an equijoin and importance sampling method, respectively, while FP3 propagates the factored particles forward using junction tree, never forming complete particles. For this study, we will mainly concern ourselves with FP2 [2].

## 2.5 Community Structure in Networks

Since social networks have been highly observed to exhibit strong community structure, we would like to observe the effect of imposing these communities as factorizations in our BK-FPF algorithm framework for tractable online inferernce. For the dialog below, we will use the term graph and network interchangeably.

### 2.5.1 Modularity Based Clustering

Let $\mathbb{G} = \{\mathbb{V}, \mathbb{E}, \mathbf{W}\}$ denote a weighted graph which is specified by set of nodes $\mathbb{V}$, set of edges $\mathbb{E}$ and non-negative and symmetric matrix of edge weights $\mathbf{W}$. Here, $\mathbf{W}$ the weighted adjacency matrix that defines the topology of the graph as well as the edge weights. We aim at finding the the decomposition of $\mathbb{V}$ into $k$ disjoint sets $\mathbb{V}_1, \ldots, \mathbb{V}_k$, *i.e.*, $\mathbb{V} = \bigcup_{i=1}^{k} \mathbb{V}_i$ and $V_i \cap V_j = \emptyset$, $\forall i \neq j$.

Let $A, B \subset \mathbb{V}$. Let $links(A, B)$ denote the total weight of edges between $A$ and $B$:

$$links(A, B) = \sum_{i \in A, j \in B} w_{ij}. \tag{6}$$

The $links(A, B)$ term is also referred to in the literature as $assoc(A, B)$ or $cut(A, B)$, although the later term is usually reserved to the case where $A$ and $B$ are disjoint.

The *degree* (or *volume*) of a set is simply the total weight of edges from all nodes in the set to all nodes in the graph:

$$degree(A) = links(A, \mathbb{V}). \tag{7}$$

Using the *degree* as a normalization term one can define the *linkratio* by

$$linkratio(A, B) = \frac{links(A, B)}{degree(A)}. \tag{8}$$

This is the proportion of edges with $B$ among those $A$ has. One can see that $linkratio(A, A)$ measures how many links remain within $A$ itself, *i.e.*, connection within $A$, and $linkratio(A, \mathbb{V}\backslash A)$ measures how many links escape from $A$, connection of $A$ with other parts of $\mathbb{V}$.

Our goal is to partition the network into clusters that are highly connected sets while minimizing the number of edges between these clusters (as in multi-class spectral clustering [6]). A recently proposed method of clustering aims at minimizing the expected number of edges between clusters (relative to some null graph) which is captured by the modularity, $Q$ [7]. This proposal leads to maximization of the *k-way modularity function*

$$Q = kmodularity(V_1, \ldots, V_k) = \frac{1}{k} \sum_{i=1}^{k} \frac{links(V_i, V_i)}{degree(V)} - \left( \frac{degree(V_i)}{degree(V)} \right)^2 \tag{9}$$

We will obtain the clustering configuration that maximizes the modularity and use this for our factorizations in the FPF algorithm.

## 2.6 Percolation Disease Dynamics

Here we propose a commonly accepted means of modeling infectious disease dynamics on a particular contact network (*e.g.,* social network) which is motivated from the field of statistical physics [10]. The fundamental parameter of disease transmission from node $i$ to $j$ is the per unit time probability of transmission given that $i$ is in the infectious state, $r_{ij}$. If we assume that our time steps are discrete (as in our DBN framework), then if we define the duration of infection for a node $i$ to be $\tau_i$, then the probability of disease transmission from $i$ to $j$ is given by $T_{ij} = 1 - (1 - r_{ij})^{\tau_i}$. Likewise, for continuous time, we obtain $T_{ij} = 1 - e^{-r_{ij}\tau_i}$. The quantity $r_{ij}$ is a function of the strength of interaction between $i$ and $j$ as well as the pathogen under investigation, $\gamma$. We will assume that $r_{ij} = p_\gamma w_{ij}$, where $0 < p_\gamma \leq 1$ is the propensity of transmission for pathogen $\gamma$ and $w_{ij}$, $0 \leq w_{ij} \leq 1$, is the strength of social interaction between $i$ and $j$ as defined by the weighted adjacency matrix characterizing the degree social contact.

Assuming that percolation is a reasonably accurate model to infectious disease dynamics, we should attempt to align our DBN parameters (transition distributions) with the transmission parameters given by percolation. There are two ways for which the $j^{th}$ individual at discrete time $t_k$ to be infected (i.e., $Z^j_{t_k} = I$). The first is that the $j^{th}$ individual has remained infected according to his/her's individual transition distribution, $P(Z^j_{t_k} | Z^j_{t_{k-1}})$, or $j$ has became infected via transmission of the virus from one of $j$'s neighbors, $\boldsymbol{\eta}(j)$ to $j$. Therefore, the conditions for becoming infected via transmission may be

given by the union of events of $\cup_{i\in\eta(j)}\{Z^i_{t_{k-1}} = I\}$. We can represent our previously defined transmission probability between $i$ and $j$ $(i \neq j)$ in terms of a transition distribution, $P(Z^j_{t_k}|Z^i_{t_{k-1}}) = T_{ij}(Z^i_{t_{k-1}}) = 1 - (1 - r_{ij})^{\mathbb{I}_{\{Z^i_{t_{k-1}}=I\}}}$, where the duration of infection, $\tau_i$, has been absorbed into $\mathbb{I}_{\{Z^i_{t_{k-1}}=I\}}$. This allows us to construct the transition distribution for the $j^{th}$ individual,

$$P(Z^j_{t_k}|\mathbf{Pa}(Z^j_{t_k})) = \frac{1}{\Phi_j}(P(Z^j_{t_k}|Z^j_{t_{k-1}}) + \sum_{i\in\eta(j)} \mathbb{I}_{\{Z^j_{t_{k-1}}=S\}}(1 - (1 - r_{ij})^{\mathbb{I}_{\{Z^i_{t_{k-1}}=I\}}}) \tag{10}$$

where $\Phi_j$ is the partition function for this distribution and the individual disease progression term is given by

$$P(Z^i_{t_k}|Z^i_{t_{k-1}}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1-p & p \\ q & 0 & 1-q \end{bmatrix}$$

We will interpret the duration of infection term, $\tau$ (see section on percolation), to be the the expected time to transition from $I$ to $R$, i.e., $\mathbb{E}[\tau] = \frac{1}{p}$ (transitioning from $I$ to $R$ is a geometric distribution). Equation 10 allows us to fold in the physics of disease transmission via percolation theory to individual Markovian progression through the $SIR$ states. This fusion of disease transmission and progression allows us to perform online inference of the hidden disease states of all $N$ individuals in the network using our DBN framework.

M. E.J Newman [10] has derived a critical threshold for epidemics based upon the average transmission probability of the disease, $T_c$ given via

$$T_c = \frac{\mathbb{E}[k]}{\mathbb{E}[k^2] - \mathbb{E}[k]} \tag{11}$$

where the the terms in the numerator and denominator are the first and second moments of the degree distribution for the network under investigation. For average transmission probabilities below the threshold, $T < T_c$ $(T = \mathbb{E}[T_{ij}])$ diseases outbreaks are usually confined to local clusters on the network, whereas if $T \geq T_c$ large-scale epidemics can occur but are not necessarily guaranteed. Because the disease trajectories across the network will potentially evolve differently for various epidemic regimes, different $T$'s, below, at, or above $T_c$, we will explore this effect on prediction accuracy and other useful metrics for active disease prediction.

# 3 Implementation

## 3.1 Divergence Criterion

Implementation of different clustering configurations using FP, relative to the gold standard distribution generated using PF with a large number of particles, will use a heavily modified

version of Kevin Murphy's Bayes Net Toolbox for (BNT, `http://bnt.sourceforge.net/`) [8], written in MATLAB. Our metric of comparison between the gold standard and approximate filtered distributions will be the root mean squared deviation under the $s^{th}$ factorization, $q^s_{t_k} = \hat{P}(\mathbf{Z}_{t_k}|\mathbf{Y}^{t_k}_{t_0}, s)$, relative to the gold standard, $p_{t_k} = P(\mathbf{Z}_{t_k}|\mathbf{Y}^{t_k}_{t_0})$ as a function of time, $t_k$. The deviation from the gold standard, exact, distribution allows us to explore a variety of approximations for an ensemble of physical processes that we will perform online filtering against. Our factorizations will include the community structure configuration that maximizes the modularity for our chosen matrix as well as the extreme approximation of each node lying within a distinct cluster, i.e., fully factorized. These two approximations will be compared against the gold standard.

## 3.2 Ground Truth Dynamics

The underlying infectious disease dynamics on our network (ground-truth) will be simulated using a percolation model of infection coupled with a Markovian model of disease progression (see section on percolation infectious disease dynamics). In order to explore the effectiveness of our approximate inference algorithms to propagate information across the network regarding disease transmission, we will observe a random $m$ sample of individuals at each iteration. Sampling all individuals will yield accurate predictions of everyone with miniscule error divergence and neglects the realistic constraints of PHD. For deviation studies, the different inference schemes will face the same nodes sampled, the same percolation dynamics, and the same random edge weights on the social network. This avoids any bias introduced into the mean squared deviation results.

## 3.3 Network Under Investigation

To begin exploring the effect of using community structure as a means of factorization for approximation inference we have decided to initially focus on one network. Since a training set does not yet exist for the PHD application, we chose a simulated network. The chosen network is a 35 node, random, small-world network obtained using the algorithm defined in [9] (See figure 1). Real social networks have been shown to exhibit a structural phenomenon known as the 'small-world effect'. This can be interpreted as relatively tight groups of individuals that communicate to other clusters via a few individuals.

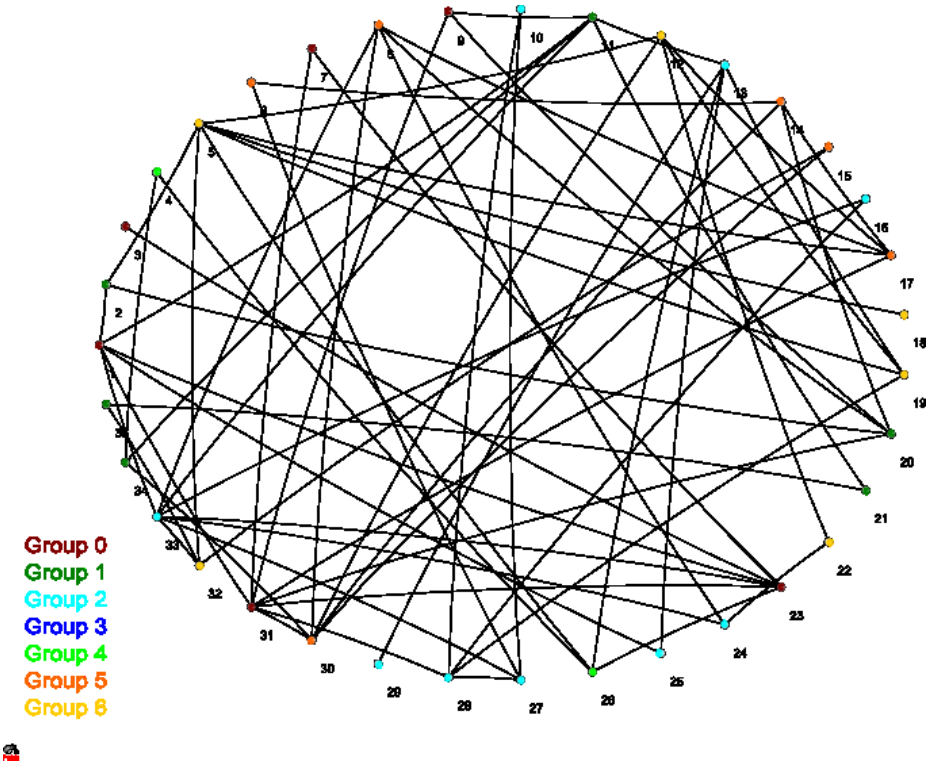


Figure 1: 35 node, random, small-world network obtained using the Wattz and Strogatz algorithm

# 4 Results and Discussion

## 4.1 Divergence of Proposed Distributions

Here, we present the results of the mean squared deviation (mean squared error) between the gold-standard distribution $p_{t_k}$ and the two approximate distributions $q^1_{t_k}$ and $q^2_{t_k}$, factorizations from modularity based clustering and fully-factorized clustering, respectively (Figure 2). The gold standard distribution, $p_{t_k}$, was computed using PF with 10,000 particles per time slice with 5 re-runs of these 10,000 particles, resulting in an average over these 5 re-runs (further sampling of state-space). The modularity clustering factorization distribution, $q^1_{t_k}$, was computed using 1,000 particles per cluster averaged over 5 re-runs per time slice while the fully-factorized distribution $q^2_{t_k}$ was determined using 100 particles per cluster with the same number of re-runs. All three of these distributions tracked the same ground truth percolation trajectory for a period of 35 time iterations.

At each time slice there were 15 randomly observed nodes that were the same for these three distributions. The propensity of disease transmission, $p_\gamma$, was taken to be equal to 0.7 (see percolation section). This resulted in the same probability rate of transmission, $r_{ij}$ across the three distributions. Our expected $\tau$ (expected time to transition from infectious

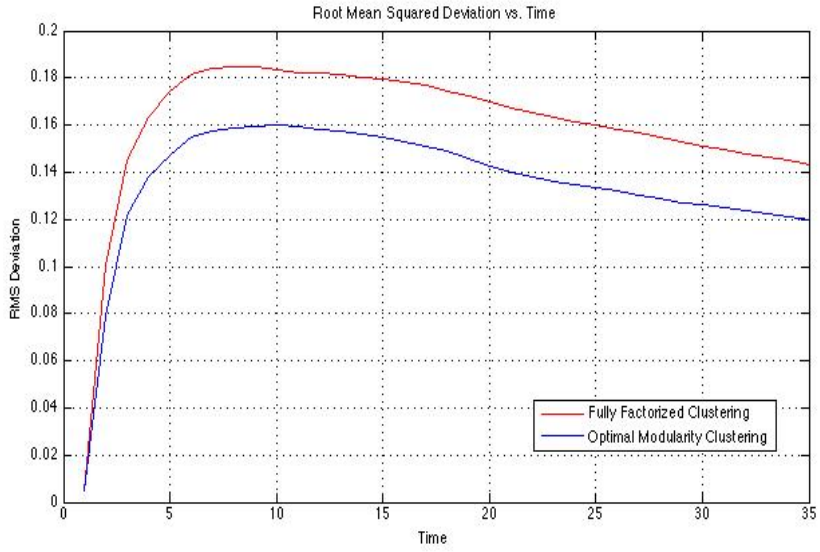


Figure 2: Root Mean Square Deviation, fully factorized modularity based

to recovered) was set to be 4. The critical average transmission probability, $T_c$, for this particular network was 0.2651 whereas the average transmission probability, given the parameters specified above, was 0.7150, well above the epidemic threshold. This was designed to produce a non-trivial (global) epidemic that affected majority of the population.

## 4.2 Prediction Error Surface

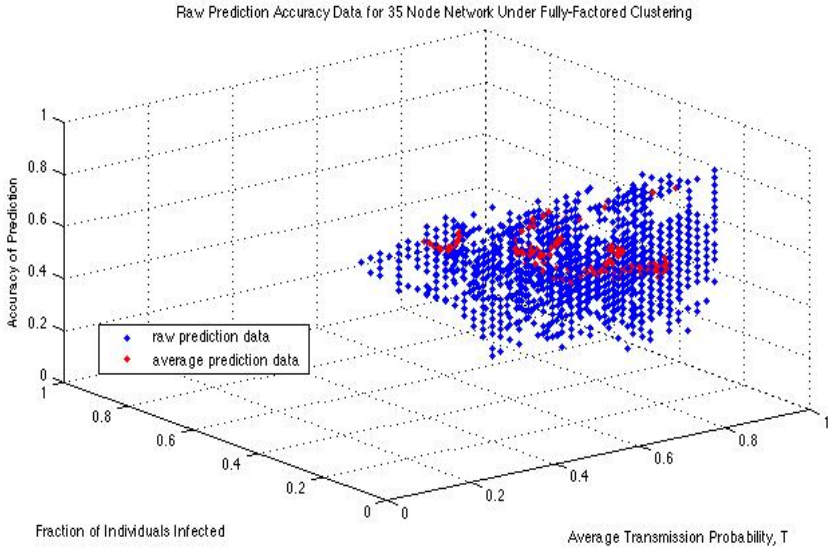


Figure 3: Prediction Error Surface, sample subset size 10, fully factorized

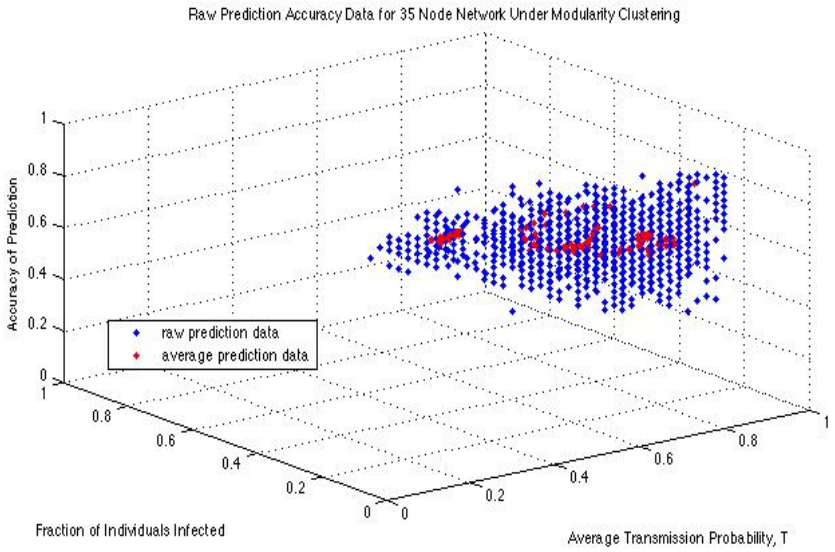


Figure 4: Prediction Error Surface, sample subset size 15, modularity based

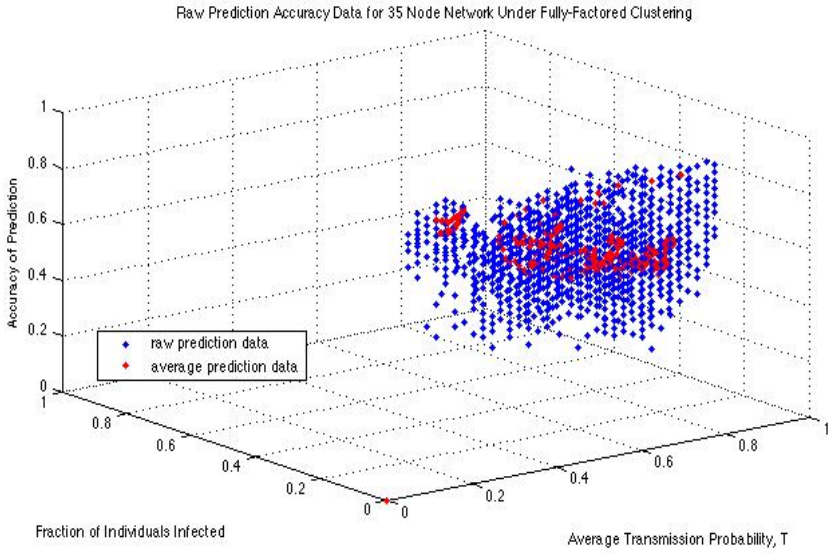


Figure 5: Prediction Error Surface, sample subset size 20, fully factorized

Here, we present prediction error surfaces for fully-factorized clustering and optimal-modularity based clustering methods. These surfaces are made using the raw prediction accuracy data for the 35-node network, given the clustering method. Prediction accuracy is plotted against fraction of individuals infected and average transmission probability. Figure 3, 5, and 7 show prediction error surfaces for varying sample subset sizes of fully factorized clustering. We see in these figures that prediction accuracy increases with increasing transmission probability, but tends to decrease with increasing fraction of infected

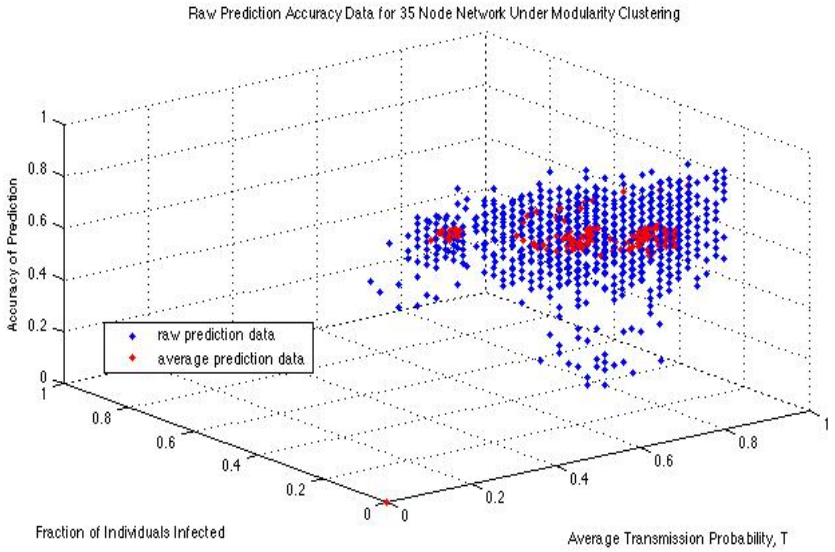


Figure 6: Prediction Error Surface, sample subset size 25, modularity based

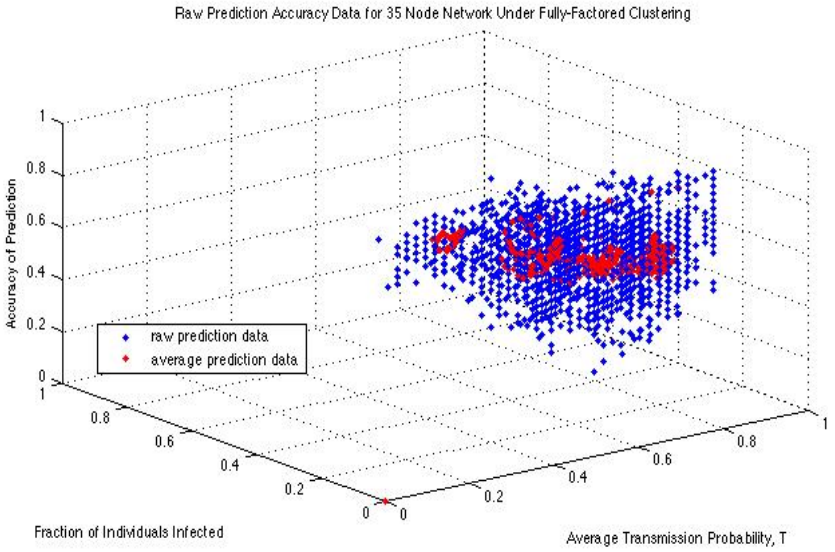


Figure 7: Prediction Error Surface, sample subset size 30, fully factorized

individuals. This pattern is apparent in all three graphs based on fully-factorized clustering. However, as we increase the sample subset size, the sample variance inherent in the data increases while bias is observed to decrese.

Using average prediction data (denoted in red), as opposed to the raw prediction data (denoted in blue) results in an attenuation effect. The trends apparent in the data are lost because of the general stability of the averages across both the fraction of infividuals affected and the average transmission probability.

As seen in Figure 4 and 6, prediction accuracies are not drastically different for optimal-modularity based clustering. However, we observe a tighter distribution meaning a smaller sample variance. Average prediction data, again, fails to represent the overall patterns in the surface.

# 5 Future Work

Given that the inference algorithms explained throughout this report have been written in matlab, we would like to extend these algorithms into C++ for its greater speed and low-level memory allocation features. This will allow much faster inference which will result in gathering better statistics and extracting useful insight to this problem. By migrating to C++, we will be able to transition to much larger social networks, i.e., on the order of hundreds to thousands of nodes. At this scale, true community structure becomes apparent whereas in 35 individuals, it may simply be a fully-connected graph. We would have much rather explored the former, but under the constraints of matlab, were confined to smaller models. Given a larger model, we anticipate having the ability to implement control and intervention strategies that aim at suppressing a particular epidemic by taking action, i.e., quarantining individuals, vaccinating them, or other possible strategies. Also, when scaling the size of the population, we are confronted with the problem of having to sample a very small subset of the population. This leads us to the problem of adaptive, of information driven, sampling strategies (given a current belief of the state at time $t_k$, who(m) to sample at time $t_{k+1}$). These issues will be addressed in future work.

# References

[1] B. Ng, L. Peshkin, and A. Pfeffer. *Factored particles for scalable monitoring*. In Proc. 18th Conf. on Uncertainty in Artificial Intelligence, Edmonton, Canada, 2002.

[2] X. Boyen and D. Koller. *Tractable inference for complex stochastic processes*. In Proc. of the Conf. on Uncertainty in AI, 1998.

[3] Boyen, X. and D. Koller (1999). *Exploiting the architecture of dynamic systems*. In Proc. AAAI.

[4] A. Doucet. *On Sequential Simulation-Based Methods for Bayesian Filtering*. Technical Report CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering, 1998.

[5] M. Newman. *Finding Community Structure in Networks Using the Eigenvectors of Matrices*. Phys. Rev. E 74, 036104 (2006).

[6] S. Yu and J. Shi. *Multiclass spectral clustering*. In Proc. of ICCV, 2003.

[7] Kevin Murphy 2001. *The Bayes Net Toolbox for Matlab*. Computing Science and Statistics, vol 33.

[8] D. Koller and U. Lerner, "Sampling in Factored Dynamic Systems," A book chapter in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J.F.G. de Freitas, and N. Gordon, Eds., Springer-Verlag 2000.

[9] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. Nature 393: 440-42 (1998)

[10] M. E. J. Newman, The spread of epidemic disease on networks. *Phys. Rev. E.* 66, 016128 (2002).

# Individual Contributions

*Patrick Harrington* - Derivation of transition distributions for DBN fused with percolation, modification of inference software (PF and FP) to allow for online inference, offline data analysis with matlab, writing.

*Yasin Senbabaoglu* - Search and design of clustering experiments with factored particles, document preparation and writing, offline data analysis with matlab.

*James Sweeney* - Network visualization, algorithm development, modification of clustering with factored particles, document preparation, offline data analysis.