# Expanding Network Communities from Representative Examples

7

ANDREW MEHLER and STEVEN SKIENA
Stony Brook University

We present an approach to leverage a small subset of a coherent community within a social network into a much larger, more representative sample. Our problem becomes identifying a small conductance subgraph containing many (but not necessarily all) members of the given seed set. Starting with an initial seed set representing a sample of a community, we seek to discover as much of the full community as possible.

We present a general method for network community expansion, demonstrating that our methods work well in expanding communities in real world networks starting from small given seed groups (20 to 400 members). Our approach is marked by incremental expansion from the seeds with retrospective analysis to determine the ultimate boundaries of our community. We demonstrate how to increase the robustness of the general approach through bootstrapping multiple random partitions of the input set into seed and evaluation groups.

We go beyond statistical comparisons against gold standards to careful subjective evaluations of our expanded communities. This process explains the causes of most disagreement between our expanded communities and our gold-standards—arguing that our expansion methods provide more reliable communities than can be extracted from reference sources/gazetteers such as Wikipedia.

Categories and Subject Descriptors: I.5.3 [**Pattern Recognition**]: Clustering

General Terms: Algorithms

Additional Key Words and Phrases: Discrete mathematics, artificial intelligence, social networks, news analysis, community discovery, graph theory

**ACM Reference Format:**

Mehler, A. and Skiena, S. 2009. Expanding network communities from representative examples. ACM Trans. Knowl. Discov. Data. 3, 2, Article 7 (April 2009), 27 pages. DOI = 10.1145/1514888.1514890 http://doi.acm.org/10.1145/1514888.1514890

## 1. INTRODUCTION

Contemporary societies are composed of many interacting communities, and so social network analysis revolves around the identification and interpretation

This work was partially supported by NSF Grants EIA-0325123 and DBI-0444815.
Author's address: A. Mehler, Department of Computer Science, Stony Brook University, Stony Brook, NY 11794-4400; email: mehler@gmail.com; S. Skiena, Department of Computer Science, Stony Brook University, Stony Brook, NY 11794-4400; email: skiena@cs.sunysb.edu.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2009 ACM 1556-4681/2009/04-ART7 $5.00
DOI 10.1145/1514888.1514890 http://doi.acm.org/10.1145/1514888.1514890

of these communities. Community discovery is a well-studied and fundamental classification problem in social network analysis [Gibson et al. 1998; Hopcroft et al. 2003; Tyler et al. 2003; Wu and Huberman 2004]. Many problems can benefit from knowledge of the underlying communities in a network, including such classic NLP applications as information retrieval and question answering (e.g., "Find all baseball players implicated in the steroid scandal"). The evolution and growth of a community can be predicted, and their interactions studied. Ideology of a community can be observed, and the flow of ideas between communities studied. Identification of communities enables us to observe community-wide sentiment, and study the degree to which a community's sentiment influences each of its members.

Community discovery in networks is typically considered as a clustering problem, where one seeks to identify dense subgraphs of relationships with relatively weak connections to outlying nodes, reviewed in Section 2.2. However, our experience with such methods has proved disappointing. Seldom do such small conductance clusters reflect homogeneous, well-defined, natural communities. This is particularly true for communities of nontrivial size in large graphs.

This problem emerged in the context of our *Lydia* news and blog analysis system [Bautin and Skiena 2007; Godbole et al. 2007; Kil et al. 2005; Lloyd et al. 2006; Lloyd et al. 2005; Lloyd et al. 2006; Mehler et al. 2006], which interprets feeds from over one thousand newspapers on a daily basis.[1] Particularly relevant here is our construction of a network of literally hundreds of thousands of people in the news, with edges between pairs of figures with statistically significant collocations/interactions. Weights on these edges measure the strength of interaction between the entity pair. Figure 1 presents a drawing of a small portion of our network around the news entity *George W. Bush*.

A theoretical explanation for the difficulties in finding large communities is provided by Leskovec et al. [2008], who demonstrate that natural networks observing power-law distributions contain core structures which mask large natural community structures. Extensive experiments on over one hundred real-world networks demonstrate that large, statistically significant small conductance clusters simply do not exist in these networks. Our news network is no different from the rest; its Network Community Profile (NCP) as defined by Leskovec et al. [2008] show that significant clusters larger than twenty vertices simply do not occur in our network.

And yet large natural communities do exist in news networks. Table I summarizes the community properties of four natural communities (baseball, basketball, and American football players, as well as movie actors) in a large network derived from news corpora comprising four years of data from essentially every daily U.S. newspaper. Observe that each of the communities has edge densities roughly two orders of magnitude greater than the network as a whole. This is a typical property of natural communities, yet does not prove sufficient support to stand out among the statistical background in an unguided search.

[1]Visit *http://www.textmap.com* for a full picture of *Lydia*'s entity/relationship extraction, sentiment analysis, and trend recognition capabilities.

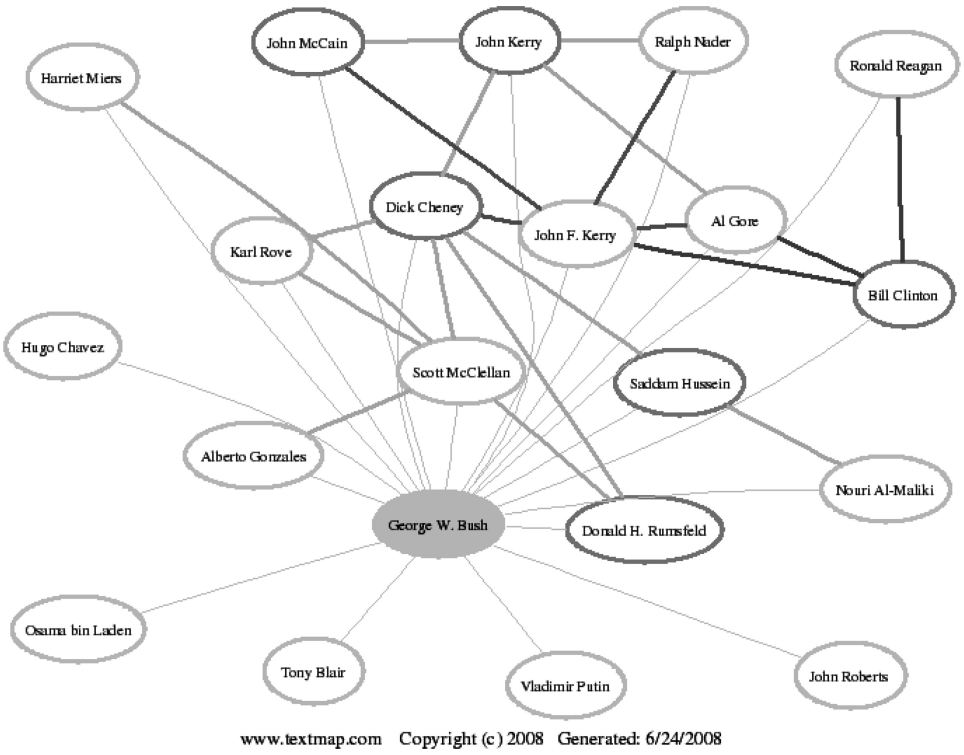


Fig. 1. A portion of the news entity network around *George W. Bush*.

Table I.
Community properties of our "dailies" news network, plus subgraphs corresponding to four natural communities. All communities have substantially higher in-community edge density than the full network.

| | Network | Baseball | Basketball | Football | Movie Stars |
|---|---|---|---|---|---|
| Vertices | 299,486 | 4,872 | 1,653 | 6,514 | 2,703 |
| Edges | 594,884 | 36,509 | 10,358 | 16,745 | 8,081 |
| In-community Degree | 3.97 | 14.98 | 12.53 | 5.14 | 5.98 |
| In-community Density | $1.30 \times 10^{-5}$ | $3.08 \times 10^{-3}$ | $7.59 \times 10^{-3}$ | $7.90 \times 10^{-4}$ | $2.21 \times 10^{-3}$ |

We assert that there is enough information in real-world social network data to amplify a small subset of a coherent community into a much larger, more representative sample. Our problem becomes identifying a small conductance subgraph containing many (but not necessarily all) members of the given seed set. This version of the community discovery problem attempts to leverage partial knowledge of a community. Starting with an initial seed set that is a sampling of a community, we seek to discover as much of the full community as possible.

Our approach is marked by incremental expansion of the seed community with retrospective analysis to determine the ultimate boundaries of our community. Our incremental method repeatedly identifies the optimal "next" unlabeled vertex $v$ to select for the community, based in some manner on the

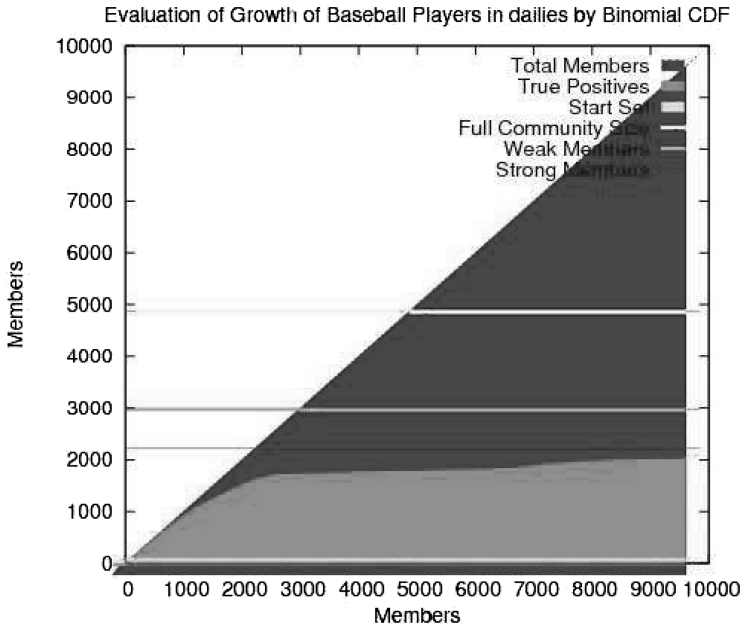


Fig. 2. Expanding the 'baseball player' community from 100 seeds. The first thousand members added to our seed group consist almost exclusively of gold-standard baseball players, although this fraction drops off during the next thousand insertions. After about two thousand insertions, the precision drops until new members are selected essentially at random.

number or strength of $v$'s neighbors who had previously been identified as members.

Figure 2 illustrates this process, expanding a seed set of 100 major league baseball players into a community of roughly 2,000 members highly enriched for baseball players. The $x$-axis represents the order of selection for our expanded community. The composition of the community at each point in time is represented by three shaded regions representing (1) the initial seed set of 100 members, (2) the selected members who are on a gold-standard roster of known baseball players, and (3) the false positives selected for the community but not on the roster.

The real problem is determining when to stop the insertion process. Figure 2 demonstrates that the first thousand or so insertions are almost all members of the gold-standard community. But with subsequent insertions, we start to see nonmembers slowly creeping in. Once too many of these impostors pollute the community, however, the grower loses all discriminatory power, adding correct members at a rate no better than random. Identifying this transition point is the critical element for reliably recognizing communities.

Our solution is based on the idea of reserving some fraction of our seed set as validation members. We then monitor the frequency with which these validation members are incorporated into the community. Assuming the community is small relative to the size of the network, we anticipate relatively frequent rediscovery of validation members during the initial "community-rich" phase

of insertion. After the phase transition, there will be less frequent rediscovery as the community has expanded beyond its natural boundaries. We can then define the community to consist of all insertions prior to the detected transition.

We note that only about a third of the full gold-standard community of 4,932 people are ultimately discovered in the example of Figure 2, even at the proper threshold. This community (like all others we have studied) contains an easily discoverable core of members as well as a large, impossible-to-distinguish group of non-connected outliers. It is the presence of these outliers which ruins the conductance of the full gold-standard community. The problem isn't as serious as it may appear, because we recover almost all of the "strong" members (defined in Section 6) with substantive connections to others in the community. We will show that many of the purported false positives can be explained by difficulties in evaluation techniques.

Our contributions in this paper are as follows:

—We present a general method for network community expansion from seed sets of members. We demonstrate our methods work well in expanding communities in real world networks, even given very small seed groups (20 to 400 members).

—We demonstrate how to increase the robustness of this general approach by bootstrapping multiple random partitions of the known-member set into seed and evaluation groups.

—We perform substantive experiments measuring performance sensitivity of several important parameters, including initial seed set size, neighborhood expansion criteria, bootstrap iteration count (governing the running time), and precision/recall trade-offs.

—The boundaries of any real-world community are fluid and imprecise. We go beyond statistical comparisons against gold standards to careful subjective evaluations of our expanded communities. We identify and explain the causes of most disagreements between our expanded communities and our gold-standards—arguing that our methods can provide more reliable communities than can be extracted from reference sources/gazetteers such as Wikipedia.

Our article is organized as follows. Section 2 reviews the extensive literature on community discovery and expansion methods in networks. Sections 3 and 4 focus on criteria for identifying the next community member and terminating the expansion process, respectively. Section 5 presents our results on performance tuning and evaluation. Section 6 gives an in-depth evaluation of the quality of our expanded memberships on four representative natural communities. Finally, we summarize our conclusions in Section 7.

## 2. RELATED WORK

Although social scientists have studied communities in social networks for generations, the emergence of the Internet provides much of the motivation for network science questions such as community identification. Before the Internet, there simply was no large and reliable network data available for study [Barabasi 2003].

Thus most previous studies of real-world communities in large scale networks arise from web data. Representative is the work of Gibson et al. [1998], who examined the link topology of the world wide web, demonstrating that communities exist on the web. These communities have "authoritative" pages, and are linked together by "hub" pages, ideas fundamental to modern search engines [Kleinberg 1999]. Tyler et al. [2003] discovered organizational community structure by studying a network formed by to/from pairs in a sample of 185,773 emails between 485 HP Lab employees. They used a divisive betweenness-based technique for discovering communities.

The two major computational problems on communities are *discovery* and *identification*. Discovery is concerned with finding a group of entities that are members of a community, while identification seeks to identify the nature of a community given its membership. We only consider the discovery problem in this paper, since the desired community's identity should be known to the supplier of its seed members.

### 2.1 Definitions of Communities

Our notion of a community is external from the network—a community is a coherent group existing in the real world. Algorithmic definitions of communities must depend on network properties, however. Generally, communities are groups of vertices that are better connected within the community than outside of it [Newman 2004], such as *Web communities*, which are defined as vertex sets each with more neighbors in the community as out of it [Flake et al. 2000; Gibson et al. 1998]. It is expected (or hoped) that each such subgraph within the network will correspond to a real-world community.

Such web communities are a specialization of graph alliances [Fernau and Raible 2007]. A defensive alliance is a set of vertices where each vertex has a majority of its neighbors in the alliance. The complexity of finding alliances of a given size $k$ is NP-complete [Cami et al. 2006; Favaron et al. 2002; Jamieson et al. 2002; Shafique 2001], but is fixed parameter tractable [Fernau and Raible 2007].

### 2.2 Community Discovery Methods

Members of natural groups in a network will tend to have a high density of connections between them, with lower connectivity between different groups. Discovering communities is typically viewed as a clustering problem, with specific techniques being more applicable to social networks [Kossinets and Watts 2005]. A large class of methods deal on a global scale, where every single vertex is assigned to a single community. An overview of these methods follows.

2.2.1 *Graph Partition Techniques.* Bisection techniques attempt to partition the network into two relatively separate subgraphs. Several methods are effective to identifying a single bisection, but work less well on graphs containing many distinct communities. An external decision must be made to indicate when to stop bisecting, that is, how many communities exist in the graph [Newman 2004]. Methods include:

—*Max Flow/Min Cut.* These methods can produce good bisections, but make no guarantees about keeping both groups of similar size. Flake et al. [2004] give a min-cut algorithm based on min-cut trees which is able to produce an arbitrary number of clusters, and can be expanded to produce a hierarchical clustering.

—*Spectral Bisection.* Spectral bisection techniques partition a graph based on the eigenvectors of its Laplacian. The Laplacian $Q$ of a graph $G$ is defined as $Q = D - A$, where $D$ is an $n \times n$ diagonal matrix with $d_{v,v} = d(v)$ and $A$ is the adjacency matrix of $G$. The spectral bisection method finds the eigenvector corresponding to the second smallest eigenvalue $\lambda_2$, and bisects the graph on whether the eigenvector entry for a vertex is positive or negative. $\lambda_2$ is also called the algebraic connectivity of a graph. A smaller value indicates a better split into two groups [Newman 2004; Pothen et al. 1990].

—*Kernighan-Lin Algorithm.* This heuristic algorithm [Kernighan and Lin 1970] attempts to greedily minimize the "external cost" of a partition, which is the sum of the cost of inter-partition edges. It starts with an initial (possibly random) partition, and determines the pair of vertices whose swap would produce the largest decrease in cost. This gives a sequence of vertex swaps which is then scanned to find the minimum. The procedure is then repeated with the new partition as the starting point, until convergence on a local minimum is achieved.

2.2.2 *Hierarchical Clustering.* Hierarchical clustering techniques are driven by an application-specific similarity measure between the groups of vertices of a network [Scott 2000]. Techniques include:

—*Agglomerative.* In this top-down approach, each vertex initially belongs to its own cluster. Clusters are merged incrementally in order of increasing cost. In *single linkage* clustering, the cost of merging two clusters depends upon the closest vertex pair spanning them. In *complete linkage* cluster, the cost is a sum of the distances of all vertex pairs spanning the clusters.
Newman [2004] gives an algorithm based on *modularity* $Q$. Given a partition of the vertices, define a matrix $e$ where $e_{ij}$ is the fraction of edges in $G$ between components $i$ and $j$. Then $Q$ is defined as

$$Q = \Sigma_i e_{ii} - \Sigma_{ijk} e_{ij} e_{ki} = Trace(e) - ||e^2||.$$

At each step, we choose to merge the two clusters that cause the greatest increase in $Q$.
Agglomerative clustering methods do not find peripheral members reliably [Newman and Girvan 2004]. An additional level of processing is needed to determine at which level the hierarchy defines the most meaningful communities.

—*Divisive.* In divisive hierarchical clustering, the entire graph $G$ begins as one cluster. Edges are removed to partition the cluster into smaller ones, as opposed to agglomerative where clusters are joined to larger clusters.
[Girvan and Newman 2002, 2004] give an algorithm based on edge betweenness centrality. The edge with highest betweenness centrality is removed

from the graph until no edges remain. Edge betweenness can be calculated in $O(mn)$, giving a total computation time of $O(m^2n)$.

Clauset et al. [2008] state that hierarchical structure is actually a defining component of social networks; sufficient to explain power law degree distributions, high clustering coefficients, and short path lengths (the small world phenomenon). The *hierarchical random graph* model is a dendogram, with probabilities at internal nodes. The probability of an edge between two leaves is equal to the value in their lowest common ancestor. This model produces networks exhibiting the properties of small-world networks. They also give a statistical-based algorithm for inferring the most likely hierarchical random graph model from a given network.

2.2.3 *Other Methods.* Not all community discover methods seek to partition the network. Hopcroft et al. [2003] give an agglomerative clustering method to find natural communities. A natural community is a stable community; a cluster that should still exist if the network is slightly perturbed. By perturbing the network, and seeing which clusters are consistently found, the natural and stable groups can be found. This work is extended in Hopcroft et al. [2004], where the goal is to track these natural communities over time. Other methods include:

—*Resistor Networks.* Wu and Huberman [2004] consider the graph as a resistor network and cluster vertices based on similar electrical potential. This method scales to extremely large graphs (linear run time), and can also be modified to extract a single community from a single node (i.e., a single seed).

—*Core Collapse Sequence.* A $k$-core is a component of a graph $G$ where each vertex has degree $k$ or larger. The core collapse sequence looks at the sequence of cores for $1 \leq k \leq n - 1$ [Scott 2000].

## 2.3 Expanding Networks From Seeds

We are not the first to consider expanding communities from seeds. Perhaps the most popular example is Google Sets [Cirasella 2007], where users can expand sets of up to five items into 15 or 30 items using connections derived from analyzing itemized lists throughout the web. Inspired by this, Ghahramani and Heller [2005] developed the idea of Bayesian sets using a statistical model of sets/communities and Bayesian inference. Our problem differs from Google sets in that we are attempting to grow much larger communities (thousands of members) where we also rely on larger seed sets (tens or hundreds of members).

Flake et al. [2000] discovered Web communities using max-flow/min-cut methods, where the source set contains seed members of the community, and the sink set known noncommunity members. They also give approximation algorithms that work on a local view of the network.

Anderson and Lang [2006] investigate methods for growing communities from seeds using random walk techniques coupled with clean-up operations based on network flow. They produce highly accurate identification of community boundaries in three different domains, although their initial seed sets

generally comprise a much larger fraction of the target community than the experiments we present here.

Sarmento et al. [2007] grow entity classes from very small seed sets. They seek to estimate the membership function $\mu(S, e)$; a measure of whether entity $e$ belongs to the same classes as the seed set $S$. This is done by a cosine similarity score on the co-occurrence vectors, where a cooccurrence means the entities appear in a list structure (such as "A, B, and C").

Our problem of growing the communities from seeds resembles minimally supervised learning and bootstrapping. *Supervised learning* uses large amounts of training data to construct a classifier. *Unsupervised learning* attempts the difficult task to construct a classifier without training data. *Minimally supervised learning* attempts to construct a classifier using a very small amount of training data. These techniques are useful for quickly constructing classifiers on lesser known domains, where a large amount of training data is unavailable.

### 2.4 Temporal Network Analysis

The communities formed by entities are not static in time. A related problem is to predict temporal changes from a given view. For instance, we would like to know the probability $p$ of a particular entity joining a particular group. Current research has shown it is possible to predict these changes in a community based on its current structure. Because very few new members will ever join any given group, there is large prior probability of *not* joining. Thus even though methods have some predictive power, accurately predicting individual membership remains elusive, but predicting overall size change is somewhat more attainable.

Backstrom et al. [2006] used a decision tree technique to predicting changes in given network properties. To train, they took snapshots of LiveJournal and DBLP coauthorship networks at different points in time. They found that the most important feature determining membership is not just who your neighbors are, but how well your neighbors are connected.

Sarkar and Moore [2005] tracked group dynamics by first reducing entities to a latent space model. This reduced dimension allow entities to be considered as spatially separated, so that Markov chain models could be used to predict movement.

### 2.5 The Lydia News Analysis System

The architecture of the *Lydia* system has been described elsewhere in detail [Lloyd et al. 2005], but is worth briefly reviewing here as the source of our network data. The major processing phases behind *Lydia* are:

—A collection of semi-customized WWW spider programs charged with retrieving articles from news sources on a daily basis. The system can also be extended to other sources such as journal databases, financial reports, and blogs [Lloyd et al. 2006].

—An NLP-based *named entity recognition* pipeline which identifies and classifies all references to proper nouns (i.e., people, places, organizations) appearing in the text.

—A variety of derivative statistical analysis based on frequencies of name entities and cooccurrence, such as coreference resolution [Lloyd et al. 2006], spatial analysis [Mehler et al. 2006], and sentiment analysis [Godbole et al. 2007].
—Applications of our database including entity search [Bautin and Skiena 2007] and question answering [Kil et al. 2005].

Artifacts introduced by the named entity recognition problem significantly increase the challenge of discovering communities, although similar challenges are likely emerge in any other algorithmically constructed network. It is simply impossible to sustain human curation on networks of almost 300,000 nodes, the size of the graph we analyze in this paper.

## 3. EXPANDING A COMMUNITY

The essential function of a community expansion method is to identify the most promising next member to add to the community. This is achieved by assigning a score to all entities in the network, and selecting the highest-scoring outside vertex to join the community. We now describe several different possible scoring criteria to rank the selection:

—*Neighbor Count.* The most obvious candidates for incorporation have many neighbors in the community. Basketball players tend to be associated with other basketball players, musicians with other musicians, etc.
—*Juxtaposition Count.* One drawback of using a simple neighbor count criteria is that each neighbor is given the same weight, regardless of the strength of the relation. The edge weights defining our network are co-occurrence frequencies of the given entity pair. Using such juxtaposition weights assigns more importance to neighbors that are more frequently associated in the text with in-community members.
—*Neighbor Ratio.* A failing of such counting scores is that the status of ubiquitous entities gets artificially elevated. A frequent entity like "George Bush" has over a thousand neighbors in our graph, and hence will have neighbors from many communities. Say six of these neighbors are chemists. The raw neighbor count score would identify George Bush as more likely to be a chemist than John Dalton, an entity that has only 8 neighbors (5 of which are chemists). But if we factor in vertex degree and use a ratio, Dalton becomes promoted to the most likely chemist.
—*Juxtaposition Ratio.* The bias to ubiquitous entities is also present in juxtaposition counts. Edges to "George Bush" tend to have high weight, simply because of the total frequency of the entity. Using a ratio helps control for high-frequency vertices.
—*Binomial Probability.* Using ratios has the problem of artificially elevating the importance of infrequent entities. An entity with 100 neighbors, 60 of which are chemists, would have a neighbor ratio of 0.6. But an entity with a single neighbor who happened to be a chemist would have a ratio of 1. We normalize for this by computing the probability $Pr[n, k]$ that an entity would

happen to have at least $k$ of its $n$ neighbors from the group by chance. So:

$$Pr[n,k] = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i},$$

where $p$ is the fraction of known-group members to network nodes. When $Pr[n,k]$ is extremely low for an observed $k$ in-group neighbors, than it can be reasoned that the entity must be a member of the community.

Experimental results on the performance of these various criteria on real networks will be presented in Section 5.2.2.

We note that these selection criteria can all be implemented efficiently. As each scoring function described above can be computed from the local neighborhood of a vertex, each can be computed for all vertices in time linear in the size of the graph. Changes in these scores can be more efficiently maintained incrementally, because when a vertex $v$ is added to the community, the only vertices whose scores change are neighbors of $v$. By keeping the vertices in a priority queue, we can efficiently identify the highest scoring member. Since a vertex can only be added once, each edge is only reconsidered during the entire series of insertions. Our algorithm only updates the scores of the neighbors of last vertex $v$, added to the community. Maintaining a priority queue of the set of $|V|$ vertices gives us a total a complexity of $O(|E|\log(|V|)$.

## 4. BOUNDING THE EXTENT OF A COMMUNITY

The techniques for growing a community presented in Section 3 do not provide a clear answer on when to stop adding members. Initially, most of the vertices added will indeed be members of the real community, but eventually we see a shift in composition after which most insertions will be erroneous. The community will have expanded outside its natural boundaries. We must stop its growth before it enters into this second phase to optimize the quality of the discovered community.

### 4.1 Stopping Rules

Identifying the proper cutoff for terminating incremental growth would be trivial if we knew the partition of our community into true positives and false positives, as suggested by Figure 2. However, all we are provided is a (small) seed subset of the community, without any other validation information.

If we did have a validation "gold-standard" subset of the community, we could monitor how frequently these members are added by the grower. In the first phase, when we identify community members with great precision, we expect to add a new validation member with frequency equal to the fraction of the community comprised by the validation set. For example, if our validation set represents 5% of the total community, we would expect to insert a validation member roughly once about every 20 insertions. Once we leave the natural boundaries of the neighborhood, we expect to rediscover validation members according to their frequency in the entire network, where they are much rarer.

Let $x_i$ denote the size of the $i$th insertion interval, namely as the difference between the discovery times of the $i$th and $(i-1)$st validation members. We find

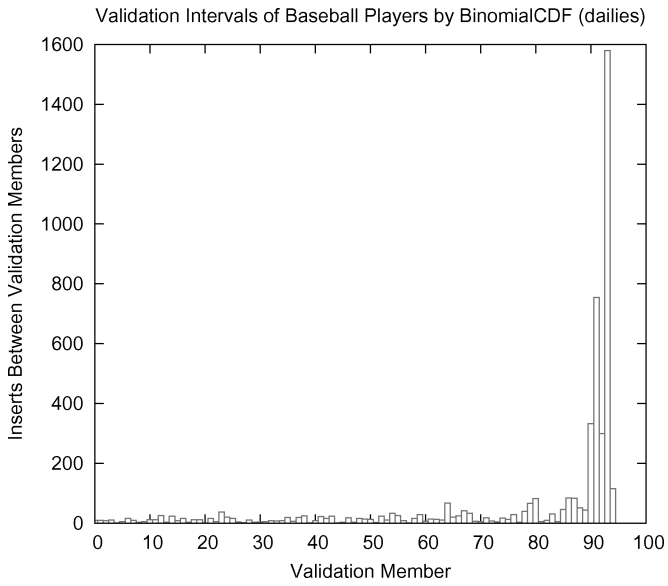


Fig. 3. Validation interval sizes on a representative graph/community. The height of the bar at position $i$ represents the number of members added between finding the $(i-1)$st and $i$th validation members. The phase transition around the insertion of the ninetieth validation member indicates we have reached the limits of the recoverable community.

the cutoff point that best splits the validation intervals into two groups to identify the correct stopping point. The expectation of initial-phase interval is the community size over the validation set size, while the second-phase frequency is graph size over validation set size. We expect the deviation of the intervals on either size of the cutoff to be small. Thus we select the stopping point $s$ that minimizes the absolute deviation; that is,

$$\text{stopping point} = argmin_k \left( \sum_{i=0}^{k} (|x_i - \mu(x_0, \ldots, x_k)|) + \sum_{i=k+1}^{n} (|x_i - \mu(x_{k+1}, \ldots, x_n)|) \right).$$

where the function $\mu$ is the arithmetic mean of its arguments.

For example, suppose validation members are identified on insertions

$$3, 7, 10, 11, 16, 18, 21, 120, 203, 290, 387, \text{ and } 506.$$

This yields an interval sequence of $X = \{3, 4, 3, 1, 5, 2, 3, 99, 83, 87, 97, 119\}$. The optimal stopping point in this example is after the seventh insertion.

Figure 3 shows a representative interval sequence for a real community on a real network. We see that the intervals do start out small but take a sudden and dramatic spike after the ninetieth validation member is found, a phase transition indicating we have left the boundaries of the community.

## 4.2 Boosting

Our iterative method is particularly sensitive to the initial partition of community members into seeds and validation sets. We seek to minimize the number

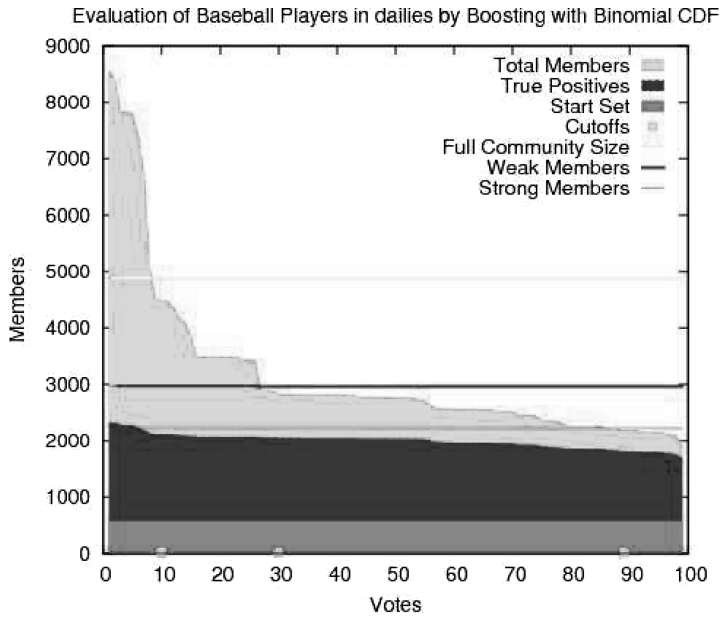


Fig. 4. Boosting Performance. Precision increases at the expense of recall as we increase the number of votes needed for an entity to be declared a member of the community. Distinct but interesting trade-offs are realized at approximately 10, 30, and 70 votes.

of seed members which must be specified, and so use a boosting technique to improve performance.

We run our growing algorithm multiple times, each time using a different partition of the input set into seed and validation members. Each vertex then accumulates a number of "votes" for how often it is identified as part of the community. Figure 4 demonstrates the value of boosting by reporting how many votes were received by vertices. The topmost shaded region represents false-positives; vertices that we incorrectly added to the community. The precision of the community members increases with the number of votes. The number of true positives proves relatively independent of the number of votes, because most are discovered in nearly every boosting run.

4.2.1 *Precision/Recall Trade-offs.* It now remains to determine which boosting cutoff to set. We use validation members to estimate precision and recall at each cutoff. The given members are divided into seed and validation sets for each boosting run. We track the number of votes these validation members get. Let $C_m(k)$ denote the set of elements appearing in at least $k$ of the $m$ boosting runs and $V$ the set of validation elements.

If we make the assumption that any vertex classified a community member in at least $b = m - \epsilon$ boosting runs is indeed a true member of the community, we can estimate the size of the still undiscovered community. The estimated fraction of the true community represented in the validation set is given by:

$$f_{val} = |C_m(b) \cap V|/|C_m(b)|.$$

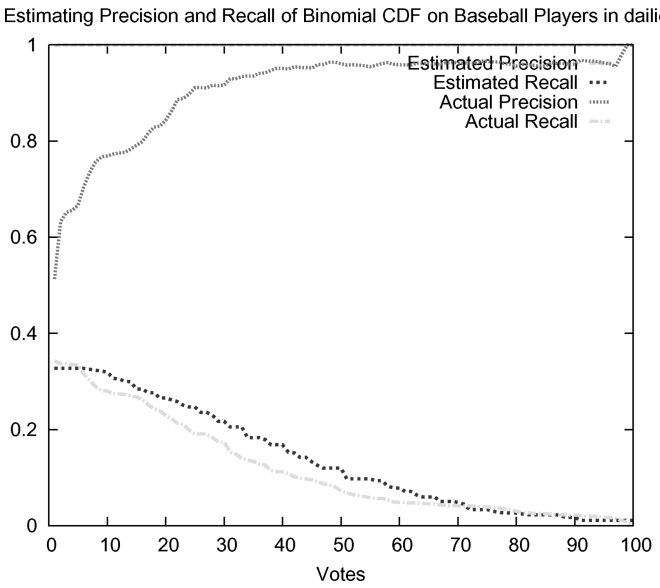


Fig. 5. Estimating Precision and Recall. The estimated precision and recall curves closely shadow the actual precision and recall.

Then at any vote cutoff, we can estimate precision and recall by looking at validation precision and recall. For a given number of votes $v$, the precision $pre_v$ is estimated as

$$pre_v \approx f_{val}|C_m(v) \cap V|/|C_m(v)|$$

and the recall estimated as

$$rec_v \approx |C_m(v) \cap V|/|V|.$$

Figure 5 shows examples of estimating the precision and recall, compared to the actual precision and recall. This approximation yields an approximation of f-score, which can be maximized to set a cutoff. Going even further, we can maximize the general F-measure for any $\beta$.

$$F_\beta = (1 + \beta^2) * (precision * recall)/(\beta^2 * precision + recall).$$

We now have a parameter ($\beta$) to tune to get appropriate precision/recall tradeoffs. Provided the precision and recall estimates are sufficiently accurate, we can maximize any general F-measure.

## 5. EXPERIMENTS IN PARAMETER OPTIMIZATION

Several decisions remain to complete the design of our community expansion algorithm. To maximize the performance of our scheme, we must optimize over the following space of parameters:

—*Neighbor Selection Method.* We compare five different scoring methods: neighbor count, juxtaposition count, neighbor ratio, juxtaposition ratio, and binomial CDF.

—*Validation Set Size.* We evaluate the trade-off on the fraction of seed members reserved for validation. Too low a fraction will not leave enough members to validate with. Too high a faction leaves us with few seed members to build the community from.

—*Number of Boosting Runs.* We anticipate that both precision and recall will increase with the number of boosting iterations. We verify this in our experiments. The running time increases linearly with the number of iterations, however. We seek the smallest number of iterations necessary to achieve quality communities.

—*Precision-Recall Trade-off.* We evaluate the effects of the f-statistic parameter $\beta$, to ensure that a useful precision/recall trade-off is achieved.

—*Given elements.* The number of seed elements presented is a function of the user's knowledge of their desired neighborhood. We assume that a certain critical mass is necessary to accurately expand neighborhoods. We evaluate our methods with different sizes of given elements to see how robust our methods are to extremely small seed sets.

Globally optimizing this five parameter space would be extremely computationally intensive. Instead, we optimize each parameter independently, using reasonable choices for the other parameters. Our initial setup uses the neighbor count grower, assigns half the seeds to the validation set, runs 100 boosting runs, uses a precision/recall trade-offs of $\beta = 1.0$, and evaluates seed sets of 20, 200, and 400 given members. For each data point, the grower was run on five different randomly assigned given sets, and the results macro-averaged.

### 5.1 Identifying Gold Standards

Properly evaluating our community expansion algorithms requires identifying gold standards of genuine communities existing within our large news-oriented network. This is a much more subtle, challenging task than may appear initially.

The most obvious approach is to consult reference lists or gazetteers, such as a table of all baseball players appearing in, say, Wikipedia. However, most curated lists will be incomplete; or at least lag in completeness relative to recent events reported in the news. Comprehensive curated lists are not readily obtainable for interesting natural communities (e.g., Democrats and Republicans), which is what motivates our expansion problem in the first place. The rosters which do exist often contain preconceptions or biases, such as occur when members are self-identified. Finally, the names used from an external list may not be consistent with the names used in the network, adding an additional layer of complexity. Still, we have opted for use of available reference lists in the absence of a better solution.[2]

[2]Another approach might be to discover the gold standard from the source text itself. Thelen and Riloff [2002] give a method for learning semantic lexicons from seed sets. Their natural language (NLP) based method uses pattern matching rules to identify new members. An article reporting "Democrat Bill Clinton announced yesterday…" identifies Bill Clinton as a member in the community "Democra." This shifts the problem of community discovery to a difficult and open NLP problem. Even if this technology proved effective, relatively few entities are identified explicitly as members of a community in news articles, particularly in the case of low-frequency entities.

Other difficulties relate to the accuracy of our underlying network data. Recall that our network was derived from an NLP-based analysis of a large news corpus. This data has several properties which complicate community identification:

—*Natural Language Processing Errors.* Our network was computed from imperfect named entity recognizers, classifiers, and coreference resolution methods. Thus we must anticipate both missing and spurious relations in the network. For example, *Lydia* sometimes segments named entities incorrectly, say tagging 'Outfielder Carlos Beltran' as a named entity. This will fail to match the gold standard lexicon entry "Carlos Beltran," and be incorrectly scored as a false positive if we classify the entity as a baseball player.

—*Entity Disambiguation.* A related problem arises when two distinct entities share the same name. Wikipedia has over thirty entries for the name "John Edwards." While current news is no doubt dominated by the philandering former North Carolina congressman, there also is an NBA player named "John Edwards," as well as a 1960's baseball player. In the statistics presented below, are penalized for not declaring "John Edwards" to be a baseball player, even though none of the media references to "John Edwards" presumably refer to him.

—*Entity Aliases.* There can be a mismatch between the name of an entity in the gold standard and that by which they are referred to in the news. For example, a reference list may identify an athlete by their legal name (e.g., baseball player "Larry Jones") as a baseball player, while media references invariably use his nickname "Chipper Jones."

—*Community Fringes.* The boundary of the baseball player community will certainly include classes of people related to baseball (e.g., managers, agents, and owners) who will not appear in a reference list. The concern here is that an accurately recognized "baseball" community will evaluate poorly against a "baseball player" gold standard.

### 5.2 Parameter Evaluation Results

We now present our experimental results on the impact of each parameter on global performance.

5.2.1 *Initial Seed Set Size.* Figure 6 shows the results for a single (no boosting) grower run on different sized, randomly selected seed sets. The true positives generally increase with the size of the seed set, but with considerable variance since each given set is randomly generated. Still the effect suggests that small seed set sizes suffice to reconstruct sizable communities.

5.2.2 *Comparing Growers.* Figure 7 illustrates the performance of the five neighbor selection criteria presented in Section 3, in experiments on seed sets with 20 and 400 entities respectively. The Binomial CDF-based score consistently achieves the highest f-score, on the strength of its producing the highest recall of any network selection criteria. The neighbors-count criteria lead to substantially higher higher precision when given a large seed set, but

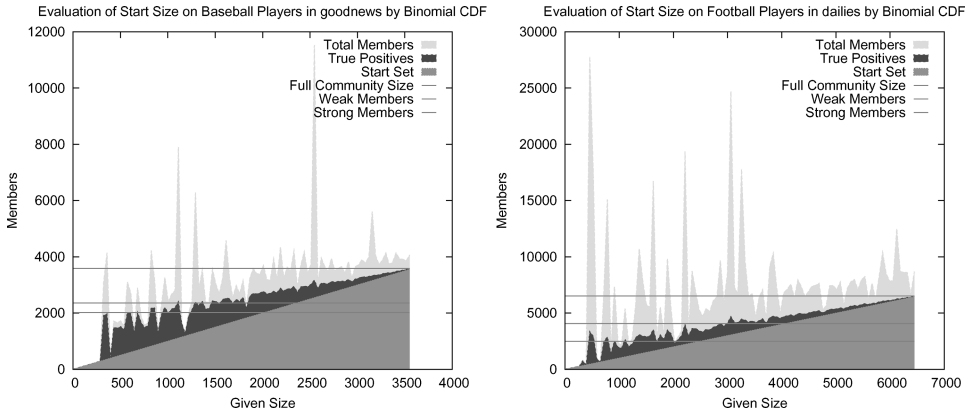


Fig. 6. Evaluation of the effect of seed set size on community expansion for baseball and football players. True positives increase slowly with larger seed set sizes.

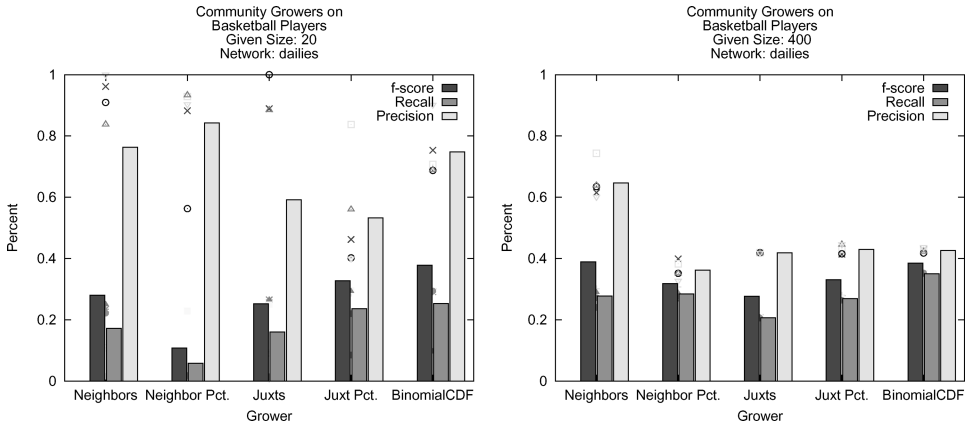


Fig. 7. Evaluating community growers on the Basketball player community, for input sets of 20 and 400 vertices.

demonstrated overall lower recall. Thus we choose Binomial CDF as our default neighbor selection criteria.

5.2.3 *Effect of Validation Set Size.* Figure 8 shows impact of varying the validation set percentage, the ratio of our given set of examples used for validation as opposed to seeds for community growth. This validation ratio was varied from 10% to 90% in increments of 10%, while requiring a minimum size of three for the validation set. Too low a percentage leaves too few members to validate with. Too high, and there are not enough seed members to grow from. As expected, there is high variance by partition for small seed sets (20 entities) but very little difference in the effect of validation size partition when given large seed sets (200 entities). A choice of 50% offers the best balance between seed and validation partition.

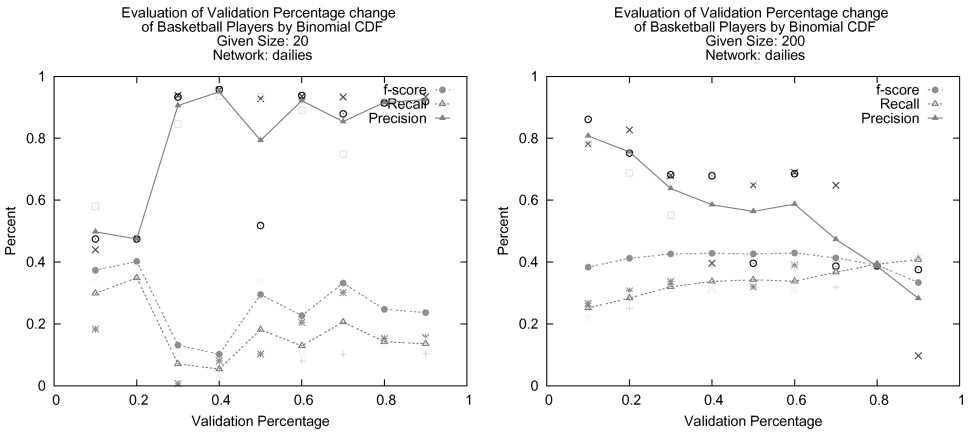


Fig. 8. Evaluation of Validation Percentage on Basketball Players, for starting sizes of 20 and 400 entities (individual data points and averages).

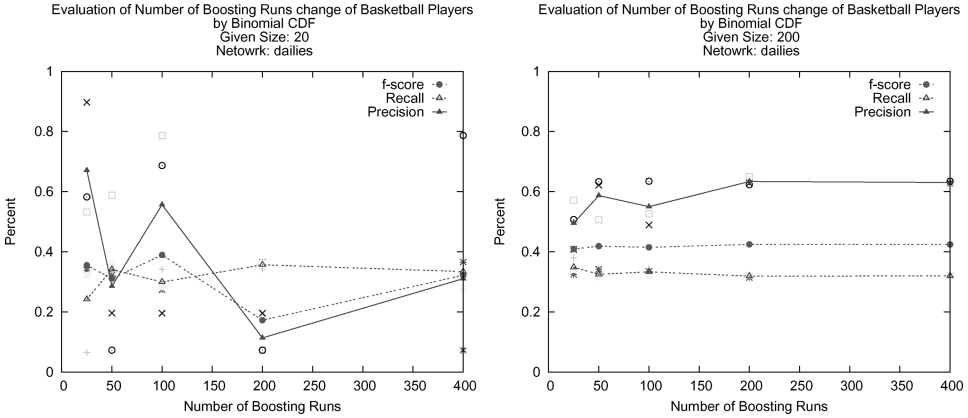


Fig. 9. Evaluating the impact of the number of boosting runs on expanding the Basketball Player community, for starting sets of 20 and 200 entities.

5.2.4 *Effects of Boosting.* Boosting provides us a tradeoff between precision and running time. We expect better results with more boosting runs, but the computation time grows linearly with the number of runs. The results shown in Figure 9 demonstrate relatively little sensitivity to the number of runs, except to reduce the variance for small initial seed sets (20 elements). We choose 100 runs as offering a reasonable performance trade-off.

5.2.5 *Precision-Recall Trade-off.* The parameter $F_\beta$ presented in Section 4.2.1 governs the precision/recall trade-off for our method. Our algorithm attempts to maximize $F_\beta$, so changing this parameter will change the behavior of the algorithm; should it try to be more precise in making membership assertions, or aim for higher coverage? Figure 10 shows results of turning this parameter for two different domains, movie stars and football players. This parameter $\beta$ governs how much more recall is weighted over precision. Thus

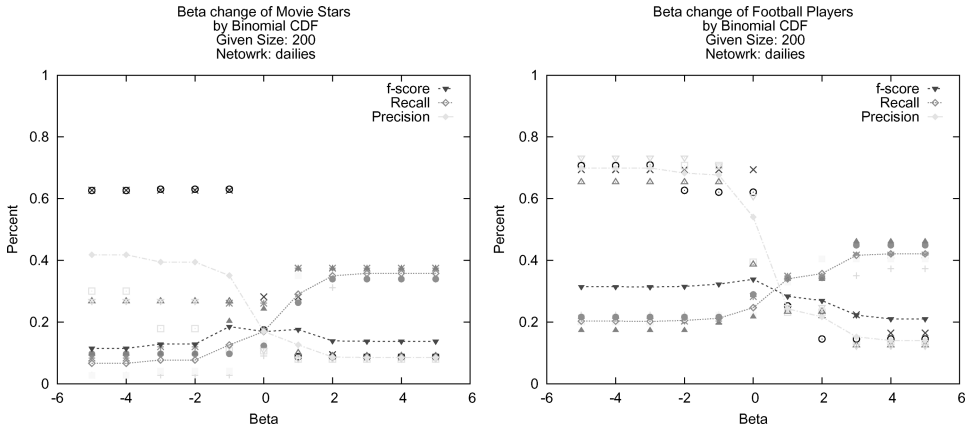


Fig. 10. Evaluation of $\beta'$ parameter on the football player and movie actors communities.

$\beta = 1$ gives an equal weighting, $\beta = 2$ means recall is weighted twice as much, and $\beta = 0.5$ means recall is weighted half as much. With the transformation, $\beta' = \beta - 1$ for $\beta > 1$, and $\beta' = (-1/\beta) + 1$ for $\beta < 1$, the operating points favoring precision have the same range as those favoring recall.

The results in Figure 10 are consistent with the notion that positive $\beta'$ weighs recall higher than precision, and negative $\beta'$ precision higher than recall. We also observe that there seem to be few optimal operating points as opposed to a continuum. The optimal points tend to be at the right, bottom ends of 'cliffs'. That is, looking at the graph we see some points of sharp decline in false positives. It is clearly better to use the points just after such a decline, as there is little or no difference in recall immediately before or after.

## 6. EXPERIMENTS IN COMMUNITY DISCOVERY

We now look more carefully at the compositions of the communities expanded from small sets of seeds. We consider the same four natural communities (baseball players, basketball players, football players, and movie actors) discussed in the previous section.

Table II presents the composition of communities reconstructed from single, randomly selected seed sets of 20, 200, and 400 initial members of each community. The precision and recall of each run are somewhat erratic, as should be expected from any process heavily dependent on the specific seed set selected. However, in each case we are able to reconstruct roughly half the community. We amplify the seed set by between 5 and 125 times on every run.

The graph theoretic properties of natural communities are important to put these recall rates into perspective. Table III presents the size (in number of members) of each community as well as measurements of connectedness, including in-community edges and average degrees. We classify the vertices as either being 'strong', 'weak', or 'isolated' based on in-community degree. *Strong* vertices have at least half their neighbors in the community, while *isolated* vertices have no neighbors in its community. *Weak* vertices are defined as neither strong nor isolated.

Table II.
Evaluations of expanded communities from seed sets of 20, 200, and 400 entities. The number of true and false positives in each community are given, with recall in parenthesis.

| seeds | | Baseball | Basketball | Football | Actors |
|---|---|---|---|---|---|
| 20 | True Positives | 1354 (.625) | 100 (.094) | 2491 (.532) | 480 (.223) |
| 20 | False Positives | 811 | 7 | 7228 | 698 |
| 200 | True Positives | 1407 (.415) | 534 (.496) | 1686 (.375) | 337 (.171) |
| 200 | False Positives | 575 | 1075 | 1536 | 652 |
| 400 | True Positives | 1496 (.469) | 392 (.477) | 1630 (.388) | 397 (.224) |
| 400 | False Positives | 737 | 829 | 1511 | 1548 |

Table III.
All four natural communities contain large fractions of isolated members (vertices with no in-community neighbors), and weakly-connected members which are difficult or impossible to recover through network analysis.

| | Network | Baseball | Basketball | Football | Actors |
|---|---|---|---|---|---|
| Vertices | 299,486 | 4,872 | 1,653 | 6,514 | 2,703 |
| Edges | 594,884 | 36,509 | 10,358 | 16,745 | 8,081 |
| Strong Vertices | — | 2,221 | 735 | 2,491 | 1,161 |
| Weak Vertices | — | 748 | 348 | 1,584 | 772 |
| Isolated Vertices | — | 1,903 | 570 | 2,439 | 770 |

The large number of weak and isolated vertices shown in Table III put our results into perspective. Over half of the entities in all four communities are weak or isolated. This means that most vertices have higher-degree outside the community than inside. Not surprisingly, the recall for our growers tops out at around the number of strong vertices in each community. Isolated vertices are impossible to discover using our methods. Thus any high-precision grower must miss all isolated vertices, which form over one-third of the community in most cases.

An interesting result is that sizes of our expanded communities tend to shrink slightly in response to larger seed sets, although precision tends to go up. We believe that this is due to the increased probability of overloaded entity names (e.g., "John Edwards" in the discussion below) entering as members of the seed sets. These entities should never have been regarded as part of the community in the first place, so they reduce the homogeneity of it. Thus growth is terminated more quickly than would have emerged in a more homogeneous community.

The false positive rates are such that true positives dominate the reconstructed communities for baseball and basketball, and form over a quarter of the basketball and actor communities. More to the point, we believe our evaluation methodology substantially over-estimates the number of false positives, as any entity not appearing in our specific, limited gold standard with an identical name is marked a false positive.

To better evaluate this, we carefully explore these false positive entities in the subsections below. They convincingly demonstrate that real precision of our communities is much higher than reflected by Table II.

Table IV.
Incorrectly classified basketball players. Names labeled with a '+' are associated with basketball, but not necessarily as players. Names labeled with a '*' are people that share a name with lesser-known basketball players. The frequency of these labels indicates that we generate more reasonable communities than simple performance scores may indicate.

| | False Positives | | False Negatives |
|---|---|---|---|
| | Laci Peterson | * | John Edwards |
| + | Roy Williams | * | Michael Jackson |
| + | Madison Square Garden | | Shaquille O'Neal |
| + | Mike Krzyzewski | * | Bob Riley |
| + | Mark Cuban | * | Michael Phelps |
| + | Van Gundy | * | Steve Smith |
| + | Greg Oden | * | Mel Gibson |
| + | Rick Pitino | | Billy Donovan |
| + | Gregg Popovich | | Pat Riley |
| + | David Stern | * | Jim Davis |
| + | Mike Montgomery | * | Greg Anderson |
| + | Jim Calhoun | * | Michael Young |
| + | Bernie Bickerstaff | * | Bernie Williams |
| + | Flip Saunders | * | Mike Davis |
| + | Jerry Buss | | Larry Johnson |
| + | Rick Barnes | * | Aaron Brooks |
| + | Van Horn | | J.J. Redick |
| + | Paul Hewitt | * | Mike Williams |
| + | John Calipari | * | John Chaney |
| + | Lawrence Frank | | Jayson Williams |

*Basketball.* Truth data for basketball players is taken from *http://www.basketballreference.com*. To get a sense of the accuracy of the community, we look at the most popular entities (measured by total news references) that are mis-classified. Table IV shows the top 20 false positives (left) and false negatives (right). Looking carefully at the false negatives,[3] we see that the grower is often the victim of an obscure basketball player having the name of someone more famous. Examining these lists leads us to believe that we generate much more reasonable communities than the performance scores may indicate.

[3]Basketball players whose names coincide with more prominent news figures include John Edwards, Bob Riley, Mike Davis, and John Chaney, most famously politicians; Michael Jackson, most famously a musician; Mel Gibson, most famously a film actor; Michael Phelps, most famously a swimmer; Jim Davis, most famously a cartoonist; Greg Anderson, most famously a personal trainer; Michael Young and Bernie Williams most famously baseball players; Aaron Brooks, most famously a football player; Mike Williams, two different football players—are all also the names of basketball players.

On the false positive side we see many basketball-related figures, such as NBA commissioner David Stern, Owners Mark Cuban and Jerry Buss, coaches Mike Krzyzewski, Van Gundy, Rick Pitino, Gregg Popovich, Mike Montgomery, Jim Calhoun, Bernie Bickerstaff, Flip Saunders, Rick Barnes, Paul Hewitt, John Calipari, and Lawrence Frank. An basketball arena, Madison Square Garden, appears as a result of poor entity categorization (most likely caused by "Madison" being considered a first name). Identifying Greg Oden as a false positive reflects an interesting omission from the gold standard. Oden is a member of the NBA, but missed his entire first season due to injuries and hence did not appear on the official list of players.

*Baseball.* Our truth data for baseball players is taken from the roster of all players from *http://www.baseball-databank.org*. We again ran our grower, and examined the most popular false positives and false negatives. The results are shown in Table V. We see much the same phenomenon that we saw for basketball players, namely that many false positives are people strongly associated with baseball.[4] Once again, inspection demonstrates that the communities identified are more reasonable than suggested by the gold standard.

*American Football.* The roster of American football players was taken from *http://www.pro-football-reference.com/*. Table VI shows misclassified football players. As usual, there is a disambiguation problem in evaluating false negatives. The false positives have accumulated many basketball players. While still in the 'athlete' category, basketball players should not otherwise related to football players. If we look closer at the actual vote count, we see that many of the basketball related entities received lower votes than football related entities.

*Movie Actors.* Truth data for movie stars was taken from the Internet Movie Database *http://www.imdb.com*. However, nearly all famous people regardless of profession appear listed as actors in IMBD, because they may have been subjects of documentaries or appear as themselves in other films. For example, Bill Clinton is listed as an actor in 15 films in IMDB.

To compile a roster of more conventional movie stars, we filtered IMDB's data to remove all actors whose movie list comprised over 25% documentaries, or who appeared in less than three other movies. Movie stars also appear to have a higher number of name clashes relative to the other communities we studied, a phenomena perhaps due to the widespread use of stage names. Since the performance of our community expansion method is sensitive to having genuine community members in the seed sets, we constructed seed set of 50 popular movie stars instead of using random selection as in the previous examples.

[4]Bud Selig is the commissioner of baseball. George Steinbrenner is the owner of the Yankees. Brian Cashman, Theo Epstein, and Jim Hendry are general managers. Tony La Russa (also appearing as "La Russa," an error in co-reference) is a manger. Scott Boras is a notorious agent.

There also appear to be several nonbaseball people linked to through recent steroid scandals. George Mitchell, a U.S. senator, and never previously involved with baseball, is now most in the news for his 'Mitchell Report, an investigation on steroids sanctioned by Major League Baseball. Similarly included in the community are Congressman Henry Waxman (part of the congressional hearings on steroids), defamed trainer Greg Anderson (who supplied many athletes with steroids), and several other athletes involved with performance enhancing drug scandals: cyclist Floyd Landis and sprinters Marion Jones and Tim Montgomery.

The false negative side again shows disambiguation problems. Our evaluator sees the name "Larry Brown" as belonging to a middle infielder that played in the late 1960's, not the current basketball coach. We see similar false negatives for Mike Tyson (heavyweight boxer), George Washington (U.S. President), Bill Richardson (governor of New Mexico), Bill Nelson (Senator from Florida), Paul Martin (Prime Minister of Canada), Michael Brown (former director of FEMA), Jim Davis (creator of Garfield), John Warner (Senator from Virginia), Tommy Thompson (Governor of Wisconsin), Paul O'Neil (former Secretary of the Treasury), Larry Johnson (basketball player) and John Fox (comedian).

"Winter Haven" is a classification error, being the name of the city where the Indians and Red Sox have spring training. "League Baseball" is also an NLP error, our pipeline mistakenly thinking the "Major" in 'Major League Baseball' is a military title.

Table V.
Incorrectly classified baseball players. Names with a + are most associated with baseball, but not as players. Names with a '#' are people associated with performance enhancing drug scandals, of which baseball played a large part. Names marked with a * share the same name as a lesser-known baseball player.

| | False Positives | | False Negatives |
|---|---|---|---|
| # | Lance Armstrong | * | Larry Brown |
| | Scott Peterson | * | Mike Tyson |
| + | Winter Haven | * | George Washington |
| + | Bud Selig | * | Bill Richardson |
| + | League Baseball | | David Wells |
| # | Floyd Landis | | Felipe Alou |
| | Eli Manning | | Miguel Tejada |
| + | George Steinbrenner | * | Bill Nelson |
| # | Marion Jones | * | Paul Martin |
| # | Greg Anderson | | Mike Brown |
| + | Brian Cashman | | Chris Carpenter |
| # | George Mitchell | * | Jim Davis |
| + | Tony La Russa | * | John Warner |
| # | Tim Montgomery | | Mark Mulder |
| + | La Russa | | Mike Lowell |
| | Bode Miller | * | Tommy Thompson |
| | Scott Boras | * | Mike Davis |
| # | Henry Waxman | * | Paul O'Neill |
| + | Theo Epstein | * | Larry Johnson |
| + | Jim Hendry | * | John Fox |

Table VI.
Incorrectly classified football players. Names labeled with a + are associated with football, but not necessarily players. Names labeled with a * are people that share a name with lesser-known football players.

| | False Positives | | False Negatives |
|---|---|---|---|
| | Kobe Bryant | * | Michael Jackson |
| | Scott Peterson | * | Tony Stewart |
| | Shaquille O'Neal | | Jimmie Johnson |
| | Michael Jordan | * | Randy Johnson |
| | Laci Peterson | * | Bob Riley |
| | Allen Iverson | | Reggie Bush |
| | Richard Nixon | * | Michael Moore |
| | LeBron James | * | Michael Brown |
| | Barry Bonds | * | Bill Nelson |
| + | Bill Belichick | | Matt Hasselbeck |
| + | Bill Parcells | * | George Allen |
| | Dwyane Wade | * | Tommy Thompson |
| | Dirk Nowitzki | * | Frank Robinson |
| | Phil Jackson | * | Michael Young |
| | Mike Tyson | * | Kevin Brown |
| | Arthur Andersen | * | Ted Williams |
| | George Washington | * | Gordon Brown |
| + | Nick Saban | * | Dan Brown |
| | Jason Kidd | * | Luis Castillo |
| | Steve Nash | * | Tim Johnson |

Table VII.
Incorrectly classified film actors, grown from manually set seeds. People marked with a '+' are movie related, if not primarily actors (Dylan and Charles being the subjects of recent films). The other false positives are entertainment related, but not movie actors. People marked with a '*' have a name clash with non-movie stars. People marked with a '⊕' have been in enough films for IMDB to call them an actor, but in everyday news are primarily associated with some other community (O'Neal with sports, Presley with music, and O'Donnell with daytime television).

| | False Positives | | False Negatives |
|---|---|---|---|
| | Michael Jackson | ⊕ | Shaquille O'Neal |
| | Lance Armstrong | * | Robert Blake |
| | Martha Stewart | * | David Wells |
| | " Friends " | ⊕ | Elvis Presley |
| | Britney Spears | * | John Howard |
| | Donald Trump | * | Richard Hamilton |
| + | Bob Dylan | * | Adam Scott |
| | Beverly Hills | | Bill Cosby |
| | David Beckham | * | John Lynch |
| | Warner Bros | ⊕ | Rosie O'Donnell |
| + | Paris Hilton | | Willie Nelson |
| | David Letterman | * | Chris Young |
| + | Steven Spielberg | * | Eddie Jones |
| | Paul McCartney | | Woody Allen |
| | Katie Couric | * | Vernon Wells |
| + | Ray Charles | | Tim McGraw |
| | Oprah Winfrey | * | Mike Smith |
| + | Martin Scorsese | | John Wayne |
| | Elton John | | Jane Fonda |
| | Simon Cowell | * | John Abraham |

The results of using these seeds are shown in Table VII. Nearly all of the false positives are entertainment-related people: movie directors, television people, or misclassified entertainment-related entities (Warner Bros., Beverly Hills).[5] On the false negative side are the disambiguation problems seen in other communities, plus people from other communities appearing in film. NBA player Shaquille O'Neal has appeared in several films. Similarly, Elvis Presley is more associated with music than acting.

## 7. CONCLUSIONS

We have proposed a new method for expanding seed sets into more encompassing communities, and validated its performance on four real communities in a large news network.

[5]Robert Blake is most famously a hockey player; David Wells a baseball pitcher; John Howard the Prime Minister of Australia; Richard Hamilton a basketball player; Adam Scott a professional golfer; John Lynch a football player; Chris Jones a baseball player; Eddie Jones a basketball player; Vernon Wells a baseball player; Mike Smith a hockey player; John Abraham a football player. David Beckham can be explained as the inspiration for the movie "Bend it like Beckham".

Several aspects of the domain make the problem challenging, including (1) weakly-connected or isolated community members, (2) the fluid boundaries which define natural communities, and (3) data irregularities inherent in the automatic construction of large networks. Proper evaluation of the accuracy of community expansion is complicated by (1) the inherent deficiencies of natural reference standards, (2) difficulties in matching named entities between the network and reference standards, and (3) the overloading of names which represent several people in several different communities.

Putting all this perspective, we consider our results quite satisfying, and are now working to use our expanded communities in all applications of *Lydia*'s news analysis. In particular, we will be using our expanded communities as part of a project to analyze temporal and relationship dynamics among hundreds of natural communities [Ward et al. 2009]. How well our method performs compared to the other community expansion methods of Section 2.3: [Anderson and Lang 2006; Flake et al. 2000; Ghahramani and Heller 2005] is an interesting and important topic for further research.

The most interesting open problem in this line of research involves doing a theoretical analysis of random graph models to determine the properties necessary for accurate community expansion. Through such an analysis, we should be able to determine the seed size necessary for accurate community reconstruction as a functions of measures of the strength of the community (e.g., ratio of in-community to out-community degree).

## ACKNOWLEDGMENTS

We thank Jure Lesovec for sending us the NCP plot for our news network, and Mikhail Bautin and Shashank Naik for assistance in generating the entity network studied in this paper.

## REFERENCES

Anderson, R. and Lang, K. 2006. Communities from seed sets. In *Proceedings of the 15th International Conference on the World Wide Web (WWW '06)*. 223–232.

Backstrom, L., Huttenlocher, D., Kleinberg, J., and Lan, X. 2006. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*. ACM Press, New York, 44–54.

Barabasi, A.-L. 2003. *Linked*. Penguin Books Ltd.

Bautin, M. and Skiena, S. 2007. Concordance-based entity-oriented search. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI '07)*. IEEE, Los Alamitos, CA, 586–592.

Cami, A., Balakrishnan, H., Deo, N., and Dutton, R. 2006. On the complexity of finding optimal global alliances. *J. Comb. Math. Comb. Comput. 58*, 23–31.

Cirasella, J. 2007. Google sets, google suggest, and google search history: Three more tools for the reference librarian's bag of tricks. *Refer. Libr. 48.1*.

Clauset, A., Moore, C., and Newman, M. E. 2008. Hierarchical structure and the prediction of missing links in networks. *Nature 453,* 7191, 98–101.

Favaron, O., Fricke, G., Goddard, W., Hedetniemi, S. M., Hedetniemi, S. T., Kristiansen, P., Laskar, R. C., and Skaggs, D. 2002. Offensive alliance graphs. *Discussiones Mathematicae—Graph Theory*.

Fernau, H. and Raible, D. 2007. Alliances in graphs: a complexity-theoretic study. In *Proceedings of the Software Seminar*, vol. 2, J. van Leeuwen, G. F. Italiano, W. van der Hoek, C. Meinel,

H. Sack, F. Plasil, and M. Bielikov, Eds. Institute of Computer Science AS CR, Prague, 61–70.

Flake, G. W., Lawrence, S., and Giles, C. L. 2000. Efficient identification of web communities. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '00)*. ACM, New York, 150–160.

Flake, G. W., Tarjan, R. E., and Tsioutsiouliklis, K. 2004. Graph clustering and minimum cut trees. *J. Internet Math. 1*, 385–408.

Ghahramani, Z. and Heller, K. A. 2005. Bayesian sets. In *Proceedings of NIPS*.

Gibson, D., Kleinberg, J., and Raghavan, P. 1998. Inferring web communities from link topology. In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia: Links, Objects, Time and Space—Structure in Hypermedia Systems (HYPERTEXT'98)*. ACM, New York, 225–234.

Girvan, M. and Newman, M. E. J. 2002. Community structure in social and biological networks. *Proc. Nati. Acad. Sci. 99*, 7821–7826.

Godbole, N., Srinivasaiah, M., and Skiena, S. 2007. Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM'07)*.

Hopcroft, J., Khan, O., Kulis, B., and Selman, B. 2003. Natural communities in large linked networks. In *proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*. ACM Press, New York, 541–546.

Hopcroft, J., Khan, O., Kulis, B., and Selman, B. 2004. Tracking evolving communities in large linked networks. *Proc. Natl. Acad. Sci. 101 Suppl 1*, 5249–5253.

Jamieson, L. H, Hedetniemi, S. T., and McRa, A. A. 2002. The algorithmic complexity of alliances in graphs. *J. Combin. Math. Combin. Comput.*

Kernighan, B. W. and Lin, S. 1970. An efficient heuristic procedure for partitioning graphs. *The Bell Syst. Tech. J. 49,* 1, 291–307.

Kil, J., Lloyd, L., and Skiena, S. 2005. Question answering with Lydia. In *Proceedings of 14th Text Retrieval Conference (TREC'05)*.

Kleinberg, J. 1999. Authoritative sources in a hyperlinked environment. *J. ACM 46*, 5, 604–632.

Kossinets, G. and Watts, D. J. 2005. Empirical analysis of an evolving social network. *Science 311*, 88–90.

Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. 2008. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th International Conference on World Wide Web (WWW'08)*. ACM, New York, 695–704.

Lloyd, L., Kaulgud, P., and Skiena, S. 2006. Newspapers vs. blogs: Who gets the scoop? In *Proceedings of the Conference on Computational Approaches to Analyzing Weblogs (AAAI-CAAW'06)*. AAAI Press, 117–124.

Lloyd, L., Kechagias, D., and Skiena, S. 2005. Lydia: A system for large-scale news analysis. In *Proceedings of the Conference on String Processing and Information Retrieval (SPIRE'05)*. Lecture Notes in Computer Science, Vol. 3772. 161–166.

Lloyd, L., Mehler, A., and Skiena, S. 2006. Identifying co-referential names across large corpora. In *Proceedings of the Conference on Combinatorial Pattern Matching (CPM'06)*.

Mehler, A., Bao, Y., Li, X., Wang, Y., and Skiena, S. 2006. Spatial analysis of news sources. *IEEE Trans. Visual. Comput. Graph. 12*, 765–772.

Newman, M. E. J. 2004. Detecting community structure in networks. *Europ. Phys. J. B 38*, 321–330.

Newman, M. E. J. and Girvan, M. 2004. Finding and evaluating community structure in networks. *Phys. Rev. E 69*, 026113.

Pothen, A., Simon, H. D., and Liou, K.-P. 1990. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl. 11,* 3, 430–452.

Sarkar, P. and Moore, A. 2005. Dynamic social network analysis using latent space models. *SIGKDD Explorations.* (Special Edition on Link Mining).

Sarmento, L., Jijkuon, V., de Rijke, M., and Oliveira, E. 2007. "More like these": growing entity classes from seeds. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM'07)*. ACM, New York, 959–962.

Scott, J. 2000. *Social Network Analysis: A Handbook*. Sage Publications.

Shafique, K. H. 2001. Partitioning a graph in alliances and its application to data clustering. Ph.D. thesis, School of Computer Science, University of Central Florida Orlando.

THELEN, M. AND RILOFF, E. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

TYLER, J. R., WILKINSON, D. M., AND HUBERMAN, B. A. 2003. Email as spectroscopy: automated discovery of community structure within organizations. *Commun. Technol.*, 81–96.

WARD, C., BAUTIN, M., AND SKIENA, S. 2009. Identifying differences in news coverage between cultural/ethnic groups.

WU, F. AND HUBERMAN, B. A. 2004. Finding communities in linear time: a physics approach. *Eur. Phys. J. B 38*, 331–338.

Received September 2008; accepted December 2008