COMPRESSIVE GRAPH CLUSTERING FROM RANDOM SKETCHES

Yuejie Chi

Department of Electrical and Computer Engineering
Department of Biomedical Informatics
The Ohio State University, USA
Email: chi.97@osu.edu

ABSTRACT

Graph clustering, where the goal is to cluster the nodes in a graph into disjoint clusters, arises from applications such as community detection, network monitoring, and bioinformatics. This paper describes an approach for graph clustering based on a small number of linear measurements, i.e. sketches, of the adjacency matrix, where each sketch corresponds to the number of edges in a randomly selected subgraph. Under the stochastic block model, we propose a computationally tractable algorithm based on semidefinite programming to recover the underlying clustering structure, by motivating the low-dimensional parsimonious structure of the clustering matrix. Numerical examples are presented to validate the excellent performance of the proposed algorithm, which allows exact recovery of the clustering matrix under favorable trade-offs between the number of sketches and the edge density gap under the stochastic block model.

Index Terms— graph clustering, sketching, convex optimization, stochastic block model

1. INTRODUCTION

An increasing number of modern datasets can be represented by an undirected graph which encodes user interactions and node connectivities within complex networks such as social networks, biological networks, and computer networks. Identifying the communities, or dense subgraphs in the graph is important to network structure inference, anomaly detection, and data mining [1]. By exploring different connectivity patterns between nodes based on whether they belong to the same cluster, graph clustering aims to recover the clusters within a graph from observations of its adjacency matrix. Many algorithms have been proposed such as spectral clustering [2], correlation clustering [3], and convex optimization [4]-[8], where exact recovery guarantees have been established recently, most notably under the stochastic block model [9].

In all the existing approaches, the graph is either fully observed which is expensive when the graph size is large, or randomly subsampled [4], or adaptive subsampled [10], which is not suitable for updating in a dynamic graph stream model [11]. Sketching a graph via linear measurements has been considered recently as a powerful tool to reduce the dimensionality of the graph [11, 12, 13], while the acquired sketches is sufficient to faithfully recover the graph or estimate its properties such as cut values, by exploiting low-dimensional structures of the graph, e.g., sparsity of the adjacency matrix [13].

In this paper, we propose a new scheme to sketch the graph, and analyze an algorithm to *exactly* recover the underlying clustering structure that generates the graph, without first reconstructing the graph, based on a number of sketches that is much smaller than the ambient dimension of the graph. This is motivated by the observation that, under popular generative models such as the stochastic block model, the graph can be regarded as a *noisy* realization of the clustering matrix that encodes the true user membership. Therefore, it is of interest to recover the clustering matrix, rather than the graph itself, from a reduced number of measurements of the graph.

Our proposed linear measurements of the graph adjacency matrix, referred to as *sketches*, are inner products between the adjacency matrix and a rank-one matrix, given as $\boldsymbol{x}_i^T \boldsymbol{A} \boldsymbol{x}_i = \langle \boldsymbol{A}, \boldsymbol{x}_i \boldsymbol{x}_i^T \rangle$, where $\boldsymbol{A}$ is the adjacency matrix and $\boldsymbol{x}_i$'s are randomly selected Bernoulli vectors. These sketches are equivalent to counting the number of edges in a randomly selected subgraph determined by the support of $\boldsymbol{x}_i$, which can be computed in a parallel and distributed manner, and updated in an online fashion if the graph dynamically inserts or delete an edge. Our graph sketching scheme is motivated by the rank-one observation model studied in [14, 15, 16], which aims to recover a low-rank symmetric matrix from quadratic measurements. Compared with unstructured linear measurements of the adjacency matrix, the proposed sketches have a lower computational and memory cost.

The proposed graph sketching scheme may naturally fit in several applications. For example, consider the friendship graph in social networks. It may be prone to privacy leakage if the server directly query a specific friendship between one user and the other. Rather, the server may query the number of users that one user is friend of within a randomized group, and none of the specific friendships is revealed through these aggregated answers. As another example, consider the IP flows in traffic monitoring [17], rather than directly storing all the pairwise links between different IPs, it is possible to only track the summary links between a group of IPs, which can be implemented in a decentralized fashion.

Our second contribution is an efficient algorithm to recover the clustering matrix from the sketches using semidefinite programming under the balanced planted partition model, a special case of the stochastic block model [9]. Inspired by [6], our algorithm promotes the sparsity and low-rankness of the clustering matrix via convex relaxation, where a surrogate matrix is properly designed to assume the role of the adjacency matrix. The algorithm then selects the matrix with the desired low-dimensional structure that maximizes its correlation with the surrogate matrix. Numerical examples are presented for the proposed algorithm to demonstrate its desirable performance. In particular, it is observed that exact recovery is achieved for a wide range of parameters, even when the graph is relatively sparse; dimensionality reduction in graph acquisition is achieved as well, where the number of sketches can be made much smaller than the ambient

This work is supported in part by NSF under the grant CCF-1422966.

dimension of the adjacency matrix.

The rest of this paper is organized as follows. Section 2 reviews several background ingredients. Section 3 presents the proposed sketching scheme and clustering algorithm. Section 4 validates the proposed approach via numerical simulations, and we conclude and outline future work in Section 5.

2. BACKGROUNDS

2.1. Stochastic Block Model

The stochastic block model (SBM) [9], or the planted partition model, is a popular generative model for studying community structures in complex networks. Consider a graph $\mathcal{G} = (V, E)$ where V is composed of a set of N nodes and E is composed of a set of random edges. Assume that each node $1 \leq i \leq N$ belongs to a non-overlapping cluster $\pi(i) \in \{1, \ldots, r\}$, where r is the total number of clusters. Define $\boldsymbol{Y} \in \{0,1\}^{N\times N}$ as the clustering matrix, where

$$Y_{i,j} = \begin{cases} 1, & \text{if } \pi(i) = \pi(j) \\ 0, & \text{otherwise} \end{cases} . \tag{1}$$

The edge set E can be represented by the adjacency matrix $\boldsymbol{A} \in \{0,1\}^{N\times N}$, where $A_{i,j}$ can be modeled as a Bernoulli random variable with different parameters depending on whether $\pi(i) = \pi(j)$:

$$A_{ij} = \begin{cases} \text{Ber}(p), & \text{if } \pi(i) = \pi(j) \\ \text{Ber}(q), & \text{otherwise} \end{cases} . \tag{2}$$

Without loss of generality we assume $p > q$, and denote the edge density gap as

$$\delta = p - q, \tag{3}$$

which serves as a key quantity in characterizing clustering performance for various graph clustering algorithms [2, 4, 5]. Fig. 1 (a) gives an example realization of the adjacency matrix $\boldsymbol{A}$ generated by the SBM with two clusters of size 50 when $p = 0.5$ and $q = 0.1$, and Fig. 1 (b) shows the sorted adjacency matrix according to the cluster membership.

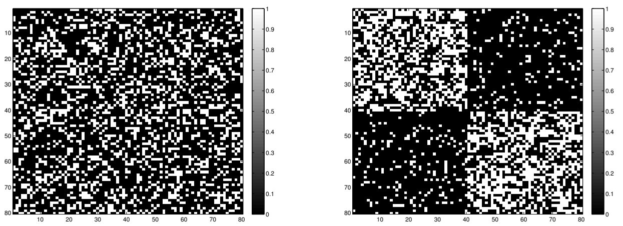
In this paper, for simplicity we focus on the balanced planted partition model, which assume the size of all clusters are the same as K, and hence $N = rK$. Our algorithm can be implemented to more general models.

2.2. Graph Clustering

The classical problem of graph clustering is that given the adjacency matrix $\boldsymbol{A}$, recovering the true membership matrix $\boldsymbol{Y}$. Of the most relevance to this paper are the semidefinite programming approaches proposed and studied in recent literature [6, 7, 8]. These approaches can be motivated as semidefinite relaxations of the maximum likelihood estimator under the SBM, and strong guarantees exist for their performance that are provably near optimal. In general, one wishes to solve the following problem:

$$\begin{aligned} \hat{\boldsymbol{Y}} &= \text{argmax}_{\boldsymbol{Z}} \ \langle \boldsymbol{A}, \boldsymbol{Z} \rangle \\ &\text{s.t.} \quad \boldsymbol{Z} \in \{0,1\}^{N\times N} \text{ is a clustering matrix,} \end{aligned} \tag{4}$$

which returns a clustering matrix $\boldsymbol{Z}$ that maximizes its correlation with the adjacency matrix $\boldsymbol{A}$. However, the constraint of being a clustering matrix is combinatorial and NP-hard, and therefore we seek convex relaxations of this constraint. Note that the true clustering matrix $\boldsymbol{Y}$ is simultaneously low-rank with $\text{rank}(\boldsymbol{Y}) = r$ and sparse with $\|\boldsymbol{Y}\|_1 = rK^2$ with binary entries, the non-convex constraint in (4) is replaced by its convex relaxation [6]



(a) (b)

Fig. 1. The stochastic block model with $p = 0.5$ and $q = 0.1$: (a) the adjacency matrix $\boldsymbol{A}$; and (b) its permutation according to the cluster membership.

$$\mathcal{Q} = \{\boldsymbol{Z} : 0 \leq Z_{i,j} \leq 1, \|\boldsymbol{Z}\|_1 = rK^2, \|\boldsymbol{Z}\|_* \leq rK\}, \tag{5}$$

where the rank constraint is replaced by the nuclear norm constraint $\|\boldsymbol{Z}\|_* \leq rK$, and the binary constraint is replaced by $0 \leq Z_{i,j} \leq 1$ for all the entries. To sum up, the algorithm proposed in [6] can then be represented as

$$\hat{\boldsymbol{Y}} = \text{argmax}_{\boldsymbol{Z}} \ \langle \boldsymbol{A}, \boldsymbol{Z} \rangle \quad \text{s.t.} \quad \boldsymbol{Z} \in \mathcal{Q}, \tag{6}$$

where the number of clusters r is assumed known.

2.3. Matrix Sketching

Sketching via linear measurements is an useful algorithmic tool in computer science [11] and compressed sensing [18, 19] to reduce the dimensionality of the data without losing its information content. In particular, when $\boldsymbol{\Sigma}$ is a symmetric matrix with low-dimensional structures such as low-rankness and sparsity, a quadratic sensing scheme can be exploited to recover $\boldsymbol{\Sigma}$ from a small number of random measurements $y_i = \boldsymbol{x}_i^T \boldsymbol{\Sigma} \boldsymbol{x}_i$, $i = 1, \ldots, m$ by solving the nuclear-norm or ℓ_1-norm regularized convex relaxation algorithms [14, 15, 16]. This quadratic sensing scheme is leveraged in this work to obtain summary information of randomly selected subgraphs by choosing $\boldsymbol{\Sigma} = \boldsymbol{A}$ as the adjacency matrix. If $p = 1$ and $q = 0$, then $\boldsymbol{A} = \boldsymbol{Y}$ is an exactly rank-r matrix, which can be recovered via nuclear norm minimization from an order of Nr measurements. However, this algorithm doesn't provide exact recovery when $p < 1$ or $q > 0$, since it is formulated to recover the matrix under sketching $\boldsymbol{A}$ rather than the clustering matrix $\boldsymbol{Y}$ that induces it.

3. GRAPH CLUSTERING FROM RANDOM SKETCHES

In this section, we first describe a graph sketching scheme that yields compressive measurements of the graph adjacency matrix, and then propose a semidefinite program to recover the clustering matrix from the sketches. The proposed algorithm is summarized in Algorithm 1.

3.1. Sketching node connectivities

Consider a graph $\mathcal{G}$ with the adjacency matrix $\boldsymbol{A}$. Our sketching scheme is non-adaptive and can be implemented in a parallel. Define the ith sketching vector $\boldsymbol{x}_i \in \{0,1\}^{N\times 1}$ where each entry $x_{i,j}$ is a

Algorithm 1 Graph Clustering From Random Sketches

```
Input: number of clusters $r$, $m$ sketching vectors $\{\boldsymbol{x}_i\}_{i=1}^m$, and sketches $\{y_i\}_{i=1}^m$ from (7);
Compute the surrogate matrix in (9);
Compute the solution $\hat{\boldsymbol{Y}}$ of the algorithm (10);
if $\hat{\boldsymbol{Y}}$ is a clustering matrix then
    output $\hat{\boldsymbol{Y}}$;
else
    declare failure.
end if
```

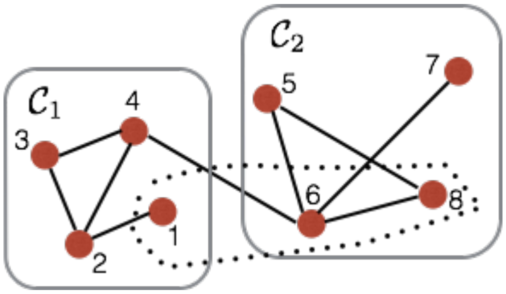


Fig. 2. Example of the measurement scheme.

Bernoulli random variable with parameter $s \in (0,1)$, then the ith sketch can be given as

$$y_i = \boldsymbol{x}_i^T \boldsymbol{A} \boldsymbol{x}_i, \quad i = 1, \ldots, m. \tag{7}$$

Denote the support of $\boldsymbol{x}_i$ as $\mathcal{I}_i$, then the ith sketch is *twice* the number of edges within the subgraph formed by the nodes in $\mathcal{I}_i$. For example, consider the graph in Fig. 2, and let

$$\boldsymbol{x}_1 = [1,0,0,0,0,1,0,1]^T$$

whose support is $\mathcal{I}_1 = \{1,6,8\}$ corresponding to the nodes in the dashed subgraph. Then it is straightforward to verify that $\boldsymbol{x}_1^T \boldsymbol{A} \boldsymbol{x}_1 = 2$ is *twice* the number of edges among $\mathcal{I}_1$. Therefore, the sketches can be computed without directly observing the subgraph, by querying each node within the subgraph "for the nodes indexed by $\mathcal{I}_i$, how many of them are you connected with?"; and then summing up their answers. This indicates that the sketch in (7) can be computed in a fully distributive manner without observing the entries in $\boldsymbol{A}$ directly.

Succinctly, we can represent the sketches (7) as

$$\boldsymbol{y} = \mathcal{X}(\boldsymbol{A}), \tag{8}$$

where $\boldsymbol{y} = \{y_i\}_{i=1}^m$ and $\mathcal{X}$ represents the linear map $\boldsymbol{A} \mapsto \boldsymbol{y}$. Our goal is to recover the clustering matrix $\boldsymbol{Y}$ given the sketches $\boldsymbol{y}$. If $m \geq N^2/2$, then (8) is overdetermined and $\boldsymbol{A}$ can be exactly recovered from (8) and then use existing graph clustering algorithm (6) to retrieve $\boldsymbol{Y}$. Therefore, we focus on the case when $m \ll N^2/2$ and (8) is under-determined.

3.2. Clustering via semidefinite programming

Motivated by the recent convex relaxations for graph clustering [6, 7, 8], we propose the following two-step algorithm for graph clustering. First, since $\boldsymbol{A}$ is not available, we formulate the surrogate matrix $\boldsymbol{S}$ as the least-norm solution of (8):

$$\boldsymbol{S} = \mathcal{X}^{\dagger}(\boldsymbol{y}) = (\mathcal{X}^*\mathcal{X})^{\dagger}\mathcal{X}^*(\boldsymbol{y}), \tag{9}$$

where $\mathcal{X}^*(\boldsymbol{y}) = \sum_{i=1}^m y_i \boldsymbol{x}_i \boldsymbol{x}_i^T$, and † denotes pseudo-inverse. Second, we replace $\boldsymbol{A}$ by the surrogate matrix in (6) and solve the following semidefinite program:

$$\hat{\boldsymbol{Y}} = \operatorname{argmax}_{\boldsymbol{Z}} \langle \boldsymbol{S}, \boldsymbol{Z} \rangle \quad \text{s.t.} \quad \boldsymbol{Z} \in \mathcal{Q}. \tag{10}$$

Interestingly, as the matrix $\boldsymbol{S}$ resembles the structure of $\boldsymbol{A}$, the above algorithm (10) can recover the clustering matrix without actually reconstructing $\boldsymbol{A}$. If the return $\hat{\boldsymbol{Y}}$ from (10) is indeed a clustering matrix, we claim it as the clustering output; otherwise the algorithm returns a failure. It is possible to employ alternative constructions of the surrogate matrix, for example, using the ridge estimation of (8) for some regularization parameter. We leave this to future work.

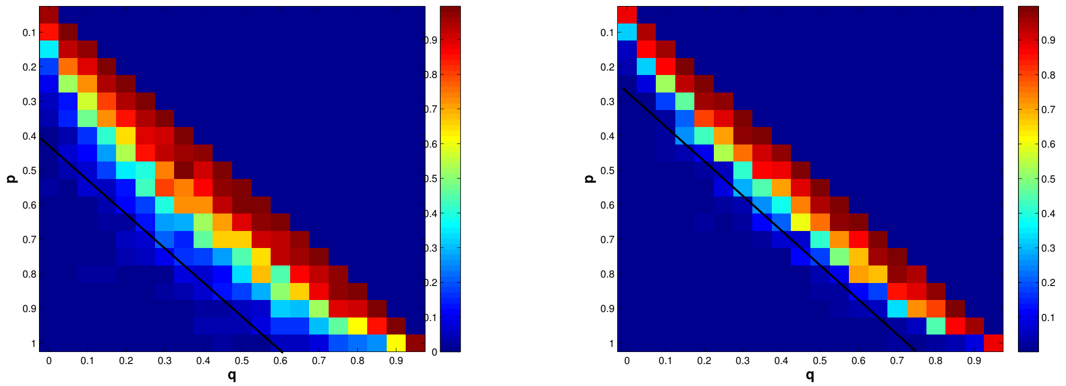
4. NUMERICAL EXAMPLES

Let $r = 2$, $K = 40$ and $N = rK = 80$. The affinity matrix $\boldsymbol{A}$ is randomly generated following the SBM with $p = 0.8$ and $q = 0.1$. We denote the solution of (10) as $\hat{\boldsymbol{Y}}$. Due to numerical inaccuracies, the matrix $\hat{\boldsymbol{Y}}$ contains continuous values, therefore we post-process it by hard-thresholding the entries in $\hat{\boldsymbol{Y}}$ against its mean value into a binary matrix. The normalized mean squared error (NMSE) is computed as $\|\hat{\boldsymbol{Y}} - \boldsymbol{Y}\|_{\mathrm{F}}^2 / \|\boldsymbol{Y}\|_{\mathrm{F}}^2$, which corresponds to the percentage of misidentified pairs. Fig. 4 shows the NMSE with respect to the number of measurements for different values of $s = 0.2, 0.4$, and 0.6. It can be seen that the reconstruction exhibits a phase transition behavior, where exact recovery of the clustering matrix is possible as soon as m exceeds certain threshold, which is much smaller than the ambient dimension of $\boldsymbol{A}$. It is also worth noticing that the performance variation with respect to s is small.

We further examine the performance of the Algorithm 1 with respect to the edge densities specified in the SBM. Let $r = 2$ and $K = 50$, and fix $s = 0.5$. For $m = 2000$ and $m = 3000$, we first generate m sketching vectors. For each (p,q) $(q < p)$, we generate the matrix $\boldsymbol{A}$ following the SBM and run Algorithm 1. Fig. 3 shows the recovery NMSE for different values of p and q when $m = 2000$ and $m = 3000$, where $p \in [0.05, 1]$ and $q \in [0, 0.95]$ with a step size of 0.05. The algorithm achieves good performance as long as the edge density gap δ is not too small, where we plotted the line $\delta = 0.4$ when $m = 2000$ and $\delta = 0.25$ when $m = 3000$ for comparison. Encouragingly, exact recovery is achieved for a wide range of (p,q) pairs, even sparse graphs when $p < 0.5$; and a higher edge density gap is allowed as m increases.

5. CONCLUSION

This paper presents a novel framework for graph clustering from a small number of linear measurements that can be regarded as pooling of a random subgraph, which is particularly suitable to scenarios when direct observations of node connectivities are impossible or expensive. Dimensionality reduction is simultaneously achieved to reduce the number of required measurements to be much smaller than the ambient dimension of the adjacency matrix. This work leaves many open questions that need to be addressed, including theoretical guarantees of the proposed algorithm, and efficient implementations to handle large-scale networks.



(a) $m = 2000$

(b) $m = 3000$

Fig. 3. The recovery NMSE with respect to the $p \times q$ plane. Only the region $q < p$, corresponding to the southwest triangle, is examined. The lines with the edge density gap $\delta = 0.4$ when $m = 2000$ and $\delta = 0.25$ when $m = 3000$ are plotted for comparison.

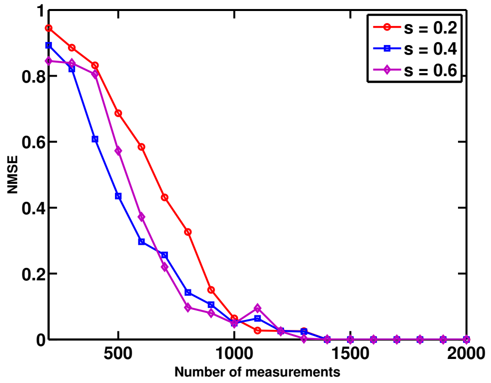


Fig. 4. The recovery NMSE with respect to the number of measurements when $r = 2$ and $L = 40$, $p = 0.8$ and $q = 0.1$.

6. REFERENCES

[1] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[2] K. Rohe, S. Chatterjee, and B. Yu, “Spectral clustering and the high-dimensional stochastic blockmodel,” *The Annals of Statistics*, vol. 39, no. 4, pp. 1878–1915, 2011.

[3] N. Bansal, A. Blum, and S. Chawla, “Correlation clustering,” *Machine Learning*, vol. 56, no. 1-3, pp. 89–113, 2004.

[4] Y. Chen, A. Jalali, S. Sanghavi, and H. Xu, “Clustering partially observed graphs via convex optimization,” *Journal of Machine Learning Research*, vol. 15, pp. 2213–2238, 2014.

[5] Y. Chen, S. Sanghavi, and H. Xu, “Clustering sparse graphs,” in *NIPS*, 2012, pp. 2213–2221.

[6] Y. Chen and J. Xu, “Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices,” *arXiv preprint arXiv:1402.1267*, 2014.

[7] A. A. Amini and E. Levina, “On semidefinite relaxations for the block model,” *arXiv preprint arXiv:1406.5647*, 2014.

[8] T. Cai and X. Li, “Robust and computationally feasible community detection in the presence of arbitrary outlier nodes,” *arXiv preprint arXiv:1404.6000*, 2014.

[9] P. W. Holland, K. B. Laskey, and S. Leinhardt, “Stochastic blockmodels: First steps,” *Social networks*, vol. 5, no. 2, pp. 109–137, 1983.

[10] S.-Y. Yun and A. Proutiere, “Community detection via random and adaptive sampling,” in *Proceedings of The 27th Conference on Learning Theory*, 2014, pp. 138–175.

[11] K. J. Ahn, S. Guha, and A. McGregor, “Graph sketches: sparsification, spanners, and subgraphs,” in *Proceedings of the 31st symposium on Principles of Database Systems*. ACM, 2012, pp. 5–14.

[12] V. Grebinski and G. Kucherov, “Optimal reconstruction of graphs under the additive model,” *Algorithmica*, vol. 28, no. 1, pp. 104–124, 2000.

[13] G. Dasarathy, P. Shah, B. N. Bhaskar, and R. Nowak, “Sketching sparse matrices,” *arXiv preprint arXiv:1303.6544*, 2013.

[14] Y. Chen, Y. Chi, and A. J. Goldsmith, “Exact and stable covariance estimation from quadratic sampling via convex programming,” *arXiv preprint arXiv:1310.0807*, 2013.

[15] ——, “Estimation of simultaneously structured covariance matrices from quadratic measurements,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 7669–7673.

[16] ——, “Robust and universal covariance estimation from quadratic measurements via convex programming,” in *IEEE International Symposium on Information Theory*, July 2014.

[17] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina, “Detection and identification of network anomalies using sketch subspaces,” in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. ACM, 2006, pp. 147–152.

[18] D. Donoho, “Compressed sensing,” *IEEE Trans. Info. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[19] E. Candés and T. Tao, “Decoding by linear programming,” *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.