Graph Completion

Tuan Dinh
University of Wisconsin - Madison

Aug 1st, 2017

---

1 Introduction

Recovering signal from partially observed samples is recently active with related development in matrix completion, collaborative filtering. Wavelet transform is a widely used transform and recently extended to the graph domain[1]. Matrix completion is currently an active research area with recent novel research and theory [2].
Most modern datasets however have additional information, either as features or as pairwise relationships between variables. For example, in the case of recommender systems, one can have demographic information or a social network for users. In sensor networks, one might have pairwise similarity information based on the actual locations of sensors. It makes sense to assume that using this additional information will aid in making predictions, and recently, several methods have been proposed to do the same [6], [7].
In this work, we consider a practical setting of a neuroimaging study. Each patient has 2 types of measurement: cheap measurement (cognition, genetic, roi) and expensive measurement (CSF NFL, Ab142, pTau, hTau). For cheap measurements, we have access to full data of every patients while we have only a small proportion of the expensive one. For example, among 100 patients, we have all 100 data points of congition data while having only 20 records of CSF data. Our goal is to recover 80 missing records from these given data.

2 Ideas

The idea behind: both cheap data and expensive data share mutual information about the underlying brain-disease relationship between patients. Therefore, we can build up a relation network based on the fully cheap data, then apply graph completion on this to recover expensive data points.

This can be modeled as an **optimization problem**:

$$|P_w(Mg - y)|_2^2 + \gamma h(g)$$

where g is the recovered signal, y is the observed data vector, P_w is the importance value, M is the projection matrix (to observed samples only) and the righthand term is the regularizer.

Assumptions and conditions: cognitive data contains relationship information about CSF, and the adjacency matrix contains true connection in terms of CSF between nodes.
Why not machine learning? the problem of small dataset and the nature of signal.

3 Problem setup, our method, and theory

We can divide the overall process into 2 main steps: Graph construction and Graph completion. Firstly, given fully cheap dataset and partially observed expensive dataset, we construct a Adjacency matrix that represents relationship between people. Using it, Graph completion part recovers the missing part of CSF signal. The whole pipeline of process is given as below:

3.1 Pre-processing

As the first step, we do feature scaling into the interval (0, 1) that makes every features equal and remove outlierst to clean noise.
Secondly, we select only a few important features among tens of features because multiple trivial features can affect the true distance metric.Features having high correlation with the target CSF are selected or using lasso to filter important features

3.2 Graph Construction

The goal of construction is to extact near-CSF-related connection between patients based on the cognitive one. The construction part can be divided into 3 steps [Tony]: metric learning, sparsification and reweight the matrix.
The important part is metric learning: how we measure CSF-similarity between two patients using their cognition, that is the kernel k so that $k(cog1, cog2) \approx sim(csf1, csf2)$. We also apply variant methods to learn this similary measures.
Metric learning: learn similarity, learn connection

- Kernel learning: rbf, laplacian, linear, polynomial, sigmoid, cosine
- Learn CSF directly (lasso, ensemble), then using predicted CSF to build the graph: few data points
- Learn CSF pair-distance, then use it to predict the weight of each connection
- Classify connection: consider only binary connection, learn from observed data points to classify each connection as 0 or 1
- MMD linear transformation
- ITML (Information Theoretic ML): using the given threshold to assign label (+/-) to each node, then using ITML to learn the metric between them
- Semi-supervised ITML

Sparsification is used to keep only strong connection. We apply kNN, thresholding - weighted or binary.

- kNN using a new sigma (Dk/3) - Tony's paper
- b-Matching: supervised learning
- Percentile threshold
- Unweighted connection

3.3 Graph Completion

Laplacian extraction, Intialize pre-used energy, Random Sampling, Recovery Optimization, Select and recover.

3.4 Post-processing

Control unreasonable prediction

3.5 Evaluation

RMSE, Error percentage, Recovery accuracy, Precision.

3.6 Baseline methods

Linear regression, Lasso.

4 Experiments

4.1 Setting

Datasets: CSF: 5 x 147, Cog: 27 features, 147 matches; Genetic: 65 - 27, 126 matches; ROI: 60 matches; Combined.
Default Params: CSF: Ab142, Available: 0.4, Max Samples:0.6, nSelected: 10, preSigma: 0.1, kernel: RBF, Threshold: 0.9, weighted: binary, KNN: 12, metric learning: ITML, rBand: 0.8, alpha - regularization: 0.1, gamma: 10
Cross-validation: Fold: 5 (shift), Iters: 100, Methods: randomly permutate.

4.2 Evaluation

5 Discussion

5.1 Analysis

- A good graph needs to be: sparse, full band
- Regularization doesn't help
- observed data always get nearly 100
- ROI $\geq$ Genetic $\geq$ Cog
- other CSFs don't really help

Questions

- Role of wavelet
- Optimization overfit
- Nature of data/problem
- former questions

5.2 TODO

- Obtain good graph - metric learning ? How to evaluate a graph being well-constructed? check if 2 signals are related? (cross-correlation); semi-supervised approach
- Strengthen optimization for uncertain graph: the current approach strongly depends on the graph (Leverage values)
- Hybrid

Table 1: Dataset 1

Dataset	Selection	Learning	Sparse	Binary	Unobserved	Observed
1	0	1	0	0	1.54961642086494E+57	41
1	0	1	0	1	231	46
1	0	1	80	0	1.10132257451605E+86	2.96574466415869E+69
1	0	1	80	1	225	47
1	0	1	90	0	1.09866432626114E+65	35
1	0	1	90	1	1104	46
1	0	3	0	0	3.71675892880196E+101	1.01917511102594E+85
1	0	3	0	1	222	47
1	0	3	80	0	1.72638716966231E+74	1.8791754897833E+58
1	0	3	80	1	282	46
1	0	3	90	0	6.36406024670921E+50	30
1	0	3	90	1	225	46
1	0	4	0	0	440	47
1	0	4	0	1	187	173
1	0	4	80	0	454	48
1	0	4	80	1	198	75
1	0	4	90	0	475	48
1	0	4	90	1	237	59
1	10	1	0	0	2.42644453862811E+32	35
1	10	1	0	1	218	60
1	10	1	80	0	4.35973959870523E+109	37
1	10	1	80	1	214	62
1	10	1	90	0	2.08978550566874E+216	41
1	10	1	90	1	246	62
1	10	3	0	0	1.74358185482125E+39	39
1	10	3	0	1	212	65
1	10	3	80	0	5.93262384782626E+48	2733
1	10	3	80	1	426	60
1	10	3	90	0	4.166170379216E+92	36
1	10	3	90	1	218	62
1	10	4	0	0	188	171
1	10	4	0	1	190	170
1	10	4	80	0	199	81
1	10	4	80	1	200	80
1	10	4	90	0	220	66
1	10	4	90	1	216	67

Table 2: Dataset 2

Dataset	Selection	Learning	Sparse	Binary	Unobserved	Observed
2	0	1	0	0	1.45882402120812E+50	63025804323189
2	0	1	0	1	229	45
2	0	1	80	0	1.43340247001959E+206	9.55489627082001E+117
2	0	1	80	1	252	49
2	0	1	90	0	2.99211738552092E+126	5710390983
2	0	1	90	1	258	47
2	0	3	0	0	1.90619837413455E+109	1.69474525741715E+92
2	0	3	0	1	233	48
2	0	3	80	0	2.69648070080857E+281	1.03591070221205E+79
2	0	3	80	1	227	45
2	0	3	90	0	5.87605647756047E+76	40
2	0	3	90	1	241	48
2	0	4	0	0	465	48
2	0	4	0	1	187	174
2	0	4	80	0	448	48
2	0	4	80	1	194	76
2	0	4	90	0	459	48
2	0	4	90	1	229	59
2	10	1	0	0	1.23852856051863E+52	37
2	10	1	0	1	216	65
2	10	1	80	0	1444906301113	37
2	10	1	80	1	212	63
2	10	1	90	0	3.3266124244703E+54	977562
2	10	1	90	1	211	61
2	10	3	0	0	10384246327	37
2	10	3	0	1	215	61
2	10	3	80	0	7.3046669499928E+76	150
2	10	3	80	1	214	62
2	10	3	90	0	23245	35
2	10	3	90	1	212	62
2	10	4	0	0	186	173
2	10	4	0	1	189	171
2	10	4	80	0	201	82
2	10	4	80	1	196	82
2	10	4	90	0	221	63
2	10	4	90	1	226	61

Table 3: Dataset 3

Dataset	**Selection**	**Learning**	**Sparse**	**Binary**	**Unobserved**	**Observed**
3	0	1	0	0	1.54903473090918E+33	1543012929119330
3	0	1	0	1	224	46
3	0	1	80	0	3.02737251340352E+23	33
3	0	1	80	1	222	45
3	0	1	90	0	2.69696418741955E+17	34
3	0	1	90	1	237	44
3	0	3	0	0	3.70926447247101E+179	2109477941248780
3	0	3	0	1	9277	44
3	0	3	80	0	1.53574797954259E+30	28270776028301
3	0	3	80	1	785	44
3	0	3	90	0	5.28506607280535E+122	9.66525191036136E+105
3	0	3	90	1	223	53
3	0	4	0	0	463	48
3	0	4	0	1	189	172
3	0	4	80	0	448	45
3	0	4	80	1	199	74
3	0	4	90	0	453	47
3	0	4	90	1	240	60
3	10	1	0	0	1.41374441607731E+90	44
3	10	1	0	1	212	62
3	10	1	80	0	1083854	37
3	10	1	80	1	219	60
3	10	1	90	0	3.40420221788906E+48	633312622368
3	10	1	90	1	215	61
3	10	3	0	0	3.45974604529664E+85	37
3	10	3	0	1	232	62
3	10	3	80	0	4.29195739957084E+43	36
3	10	3	80	1	216	59
3	10	3	90	0	3.11387913408208E+31	38
3	10	3	90	1	216	62
3	10	4	0	0	186	172
3	10	4	0	1	187	172
3	10	4	80	0	200	78
3	10	4	80	1	199	81
3	10	4	90	0	219	65
3	10	4	90	1	220	65

Table 4: No learning metric

Dataset	Selection	Learning	Sparse	Binary	Un	Ob
1	0	0	80	0	8.37534530699594e+25	31.2283285940655
1	0	0	80	1	169.811589282339	67.1314233865849
1	0	0	90	0	3.92689474545834e+27	31.2728103401116
1	0	0	90	1	8273.9478226901	48.2890713851527
1	10	0	80	0	188.89398953533	54.3337434394667
1	10	0	80	1	183.04721606027	76.3197172110488
1	10	0	90	0	210.232897671081	42.0337774719589
1	10	0	90	1	192.826607703433	52.8589041113998
2	0	0	80	0	8.47008277251517e+27	28.5544653381159
2	0	0	80	1	169.940921002443	66.3176891608828
2	0	0	90	0	1.03600604652679e+28	27.3782107285531
2	0	0	90	1	200.374401628382	48.7501166393764
2	10	0	80	0	187.517883561345	55.1083756739318
2	10	0	80	1	179.418694210083	77.3653994805961
2	10	0	90	0	201.61394698689	41.2969962890936
2	10	0	90	1	189.715486090034	52.7498314968634
3	0	0	80	0	5.44110187416378e+24	30.7053754138505
3	0	0	80	1	169.641393156147	65.4419378958738
3	0	0	90	0	1.13717210734559e+27	26.7336754955024
3	0	0	90	1	217.120328737184	49.6705866635551
3	10	0	80	0	190.905175653626	54.8149733968542
3	10	0	80	1	178.03491250063	78.556383990529
3	10	0	90	0	208.987455065167	42.7130502325773
3	10	0	90	1	192.002262018352	52.0688875155099

Table 5: Random Matrix

Sparse	Binary	Un	observed
0	0	640.2203	34.7778
0	1	187.9999	172.1380
90	0	665.6549	33.7123
90	1	232.1691	60.5835

Table 6: Baseline Methods

Dataset	Selection	Baseline	Unobserved	Observed
1	0	1	199.1077	148.6310
1	1	1	200.7876	147.3206
2	0	1	468.6379	79.0458
2	1	1	480.6175	77.9348
3	0	1	362.6294	0
3	1	1	365.9741	0
1	0	2	200.2997	147.5909
1	1	2	202.8420	147.0311
3	0	2	223.1343	5.4357
3	1	2	216.0443	5.4199
1	0	3	247.0391	83
1	1	3	249.6211	82.8922
3	0	3	245.5798	76.6625
3	1	3	238.2402	79.0812