Contents

1 SVM vs Neural Networks PartI

1.1 Problem

Select at least two datasets from the DataA,and then investigate classication performances of Support Vector Machine (SVM) and neural networks (e.g., MLP) on the selected data sets. Try different experimental settings, e.g., varying the sample size of the training set, trying data sets with different dimensions, and other congurations that may affect the performance in your mind. You may also try different kernels for SVM.

1.2 Introduction

In the first part, I evaluate the performance of Support Vector Machine(SVM) and Multi-Layer Perception(MLP) on different datasets. In the second part, I apply SVM on CIFAR-10 dataset and compared with the state-of-art deep learning method. Besides, I discuss the strengths and weaknesses of SVM on big datasets.

1.3 Data Description

name	class	data	features
diabetes	2	768	8
breast-cancer	2	683	10

1.4 Models

1.4.1 SVM

Given training vectors $x_i \in \mathbb{R}^p$, i=1,...,n,in two classes, and a vector $y \in \{1, -1\}^n$,SVC solves the following primal problem:

$$\min_{\omega,b,\zeta} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}\zeta_i$$

$$s.t.\ y_i(\omega^T\Phi(x_i) + b) \geq 1 - \zeta_i,$$

$$\zeta_i \geq 0, i = 1, \ldots, n$$

The principle of SVM is shown in Figure 1.4.1.

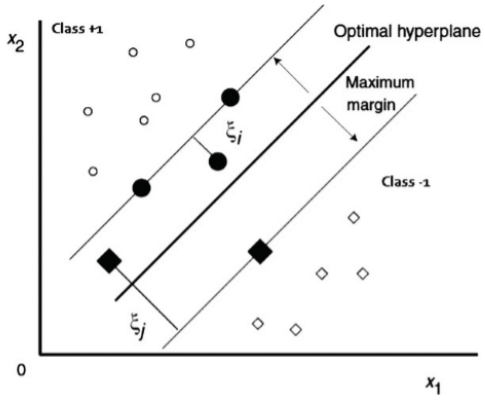

Figure 1: SVM

1.4.2 MLP

Multi-layer Perception(MLP) is a supervised learning algorithm that learns a function $f() : R^m \rightarrow R^m$ by training on the dataset, where m is the numbers of dimensions for input and n is the number of dimensions for output. Given a set of features $X = x_1, x_2, \ldots, x_m$ and a target y, it can learn a non-linear function approximator for either classification. It is different from logistic regression, in that between the input and output layer, there can be one or more non-linear layers, called hidden layers. Figure 1.4.2 demonstrates a one hidden layer MLP with scalar output.

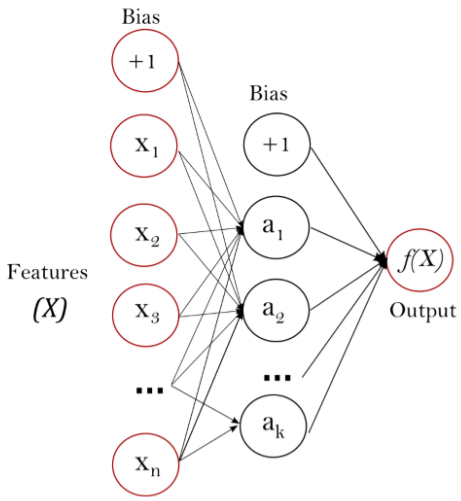

Figure 2: One hidden layer MLP

dataset	model	best test size	best accuracy
diabetes	svm	0.4	0.669
	mlp	0.1	0.766
breast-cancer	svm	0.4	0.635
	mlp	0.5	0.643

Table 1: Test size

1.5 Experiment

In my experiment, I use diabetes and Iris as my dataset to evaluate the performance of SVM and MLP. I use *sklearn.svm.SVC* and *sklearn.neural_nerwork.MLPClassifier* as my model.

1.5.1 Training Size

Diabetes and Iris has no original testing samples so I use *train_test_spilit* in sklearn to split a part of data from the original data. The result of different size of data is shown in Figure 3 and Figure 4. I use the original model from sklearn without given any parameters at first. And the best result here is in Table 1. I think it is necessary to find the best parameters so the comparison between SVM and MLP will be more objective. I choose 40% as my test size at last for this size performs well in most occasions and is representative enough.

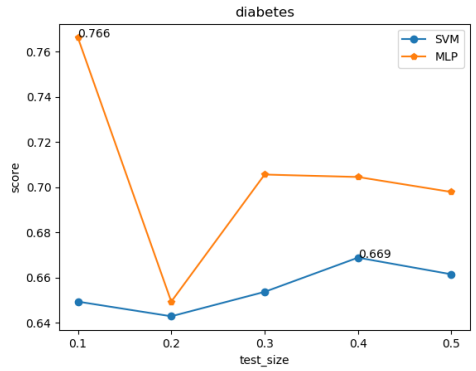

Figure 3: diabetes

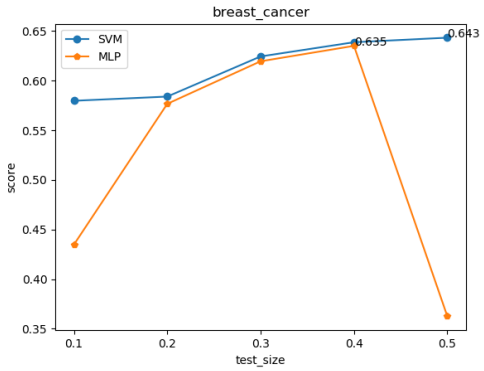

Figure 4: breast

1.5.2 hyper parameters

The hyper parameters of SVM is shown in Tabel 2. And hyper parameters of MLP is shown in Table 3. To find the best parameter I use the *GridSearchCV* provided by scikit-learn. *GirdSearchCV* will exhaustively search over specified parameter values for an estimator.

Hyper parameters in SVM include Penalty parameter C,kernel and gamma which represent the kernel coeffcient. Hyper parameters in MLP include hidden_layer_sizes, learning_rate, solver, alpha(L2 penalty parameter) and activation.

Optimal solution for SVM The GirdSearchCV found the best hyper-parameters for SVM in given ranges and the effect of C is listed in Figure 5.

kernel	'linear','rbf','poly','sigmoid'
C	0.01,0.03,0.06,0.1,0.3,0.6,1.0,3.0,10.0,100.0,1000.0
gamma	1e-3,1e-4,'auto'

Table 2: The parameters of SVM

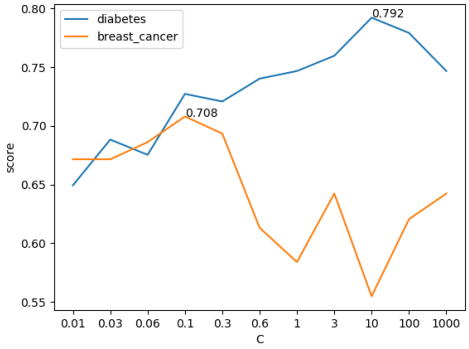

Figure 5: different C in SVM

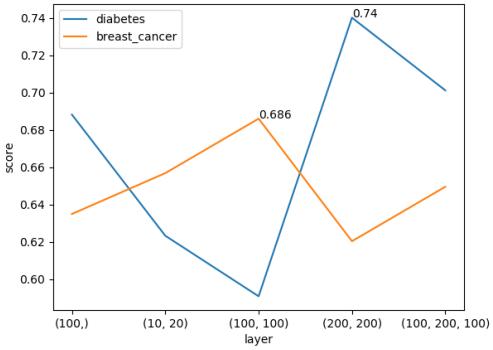

Figure 6: different layers in MLP

Optimal solution for MLP The GirdSearchCV found the best hyper-parameters for MLP in given ranges and the effect of layers is listed in Figure 5.

solver	'lbfgs','sgd','adam'
hidden_layer_sizes	(100,),(10,20),(100,100),(200,200),(100,200,100)
alpha	1e-3,1e-4
activation	'identity','logistic', 'tanh', 'relu'
learning_rate	'constant','invscaling','adaptive'

Table 3: The parameters of MLP

model	datasets	best parameters	score
SVM	diabetes	c:10, gamma:0.0001, kernel:rbf	0.792
	breast-canceer	c:0.01, gamma:auto, kernel:rbf	0.708
MLP	diabetes	alpha:0.0001, hidden_layer_sizes:(200,200) learning_rate:'adaptive', slover:'adam'	0.740
	breast_cancer	alpha:0.001, hidden_layer_size:(100,100) learning_rate: 'adaptive', solver:'adam'	0.686

Table 4: Caption

Results The results of the best hyper parameters for SVM and MLP in shown in Table 4. Actually the optimal choice is not unique. Sometimes changing a parameter only has a slight disturbance to the result while sometimes the same parameter performs differently. So I chose the most frequently occurring optimal solution.

1.5.3 data dimension

In this part, I use PCA to realize the dimension reduction.Principal component analysis (PCA), is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. In this part, I adjust the models with the optimal solution in Hyper Parameters part. I use *sklearn.decomposition.PCA* to realize my implementations. And the result is shown in Figure 7 and Figure 8. As we can see, suitable dimension reduction will improve the performance. And MLP seems works not as well as SVM after adjusting the parameters. The result is shown in Table 5.

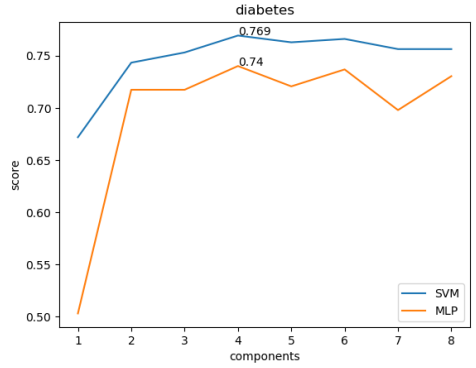

Figure 7: different components in diabetes

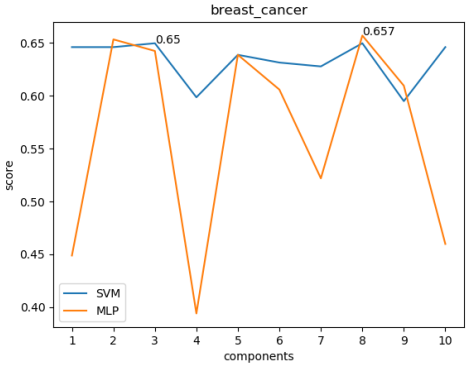

Figure 8: different components in breast_cancer

datasets	model	best_components	best_score
diabetes	SVM	4	0.769
	MLP	4	0.740
breast_cancer	SVM	3	0.650
	MLP	8	0.657

Table 5: pca results

datasets	model	Scaele(Y/N)	score
diabetes	SVM	N	0.656
		Y	0.675
	MLP	N	0.591
		Y	0.945
breast cancer	SVM	N	0.578
		Y	0.698
	MLP	N	0.628
		Y	0.964

Table 6: Scale Comparison

1.5.4 data scale

In this part, I scale the datasets and compare their performances before and after scale. I use StandardScaler from sklearn to implement the scale. The result is shown in Table 6. It is amazing that data scale has a huge improvement on MLP. However, it depends both on the datasets and the models. For breast_cancer dataset, both datasets has a large improvement. And MLP benefits more from scale then SVM.

1.6 Conclusion

So we can get the total results of SVM and MLP for my datasets. I place the best results in Table 7. I choose the test size as 0.4 at last for it is reasonable to divide the datasets and always perform well. It is clearly that the SVM performs better than MLP before the data was scale and MLP is optimized. After I give MLP enough optimization, this deep learning method performs quite well. Besides, as the two datasets both has only two classes, SVM works well in the classification problems most time. I think MLP has more potential than SVM for its huge space for improvement but SVM is more universal for different kinds of problems.

best score / model \ datasets	diabetes	breast_cancer
SVM	0.792	0.698
MLP	0.945	0.964

Table 7: Best Score

2 SVM vs Neural Networks Part II

2.1 Problem

Select at least one dataset from DataB,and applied SVM on it. Compare the SVM performance with the deeplearning algorithm benchmarks. Discuss the strengths and weaknesses of SVM on big data sets.

2.2 Introduction

In this experiment, I will use CIFAR-10 as my dataset to evaluate the performance of SVM. The data description is showed in Table 2.3 and Picture 9. For it takes quite a long time to run a test, I decide to use the optimal parameter that I find in the dataset diabetes. I use PCA and StandScaler from sklearn to improve the performance of SVM.

2.3 Data Description

name	class	training samples	testing samples	features
CIFAR-10	10	50000	10000	3072

Table 8: CIFAR-10

2.4 Experiment

- kernel: 'rbf'
- gamma: 1e-4
- C:10
- Optimization: PCA, StandScaler

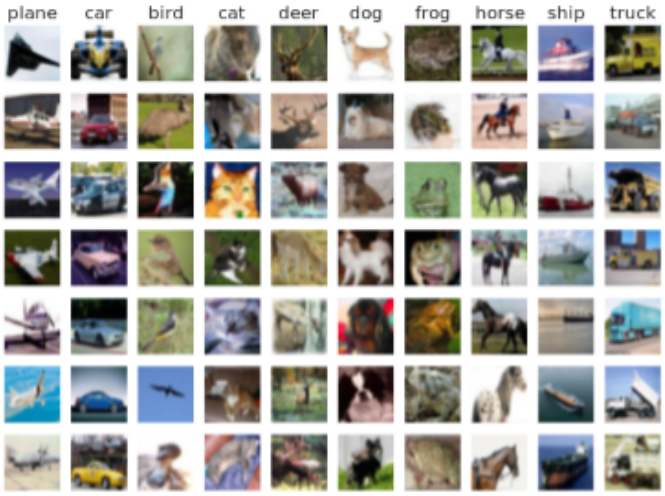

Figure 9: CIFAR-10

Stand Scaler
StandScaler: Standardize features by removing the mean and scaling to unit variance. After the standscaler all data has a mean 0 with a variance of 1.

$$x_i = \frac{x_i - \overline{x}}{\sigma}$$

PCA
There are 3072($3 \times 32 \times 32$) features in the original data. The principle of PCA in descibed in the data dimension part. I compress the features for $10\%, 30\%, 50\%, 70\%, 90\%$ and test their performance. It takes a long time for SVM to deal with 3072 features and PCA will excel the process a lot. So I want to get a balance between the speed and data dimension and enlarge the accuracy as much as possible.

2.5 Comparison

The comparison between SVM and other deep learning methods is demonstrated on Figure 10. We can clearly get from the chart here that there is a huge gap between SVM and other deep learning methods. Although SVM performs better after scale, it still has a large distance with NiN and other deep learning methods. The results of running PCA on CIFAR-10 is shown in Figure 11. As the accuracy of SVM before scale is only 0.103, there is no significance to apply PCA on it so I use PCA and scale together to test the performance. Sadly, data dimension makes SVM performs poorly even if I compress the data features to 90%. I think it is because although there is 3072 features, the original picture is just

$32 \times 32 \times 3$ so data dimensionl like PCA will destroy some inner connections in the small picture. Maybe PCA will suit for some more exquisite pictures.

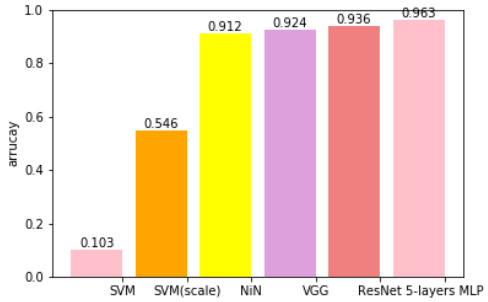

Figure 10: Comparison on SVM with deep learning methods

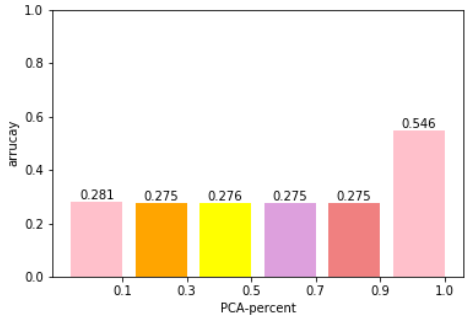

Figure 11: comparison on SVM running on PCA

2.6 Strengths and Weaknesses of SVM on Big Datasets

With comparison to other deep learning networks, SVM got a bad performance even with optimization running on CIFAR-10. And SVM demonstrates its strengths and weakness in this process.

Strength

1. SVM has a regularization parameter, which will avoid overfitting more likely.
2. SVM has the kernel trick, and a suitable kernel will improvement its performance greatly.
3. SVM has been translated to a convex optimization problem so it will be more likely to obtain a global optimal solution and deep learning method sometimes will struck in the local optimal solution.
4. SVM is an approximation to bound on the test error.
5.

Weakness

1. SVM only convers the determination of parameters for a given value.So it is more likely transform the problem to model selection.
2. Kernel models is quite sensitive to over-fitting models.

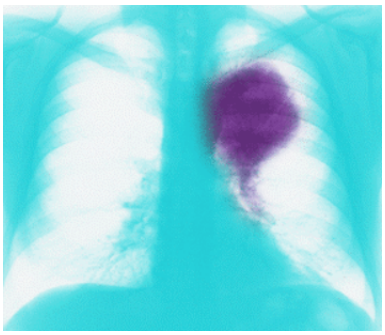

Figure 12: Lung Cancer

3. SVM use quadratic programming to find the hyper-parameters, which would consume quantities of memories and time.

4. The cost of parameters adjustment is quite tremendous and SVM is useful with binary classification while most big datasets has many class. And SVM performs not good in multi-class classification.

3 Casual discover algorithms

3.1 Problem

According to data released by the World Health Organization (WHO), lung cancer ranks first in cancer worldwide, both in terms of morbidity and mortality. Studies have shown that 90% of lung cancer patients have had a history of smoking. However, relevant studies have shown that less than 20% of the smokers in the smokers are in the total smokers, and 10%15% of the non-smokers are also afflicted with lung cancer. Is there a causal relationship between smoking and lung cancer? To solve the problem, I use dataset LUCAS from http://www.causality.inf.ethz.ch/home.php.

3.2 Datasets

I use datset LUCAS to find the casual relation for lung cancer. The dataset has 2000 samples and 12 variables.
Variables

1. Smoking

2. Yellow Fingers

3. Anxiety

4. Peer Pressure
5. Genetics
6. Attention Disorder
7. Born On Even Day
8. Car Accident
9. Fatigue
10. Allergy
11. Coughing

3.3 Algorithm

3.3.1 Introduction

LiNGAM is a linear continuous, non-Gaussian, directed acyclic causal model. Therefore, after the variables are arranged in a causal order, the variable located at the back cannot be the dependent variable of the previous variable. In practical applications, the order of the observed variables is random and different from the causal order. According to the order of the samples in the observation, the sample is recorded as $\{x_1, x_2, \ldots, x_n$,Record the causal order as $k(i), i \in [1, n]$. It represents the position of the i_{th} variable in the causal order in the observation order. The generation process of the name variable can be described as:

$$x_i = \sum_{k(j)<k(i)} b_{ij} + e_i, i, h \in [1, n] \tag{1}$$

the noise e_i are none-Gaussian distributed, and e_i is independent of each other. If b_{ij} is not zero, there is a side where x_j points to x_i. LiNGAM is expressed in matrix form as

$$x = Bx + e \tag{2}$$

3.3.2 Desciption

LiNGAM is a classic algorithm for structural inference in the LiGNAM model. In general, the LiNGAM is divided into estimation and pruning phase,

Estimation The first phase is the estimation phase, which leads to a causal sequence and preliminary estimate the entire causal structure (matrix coefficients). From equation 2,$Wx = e$ can be derived. $W = (I - B)$,I is the identity matrix, W is the mixing matrix. The LiNGAM algorithm first calculates the separation matrix W' f based on the observed data, but there are two problems toward W'

1. The diagonal in W' may be 0, which is contradicted to that all diagonals are 1
2. the variables are randomly arranged, and cause and effect The order does not match

In order to solve these two problems, LiGNAM eorders W' so that the diagonal of W' is not 0, then LiGNAM will let the row divided by the value of the diagonal of the row(normalized) to get W", and $B = E - W''$ is calculated. Then the coefficient matrix corresponding to the causal order is a lower triangular matrix with diagonals of 0. With this constraint, the algorithm re-arranges the rows and columns of B to get B', so that B' is expanded to the lower triangular matrix, thus obtaining the causal order. Finally LiNGAM derives the specific matrix coefficients using the least squares method. The gained directed graph from the estimation phase corresponding to the matrix B is fully associative, but is common in practice that the figure is sparse, so it is a necessary step for prunning.

Pruning The second stage is the pruning phase, which is the test of each edge that is not 0 in the relational matrix derived from the estimation phase. The algorithm MODELFIT is raised by Shimizu. Firstly,The algorithm uses Wald to test the level of significance of each side. For each edge that fails the Wald test, MODELFIT root mean square test to verify the difference between the model before and after cutting off the edge. MODELFIT decides whether to pruning based on the significant level of difference between the model before and after cutting off an edge.There are two disadvantages of the algorithm

1. Every time MODELFIT cuts an edge, it needs to check the significance of the model difference. This operation takes a lot of time to calculate matrix inversion and matrix rooting operations ($O(n^3)$;
2. Contrast experiments show that the MODELFT algorithm has a relatively low accuracy in sparse graphs.

3.4 Result

I use Tetrad to implement the LINGAM algorithm: Tetrad can be referred though www.phil.cmu.edu/tetrad/. And the relation is shown in Figure 13. Arrow in the picture means causality And we can see more clearly through Figure 14 get the direction reasons for Lung cancer. But there is a mistake between Genetics and Coughing. Is there any connection between them? I suspect it. This echoes the above mentioned and sometimes LiNGAM will

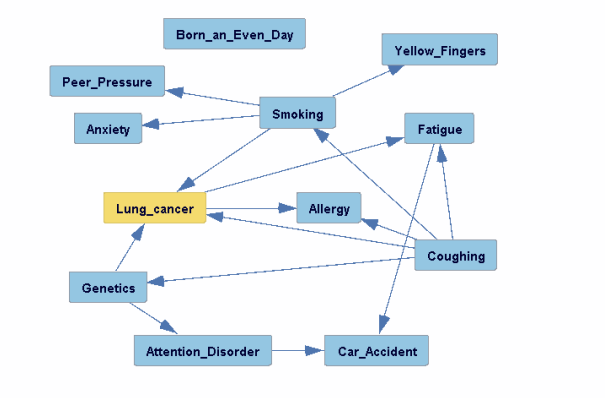

Figure 13: Total Casual Relation

go wrong in the sparse graph. But most part of the connections is reasonable like Figure 15 and Figure 16.

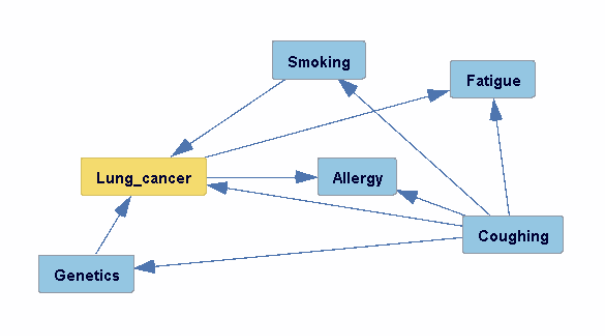

Figure 14: Caption

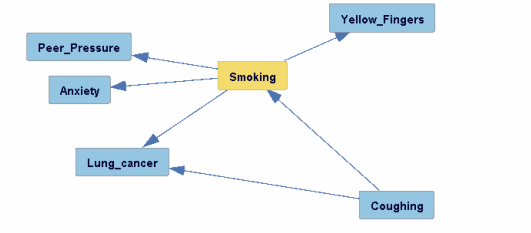

Figure 15: Smoking

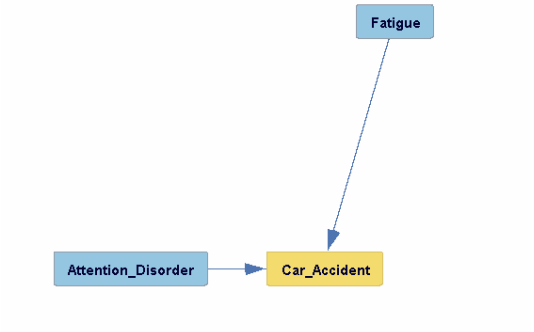

Figure 16: Car accident

A Problem1

The code for the SVM vs. Neural Networks on small datasets is in the $problem1_part1.py$. The code for the SVM vs. Neural Networks on small datasets is in the $problem1_part2.py$. $CIFAR-preprocess.py$ record the process to deal with CIFAR-10. The pictures in pic store the results that I used.

B Problem2

I placed the Directed Acyclic Graph produced by LiNAGM.