CS420 Machine Learning – Assignment 3

Xiang Gu
5130309729

1 Support Vector Machines (SVM) vs. Artificial Neural Networks (ANN)

1.1 SVM and MLP on a1a and w8a Dataset

I experimented and compared results of SVM and a MLP classifier on two simple dataset – a1a and w8a. Both datasets are used are binary classification.

Specifically, a1a is a dataset that contains 1605 training samples and 30956 test samples where each sample is represented by a 123-dimensional feature and a binary label (-1 or +1). w8a is a larger dataset with 49749 training samples and 14951 test samples where each sample is represented by a 300-dimensional feature and a binary label (-1 or +1).

On each dataset, I applied the sklearn.svm.SVM classifier with default parameter setting, and a three-layer MLP (inputs, 10, 5, 1) to them with sklearn.neural_network.MLPClassifier.

sklearn.svm.SVC(gamma='auto', random_state=1)

sklearn.neural_network.MLPClassifier(hidden_layer_sizes=(10, 5), max_iter=800, random_state=1)

After training on the training set, I tested both learned classifier on the test set and the results are summarized in the following table.

	a1a	w8a
SVM	0.835734	0.974449
MLP	0.809633	0.994448

1.2 SVM on Large Dataset (CIFAR-10)

I also applied SVM on some large dataset. I chose the CIFAR-10 image dataset that is commonly used for classification. CIFAR-10 contains 50000 training samples/images and 10000 test samples/images where each sample/image is a 32 by 32 color image in 10 classes. In other words, this is a multi-class classification problem.

Due to computational limitation, I did not train a SVM with all those available 50000 images but only 10000 of it. Again, I used the sklearn implementation of SVM. In particular, for efficiency reasons, I chose the LinearSVM class to construct a SVM classifier with linear kernel.

sklearn.svm.LinearSVC(C=0.1, max_iter=6000, random_state=1)[1]

Then I tested the (10-way) classification accuracy on the test set and I got the following accuracy, shown in the table below when compared with some of the benchmark baselines in the literature: [2] We can see that our naive and simple method performs very poorly on this dataset. An 20% accuracy rate means it is just 10% better than random guessing. One of the reason why it did not perform well is probably that this dataset is not linear separable with linear kernel. It might be helpful if we try other non-linear kernel (e.g. RBF).

When applied to large dataset like CIFAR-10, the advantage of SVM is fast speed and low cost: Both the training and test process of using SVM are faster than using deep neural networks. This makes real-time processing possible with SVM. Of course, the disadvantage is of course the performance loss when compared with deep learning models. When one can afford necessary computation and seek extreme performance, deep learning models are usually the direction to pursue but SVM are still OK because of its acceptable performance and, more importantly, lightening speed.

[1] I used this many iterations to ensure convergence of the algorithm.

[2] kindly collected by Rodrigob Benenson

Result	Method	Venue
96.53%	Fractional Max-Pooling	arXiv 2015
95.59%	Striving for Simplicity: The All Convolutional Net	ICLR 2015
94.16%	All you need is a good init	ICLR 2016
94%	Lessons learned from mutually classifying CIFAR-10	Unpublished 2011
....	....	...
20.79%	Linear SVM (our method)	This Assignment

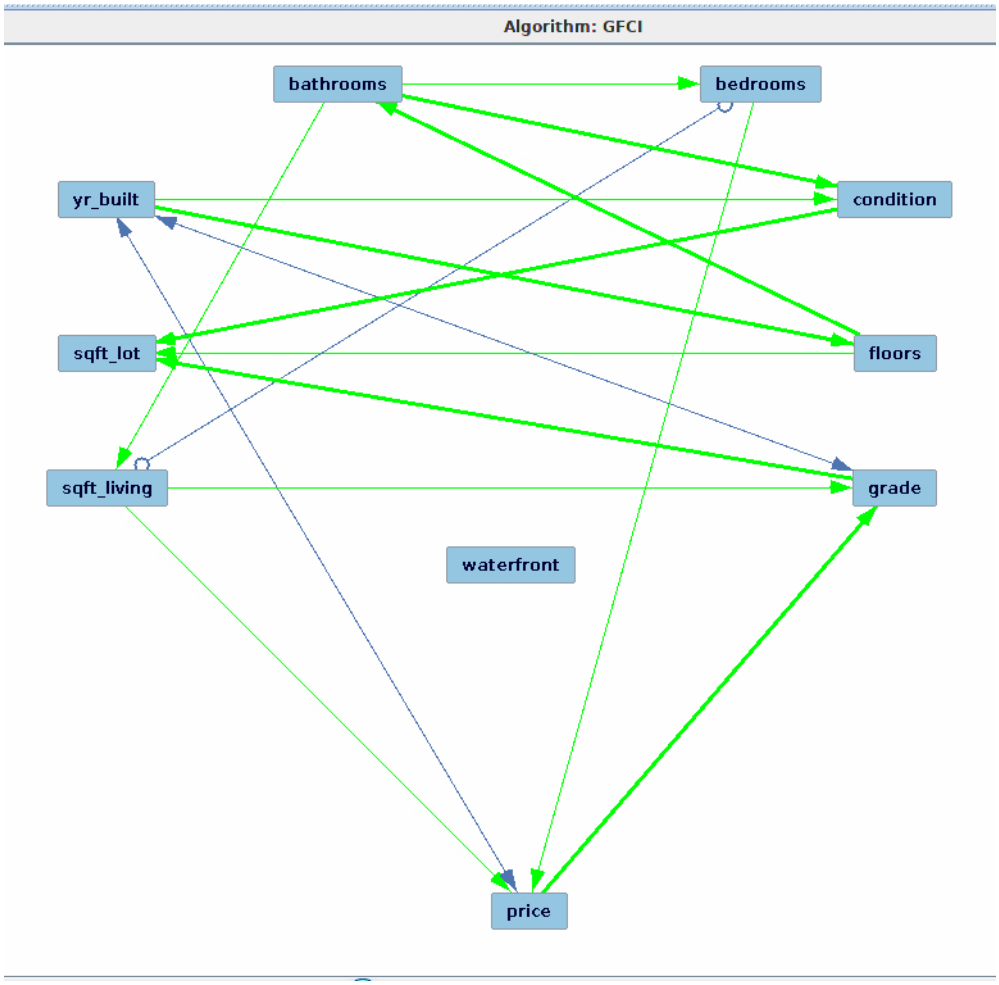

Figure 1: The causal graph found by the GFCI algorith. A green edge means that there is no latent confounder, and a bold edge means that it is definitely direct.

2 Causal Discovery Algorithm

For this question, I used the King County House dataset [3]. This dataset contains 21613 house records in the King County area including Seattle. Each house record in the original dataset contains 19 house features such as num_bedrooms, num_bathrooms, square_footage_living_room, etc. I deleted some of those features to 10 features [4].

I then used the Tetrad software [5] to search for an graphical causal model for this dataset. Namely, I used the GFCI algorithm with alpha = 0.05 and one-edge faithfulness = NO. The result is presented in figure 1. As we can see from the result, the number of bedrooms and the area of living rooms are two causes of the prices without latent confounder. There is an unmeasured confounder (call it L) between yr_built and price. Surprisingly, the condition of the house and the (official) grade given to the house is not a cause of price, although we might think so. Also, waterfront (whether this

[3] downloadable from this kaggle page.
[4] Remaining features can be found in the causal relationships graph.
[5] An open source software project for graphical causal models.

house has a view to a body of water) is not a cause of the price, although my experiences are that hotels in Sanya, China are definitely more expensive if they are built in waterfront areas. But this dataset, collected from the United State for House sales, does not show any causal relation between waterfront and the price.