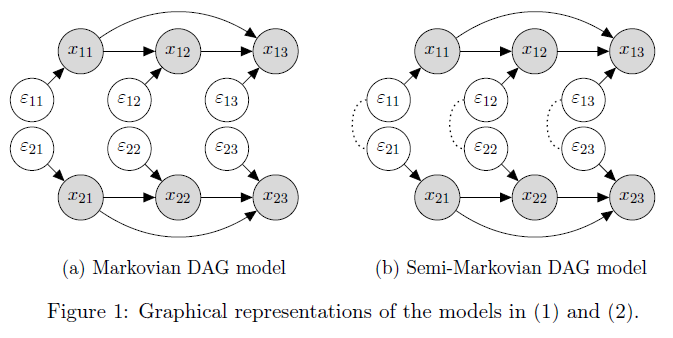
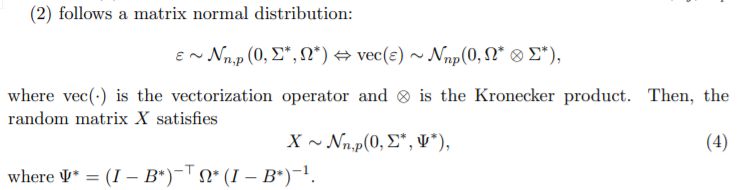
Preparation for Meeting 3

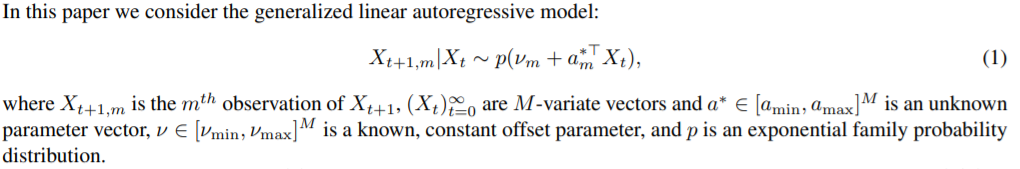
Things done:

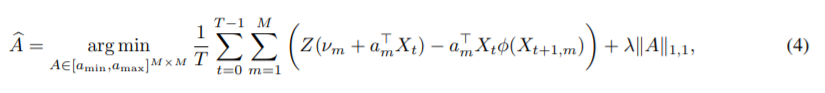
* Read paper “Learning Gaussian DAGs from Network Data”
  + Interesting approach, by assuming a matrix-normal distribution, they can include dependence between the different *n* individuals for a particular feature.



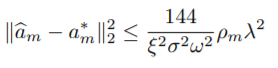
* + They model the data X as follows:
  + The parameters that need to be estimated are B\* (the WAM), Omega\*, and Sigma\* or equivalently, Theta\* = inv(Sigma\*).
  + They also claim that to impose sparsity, such a constraint needs to be applied directly on B\*, not on Psi\*.
  + It is important to note that they need to know a topological ordering for many
  + To solve to model, they minimize the following penalty function:



* + If they assume, they have a topological ordering, they can solve this minimization problem quite easily using block coordinate descent. They do this by first *pre-estimating* Omegas\*, and then iteratively using *block coordinate descent* to solve this problem. They estimate Omegas\* using the natural estimator introduced by Yu and Bien.
  + Also, when there is no topological ordering known, they can still pick a random permutation and using the Cholesky decomposition of Theta, they can *decorrelate* rows of X, making methods that assume independence more applicable and yield better results.
  + There are also a lot of error bounds proven for this.
* Also implemented the minimizations methods in the paper (WIP), was mainly because I had time left and was interested in replicating results / getting my hands dirty with these methods. Also added ROC curve, but scores are not as good as them because no block structure.
* Read paper “Inference of High-dimensional Autoregressive Generalized Linear Models”
  + More theoretical, it is about deriving a sparsity regularized MLE in a non-Gaussian setting. The model was as follows: 
  + To solve this, they could use the LASSO estimator, but rather use the REML.



* + Under some assumptions, they show that

.

* + However, they need to know the distribution to get bounds on the five parameters. They fill these in for the Bernoulli and the Poisson distribution.
  + As a conclusion, they show that they need much less data, the order of rho^3 log(M), is needed when the network is sparse.
* Wrote a notebook that generates a VAR model and uses two methods to infer the structural network:
  + Fitting a VAR(1) model and use the granger causality test to test causality. Set threshold of significance to determine an edge.
  + Minimizing ||X[t]- AX[t-1]||\_2^2 + lambda |A|\_1 to find A.
* Also wrote a notebook that uses the two aforementioned methods on a binary and on a discrete dataset, but the results are not as what I had hoped.
* Investigated BNlearn, a module in python where you can import useful datasets with ground truths to see how methods perform. Also includes fitting of other methods (exhaustivesearch, hillclimbsearch, chow-liu, PC).
* Investigated the pcalg module, which can perform the PC algorithm on data.