*Meeting 14*

*Duration: 75 Minutes. Present: Rui (First 60 minutes), Alex, and Martin.*

***Thesis***

*Wrote a large part of introduction and problem setting. Will send to Rui and Alex to have a look, but of course not a full version of anything.*

***OMP***

*However, another question then is, when we have a sparse DAG, when do we stop? We can continue adding edges until we violate the DAG assumption; then we stop.*

Implemented this.

*Alternatively, we can continue adding edges. When we see that we are about to add an edge that violates the DAGness, we discard this edge, and continue. We fix this constant to zero. We continue until we have a dense DAG (or similarly, until we have visited all nodes). This is a more principled approach, as we now also have an ordering of importance of the edges in our DAG. Then, when we want to prune our DAG, we can have a more principled approach than simply removing the smallest non-zero edge weight. Removing the smallest non-zero edge weight is not a principled approach, as a small edge weight does not imply a small importance. Using this OMP approach, we have a principled way of ordering the edges by importance. We can then prune by removing the edges from least important to most important.*

Implemented this.

*However, when do we stop pruning? This is a difficult choice. We can e.g. prune all edges who do not significantly improve the model fit (e.g. BIC / AIC). We can e.g. prune all edges that do not contribute enough to the loss value (e.g. by “curvature” in the loss function). The less you prune, the more predictive your model is.*

Since we have no ground truth, we can not pick any structural performance measures, e.g. highest accuracy. We do, however, have access to the loss. If we pick the one that minimizes the loss, then we will have a complete dense DAG, but generally the latest edges will contribute very little.

In general, the first edges contribute by far the most to the score / loss. The last “true” edges contribute very little, so it is difficult to say when to stop if we want to unveil all data generating edges.

When the time series have different variability, the last “true” edges contribute even less, so it makes it even more difficult.

***NOTEARS***

*NOTEARS was investigated a bit more. Proposition 3 of the papers was quite meaningless. No scale invariance, see notebooks of Alex.*

*Also interesting: Stationary points of h(W) where h(W) =/= 0. Furthermore, plots of h(W) as a function of w\_21 and w\_12.*

Function was plotted.

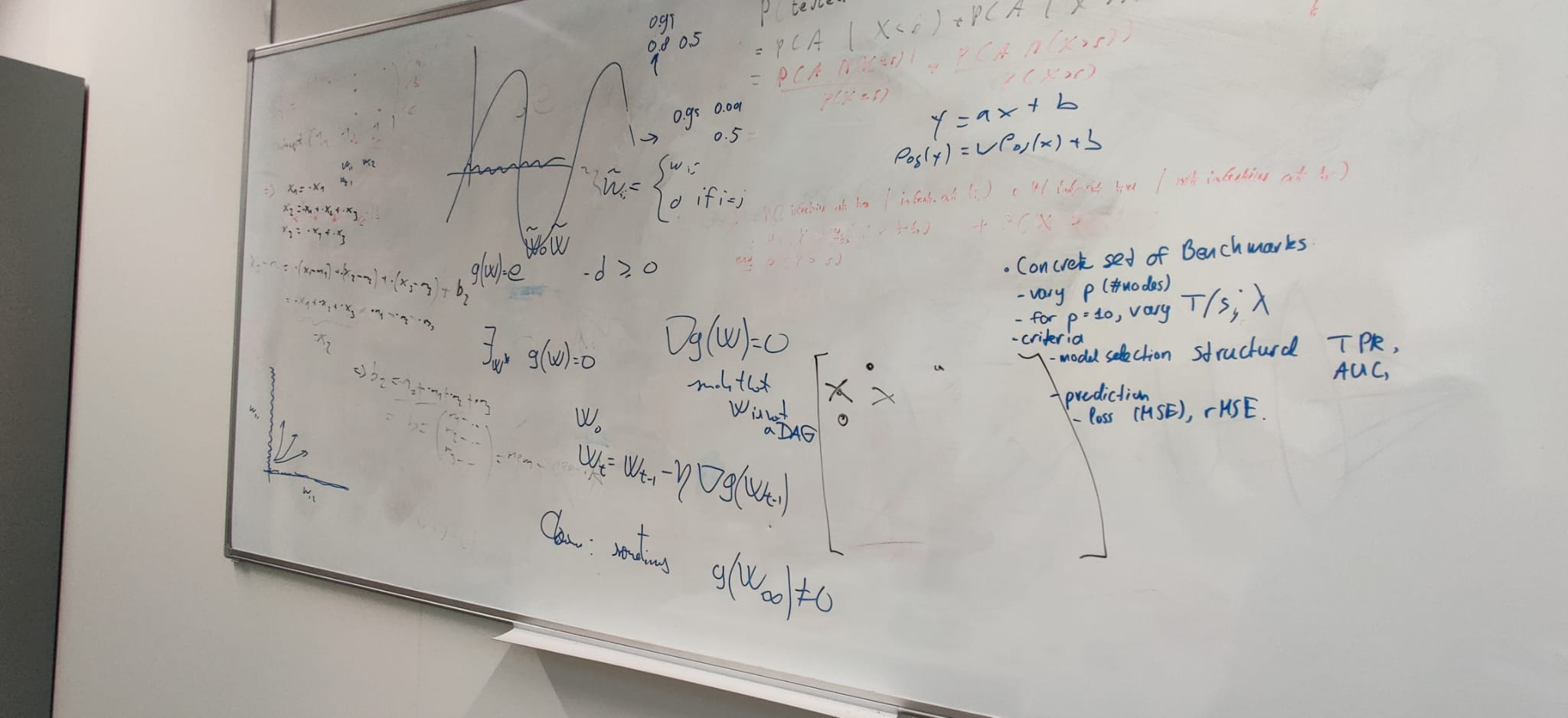
***Benchmarks***

*We discussed that it is good to have some concrete set of benchmarks. E.g., fix p = 10, and vary some other parameters, such as the number of samples T, the number of edges s. Get some good comparisons of all methods; OMP, NOTEARS, LASSO, OLS.*

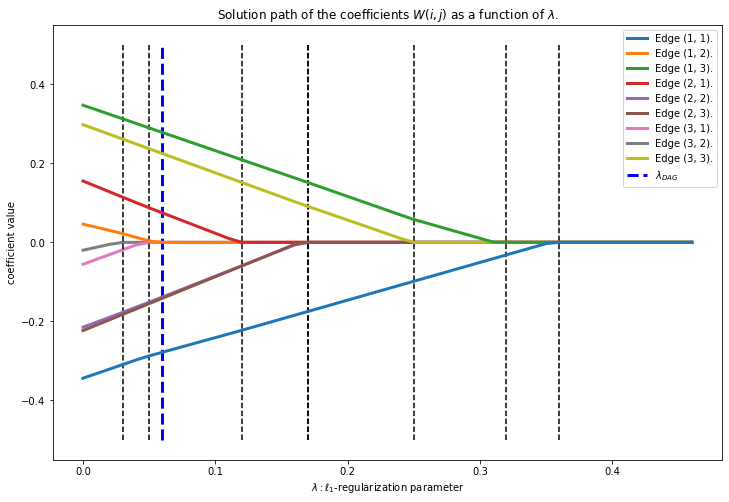
*Ideal: Find instances where each of the methods is not that good, to see the pros and cons.*

***Causal or Predictive***

*2D-Example where one time series had a much larger range, such that we were unable to detect the underlying generating matrix. However, we found a better matrix, in the sense that it had a higher predictive power. This showcases that our models are focused on getting the best predictive model under our constraints, and we do not necessarily focus on the true generating graph.*

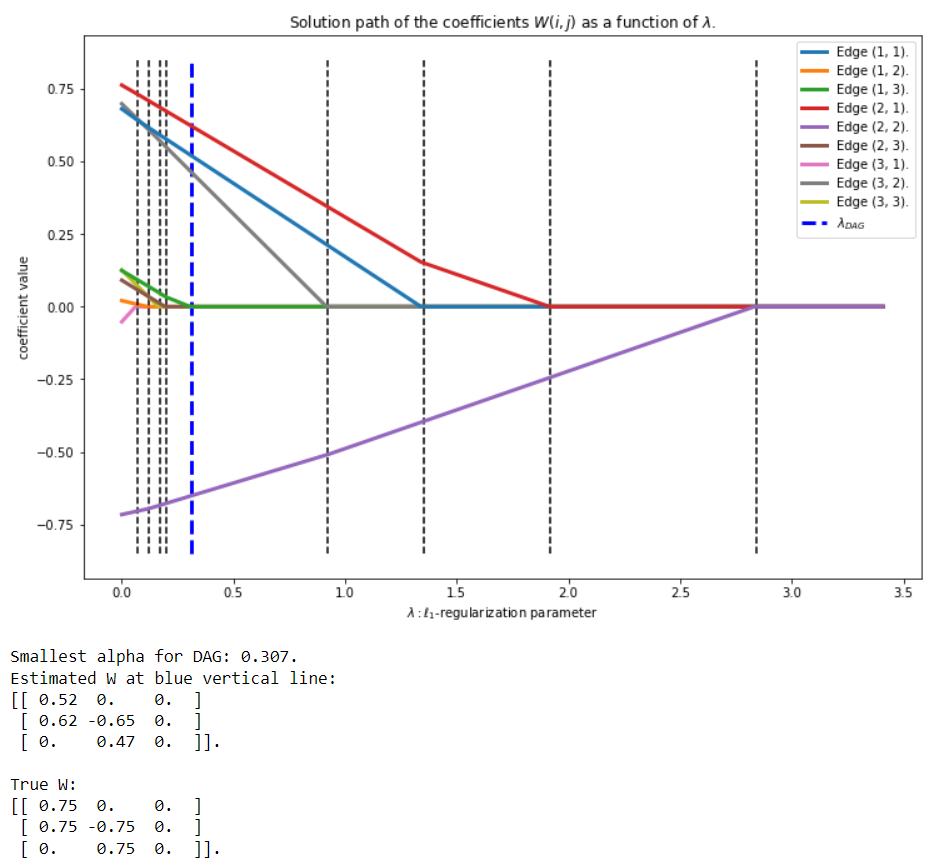
**

Extra: Plotted solution paths for LASSO:



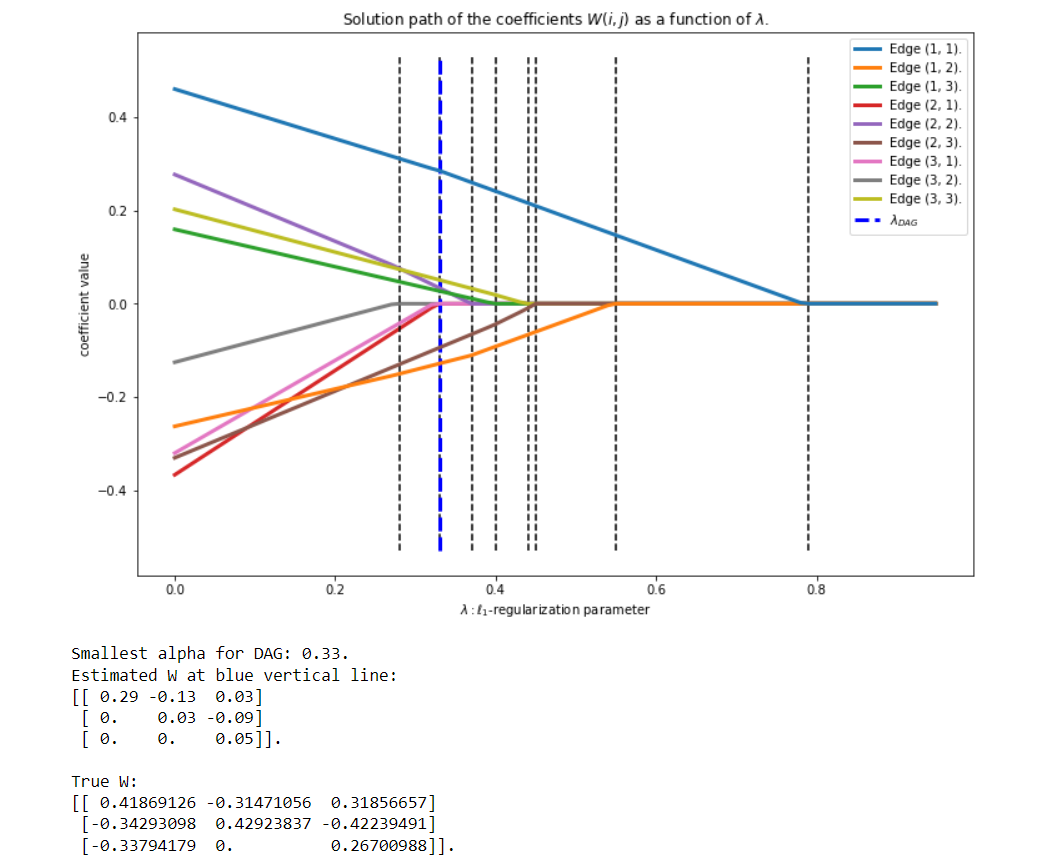
We see that the strongest four edges are the last ones to disappear, and that three edges (1, 1), (1, 2), and (3, 1) disappear very fast. After that follows edge (2, 1). Edge (2, 2) follows just barely before the first true edge (2, 3). The last three true edges (3, 3), (1, 3) and (1, 1) require a significantly larger lambda. The smallest lambda for which G(W) is a DAG is shown in blue vertical dotted line.

Other Figures: ("X\_s4\_n100\_T3\_random\_matrix\_1")



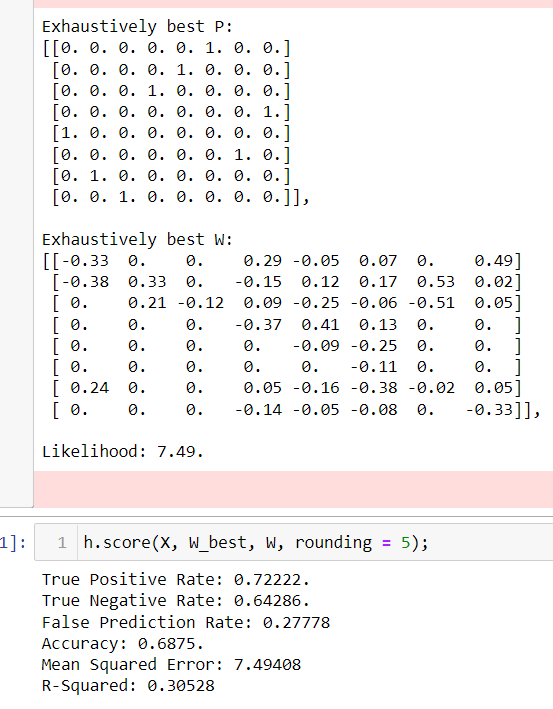
Same Behavior, but a clearer division as the coefficients are larger. We can clearly see a difference between the non-true edges and the true edges in terms of black vertical lines.

Other figure: ("X\_s8\_n100\_T3\_random\_matrix\_0"):



We see that this is less nice, as there is a model mismatch to begin with. Furthermore, the lines cross each other, because some start at smaller values (e.g. orange (1, 2)), but turns out to be quite important and hence it degrades smaller than non-important edges that were larger in absolute value, such as edge (2, 1).

Benchmarks



TODO

OMP: Find cut-off with train / test data, the noise should be cancelled out.

Read paper, find use of indepence, how can we circumvent it / replace it? Our method is not that “depdendent”, only depends on previous timestep, perhaps Martingale.

Investigate model mismatch / DAGness was not true in data generation.

Writing.

Write Update for Rui.

Future:

Nonlinear violations (with respect to data generation).

Kernel Matching Pursuit.

Neural Networks.