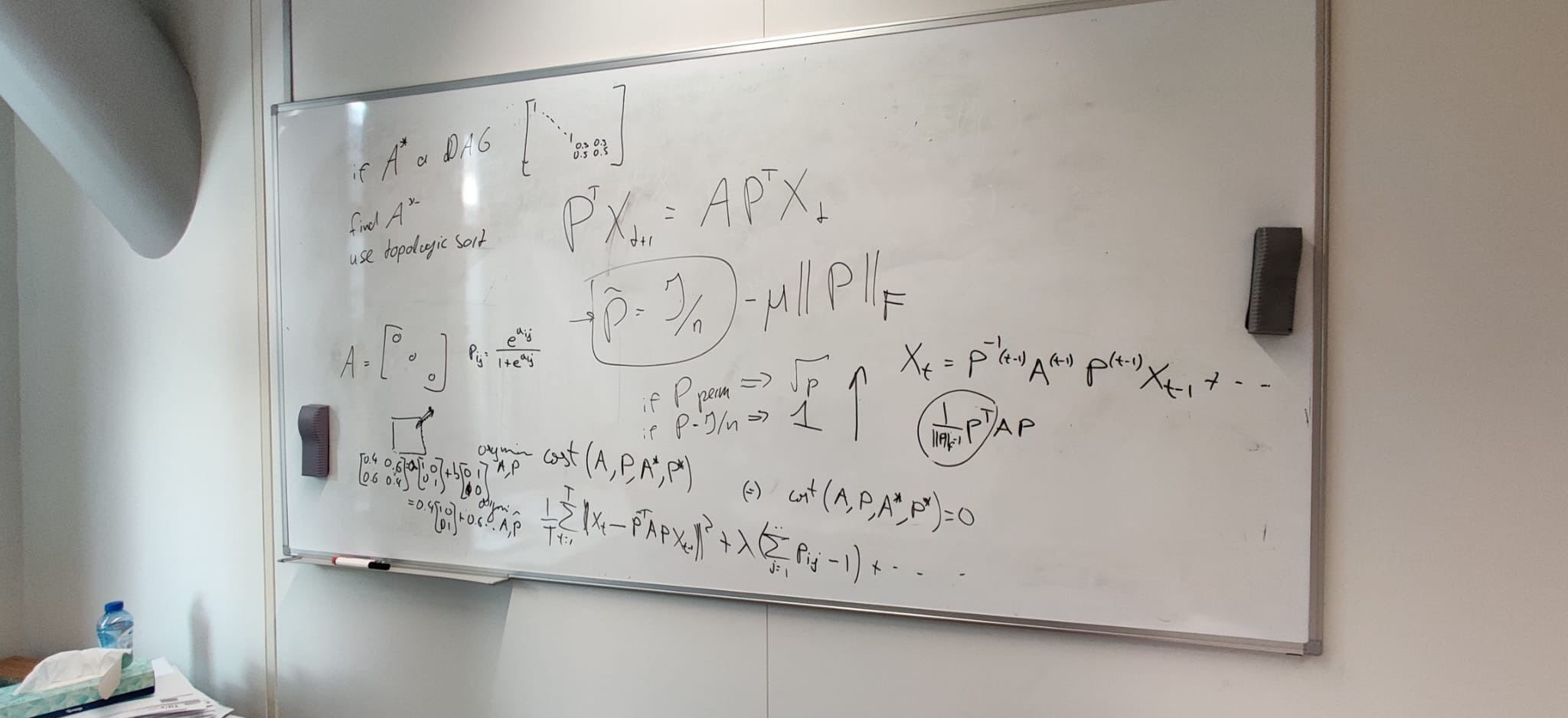
Meeting 9 Prep

Interesting discussions, discussed the paper where they use PX = APX. Also discussed that the problem is easy for VARs in the population setting (when A\* is DAG), by simply estimating A\* with A (which should by consistency be equal) and then doing topological sort.

Interesting things to try out for next time:

* EM algorithm! E-step: Impute P. M-step: Maximize A.
* Project p as a sigmoid to prevent getting out of doubly stochastics.
  + Lambda that they must sum to one.
* Use a penalty like 1 over the frobenius norm minus 1.
* Write DS matrix as convex combination of permutation matrices.
  + Consequently, learn these parameters.
* Write down some derivations stuff in latex, write literature summaries.



# Write DS matrix as convex combination of permutation matrices.

* Done, algorithm not so difficult, but requires iterating over all permutation matrices in the worst case. Came up with a way to make it significantly faster. Works for 100s of nodes in seconds.
* Is not unique, e.g. three dimensional case
  + P\_DS:

[[0.5 0.3 0.2]

[0.2 0.5 0.3]

[0.3 0.2 0.5]]

* + Two solutions (see notebook).
* Furthermore, the decomposition seems to make sense.
* Learning the coefficients seemed to work okay for four dimensions, but for more the results were not that good. Many problems arise from the fact that even an incorrect permutation can reach a very close to optimal score, e.g. 3.01 rather than 3.0000.

# Project P as sigmoid

Interesting idea, rather than *constrain* p\_{ij} to be between 0 and 1, we can also *project* our estimates of p, for example by using the *sigmoid* function. The advantages is that we can let our estimates of p be *unconstrained*, but a disadvantage might be numerical stability issues. The only constraints be *require* is that each row must sum to one, and each column must sum to one.

This has been prototyped, but the results are not ideal. Many times, the solver arrives at numerical stability issues as the Hessian(?) is non-singular. Furthermore, when it finds a solution, it is almost always not the one we want. Sporadically (and with that I mean 1400 attempts).

So, the idea is good, it reduces it to an ECP, but a more sophisticated method than the scipy minimize approach might be necessary.

# EM-Algorithm

Derived log-likelihood for a VAR(1) model.

Wrote down E and M step formally. However, what is it?

Sample