15.095 - Final Project

Robust Expectation-Maximization

Aniruddha Deshpande[1] and Alex Birch[1]

[1]Massachusetts Institute of Technology

December 6, 2021

1 Introduction - EM algorithm

The Expectation-Maximization(EM) algorithm is a method to find the maximum likelihood estimate of model parameters specifically for the case where only part of the data is observable and the rest is not. It is most commonly used to estimate parameters for Gaussian mixture models (GMM) and Hidden Markov models (HMM). Applications of GMMs involve soft-clustering, and HMMs are used for modelling in a variety of applications ranging from speech recognition to financial models. The general theoretical framework for the EM algorithm is as follows. Let $z \sim P_Z(z;\theta)$ be the complete data. $y \sim P_Y(y;\theta)$ is the observed data. The underlying assumption here is that $y = g(z)$ ie. it can be deterministically obtained if z is known. However since only y is observed our aim is to find θ, the set of parameters that maximize the log-likelihood of the observed data. So, we have:

$$\theta_0 = argmax_{\theta\in\Theta} log P_Y(y;\theta) \tag{1}$$

Using Baye's rule and the fact that y is a deterministic function of z, we and taking expectation on either side we get the following.

$$\log P_Y(y;\theta) = U(\theta,\theta') + V(\theta,\theta') \tag{2}$$

where:

$$U(\theta,\theta') = E_{P_{Z|Y}(.|y;\theta')}(P_Z(z;\theta)) \tag{3}$$

$$V(\theta,\theta') = -E_{P_{Z|Y}(.|y;\theta')}(P_{Z|Y}(z|y;\theta)) \tag{4}$$

Using Gibbs inequality and some simplification, we get the following inequality which justifies the core of the EM algorithm.

$$\log P_Y(y;\theta) \geq (U(\theta,\theta') - U(\theta,\theta')) + \log P_Y(y;\theta') \tag{5}$$

This inequality shows that for any current parameter estimate θ', if we find θ such that $U(\theta,\theta') \geq U(\theta,\theta')$, then the log-likelihood of y with the new θ is greater than that with θ'. The EM algorithm is then given by :-

1. Initialise parameter estimate: θ^0
2. E-step:

$$U(\theta,\theta^i) = E_{P_{Z|Y}(.|y;\theta^i)}(P_Z(z;\theta)) \tag{6}$$

3. M-step:

$$\theta^{i+1} = argmax_{\theta\in\Theta} U(\theta,\theta^i) \tag{7}$$

The key idea of this project is to change the M-step and make it robust/optimistic to noise in the observed data. The M-step in our re-worded algorithm would be.

- Robust M-step:

$$\theta^{i+1} = argmax_{\theta\in\Theta} min_{y\in U} U(\theta,\theta^i) \tag{8}$$

- Optimistic M-step:

$$\theta^{i+1} = argmax_{\theta\in\Theta} max_{y\in U} U(\theta,\theta^i) \tag{9}$$

2 Hidden Markov model

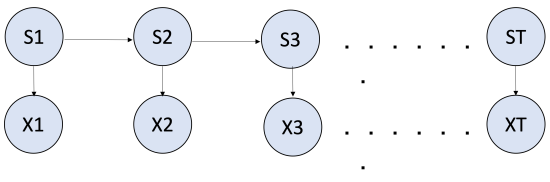

Figure 1: HMM Structure showing the sequence of states at each iteration T

In this project we'll be applying the modified EM algorithm in the context of HMMs. The specialized EM algorithm in this case is also called

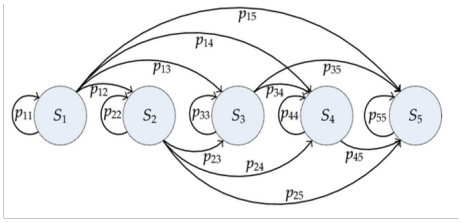

Figure 2: HMM Structure showing state transition probabilities[1]

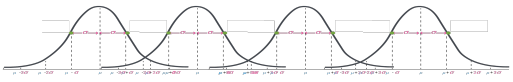

Figure 3: Plot showing the superposition of 1D Gaussian distributions from the HMM emissions[2]

the Baum-Welch algorithm. An HMM has structure as seen in Figs. 1 and 2. At each time step the system is in a particular state $S^t = i$ for $i \in [m]$. For each possible state i there is a unique emission distribution $b_i(.)$ corresponding to it. The observed data X_t at time t is then sampled from the distribution $b_i(.)$ depending on the current state, this results in a superposition of the distributions as shown in Fig. 3. There are 2 sets of parameters corresponding to the HMM, namely $\mathbf{A} \in R^{mxm}$ the transition probability matrix and $\pi \in R^m$ the initial state probabilities. $A_{ij} = P(S_{t+1} = j | S_t = i)$ is the probability of the next state being j given the current state is i. π_i refers to the probability that the initial state of the system will be i. We will be working with a specific case of the HMM model, where the emission distribution $b_i()$ is a normal distribution with a different mean and co-variance depending on the state, ie. $b_i(.) = \mathcal{N}(\mu_i, \Sigma_i)$. Comparing with the analysis in the introduction section, we have observed data $Y = X$ and complete data $Z = (X, S)$ For our analysis we assume that the emission distributions for each state are known, and we are only trying to learn the state transition probability matrix and the initial probabilities. We first develop the modified EM algorithm for the 1D case for simplicity and then generalize it to the multi-dimensional case.

3 Baum-Welch - EM algorithm for HMM parameter estimation

The problem setup is as follows. We have an observed sequence $(x_1, x_2..x_T)$. Corresponding to this is a sequence of unobserved hidden states $(s_1, s_2, ..s_T)$. Before developing the algorithm for this case and modifying it for robustness, let us first define some useful quantities for going forward.

$$\alpha_j(1; x_1) = \pi_j b_j(x_1), \forall j \in [m] \tag{10}$$

$$\alpha_j(t, x_1..x_t) = \sum_{i=1}^{m} \alpha_{t-1}(i) A_{ij} b_j(x_t), \forall j \in [m] \tag{11}$$

$$\beta_i(T) = 1, \forall i \in [m] \tag{12}$$

$$\beta_i(t, x_{t+1}..x_T) = \sum_{j=1}^{m} \beta_j(t+1) A_{ij} b_j(x_{t+1}), \forall i \in [m] \tag{13}$$

These quantities are computed in each iteration in the E-step of the algorithm. At the $(l+1)^{th}$ iteration it uses the parameter estimates θ^l from the previous step to compute these quantities. This includes the choice of TPM and initial probabilities. This computation is done recursively using the forward-backward algorithm (ADD reference here).

Now, following the framework provided in the introduction and [3]:

$$\begin{aligned} U(\theta, \theta') = &\sum_{i=1}^{m} \alpha_i(1; x_1) \beta_i(1; x_1..x_T) \log \pi_i b_i(x_1) \\ &+ \sum_{i=1}^{m} \sum_{t=1}^{T-1} \log(b_i(x_{t+1})) \alpha_i(t+1; x_1..x_{t+1}) \beta_i(t+1; x_{t+2}...x_T) \\ &+ \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{t=1}^{T-1} log(A_{ij}) [\alpha_i(t; x_1..x_t) A'_{ij} b'_j(x_{t+1}) \\ &\beta_j(t+1; x_{t+2}...x_T)] \end{aligned} \tag{14}$$

This formulation is for learning parameters over one sequence. It can be easily generalized to the case of multiple sequences by re-writing the same thing for each sequence and adding it up. The regular EM algorithm then proceeds by maximizing this function over pi_i, A_ij and the parameters of $b_j(.)$ respectively for all states.

4 Robust M-step

We now develop the robust algorithm for the general case. We want to solve the following robust optimization problem.

$$max_{\theta \in \Theta} [min_{\Delta x \in U} U(\theta, \theta')] \tag{15}$$

First, we note that a number of occurrences of the observed data terms x_t in $U(\theta, \theta')$ occur within the α and β terms. Trying to minimize over x in these functions would be very difficult as they are computed recursively and are highly non-linear. Instead we use an approach similar to the parameter update for the regular EM algorithm, where the α and β terms are computed using the parameter (and Δx) estimates from the previous iterations. To make this clearer, re-writing equation 14.

$$U(\theta, \theta', \mathbf{x}, \mathbf{x'}) = \sum_{i=1}^{m} \alpha_i(1; x'_1)\beta_i(1; x'_1..x'_T) \log \pi_i b_i(x_1)$$
$$+\sum_{i=1}^{m}\sum_{t=1}^{T-1} \log(b_i(x_{t+1}))\alpha_i(t+1; x'_1..x'_{t+1})\beta_i(t+1; x'_{t+2}...x'_T)$$
$$+\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{t=1}^{T-1} log(A_{ij})[\alpha_i(t; x'_1..x'_t)A'_{ij}b'_j(x_{t+1'})$$
$$\beta_j(t+1; x'_{t+2}...x'_T)] \quad (16)$$

and we minimize this over **x** and assume the **x'** to be constant.

The robust algorithm then becomes.

- **Initialize:** θ^0, $\Delta x = 0$
- for iter=1:num-iters
  - E-step: Compute α and β using parameter estimates and Δx from the previous iteration.
  - M-step 1: Inner minimization
  $$x_{next} \longleftarrow argmin_{\mathbf{x}\in U} U(\theta, \theta', \mathbf{x}, \mathbf{x'}) \quad (17)$$
  - Re-compute α and β terms using x_{next}
  - M-step 2: Outer Maximization
  $$\theta_{next} \longleftarrow argmax_{\theta\in\Theta} U(\theta, \theta') \quad (18)$$
  - $x' \longleftarrow x_{next}$
  - $\theta' \longleftarrow \theta_{next}$

For the optimistic EM algorithm, the inner minimization is replaced by an inner maximization instead. Note: It is not clear just from this formulation as to what kind of convergence properties this algorithm would have.

5 Formulating robust M-step for HMM with 1D Gaussian emissions

Now we specialize the above for the case where the emission distributions are $b_i(.) = \mathcal{N}(\mu_i, \sigma_i)$. Substituting the normal distribution pdf into the equation 16 and simplifying the terms we get the following equation.

$$U(\theta, \theta', x, x') =$$
$$\sum_{i,j=1,1}^{m,m}\sum_{t=1}^{T-1} \log A_{ij}\alpha_i(t)A'_{ij}b'_j(x'_{t+1})\beta_j(t+1)$$
$$+\sum_{i=1}^{m} \alpha_i(1)\beta_i(1)log(\pi_i)$$
$$-\sum_{i=1}^{m}\sum_{t=1}^{T} \frac{1}{2}\alpha_i(t)\beta_i(t) \log 2\pi\sigma_i^2$$
$$+\sum_{i=1}^{m}\sum_{t=1}^{T}(-\frac{\alpha_i(t)\beta_i(t)}{2\sigma_i^2})(x_t - \mu_i)^2 \quad (19)$$

Note that to optimize over x, the only term that matters is the last one (we ignore the first term as x appears in a term that is recursively computed in it).

$$\min_{|\Delta x_t|\leq r; \forall t\in[T]} \sum_{i=1}^{m}\sum_{t=1}^{T}(-\frac{\alpha_i(t)\beta_i(t)}{2\sigma_i^2})(x_t+\Delta x_t-\mu_i)^2 \quad (20)$$

So, the inner minimization problem (or maximization) is actually a negative quadratic objective function subject to the constraints of the uncertainty set. This makes the objective non-convex. We choose the uncertainty set such that

$$||\Delta x_t||_2^2 \leq r^2; \forall t \in [T] \quad (21)$$

In the 1D case this simplifies to a linear constraint and the inner optimization problem is solved pretty fast by optimizer like Gurobi in each step. We do this inner optimization, replace the value of x used and then follow the standard EM algorithm parameter update rule for M-step 2 [3].

For the multi-dim Gaussian case, the inner optimization will still be a quadratic function and should be simple enough to optimize over.

6 Dataset

We evaluate our robust EM algorithm on a synthetic dataset that we generated. We used the HMMBase Package in Julia to generate it. The true HMM we used has 5 states, 1D Gaussian emissions corresponding to each state and randomly chosen true initial probabilities π and state transition probability matrix A. We use 10 sequences, each of length 500. We use all emission distributions having variance 1 and means [0,2,4,6,8]. We also generate different noisy versions of this data by adding white noise of mean

0 and variances from 0.1 to 1 in steps of 0.1. This is to evaluate the performance of our algorithm in the presence of noise, and how the amount of noise affects relative performance. Throughout our experiments going forward we assume the emission distributions to be known, and aim to estimate the initial probability vector and state transition probabilities from the data.

7 Evaluation metrics

We use the following evaluation metrics to compare performances of the standard EM algorithm with the robust and optimistic versions of the algorithm for varying uncertainty sets.

- $D(A_{true}||A_{est})$ - Average KL divergence between rows of the true state transition matrix with the rows of the estimated matrix. This makes sense because each row i corresponds to a probability distribution, which is the probability of moving to state j in the next timestep, given current state is i. This is essentially a distance metric between the true A matrix and the estimated A matrix. Smaller value indicates better performance.
- **State estimation accuracy** Given the parameters of the HMM (A and π), the Viterbi algorithm can be used to decode the HMM for a sequence of observations, and estimate what the underlying state sequence would be. We run this for all the different parameter estimates and compare the estimated sequence with the true underlying sequence, and report the accuracy as the fraction of states estimated correctly to the total number of states. This metric measures how well the estimated parameters do at the final end task of an HMM which is to estimate the states. Larger value indicates better performance.

8 Experiments

We did the following experiments to assess the performance of the modified EM algorithm.

- Run the standard EM algorithm and get parameter estimates for all sets of noisy data
- Run the robust and optimistic EM algorithms for all noisy data with uncertainty parameter r taking values in [0.01, 0.02, 0.05, 0.1, 0.5] and obtain parameter estimates for each case
- Compare parameters obtained in all cases with the true underlying parameters using the KL divergence metric described in the previous section
- Compare State estimation accuracy for all cases
- Compare State estimation accuracy for all cases on corresponding test datasets
- Do above runs for 5 different random initializations and report average scores over random initializations (Ran out of time to do this :( )
- Timing analysis to compare slow-down caused due to additional inner optimization for the robust and optimistic case.

9 Results

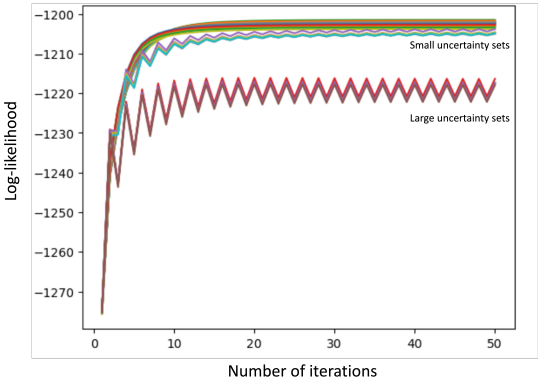

Figure 4: Plot showing the log-likelihood convergence of the robust model.

Fig. 4 shows the model converging to a maximum log-likliness. Notice the periodic nature of this, which grows with the size of the uncertainty set.

Fig. 5 shows a heatmap of our training set results for the robust EM methods for the state estimation accuracy evaluation metric. Fig. 7 shows a heatmap of the parameter estimation error for the models. From these we observe a relationship between the performance and the noise, with the performance uniformly decreasing with the size of the noise as expected. We also observe that for large values of r, the performance also drops off, potentially indicating that the size of the uncertainty set is too large, making the robustness too conservative.

Also shown are line plots to show how uncertainty set size affects the performance metrics for a particular noise level, seen in Figs. 6 and 8. From these we observe *sweet-spots* in the performance, where at a given noise and a given r the robust model outperforms the baseline.

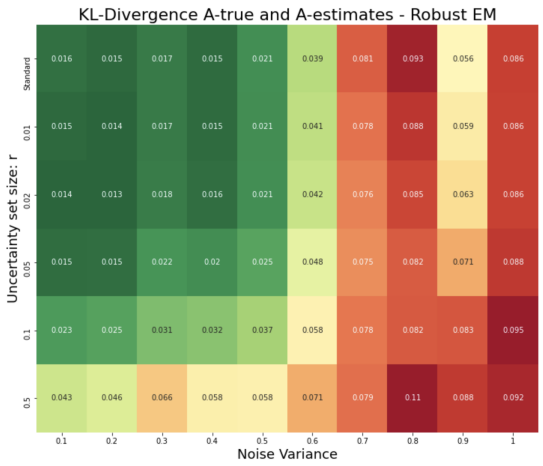

Figure 5: Heatmap showing the robust EM methods state estimation performance against noise and uncertainty set size, on the training data.

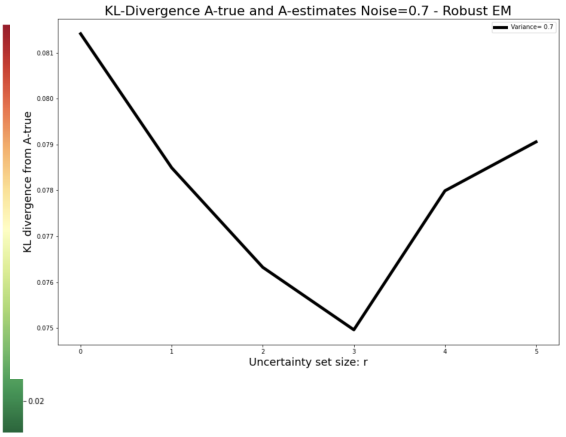

Figure 6: Plot showing the robust EM methods state estimation performance against uncertainty set size for noise variance = 0.7, on the training data. Here we observe a *sweet-spot* at which the robust method improves over the standard method

Figs. 9 and 10 show the State Estimation accuracy on the test set. There are similar observations to be made here as in the training set. Seeing this behaviour on the test set also indicates that this method does in some cases give better out of sample state estimation accuracy than the standard EM algorithm.

Fig. 11 shows a heatmap of our training set results for the optimistic EM methods for the state estimation accuracy evaluation metric. Fig. 13 shows the parameter estimation error ($D(A_{true}||A_{est})$)) in the models. For this method the KL-divergence metric does not improve as compared to the standard EM algorithm From these we also observe the clear relationship between the performance and the noise, as for robust. We also observe that the state estimation accuracy metric does show improvement in some cases as compared to the standard method, in the training set as well as the test set. For large values of r, the performance drops off, as with robust.

Also shown are line plots to show how uncertainty set size affects the performance metrics, seen in Figs. 12 and 14. From these we observe *sweet-spots* in the performance, where at a given noise and a given r the optimistic model outperforms the baseline.

Figs. 15 and 16 show the State Estimation accuracy on a test set. Here we observe many of the same results as in the training set.

We also observe the scalability of these methods in Table 1. Here we see the time complexity of the different approaches. From this we observe that although the robust approach takes more time at these lower complexities, it actually scales better than the standard approach and the optimistic approach with the number of states, and comparably well with the number of sequences.

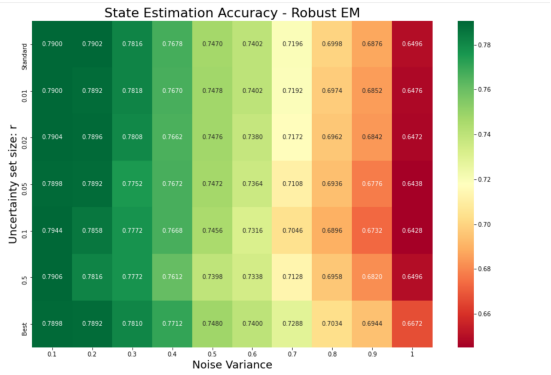

Figure 7: Heatmap showing the robust EM methods performance on the KL divergence from the true A parameter, as a function of noise and uncertainty set size

# States	Sequence Length	# Sequences	Standard (s)	Robust(s)	Optimistic (s)
5	500	10	8.62	134.42	33.32
10	500	10	20.75	186.68	59.35
20	500	10	96.63	369.91	150.81
2	100	5	2.69	32.56	2.12
2	200	5	3.34	56.07	3.36
2	400	5	1.57	70.89	5.08
2	50	10	0.51	19.72	2.20
2	50	20	0.83	38.48	3.78
2	50	40	1.40	73.05	6.64

Table 1: Time Complexity of Standard, Robust and Optimistic EM method, as we vary the number of states, Sequence length and number of sequences, to show scalability

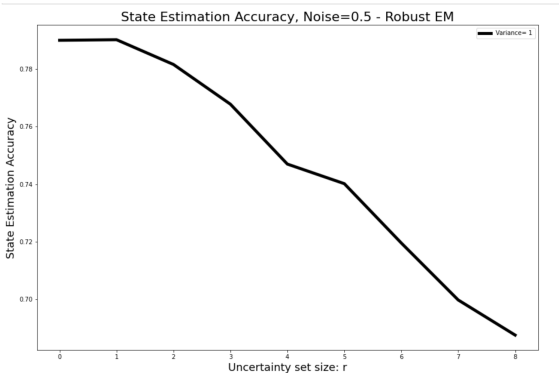

Figure 8: Plot showing the robust EM methods state estimation performance on the KL divergence from the true A parameter, for noise variance = 0.5 Here we observe a *sweet-spot* at which the robust method improves over the standard method.

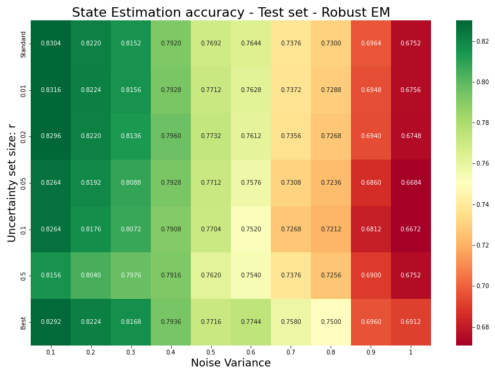

Figure 9: Heatmap showing the robust EM methods state estimation performance against noise and uncertainty set size, on the testing data.

10 Future Work

There are quite a few directions to further explore from here:

- Generalize formulation for the n-dimensional Gaussian emissions case
- Test for case where parameters of the emission distributions are also to be estimated
- Generalize for the case of Mixture of Gaussians emissions. This would be particularly interesting because a lot of interesting applications model emissions as Gaussian mixtures. Eg. Speech recognition

References

[1] North London Collegiate School. "what if i told you that 1=2? the banach-tarski paradox". *[https://nlcshub.com/stem/mathematics/1115/] - accessed: 01/12/21.*

[2] Gaussian distribution image courtasy of calcworkshop,. *[https://calcworkshop.com/exploring-data/normal-distribution/.] - accessed: 01/12/21.*

[3] Jeff A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *CTIT technical reports series*, 1998.

11 Appendix

Contributions to work:

Alexander Birch - Generation of Hidden Markov Chain synthetic data and results/plotting.

Aniruddha Deshpande -Problem formulation, robust/optimistic algorithm formulation and ju-

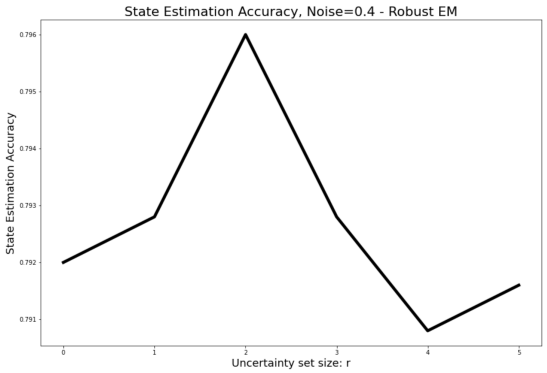

Figure 10: Plot showing the robust EM methods state estimation performance against uncertainty set size for noise variance = 0.7, on the testing data. Here we observe a *sweet-spot* at which the robust method improves over the standard method

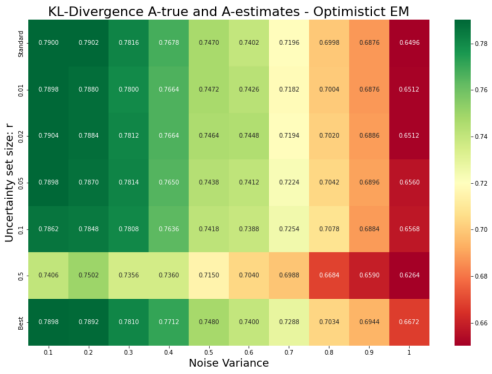

Figure 11: Heatmap showing the optimistic EM methods state estimation performance against noise and uncertainty set size, on the training data.

lia function implementation, evaluation metrics function implementation

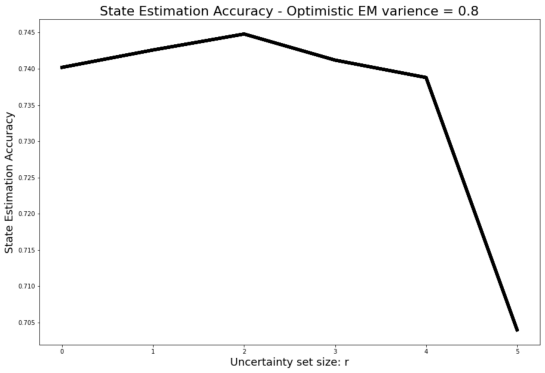

Figure 12: Plot showing the optimistic EM methods state estimation performance against uncertainty set size for noise variance = 0.7, on the training data. Here we observe a *sweet-spot* at which the optimistic method improves over the standard method

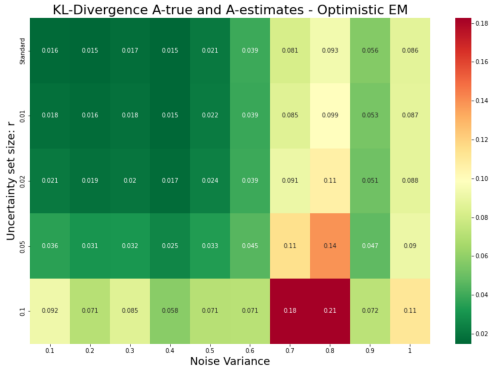

Figure 13: Heatmap showing the optimistic EM methods performance on the KL divergence from the true A parameter, as a function of noise and uncertainty set size, on the training data.

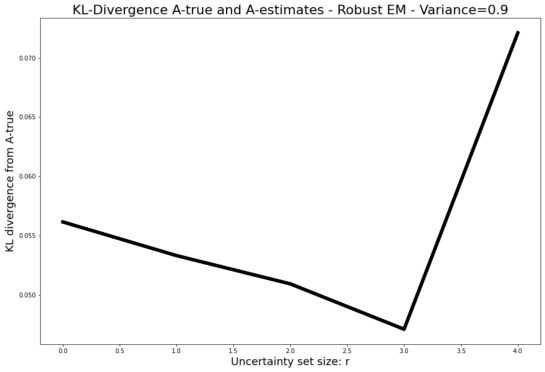

Figure 14: Plot showing the optimistic EM methods state estimation performance on the KL divergence from the true A parameter, for noise variance = 0.5, on the training data. Here we observe a *sweet-spot* at which the optimistic method improves over the standard method.

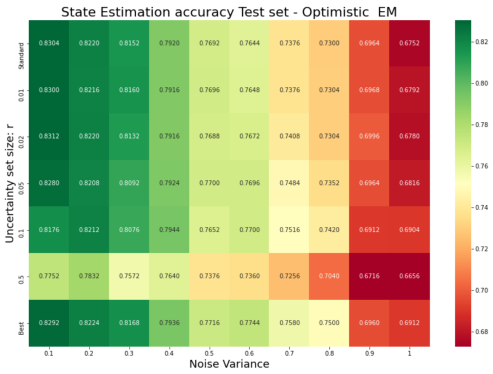

Figure 15: Heatmap showing the optimistic EM methods state estimation performance against noise and uncertainty set size, on the testing data.

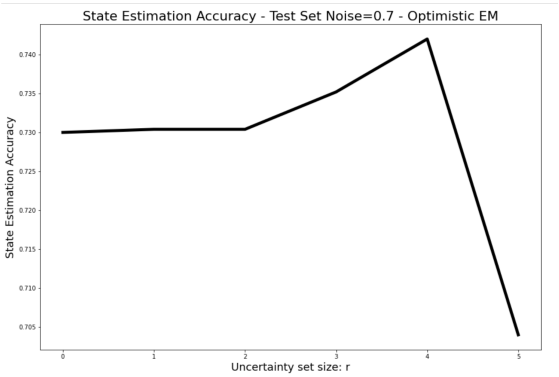

Figure 16: Plot showing the optimistic EM methods state estimation performance against uncertainty set size for noise variance = 0.7, on the testing data. Here we observe a *sweet-spot* at which the optimistic method improves over the standard method