CS229 PROJECT - PHYSICAL SCIENCES
MARKOV MODEL IN TIME FOR TRANSPORT IN POROUS MEDIA

Amir Hossein Delgoshaie (SUid: amirdel), Yang Wong (SUid: zhiyangw) *December 14, 2016*

1 Introduction

Transport in porous media is a highly uncertain physical phenomena. Modeling this type of flow regime is important in a wide range of fields such as contaminant flow, ground-water pollution or flow in fractures from fracking. However all of these physical problems suffer from a great deal of uncertainty. This uncertainty stems from the lack of data of the governing parameters of the porous medium. More specifically diffusion type models have been developed for this problem. However due to the uncertainty in the model parameters the diffusion coefficient is considered to be a random field, i.e.

$$u = -\frac{K(x,y)}{\mu}\nabla p$$
$$\nabla . u = q$$

where $K(x,y)$ is permeability, which describes the geology of the model, μ is the fluid viscosity, u is the fluid velocity and $p(x,y)$ is the pressure field. Since $K(x,y)$ is highly uncertain, many realizations of a geological model are used to describe the statistics of the contaminant flow in a Monte Carlo framework. This current physics-based framework to model each realization is computationally expensive, thus not suitable for large problems that are of practical interest. This project aims to use machine learning to replace the physics-based part of the workflow.

This physical model is used to generate training data for our statistical model. More specifically, we consider many realizations of a smaller physical domain that is feasible to model. Then we track contaminant fluid particles in those realizations by evolving their velocity using the resulting velocity field in the physics-based model. The velocity process for each particle serves as our training data. We then train a Markov model that captures the physics of the velocity evolution. Using this procedure we can easily produce at least 10^6 velocity transitions to train the model. This idea has been previously tested for velocity processes [2, 3] and shown to be very promising. The key question to answer is how to compute the transition probabilities for the Markov model. In this work, we model these transition probabilities using both transition matrices (using discrete states) and kernel density estimation (KDE).

2 Problem description

In this project we analyzed velocity statistics from particles released in different realizations of a random network. The studied network are structured networks where the transmissibility between two nodes is a random variable with a log-normal distribution [5]. A schematic of such network is shown in Fig. 1. This is representative of the pore-network in a typical geological rock. In each realization of the network, multiple particles are released at the center of the left boundary. The particles follow the flow in the network using the following rules: 1- at each node the particle can choose from all the links connected to nodes with lower pressure values. 2- The probability of choosing a link is proportional to the flow in that link.

The raw attributes generated from this particle tracking procedure are $x_n^{(i)}$ and $y_n^{(i)}$ and $t_n^{(i)}$, where $x_n^{(i)}$ and $y_n^{(i)}$ are the position of particle i after n transitions and $t_n^{(i)}$ is the elapsed time to get to that position. Given these trajectories for all particles we can obtain the concentration map for the contaminant at a given time less than $t_{max} = \min_i \max_n t_n^{(i)}$, which is the largest time where we have the position of all particles. Alternatively, one can obtain the distribution of $t_h(\mathbf{X})$, the time required for the contaminant particles to reach a certain location $\mathbf{X}$. This

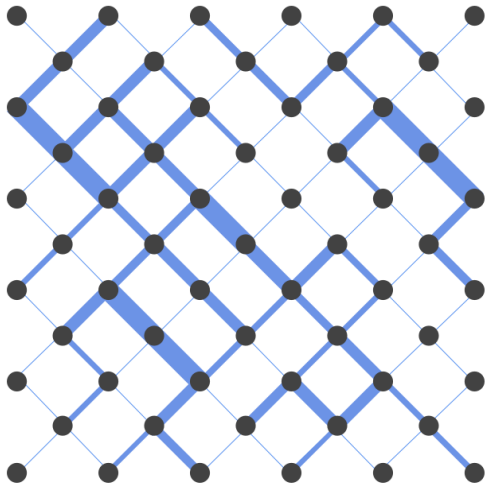

Figure 1: Schematic of the random network. Adapted from [2].

is also commonly known as the breakthrough curve. This is a commonly used statistics in the hydrology industry to examine the flow of a particular contaminant. However, finding the statistics of t_h for positions far away from the injection well is computationally very expensive using physics-based simulations. Our goal here is to train a Markov model that can efficiently predict $t_h(\mathbf{X})$ were $\mathbf{X}$ is further than the points observed in the training set. This will allow us to make predictions of particle positions as a Markov model rather than the typical computationally expensive physics simulations.

To achieve this goal, we assume the velocity is stationary and model the velocity of the particles, $\mathbf{v}_n^{(i)}$, as a Markov process. One can use the velocities obtained directly from the training data $(x_n^{(i)}, y_n^{(i)}, t_n^{(i)})$.

$$\mathbf{v}_n^{(i)} = [v_x^{(i)}, v_y^{(i)}]_n^T \text{ ,where} \tag{1}$$

$$v_x^{(i)} = \frac{x_{n+1}^{(i)} - x_n^{(i)}}{t_{n+1}^{(i)} - t_n^{(i)}}, \tag{2}$$

$$v_y^{(i)} = \frac{y_{n+1}^{(i)} - y_n^{(i)}}{t_{n+1}^{(i)} - t_n^{(i)}}. \tag{3}$$

A Markovian model based on these velocities would allow us to efficiently march particles with length steps equal to the length of the links, l, in the underlying simulations. However l is typically much smaller than scale of distances of practical interest. A key idea used here to further improve computational efficiency is to combine multiple velocity transitions together. This is done by averaging the velocity over time steps Δt_s which we choose as the stencil time. The stencil time is chosen such that it is at least one order of magnitude larger than $\overline{\delta t}$, which is the median time required for one velocity transition. We choose the median time as a time scale to have a measure that is not sensitive to the outliers in the transition time data.

3 Methodology

The time averaged velocity processes obtained from the raw attributes are first converted to polar coordinates.

$$\overline{\mathbf{v}}_n^{(i)} = [v_n^{(i)}, \theta_n^{(i)}], \tag{4}$$

where $v_n^{(i)}$ is the magnitude of the velocity vector and $\theta_n^{(i)}$ is the direction angle. What remains is to estimate the transition probability $(P(\overline{\mathbf{v}}_{n+1}^i | \overline{\mathbf{v}}_n^i))$ between different v and θ pairs. Once we have the transition probabilities and initial velocities, we can model the movement of these particles as a Markov process and predict the same concentration maps and other results from a physics based simulator but for a significantly smaller computational cost.

We estimate the transition probabilities using discrete transition matrices and kernel density estimation. For each method we perform simulations and calculate the prediction error in two settings: 1) we use all the available data and test on the training set and 2) we use only 20 percent of velocity time series available to us for training the model and we perform testing on later times.

3.1 Modeling using a discrete transition matrix

In this method we estimate the transition probability using discrete bins for v and θ. The observed velocity vectors are then binned into n_v velocity and n_θ angle bins. To further differentiate between the low and high velocity values, bins for the velocity are constructed using $\log(v^{(i)})$. The velocity bin edges are chosen such that each bin has approximately equal probability. To capture the effect of very low and high velocity values the velocity classes corresponding to these values are further refined. The angle is discretized linearly. Here we use $n_v = n_\theta = 100$.

The state space for the Markov model is then defined as $S = \{(v_j, \theta_k)|\ 1 \leq j \leq n_v, 1 \leq k \leq n_\theta\}$. The transition probabilities for this model are estimated by counting the transitions from velocity classes in the training data. This corresponds to the maximum likelihood estimation of the model parameters.

3.1.1 Testing the Markov Assumption

Before using these transition probabilities to predict the contaminant flow, the Markov assumption of our model was first tested. The discrete version of the Chapman-Kolmogorov equation [1], is verified for lag five transitions. Here we test how well the Markov assumption can predict transitions with lags greater than one. Fig. 2 shows a comparison between the transitions probability from the model and the observed probability from the data. We performed this test for other velocity and angle classes. The results confirm that our assumption for using a Markov model is indeed consistent with the patterns observed in the data.

3.1.2 Plume Location Prediction

In this section, the transition matrix and the initial velocity distribution obtained from the training data were used to simulate the evolution of the contaminant plume. The results presented in this section are for the case of $\Delta t_s = 10\overline{\delta t}$.

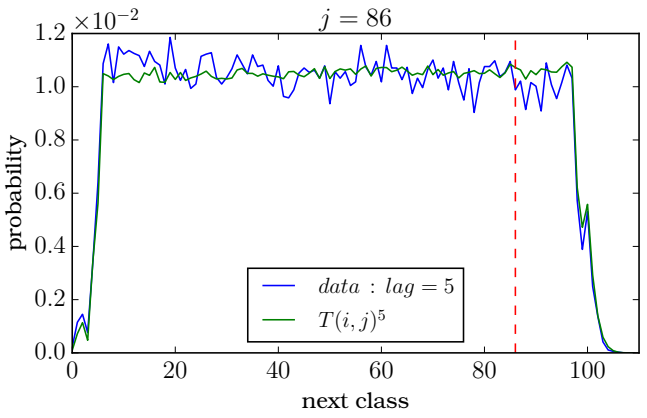

Figure 2: Probability of the velocity class after five transitions from a fast initial velocity class.

This corresponds to averaging over periods that are ten times the transition time scale. The simulations are also performed in two dimensions. Fig. 3 and Fig. 4 show the comparison between the contaminant plume and the results obtained from the Markovian model in 2D. The Markov model shows a very close agreement with the data. It specifically captures important features such as the position of leading and trailing edges of the plume very well. In order to better visualize and compare the plume evolution, in Fig. 5 we plot the projection of the plume on the x axis. For the results presented in this section, the model is trained on the entire data set, using transition data from all parts of the domain at all time. We observe that the error on the training set is very small.

3.2 Modeling using kernel density estimation

In this section we describe the kernel density estimation (KDE) method to estimate the transition probability from one velocity value to another. Using the fastkde package [4], the transition probabilities were obtained much faster compared to the discrete method and we do no need to make any prior definitions for binning the data. This reduces the amount of user-defined input required for the training of the data.
In order to test our method, we use data only from

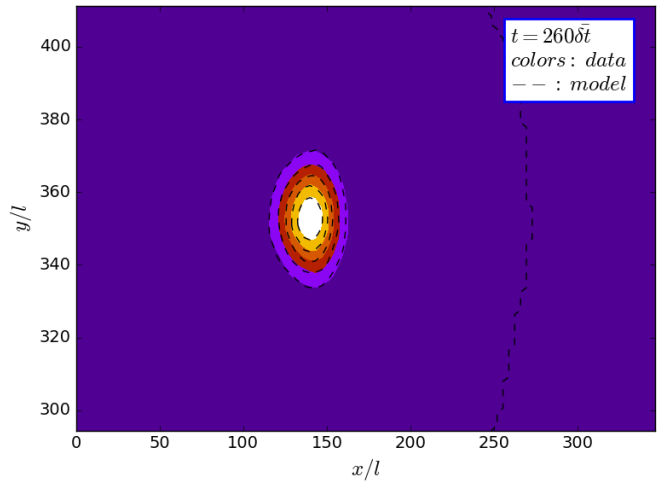

Figure 3: 2D Comparison of the contaminant plume evolution prediction: discrete model vs. data. Testing performed on the raining set.

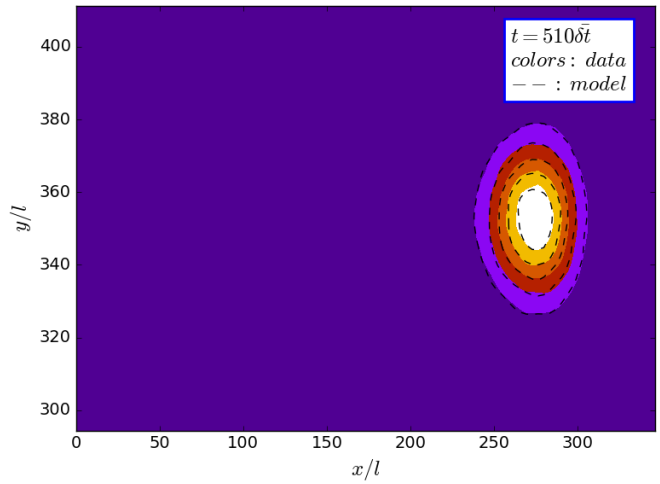

Figure 4: Same as Fig. 3 but for a later time.

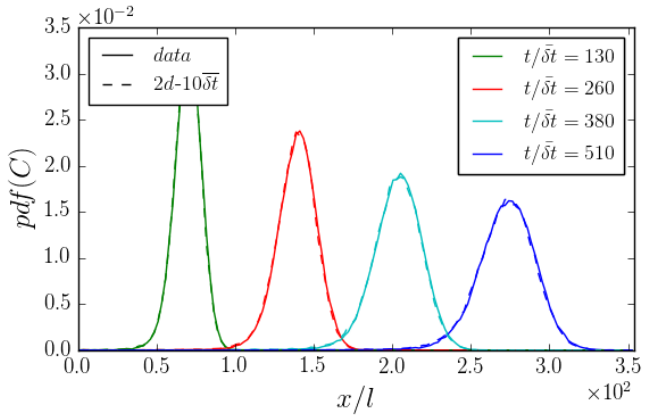

Figure 5: Comparison of the contaminant plume evolution predicted by the discrete model and the data, projected on the x axis. Testing performed on the training set.

the breakthrough time.

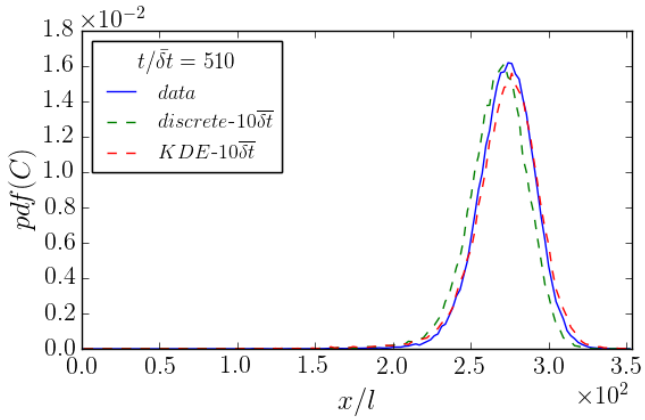

Figure 6: Comparison of the contaminant plume evolution projected on the x axis: KDE vs. discrete model for a time significantly larger than the test period.

20 percent of the time were the fastest particle is still in the domain to train our model. This corresponds to using points with $t_n^{(i)} \leq 0.2t_h(L)$, where L is the domain length. Fig. 6 shows the comparison of the plume obtained with the KDE model and the discrete model after $580\overline{\delta t}$. The breakthrough time for $x = 0.75L$ is compared in Fig. 7. These figures suggest that with less data to train on the KDE model performs better in predicting the contaminant plume location and

We define the prediction error as the max norm of the difference between the predicted plume and the data at $t = 510\overline{\delta t}$, divided by the maximum concentration at that time. The test time is significantly larger than the training period ($0 \leq t \leq 120\overline{\delta t}$). Table 1 summarizes the errors for both models. We found that both methods have small training error, but the KDE model performs significantly better on the test set. This is consistent

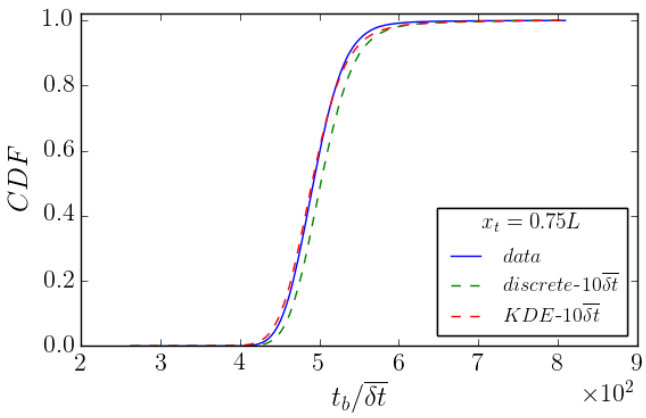

Figure 7: Comparison of the break through time for $x = 0.75L$: KDE vs. discrete model.

with previous results that showed the KDE model makes better predictions on the plume spreading and breakthrough for larger times. The KDE method performs better than the discrete model in the test error, due to the fact it is fitting a function to the probabilities and the continuity between velocity classes is better maintained. This results in better predictions for regimes that the model has not yet seen. As it also has less model parameters to fit, it would suffer from less overfitting. Having these extra parameters for the discrete case is reflected in the slightly better fit for the training data. It is also observed that the performance of the KDE model does not change as much when testing compared to the performance on the training data, whereas we observe a big change in the performance of the discrete model.

	training error	test error
discrete	0.042	0.206
KDE	0.128	0.106

Table 1: summary of test and training error for both models

4 Conclusions

We used a Markovian model in time to study contaminant particle dispersion in a porous media using both discrete transition matrices and kernel density estimation. By averaging multiple particle transitions, our model is able to make prediction more efficiently. Here the model is one order of magnitude faster, since on average we combine 10 particle transitions compared to the previously proposed models. Moreover, We showed that by using only a small fraction of the velocity time series data, we are able to make predictions for significantly later times. These results show that the studied model is indeed very promising in making prediction on large scale problems using limited data from small domains. It was also found that the KDE method was able to predict flow better and obtained a lower testing error than the discrete method.

5 Acknowledgements

The authors acknowledge professor Hamdi Tchelepi from the Energy Resources Engineering department for helpful discussions.

References

[1] Crispin W Gardiner et al. *Handbook of stochastic methods*, volume 3. Springer Berlin, 1985.

[2] Peter K Kang, Marco Dentz, Tanguy Le Borgne, and Ruben Juanes. Spatial markov model of anomalous transport through random lattice networks. *Physical review letters*, 107(18):180602, 2011.

[3] Daniel W Meyer and Hamdi A Tchelepi. Particle-based transport model with markovian velocity processes for tracer dispersion in highly heterogeneous porous media. *Water Resources Research*, 46(11), 2010.

[4] Travis A. O'Brien, William D. Collins, Sara A. Rauscher, and Todd D. Ringler. Reducing the computational cost of the ecf using a nufft: A fast and objective probability density estimation method. *Computational Statistics and Data Analysis*, 79:222 – 234, 2014. ISSN 0167-9473. doi: http://dx.doi.org/10.1016/j.csda.2014.06.002.

[5] JE Warren, P Jj Root, et al. The behavior of naturally fractured reservoirs. *Society of Petroleum Engineers Journal*, 3(03):245–255, 1963.