

Distributed HMM

Megha Gupta

May 16, 2014

1 Automata and their products

Distributed networks can be modelled using interacting automata. Benveniste defines automaton as a quadruple, $\hat{A} = (X, X_0, A, T)$ where X is a finite state of sets, X_0 is the subset of initial states, A is a finite set of messages, T is a set of transitions of the form $t = \{x_-, a, x\}$ where x_- is the previous state, a is the message label on which the state transitions to the next state x . The figure 1 below explains the automata with an example.

For automaton R , $X_R = \{2; R1, R2\}$, $X_{0R} = \{R1\}$, $A_R = \{a, b\}$, $T_R = \{R1, a, R1; R1, b, R2; R2, a, R2; R2, b, R1\}$
For automaton S , $X_S = \{3; S1, S2, S3\}$, $X_{0S} = \{S1\}$, $A_S = \{a, b\}$, $T_S = \{S1, a, S1; S1, b, S2; S2, a, S2; S2, b, S3; S3, a, S3; S3, b, S1\}$

The product of two automata $\hat{A} = R \times S$ is defined as follows:

$$X = X_R \times X_S$$

$$X_0 = X_{0R} \times X_{0S}$$

$$A = A_R \cup A_S$$

Benveniste uses a notion of stuttering transition which helps to distinguish between local and global time by inserting dummy transitions between two transitions of a local automaton attached to a node. This stuttering transition does nothing but lets the rest of the world progress.

A	R1	R2
R1	0.6	0.4
R2	0.3	0.7

Table 1: Transition probability, A

B	a	b
R1	0.2	0.8
R2	0.5	0.5

Table 2: Observed probability, B

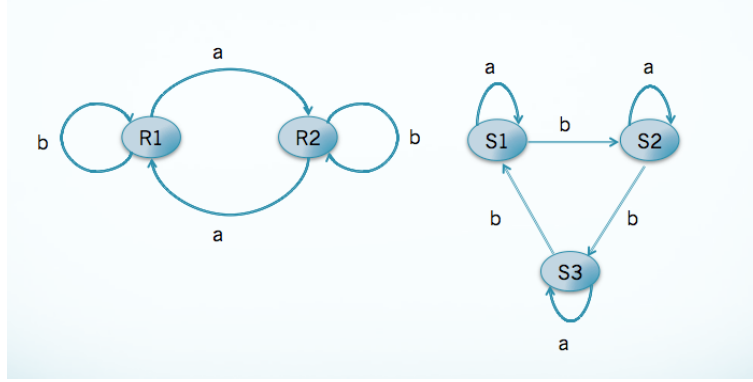


Figure 1: Automata R and S

	R1	R2
π	0.4	0.6

Table 3: Initial state probability, π

Talking in terms of HMM, requires us to equip products of automata with probabilities. Benveniste defines HMM as a triple (\hat{A}, μ, π) where $\hat{A} = (X, X_0, A, T)$ is an automaton, μ is the initial state probability, π is factored as state transition probability π_x and message transition probability π_A . He uses a random arbiter α , with values first, second, third to choose automaton to initiate transition. If $\alpha = \text{first}$ then first automaton chooses any transition having a private message whereas second automaton performs a stuttering transition, and vice versa for $\alpha = \text{second}$. If $\alpha = \text{both}$, then both automata agree on some shared message and move accordingly.

Using the traditional HMM notation of the parameters $\lambda = \{A, B, \pi\}$ where A is the transition probability, B is the observed probability, π is the initial state probability. For automata R, we have the values of A, B, π as shown in table 1, 2, 3 respectively.

2 Product of HMM

Product of HMM is a way of combining HMM's to form distributed state time series model. The figure 2 is a product of two HMMs shown in 1. For $P = R \times S$, the quadruple becomes

$X = \{6; R1S1, R1S2, R1S3, R2S1, R2S2, R2S3\}$

$X_0 = \{R1S1\}$

$A = \{a, b\}$

The rules for synchronised product construction are :

1. $\langle p, q \rangle \xrightarrow{-a-} \langle p', q' \rangle$ if $a \in A_R \cap A_S$ and $p \xrightarrow{-a-} p'$ and $q \xrightarrow{-a-} q'$
2. $\langle p, q \rangle \xrightarrow{-a-} \langle p', q \rangle$ if $a \in A_R, a \notin A_S$ and $p \xrightarrow{-a-} p'$

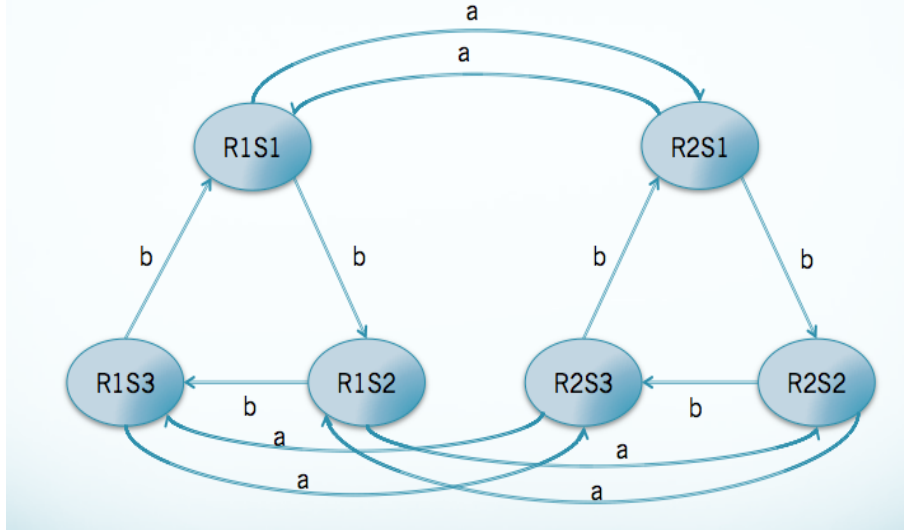


Figure 2: Product of HMMs, $P = R \times S$

3. $\langle p, q \rangle \xrightarrow{a} \langle p, q' \rangle$ if $a \notin A_R$, $a \in A_S$ and $q \xrightarrow{a} q'$

3 Training Product of experts by minimising contrastive divergence

High dimensional distributions are approximated as a product of one dimensional distributions. The product of individual distributions which is uniguassian or multivariate guassian will also be multivariate guassian. If the individual models are more complicated and contain one or more hidden variables, multiplying their distributions together and renormalizing them can be very powerful. These individual models are called "experts". The product of experts produce sharper distribution than the individual distributions[1].

4 Proof of concept on REDD House 2

4.1 Aim

To represent streams of energy consumption data from n^1 appliances by product(s) of k HMMs.

¹ $n=2$

4.2 Method

- **Data** The Reference Energy Disaggregation Data Set (REDD) is used in empirical analysis. The data contains power consumption from real homes, for the whole house as well as for each individual circuit in the house (labeled by the main type of appliance on that circuit). It is intended for use in developing disaggregation methods, which can predict, from only the whole-home signal, which devices are being used. The REDD data set contains two main types of home electricity data: high-frequency current/voltage waveform data of the two power mains (as well as the voltage signal for a single phase), and lower-frequency power data including the mains and individual, labeled circuits in the house. The main directory consists of several house_*i* directories, each of which contain all the power readings for a single house. Each house subdirectory consists of a labels.dat and several channels_*i*.dat files. The labels file contains channel numbers and a text label indicating the general category of device on this channel. Each channel_*i*.dat file has two columns containing UTC timestamps (as integers) and power readings (recording the apparent power of the circuit) for the channel. Experiments reported here use the House 2 data from REDD. It has 11 channels where each channel corresponds to the following appliance:

1. mains_1
2. mains_2
3. kitchen_1
4. lighting
5. stove
6. microwave
7. washer_dryer
8. kitchen_2
9. refrigerator
10. dishwasher
11. disposal

The dataset has 318759 records and 2 columns. We randomly sample 300 records for our experiment. Time series data from two appliances are represented as product of k HMMs.

- **Time Series** : The time series data of the microwave, dryer, kitchen_2 and refrigerator are plotted below in Figures 3, 4, 5, 6.
- **Code** The implementation of the product of experts model is obtained from Iain Murray’s website [?]. It implements the technique described in Geoff Hinton’s paper [1].

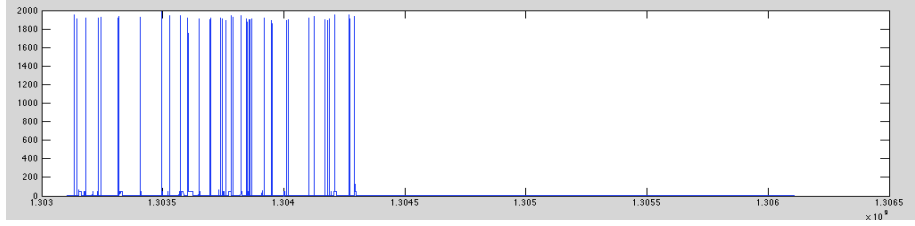


Figure 3: Microwave

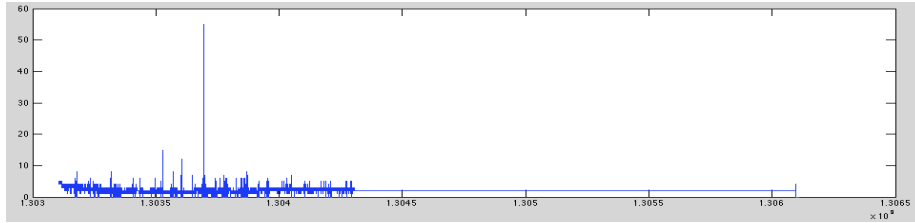


Figure 4: washer_dryer

- **Additional details** Some additional details regarding experiments:
 1. The product of HMMs model (PoHMM) minimizes “contrastive divergence” as described in the paper [1].
 2. The number of experts, k used here is 15. This is set somewhat arbitrarily and needs to be experimented on.
 3. Learning rate is $\epsilon = \frac{1}{300}$.

4.3 Results

This section shows the graphs for different combinations of devices representing how well the learning by product of experts on real data takes place. Experts try to fit the generated data (fantasy data) to the means calculated from the real data. The results of the following pairwise appliances, microwave & washer, Kitchen_2 & refrigerator and microwave & refrigerator are plotted below in Figures 7, 8, 9. The black dots represent the fantasy data that is generated from the product of experts. Mean computed by the real data is shown by red dots and ellipses in blue show the one standard deviation contours of the Gaussians in each expert. There are 15 unigauss experts each of which is a mixture of a uniform and a single, axis aligned gaussian. These experts are initialized with randomly located, circular gaussians that have the same variance as the data. In the fitted model, each tight data cluster is represented by the intersection of two Gaussians which are elongated along different axes.

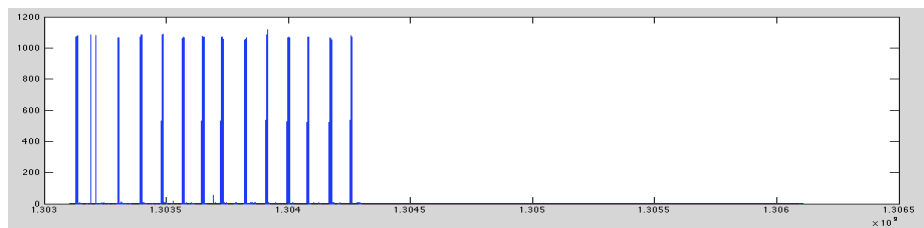


Figure 5: Kitchen_2

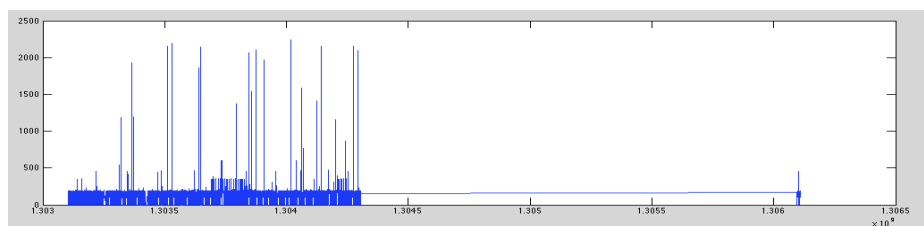


Figure 6: refrigerator

References

- [1] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. Technical Report GCNU TR 2000-004, Gatsby Computational Neuroscience Unit, University College London, 2000.

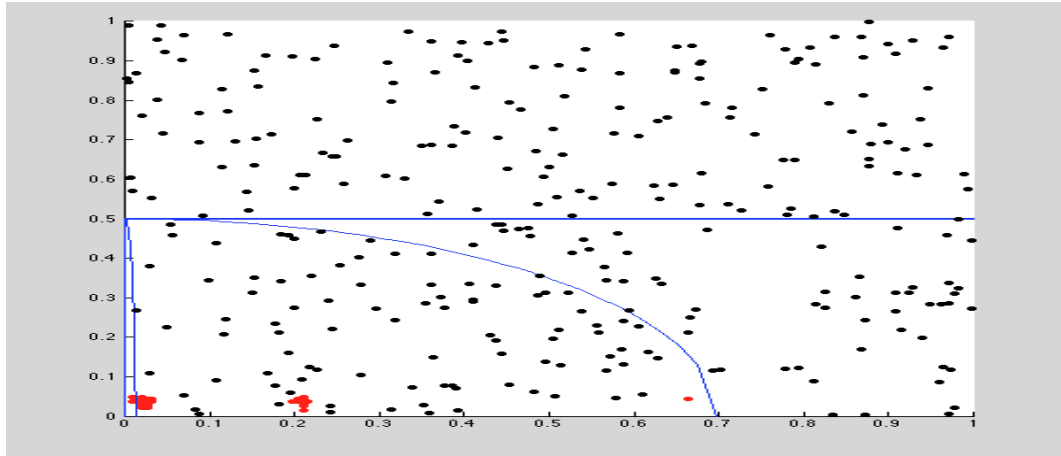


Figure 7: Training product of experts for microwave and washer_dryer

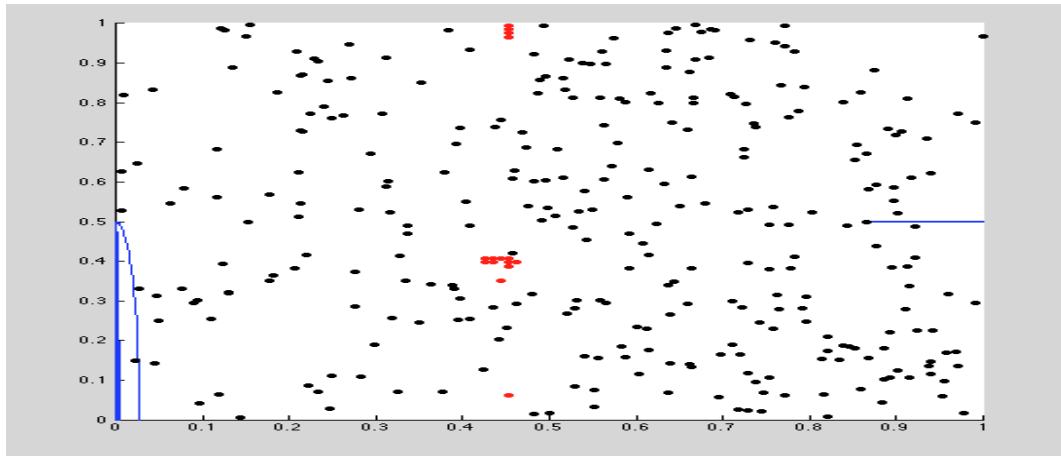


Figure 8: Training product of experts for Kitchen_2 and refrigerator

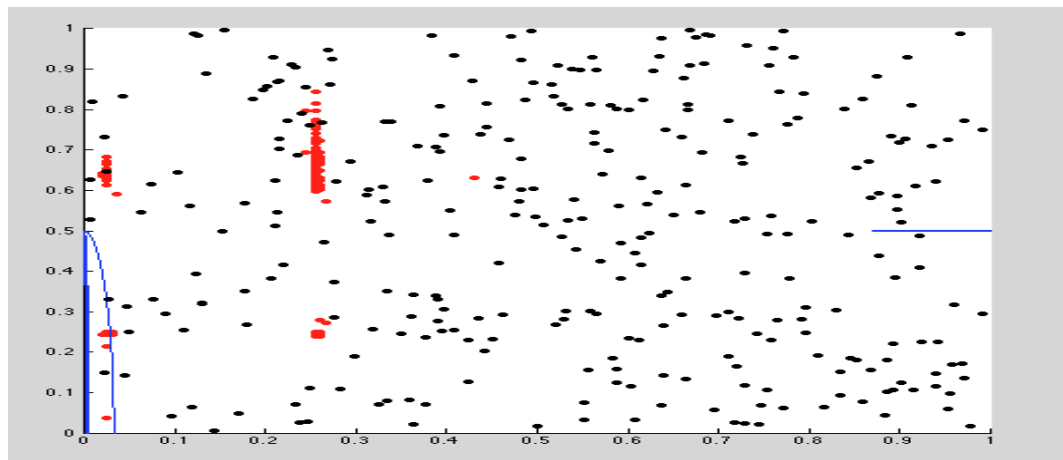


Figure 9: Training product of experts for microwave and refrigerator