A Appendix to Online Clustering with Experts

Further discussion of experiments. Here we further discuss experimental results reported in the paper. Interestingly, we observe that OCE (and in particular Learn-α) tracks the best expert much more effectively on all the real data sets than on the 25 Gaussian experiment. This is a simulated data set from a mixture of Gaussians that is fixed, and where k is known to be 25. While OCE makes a favorable showing on the 3-experts experiment, in the 6-expert experiment, experts 4.-6. incur loss that is orders of magnitude lower than the remaining ones. We drilled down on the performance of the OCE algorithms with low α values, including Static-Expert, as well as Learn-α, and observed that the means were simply hurt by the algorithms' uniform priors over experts. That is, in a few early iterations, OCE incurred costs from clusterings giving non-trivial weights to all the experts' predictions, so in those iterations costs could be orders of magnitude higher than those of experts 4.-6. Moreover, as mentioned above, our regret bounds instead upper bound loss.

Additional experimental details. Experts 4.-6. are 3 variants of k-means# algorithm [3]. In particular they are: 4. k-means# that outputs $3 \cdot k \cdot \log k$ centers. 5. k-means# that outputs $2.25 \cdot k \cdot \log k$ centers. 6. k-means# that outputs $1.5 \cdot k \cdot \log k$ centers. Although OCE can start training from the first observation, via our analysis treating smaller window sizes (encountered at the beginning and the end of the sequence), in the experiments we used the first batch of 200 points as input to all the clustering algorithms, and started training OCE variants after that. The parameter R was not tuned but was set as follows: $R^2 = 10000$ for all data sets except Intrusion and Spambase, in which $R^2 = 1000000000$.

Additional experimental results are provided in Section C.

B Additional proofs

First we provide some lemmas that will be used in subsequent proofs. These follow the approach in [26]. We use the short-hand notation $L(i,t)$ for $L(x_t, c_t^i)$, which is valid since our loss is symmetric with respect to its arguments.

Lemma 6.
$-\sum_{t=1}^{T} \log \sum_{i=1}^{n} p_t(i) e^{-\frac{1}{2}L(i,t)} = -\log[\sum_{i_1,\dots,i_T} p_1(i_1) e^{-\frac{1}{2}L(i_1,1)} \prod_{t=2}^{T} e^{-\frac{1}{2}L(i_t,t)} P(i_t|i_{t-1},\Theta)]$

Proof. First note that following [25], we design the HMM such that we equate our loss function, $L(i,t)$, with the negative log-likelihood of the observation given that expert i is the current value of the hidden variable. In the unsupervised setting, the observation is x_t. Thus $L(i,t) = -\log P(x_t|a_i, x_1, ..., x_{t-1})$. Therefore, we can expand the left hand side of the claim as follows.
$-\sum_{t=1}^{T} \log \sum_{i=1}^{n} p_t(i) e^{-\frac{1}{2}L(i,t)} = -\sum_{t=1}^{T} \log \sum_{i=1}^{n} p_t(i) P(x_t|a_i, x_1, ..., x_{t-1})$

$$\begin{aligned}
&= -\sum_{t=1}^{T} \log P(x_t|x_1, ..., x_{t-1}) \\
&= -\log p_1(x_1) \prod_{t=2}^{T} P(x_t|x_1, ..., x_{t-1}) \\
&= -\log P(x_1, ..., x_T) \\
&= -\log[\sum_{i_1,\dots,i_T} P(x_1, ..., x_T|i_1, ..., i_T) P(i_1, ..., i_T|\Theta)] \\
&= -\log[\sum_{i_1,\dots,i_T} p_1(i_1) P(x_1|i_1, ..., i_T) \prod_{t=2}^{T} P(x_t|i_1, ..., i_T, x_1, ..., x_{t-1}) P(i_t|i_1, ..., i_{t-1}, \Theta)] \\
&= -\log[\sum_{i_1,\dots,i_T} p_1(i_1) P(x_1|i_1) \prod_{t=2}^{T} P(x_t|i_t, x_1, ..., x_{t-1}) P(i_t|i_{t-1}, \Theta)] \\
&= -\log[\sum_{i_1,\dots,i_T} p_1(i_1) e^{-\frac{1}{2}L(i_1,1)} \prod_{t=2}^{T} e^{-\frac{1}{2}L(i_t,t)} P(i_t|i_{t-1}, \Theta)]
\end{aligned}$$

□

Lemma 7.

$$\sum_{i=1}^{n} \rho_i^* D(\Theta_i^* \| \Theta_i) = D(\alpha^* \| \alpha)$$

when $\theta_{ij} = 1 - \alpha$ for $i = j$ and $\theta_{ij} = \frac{\alpha}{n-1}$ for $i \neq j$, $\alpha \in [0,1]$, $\sum_{i=1}^{n} \rho_i^* = 1$.

Proof.

$$\sum_{i=1}^{n} \rho_i^* D(\Theta_i^* \| \Theta_i) = \sum_{i=1}^{n} \rho_i^* \sum_{j=1}^{n} \theta_{ij}^* \log \frac{\theta_{ij}^*}{\theta_{ij}} = \sum_{i=1}^{n} \rho_i^* [\theta_{ii}^* \log \frac{\theta_{ii}^*}{\theta_{ii}} + \sum_{j \neq i}^{n} \theta_{ij}^* \log \frac{\theta_{ij}^*}{\theta_{ij}}]$$

$$= \sum_{i=1}^{n} \rho_i^* [(1-\alpha^*) \log \frac{1-\alpha^*}{1-\alpha} + \sum_{j \neq i}^{n} \frac{\alpha^*}{n-1} \log \frac{\frac{\alpha^*}{n-1}}{\frac{\alpha}{n-1}}] = \sum_{i=1}^{n} \rho_i^* [(1-\alpha^*) \log \frac{1-\alpha^*}{1-\alpha} + \alpha^* \log \frac{\alpha^*}{\alpha}]$$

$$= \sum_{i=1}^{n} \rho_i^* D(\alpha^* \| \alpha) = D(\alpha^* \| \alpha) \sum_{i=1}^{n} \rho_i^* = D(\alpha^* \| \alpha)$$

□

B.1 Proof of Theorem 1

Proof. Using Definitions 3 and 4, we can express the loss of the algorithm on a point x_t as

$$L(x_t, \texttt{clust}(t)) = \frac{\| \sum_{i=1}^{n} p(i)(x_t - c_t^i) \|^2}{4R^2}$$

Then the following chain of inequalities are equivalent to eachother.

$$\begin{aligned}
L(x_t, \texttt{clust}(t)) &\leq -2\log\sum_{i=1}^{n} p(i)e^{-\frac{1}{2}L(x_t,c_t^i)}\\
\frac{\|\sum_{i=1}^{n} p(i)(x_t - c_t^i)\|^2}{4R^2} &\leq -2\log\sum_{i=1}^{n} p(i)e^{-\frac{1}{2}\frac{\|x_t-c_t^i\|^2}{4R^2}}\\
e^{\frac{\|\sum_{i=1}^{n} p(i)(x_t-c_t^i)\|^2}{4R^2}} &\leq \left(\sum_{i=1}^{n} p(i)e^{-\frac{1}{2}\frac{\|x_t-c_t^i\|^2}{4R^2}}\right)^{-2}\\
\sum_{i=1}^{n} p(i)e^{-\frac{1}{2}\frac{\|x_t-c_t^i\|^2}{4R^2}} &\leq e^{-\frac{1}{2}\frac{\|\sum_{i=1}^{n} p(i)(x_t-c_t^i)\|^2}{4R^2}} \qquad (2)
\end{aligned}$$

Let $v_t^i = \frac{x_t - c_t^i}{2R}$. Since $\|x_t\| \leq R$ and $\|c_t^i\| \leq R$ then $v_t^i \in [-1\ \ 1]^d$. Equation (2) is equivalent to

$$\sum_{i=1}^{n} p(i)e^{-\frac{1}{2}\|v_t^i\|^2} \leq e^{\frac{-1}{2}\|\sum_{i=1}^{n} p(i)v_t^i\|^2}$$

This inequality holds by Jensen's Theorem since the function $f(v_t^i) = e^{-\frac{1}{2}\|v_t^i\|^2}$ is concave when $v_t^i \in [-1\ \ 1]^d$. □

B.2 Proof of Theorem 2

Proof. We can proceed by applying Theorem 1 to bound the cumulative loss of the algorithm and then use Lemma 6. As we proceed we follow the approach in the proof of Theorem 2.1.1 in [26].

$$\begin{aligned}
L_T(\texttt{alg}) = \sum_{t=1}^{T} L(x_t, \texttt{clust}(t))&\\
\leq\ & -\sum_{t=1}^{T} 2\log\sum_{i=1}^{n} p_t(i)e^{-\frac{1}{2}L(i,t)}\\
=\ & -2\log P(x_1, ..., x_T)\\
=\ & -2\log\sum_{i=1}^{n} P(x_1, ..., x_T|a_{i,1}, ..., a_{i,T})P(a_{i,1}, ..., a_{i,T})\\
=\ & -2\log\sum_{i=1}^{n} p_1(i)P(x_1|a_{i,1})\prod_{t=2}^{T} P(x_t|a_i, x_1, ..., x_{t-1})\\
=\ & -2\log\frac{1}{n}\sum_{i=1}^{n} e^{-\frac{1}{2}L(i,1)}\prod_{t=2}^{T} e^{-\frac{1}{2}L(i,t)}\\
=\ & -2\log\frac{1}{n}\sum_{i=1}^{n} e^{-\frac{1}{2}\sum_{t=1}^{T} L(i,t)}\\
=\ & -2\log\frac{1}{n} - 2\log\sum_{i=1}^{n} e^{-\frac{1}{2}\sum_{t=1}^{T} L(i,t)}\\
\leq\ & L_T(a_i) + 2\log n
\end{aligned}$$

The last inequality holds for any a_i, so in particular for a_i^*. □

B.3 Proof of Theorem 3

Proof. By applying first Theorem 1 and then Lemma 6 and following the proof of Theorem 3 (Main Theorem) in the [26] we obtain:

$$L_T(alg) = \sum_{t=1}^{T} L_t(alg)$$

$$\leq \sum_{t=1}^{T} -2\log\sum_{i=1}^{n} p_t(i)e^{-\frac{1}{2}L(i,t)}$$

$$= -2\log[\sum_{i_1,...,i_T} p_1(i_1)\prod_{t=2}^{T} P(i_t|i_{t-1},\Theta)\prod_{t=1}^{T} e^{-\frac{1}{2}L(i_t,t)}]$$

$$= -2\log[\sum_{i_1,...,i_T} P(i_1,...,i_T|\Theta)\prod_{t=1}^{T} e^{-\frac{1}{2}L(i_t,t)}]$$

where

$$P(i_1,...,i_T|\Theta) = p_1(i_1)\prod_{t=2}^{T} P(i_t|i_{t-1},\Theta)$$

Notice that also:

$$P(i_1,...,i_T|\Theta) = p_1(i_1)\prod_{i=1}^{n}\prod_{j=1}^{n}(\theta_{ij})^{n_{ij}(i_1,...,i_T)}$$

where $n_{ij}(i_1,...,i_T)$ is the number of transitions from state i to state j, in a sequence $i_1,...,i_T$ and $\sum_j n_{ij}(i_1,...,i_T)$, is the number of times the sequence was in state i, except at the final time-step. Thus $\sum_j n_{ij}(i_1,...,i_T) = (T-1)\hat{\rho}_i(i_1,...,i_T)$, where $\hat{\rho}_i(i_1,...,i_T)$ is the empirical estimate, from the sequence $i_1,...,i_T$, of the marginal probability of being in state i, at any time-step except the final one. It follows that: $n_{ij}(i_1,...,i_T) = (T-1)\hat{\rho}_i(i_1,...,i_T)\hat{\theta}ij(i_1,...,i_T)$ where $\hat{\theta}_{ij}(i_1,...,i_T) = \frac{n_{ij}(i_1,...,i_T)}{\sum_j n_{ij}(i_1,...,i_T)}$ is the empirical estimate of the probability of that particular state transition, on the basis of $i_1,...,i_T$. Thus:

$$P(i_1,...,i_T|\Theta) = p_1(i_1)\prod_{i=1}^{n}\prod_{j=1}^{n}(\theta_{ij})^{(T-1)\hat{\rho}_i(i_1,...,i_T)\hat{\theta}_{ij}(i_1,...,i_T)}$$

$$= p_1(i_1)e^{(T-1)\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{\rho}_i(i_1,...,i_T)\hat{\theta}_{ij}(i_1,...,i_T)\log\theta_{ij}}$$

Thus:

$$L_T(alg) \leq -2\log[\sum_{i_1,...,i_T} P(i_1,...,i_T|\Theta)\prod_{t=1}^{T} e^{-\frac{1}{2}L(i_t,t)}]$$

$$= -2\log[\sum_{i_1,...,i_T} p_1(i_1)e^{(T-1)\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{\rho}_i(i_1,...,i_T)\hat{\theta}_{ij}(i_1,...,i_T)\log\theta_{ij}}\prod_{t=1}^{T} e^{-\frac{1}{2}L(i_t,t)}]$$

Let $i_1^{'},...,i_T^{'}$ correspond to the best segmentation of the sequence into s segments meaning the s-partitioning with minimal cumulative loss. Obviously then the hindsight-optimal (cumulative loss minimizing) setting of switching rate parameter α, given s, is: $\alpha^{'} = \frac{s}{T-1}$ and since we are in the fixed-share setting: $\theta_{ij}^{'} = 1-\alpha^{'}$ for i = j and $\theta_{ij}^{'} = \frac{\alpha^{'}}{n-1}$ for $i \neq j$. We can continue as follows:

$$\leq -2\log[p_1(i_1^{'})e^{(T-1)\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{\rho}_i(i_1^{'},...,i_T^{'})\hat{\theta}_{ij}(i_1^{'},...,i_T^{'})\log\theta_{ij}}\prod_{t=1}^{T} e^{-\frac{1}{2}L(i_t^{'},t)}]$$

$$= -2\log p_1(i_1^{'}) - 2(T-1)\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{\rho}_i(i_1^{'},...,i_T^{'})]\hat{\theta}_{ij}(i_1^{'},...,i_T^{'})\log\theta_{ij} + \sum_{t=1}^{T} L(i_t^{'},t)$$

$$= 2\log n - 2(T-1)\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{\rho}_i(i_1^{'},...,i_T^{'})\hat{\theta}_{ij}(i_1^{'},...,i_T^{'})\log\theta_{ij} + \sum_{t=1}^{T} L(i_t^{'},t)$$

$$= \sum_{t=1}^{T} L(i_t^{'},t) + 2\log n - 2(T-1)\sum_{i=1,j=i}^{n}\hat{\rho}_i(i_1^{'},...,i_T^{'})\hat{\theta}_{ij}(i_1^{'},...,i_T^{'})\log\theta_{ij}$$

$$-2(T-1)\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\hat{\rho}_i(i_1^{'},...,i_T^{'})\hat{\theta}_{ij}(i_1^{'},...,i_T^{'})\log\theta_{ij}$$

$$=\sum_{t=1}^{T}L(i_t^{'},t)+2\log n-2(T-1)(1-\alpha^{'})\log(1-\alpha)\sum_{i=1,j=i}^{n}\hat{\rho}_i(i_1^{'},...,i_T^{'})$$

$$-2(T-1)\frac{\alpha^{'}}{n-1}\log(\frac{\alpha}{n-1})\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\hat{\rho}_i(i_1^{'},...,i_T^{'})$$

$$=\sum_{t=1}^{T}L(i_t^{'},t)+2\log n-2(T-1)(1-\alpha^{'})\log(1-\alpha)-2(T-1)\alpha^{'}\log(\frac{\alpha}{n-1})$$

$$=\sum_{t=1}^{T}L(i_t^{'},t)+2\log n-2(T-1)(1-\alpha^{'})\log(1-\alpha)-2(T-1)\alpha^{'}\log\alpha+2(T-1)\alpha^{'}\log(n-1)$$

$$=\sum_{t=1}^{T}L(i_t^{'},t)+2\log n-2(T-1)(1-\alpha^{'})\log(1-\alpha)-2(T-1)\alpha^{'}\log\alpha+2s\log(n-1)$$

$$=\sum_{t=1}^{T}L(i_t^{'},t)+2\log n+2s\log(n-1)-2(T-1)((1-\alpha^{'})\log(1-\alpha)+\alpha^{'}\log\alpha)$$

$$=\sum_{t=1}^{T}L(i_t^{'},t)+2\log n+2s\log(n-1)+2(T-1)(H(\alpha^{'})+D(\alpha^{'}\|\alpha))$$

□

B.4 Proof of Lemma 2

Proof. Given any b, it will suffice to provide a sequence such that any b-approximation algorithm cannot output x_t, the current point in the stream, as one of its centers. We will provide a counter example in the setting where $k = 2$. Given any b, consider a sequence such that the stream before the current point consists entirely of n_1 data points located at some position A, and n_2 data points located at some position B, where $n_2 > n_1 > b-1$. Let x_t be the current point in the stream, and let it be located at a position X that lies on the line segment connecting A and B, but closer to A. That is, let $\|A-X\| = a$ and $\|B-X\| = c$ such that $0 < a < \frac{n_2}{n_2+1}c$. This is reflected by the figure, which includes some additional points D and E:

———A———D—————X——————E——————————-B—————-

where A,D,X,E,B are lying on the same line. Let $D \in [A,X]$ and $E \in [X,B]$. Let for particular location of D and E, $\|A-D\| = a_1$, $\|D-X\| = a_2$, $\|X-E\| = c_1$ and $\|E-B\| = c_2$, such that $a_1+a_2 = a$ and $c_1+c_2 = c$.

We will first reason about the optimal k-means clustering of the stream including x_t. We will consider cases:
1) Case 1: optimal centers lie inside the interval (A,X). Any such set cannot be optimal since by mirror-reflecting the center closest to X with respect to X (such that it now lies in the interval (B,X) and has the same distance to X as before) we can decrease the cost. In particular the cost of points in B will only decrease, leaving the cost of points in A, plus the cost of X, unchanged.
2) Case 2: optimal centers lie inside the interval (B,X). In this case we can alternately mirror-reflect the closest center to X with respect to X and then with respect to A (reducing the cost with each reflection) until it will end up in interval $[A,X)$. The cost of the final solution is smaller than when both centers were lying in (B,X), because while mirror-reflecting with respect to X, the cost of points in A can only decrease, leaving the cost of points in B and point X unchanged and while mirror-reflecting with respect to A, the cost of point X can only decrease, leaving the cost of points in A and in B unchanged.

Thus the optimal set of centers (let's call them: C_1, C_2) must be such that: $C_1 \in [A, X]$ and $C_2 \in [B, X]$. The figure above reflects this situation and is sufficiently general; thus $D = C_1$ and $E = C_2$. We will now consider all possible locations of such centers (C_1, C_2) and their costs:

1) (A,X): cost $= n_2(c_1 + c_2)^2$
2.1) (A,E) where $E \in (X, B)$: cost $= (a_1 + a_2)^2 + n_2 c_2^2$ when $c_1 \geq a$
2.2) (A,E) where $E \in (X, B)$: cost $= c_1^2 + n_2 c_2^2$ when $c_1 < a$
3) (A,B): cost $= (a_1 + a_2)^2$
4) (D,X) where $D \in (A, X)$: cost $= n_1 a_1^2 + n_2(c_1 + c_2)^2$
5.1) (D,E) where $D \in (A, X)$ and $E \in (X, B)$: cost $= n_1 a_1^2 + a_2^2 + n_2 c_2^2$ when $a_2 \leq c_1$
5.2) (D,E) where $D \in (A, X)$ and $E \in (X, B)$: cost $= n_1 a_1^2 + c_1^2 + n_2 c_2^2$ when $a_2 > c_1$
6) (D,B) where $D \in (A, X)$: cost $= n_1 a_1^2 + a_2^2$
7) (X,E) where $E \in (X, B)$: cost $= n_1(a_1 + a_2)^2 + n_2 c_2^2$
8) (X,B) where $E \in (X, B)$: cost $= n_1(a_1 + a_2)^2$

Notice:
cost(1) > cost(2.2) thus optimal configuration of centers can not be as in case 1)
cost(7) > cost(8) thus optimal configuration of centers can not be as in case 7)
cost(4) > cost(5.2) thus optimal configuration of centers can not be as in case 4)
cost(8) > cost(6) thus optimal configuration of centers can not be as in case 8)
cost(5.2) > cost(2.2) thus optimal configuration of centers can not be as in case 5.2)
cost(5.1) > cost(6) thus optimal configuration of centers can not be as in case 5.1)
cost(2.1) > cost(3) thus optimal configuration of centers can not be as in case 2.1)

Consider case (2.2): cost $= c_1^2 + n_2 c_2^2 = c_1^2 + n_2(c - c_1)^2$ and $c_1 \in (0, a)$. Keeping in mind that $0 < a < \frac{n_2}{n_2+1}c$, it is easy to show that cost $> a^2 + n_2(c - a)^2 > n_2 a^2 > n_1 a^2 > a^2 = \text{cost}(case(3))$. Thus the optimal configuration of centers can not be as in case 2.2). Therefore, the only cases we are left to consider are cases 3 and 6. In both cases, one of the optimal centers lies at B. Let this center be C_2. Since $a < c$, the remaining points (in A and X) are assigned to center C_1 whose location can be computed as follows (please see figure below for notation simplicity):

A—C_1———X—————B $= C_2$

Let $\|A - C_1\| = \delta$ and $\|A - X\| = a$ (as defined above). Since we proved that C_2 must be fixed at B, the points at B will contribute 0 to the objective, and we can solve for C_1 by minimizing the cost from points in A, plus the cost from X: $\min_\delta\{n_1\delta^2 + (a - \delta)^2\}$. The solution is $\delta = \frac{a}{n_1+1}$. Thus the total optimal cost is:

$$OPT = n_1\delta^2 + (a - \delta)^2 = \frac{n_1 a^2}{(n_1 + 1)^2} + \frac{n_1^2 a^2}{(n_1 + 1)^2}$$

We will now consider any 2-clustering of set A, X, B when one cluster center is x_t, which is therefore located at X. We can lower bound the k-means cost of any two set of centers that contain X, as follows: $\text{cost}(\{X, \hat{c}\}) \geq n_1 a^2$ for any $\hat{c}$; the minimal cost is achieved when the other center is located at B (when one of the centers is located in X, the location of the other center that gives the smallest possible k-means cost can be either A (case 1), D (case 4), E (case 7) or B (case 8), where case 8 has the smallest cost from among them).

Violating the b-approximation assumption occurs when $\text{cost}(\{X, \hat{c}\}) > b * OPT$. Given the above, it would suffice to show $n_1 a^2 > b * OPT$. That is:

$$n_1 a^2 > b\left(\frac{n_1 a^2}{(n_1 + 1)^2} + \frac{n_1^2 a^2}{(n_1 + 1)^2}\right) \Leftrightarrow b < (n_1 + 1)$$

This holds, as we chose $n_1 > b - 1$ in the beginning. Therefore the b-approximation assumption is violated. □

B.5 Proof of Lemma 4

Proof. For ease of notation, we denote by $\Phi_{(t-W,t>}$ the k-means cost of algorithm a on the data seen in the window $(t - W, t >$ (omitting the argument, which is the set of centers output by algorithm a at time t). Since a is b-approximate then:

$$\forall_W \quad \Phi_{(t-W,t>} \leq b \cdot OPT_{(t-W,t>} \tag{3}$$

For any W such that $W < T$, we can decompose T such that $T = mW + r$ where $m \in \mathbb{N}$ and $r \leq W$. Notice then that for any $j \in \{0, ..., W-1\}$ the following chain of inequalities holds as the direct consequence of Equation 3:

$$\begin{aligned} \Phi_{(T-W+j,T>} &+ \Phi_{(T-2W+j,T-W+j>} \\ &+ \Phi_{(T-3W+j,T-2W+j>} \\ &+\ldots+ \Phi_{(T-mW+j,T-(m-1)W+j>} \\ &+ \Phi_{<1,T-mW+j>} \\ &\leq b \cdot OPT_{(T-W+j,T>} \\ &+ b \cdot OPT_{(T-2W+j,T-W+j>} \\ &+ b \cdot OPT_{(T-3W+j,T-2W+j>} \\ &+...+ b \cdot OPT_{(T-mW+j,T-(m-1)W+j>} \\ &+ b \cdot OPT_{<1,T-mW+j>} \\ &\leq b \cdot OPT_{<1,T>} \end{aligned} \tag{4}$$

where the last inequality is the direct consequence of Lemma 3.

Notice that different value of j refers to different partitioning of the time span $< 1, T >$. Figure 2 illustrates an example when T = 10 and W = 3.

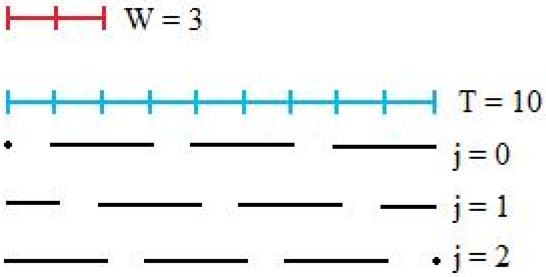


Figure 2: Different partitioning of time span T = 10 into time windows, W = 3, with respect to different values of j.

Let W_t refer to the data chunk seen in time span $< \max(1, t-W+1), t >$. We finish by showing that,

$$\begin{aligned} \sum_{t=1}^{T} \Phi_{W_t} &\leq \sum_{j=0}^{W-1} \{\Phi_{(T-W+j,T>} \\ &+ \Phi_{(T-2W+j,T-W+j>} \\ &+ \Phi_{(T-3W+j,T-2W+j>} \\ &+\ldots+ \Phi_{(T-mW+j,T-(m-1)W+j>} \\ &+ \Phi_{<1,T-mW+j>}\} \\ &\leq b \cdot W \cdot OPT_{<1,T>} \end{aligned} \tag{5}$$

The left hand side of the first inequality (5) sums the losses over only a subset of all the windows that are induced by partitioning the time span T using all possible values of j. The final inequality follows by applying (4). □

To illustrate some of the ideas used in this proof, Figures 2 and 3 provide schematics. Figure 3 shows the windows over which the loss is computed on the left hand side of inequality (5), which is a subset of the set of all windows induced by all possible partitioning of time span T using all possible values of j, which is shown in Figure 2.

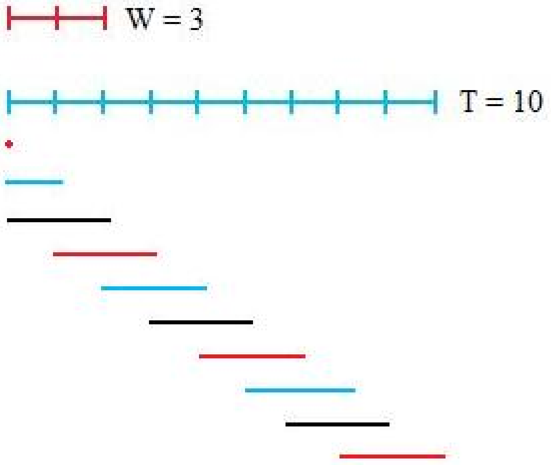


Figure 3: Illustration of the time spans over which the loss is being computed in Equation 5. T = 10, W = 3. Colors red, blue and black correspond to different partitionings, with respect to j, of time span T illustrated in Figure 2.

B.6 Proof of Lemma 5

Proof. The loss of expert i at time t is defined in Definition 3 as the scaled component of the k-means cost of algorithm a's clustering at time t. That is, $\Phi_t(C_t) = \sum_{x'_t \in W_t} \min_{c \in C_t} \|x_{t'} - c\|^2 = \min_{c \in C_t} \|x_t - c\|^2 + \sum_{x'_t \in W_t \setminus x_t} \min_{c \in C_t} \|x_{t'} - c\|^2 = 4R^2 \cdot L_t(C_t) + \sum_{x'_t \in W_t \setminus x_t} \min_{c \in C_t} \|x_{t'} - c\|^2$.
Therefore, since all terms in the sum are positive, $4R^2 \cdot L_t(C_t) \leq \Phi_t(C_t)$, and $L_t(a) \leq \frac{\Phi_t}{4R^2}$. where in the second inequality we substitute in our simplifying notation, where C_t are the set of clusters generated by algorithm a at time t, and Φ_t is algorithm a's k-means cost on W_t. Now, summing over T iterations, and applying Lemma 4, we obtain $\sum_{t=1}^{T} L_t(a) \leq \frac{b \cdot W}{4R^2} OPT_{<1,T>}$. □

B.7 Proof of Theorem 7

Proof. The theorem directly follows from Theorem 3, Lemma 4 and Lemma 5. Notice that both Lemma 4 and Lemma 5 hold in a more general setting when in each time window the identity of the expert (b-approximate algorithm) may change in an arbitrarily way. However we did not provide the generalized proofs of those lemmas since it would only complicate the notation. □

B.8 Proof of Theorem 4, Corollary 1, and Theorem 8

Lemma 8.

$$L_T^{log}(\Theta) = \sum_{t=1}^{T} L^{log}(p_t, t) = -\log\left[\sum_{i_1,\ldots,i_T} p_1(i_1) e^{-\frac{1}{2}L(i_1,1)} \prod_{t=2}^{T} e^{-\frac{1}{2}L(i_t,1)} P(i_t|i_{t-1},\Theta)\right]$$

Proof. It follows directly from Lemma 6. □

Lemma 9.

$$L_T^{log}(\Theta) - L_T^{log}(\Theta^*) = -\log\left[\sum_{\vec{s}} Q(\vec{s}|\theta^*) \exp\left\{T' \sum_{i=1}^{n}\sum_{j=1}^{n} \hat{\rho}_i(\vec{s})\hat{\theta}_{ij}(\vec{s}) \log(\frac{\theta_{ij}}{\theta^*_{ij}})\right\}\right]$$

where $\vec{s} = i_1, \ldots, i_T$ and $Q(\vec{s}|\Theta^*)$ is the posterior probability over the choices of experts along the sequence, induced by hindsight-optimal Θ^*.

Proof. It follows directly from applying proof of Lemma A.0.1 in [26] with redefined $\phi(\vec{s})$ such that $\phi(\vec{s}) = \prod_{t=1}^{T} e^{-\frac{1}{2}L(i_t,t)}$. □

Lemma 10.

$$L_T^{log}(\Theta) - L_T^{log}(\Theta^*) \le (T-1)\sum_{i=1}^{n} \rho_i^* D(\Theta_i^* \| \Theta_i)$$

$$\le (T-1)\max_i D(\Theta_i^* \| \Theta_i)$$

Proof. It holds by Theorem 3 in [26]. □

Lemma 11.

$$L_T^{\log}(\alpha) \le L_T^{\log}(\alpha^*) + (T-1)D(\alpha^* \| \alpha)$$

Proof. It holds by Lemma 7, Lemma 10. □

Lemma 12.

$$L_T(\alpha) \le 2L_T^{\log}(\alpha^*) + 2(T-1)D(\alpha^* \| \alpha)$$

Proof. It holds by Lemma 11 and Lemma 1. □

Lemma 13.

$$L_T(\theta) \le \frac{bW}{2R^2} OPT_T + 2(T-1)\sum_{i=1}^{n} \rho_i^* D(\theta_i^* || \theta_i)$$

Proof.

$$L_T^{log}(\theta^*) = \sum_{t=1}^{T} L^{log}(p_t, t)_{|\theta^*} = (\sum_{t=1}^{T} -\log \sum_{i=1}^{n} p_t(i) e^{-\frac{1}{2}L(i,t)})_{|\theta^*} =$$

$$= (-\sum_{t=1}^{T} \log \sum_{i=1}^{n} p_t(i) e^{-\frac{1}{2}\| \frac{x_t - c_t^i}{2R} \|^2})_{|\theta^*}$$

Define as previously $v_t^i = \frac{x_t - c_t^i}{2R}$ and notice $v_t^i \in [-1;1]^d$.

Thus we continue:

$$= (-\sum_{t=1}^{T} \log \sum_{i=1}^{n} p_t(i) e^{-\frac{1}{2}\|v_t^i\|^2})_{|\theta^*}$$

By the proof of Lemma 4 $\|v_t^i\|^2 \le \frac{\phi_t^i}{4R^2}$ where ϕ_t^i is the k-means cost of algorithm i^{th} clustering (i^{th} expert) at time t.

Thus we can continue as follows (we omitt conditioning by θ^* since it holds for any $\vec{p_t}$):

$$\le -\sum_{t=1}^{T} \log \sum_{i=1}^{n} p_t(i) e^{-\frac{1}{2}\frac{\phi_t^i}{4R^2}} \le -\sum_{t=1}^{T} \log \sum_{i=1}^{n} p_t(i) e^{-\frac{1}{2}\frac{\max_i \phi_t^i}{4R^2}}$$

$$\le -\sum_{t=1}^{T} \log e^{-\frac{1}{2}\frac{\max_i \phi_t^i}{4R^2}} = \sum_{t=1}^{T} \frac{1}{2}\frac{\max_i \phi_t^i}{4R^2} = \frac{1}{8R^2}\sum_{t=1}^{T} \max_i \phi_t^i \le \frac{bW}{8R^2} OPT_T$$

The last inequality follows from the proof of Theorem 7. Thus finally:

$$L_T(\theta) \leq 2L_T^{log}(\theta) \leq 2L_T^{log}(\theta^*) + 2(T-1)\sum_{i=1}^{n}\rho_i^* D(\theta_i^*||\theta_i) \leq \frac{bW}{4R^2}OPT_T + 2(T-1)\sum_{i=1}^{n}\rho_i^* D(\theta_i^*||\theta_i)$$

□

B.9 Proof of Theorem 5

Proof. The logloss per time step of the top-level algorithm (for the ease of notation we skip index log), which updates its distribution over α - experts (Fixed-Share (α_j algorithms) is:

$$L^{top}(p_t^{top}, t) = -\log\sum_{j=1}^{m} p_t^{top}(j)e^{-L^{\log}(j,t)}$$

where

$$L^{\log}(j,t) = -\log\sum_{i=1}^{n} p_j(i)e^{-\frac{1}{2}L(i,t)}$$

thus:

$$L^{top}(p_t^{top}, t) = -\log\sum_{j=1}^{m} p_t^{top}(j)\sum_{i=1}^{n} p_j(i)e^{-\frac{1}{2}L(i,t)}$$

The update is done via the Static Expert algorithm. When running Learn-$\alpha(m)$, m possible values α_j are tested. Following [25, 26], the probabilistic "prediction" of the algorithm is defined as:

$$\sum_{j=1}^{m} p_t^{top}(j)\sum_{i=1}^{n} p_j(i)e^{-\frac{1}{2}L(i,t)}$$

Let us first show that this loss-prediction pair are $(1,1)$-realizable, thus they satisfy:

$$L^{top}(p_t^{top}, t) \leq -\log\sum_{j=1}^{m} p_{top}(j)e^{-L^{\log}(j,t)}$$

Thus we have to prove that following holds:

$$-\log\sum_{j=1}^{m} p_t^{top}(j)\sum_{i=1}^{n} p_j(i)e^{-\frac{1}{2}L(i,t)} \leq -\log\sum_{j=1}^{m} p_t^{top}(j)e^{-L^{\log}(j,t)}$$

thus we have to prove:

$$-\log\sum_{j=1}^{m} p_t^{top}(j)\sum_{i=1}^{n} p_j(i)e^{-\frac{1}{2}L(i,t)} \leq -\log\sum_{j=1}^{m} p_t^{top}(j)e^{\log\sum_{i=1}^{n} p_j(i)e^{-\frac{1}{2}L(i,t)}}$$

which is equivalent to

$$-\log\sum_{j=1}^{m} p_t^{top}(j)\sum_{i=1}^{n} p_j(i)e^{-\frac{1}{2}L(i,t)} \leq -\log\sum_{j=1}^{m} p_t^{top}(j)\sum_{i=1}^{n} p_j(i)e^{-\frac{1}{2}L(i,t)}$$

where the last inequality holds. Now, since our logloss-prediction pair are (1,1)-realizable, by Lemma 1 in [19] we have:

$$L_T^{top} \leq \min_{\{\alpha_j\}} L_T^{\log}(\alpha_j) + \log m$$

where $\{\alpha_j\}$ is the discretization of the α parameter that the Learn-α algorithm takes as input. Now, by applying Corollary 1 we get:

$$L_T^{\log}(\texttt{alg}) \leq L_T^{\log}(\alpha^*) + (T-1)\min_{\{\alpha_j\}} D(\alpha^* \| \alpha_j) + \log m$$

□

C Additional Experimental Details

To provide qualitative results on OCE's performance, in Figures 4-5 we show clustering analogs to learning curves from our experiments in the predictive setting. These curves generated the statistics for Table 1. We plot the batch k-means cost of each expert, and the OCE algorithms, on all the data seen so far, versus t. While Fixed-Share algorithms with high values of α suffer large oscillations in cost, Learn-α's performance tracks, and often surpasses that of the Fixed-Share algorithms.

We also demonstrate the evolution of the weights maintained by the Fixed-Share and Learn-α OCE algorithms. In Figures 6-7 we show the evolution over time of the weights maintained by the OCE Fixed-Share algorithm over the experts (clustering algorithms). For smaller values of α we observe an inversion in the weight ordering between experts 4 and 5 at around iteration 48 for this particular experiment. For α values closer to 1/2 and 1 there is a higher amount of shifting of weights among experts. In Figure 8 we show the evolution over time of the weights maintained by the OCE Learn-α algorithm over α-experts (Fixed-Share algorithms run with a different setting of the α parameter). The experiment was performed with 45 α-experts (Fixed-Share algorithms with different values of the α parameter). Lower α values received higher weights. One value of α (the lowest) receives an increasing share of the weight, which is consistent with the fact that the Static-Expert algorithm is used to update weights over α-experts.

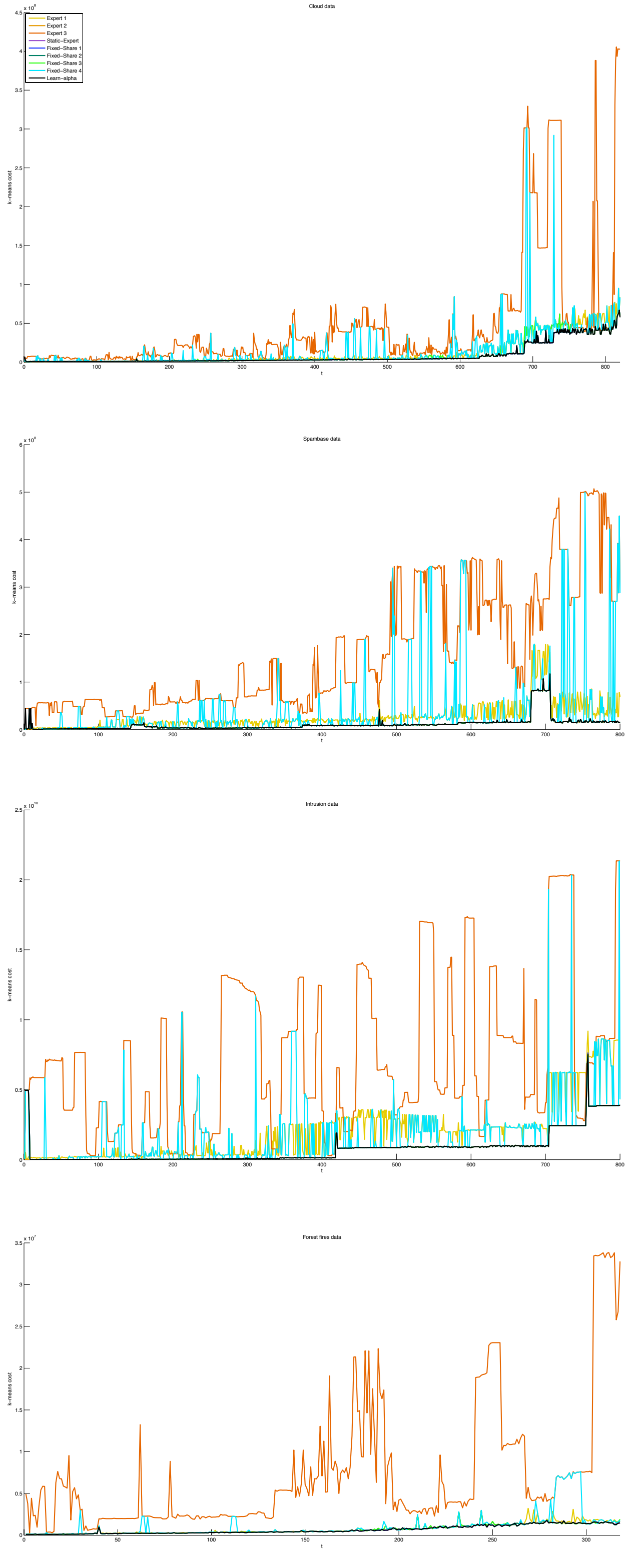


Figure 4: Clustering analogs to learning curves; k-means cost vs. t; Top to bottom: Cloud, Spambase, Intrusion, Forest fires. Legend in upper left.

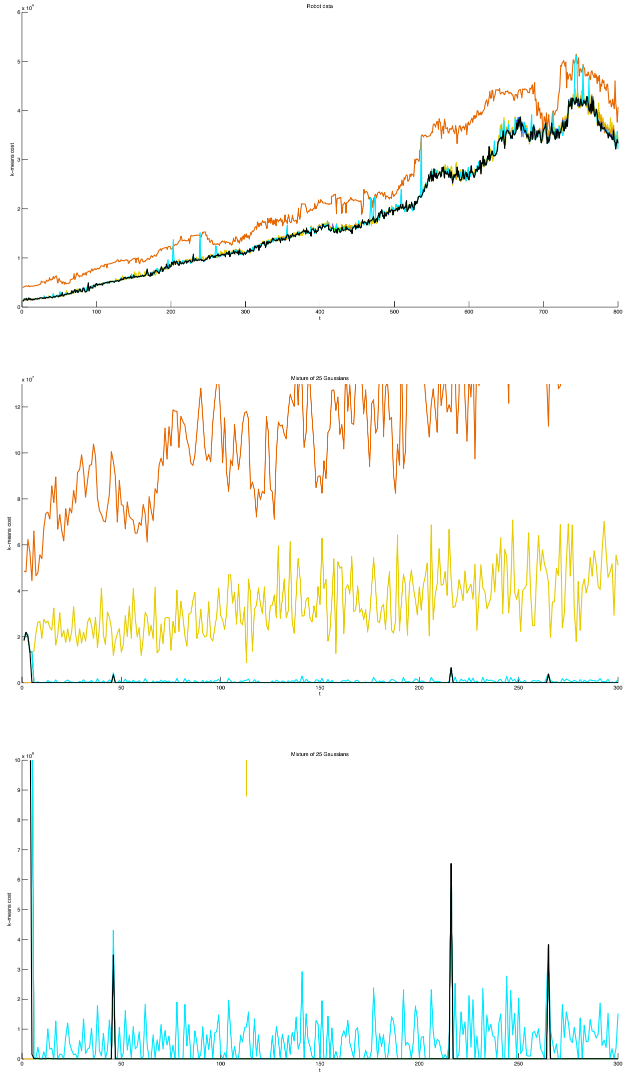


Figure 5: Clustering analogs to learning curves; k-means cost vs. t; Top to bottom: Robot data, 25 Gaussians data, Zooming in on y-axis of 25 Gaussians. Legend in Figure 5.

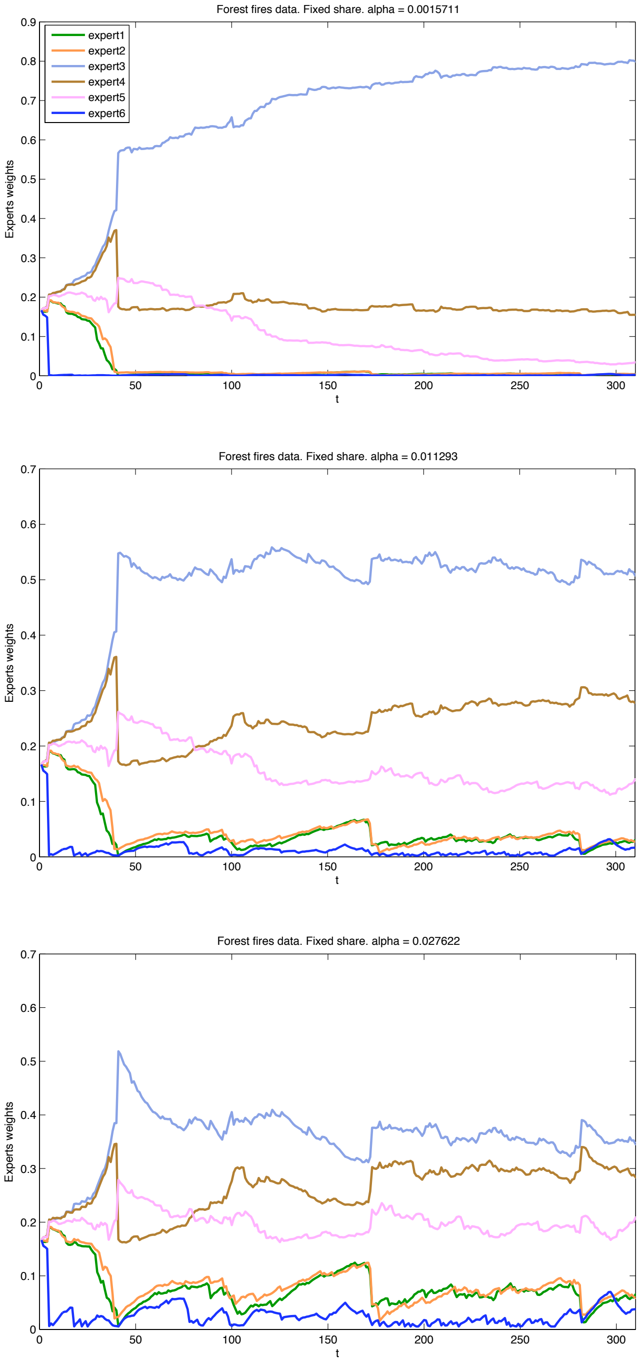
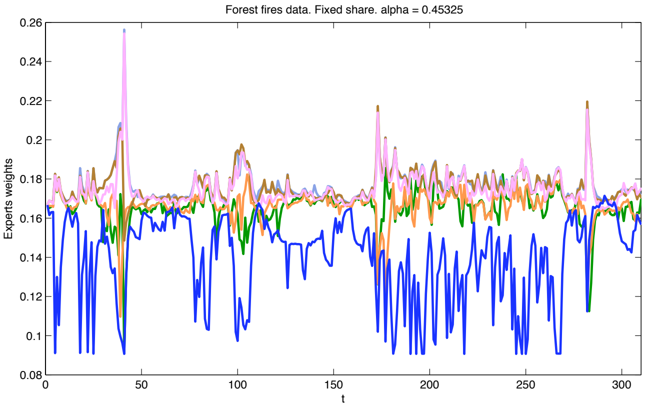


Figure 6: Evolution of weights over experts for Fixed-Share algorithms using different (low) values of the α parameter; Forest fires data. 6-experts. Legend in top left.



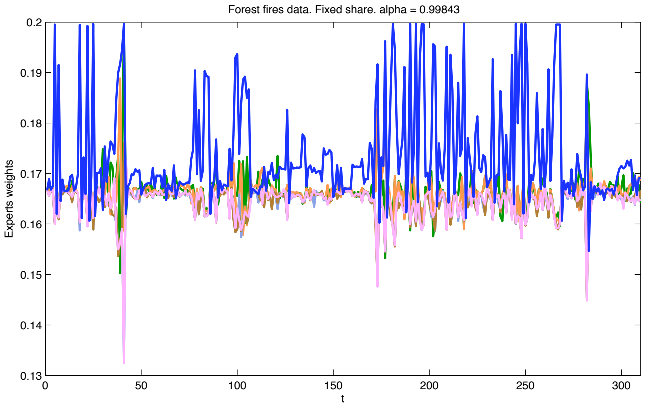


Figure 7: Evolution of weights over experts for Fixed-Share algorithms using different values of the α parameter (Top: close to 1/2; Bottom: close to 1); Forest fires data. 6-experts. Legend in Figure 6.

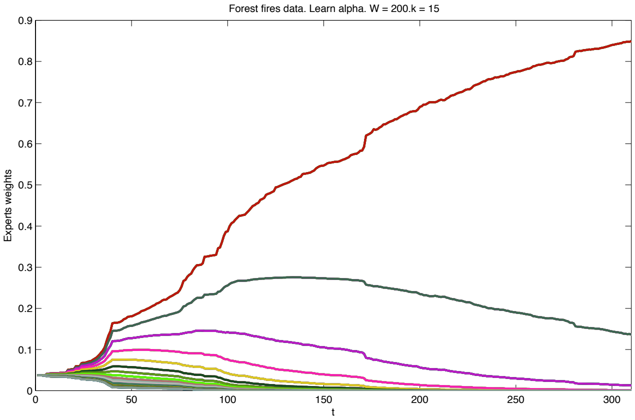


Figure 8: Evolution of weights maintained by Learn-α, over its α-experts, in the 6-expert Forest-Fires experiment. Lowest values of α receive highest weight.