Regret Dynamics in Online Clustering

Marco Jiralerspong [1] **Andjela Mladenovic** [1]

Abstract

Many real world applications make use of unsupervised learning approaches (e.g. clustering), data that arrives in an online fashion. In this paper, we study the problem of online clustering within a regret minimization framework. Building on prior work, we first investigate and implement a Follow-The-Leader (FTL) k-means algorithm—in an attempt to reproduce results in (Cohen-Addad et al., 2021)—and find differences in the regret dynamics for the MNIST dataset and for their worst-case scenario. We then expand the analysis from FTL k-means to a FTL Gaussian Mixture Model (GMM) and provide strong intuition that the worst-case linear regret still holds. Empirically, we demonstrate interesting differences in regret dynamics for MNIST between a GMM with spherical and diagonal covariance matrices.

1. Introduction

One of the main problems in unsupervised learning is identifying structure within data such that similar data points are grouped together. Consequently, the clustering of data points represents a fundamental and challenging task where even classical approaches, e.g. k-means, are known to be NP-hard optimization problems (Vattani, 2009).

In this report, we consider clustering in an online setting where data points arrive sequentially—i.e. one at a time—and needs to be assigned to a cluster (new or existing) without future knowledge of the remaining data points. The online variant of this problem is motivated by numerous applications in fields such as online dynamic pricing (Miao et al., 2019), video reconstruction (Yang et al., 2014) and real-time speech recognition (Higuchi et al., 2017).

It is customary to analyze algorithms operating in online settings with respect to either *regret* or *competitive ratio*. The former quantifies the difference in cost between the algorithm and the optimal offline *static* solution (Orabona, 2019), while the later quantifies the difference in cost between an online algorithm and the optimal offline *dynamic* solution (Albers, 1996). Following prior work (Cohen-Addad et al., 2021), in this report, we choose to perform a regret based analysis. As there are several clustering objective functions, a natural starting point for our work is the k - means and GMM objectives.

2. Related Work

While online clustering is not a particularly novel concept, the vast majority of the current literature mainly addresses four main themes.

The first theme is the efficient updating of an online clustering model as new data comes in. Ideally, instead of storing all incoming data points and retraining the model each time a new data point comes in, it would be preferable to instead derive some update rules for the model. In (Declercq & Piater, 2008), describe using 2 levels of GMMs, the former to approximate the past data stream and the latter to manage updates based on new data points. On the other hand, in (Song & Wang, 2005) and (Hasan & Gan, 2009), algorithms for updating a GMM based solely on new data points and the current model parameters are described.

Other papers discuss the dynamic updating of the number of clusters in an online algorithm (i.e. in scenarios where the number of clusters is not known in advance). For online GMMs this is usually done by merging or splitting clusters based on a certain measure. In particular (Song & Wang, 2005) considers the W and Hotteling statistics to determine when to merge components whereas (Ueda et al., 1998) considers the density of clusters.

The third theme is the application of online clustering to various real world problems. (Yang et al., 2013; 2014) both describe the use of online GMMs for reconstructing blurry videos (here, data points have some level of time-dependence and are thus weighted by how far ago they occurred when training the model). On the other hand (Gentile et al., 2014) discusses online clustering in the context of a bandit problem where it is assumed that users of a recommendation system can be grouped into clusters with similar

[1]Département d'informatique et de recherche opérationnelle , Université de Montréal. Correspondence to: Andjela Mladenovic <andjela.mladenovic@mila.quebec>.

IFT 6269 project report - Université de Montréal

taste. On the business side, (Miao et al., 2019) provides an algorithm for pricing items with low sales by grouping them into clusters to determine appropriate pricing. The algorithm is then used on real items being sold on Alibaba and shown to provide a measurable increase in revenue.

Finally, a few recent papers discuss online clustering within a regret minimization framework. (Cohen-Addad et al., 2021) examines the performance of an online k-means algorithm based on Follow The Leader (FTL) from a theoretical perspective (deriving a counter-example that has linear regret) and a practical perspective (computing the regret of the algorithm on various artificial examples as well as MNIST). (Choromanska & Monteleoni, 2012) on the other hand provides various examples of online clustering algorithms using an ensemble of experts (various types of different k-means algorithms) and perform a regret-based analysis on the performance of these algorithms.

However, as of yet and to the best of our knowledge, there is no paper that examines online GMM models within the context of a regret minimization framework. As such, by extending the regret analysis of previous papers to additional model classes (GMMs) the work in this paper is novel.

Our Contributions: As a starting point for our analysis we attempt to reproduce the results of (Cohen-Addad et al., 2021) for online k-means and find our implementations yields notable differences. We then extend our empirical analysis to online GMMs and find they mostly perform similarly to k-means. Specifically, we focus on studying the application of a prominent algorithm known in the literature as Follow-The-Leader(FTL) in online k-means and online GMM. We assume that FTL has oracle access to a k-means solver and a GMM solver. Despite this we show the existence of a counter example—in an analogous manner to the counter example in online k-means—where the online GMM incurs linear regret. Finally, we also perform experiments for FTL k-means and FTL GMM on natural datasets (generated Gaussian clusters, MNIST and 2 other well-known datasets).

3. Background

Regret is defined as $R(\omega_{1:T})$:

$$R(\omega_{1:T}) = \sum_{t=1}^{T} f_t(\omega_t) - \min_{u \in S} \sum_{t=1}^{T} f_t(u), \tag{1}$$

where $f_t(\omega_t)$ is the loss incurred at time t by taking action ω_t. In FTL, we seek to minimize regret by playing the strategy that minimizes loss over past rounds (i.e. finding the optimal strategy based on the data we've so far). Specifically, FTL selects ω_t as a solution to the following optimization problem:

$$\omega_t = \arg\min_{\omega \in S} \sum_{i=1}^{t-1} f_i(\omega). \tag{2}$$

4. Online Setting

In this section, we first highlight the online k-means setting proposed in Cohen-Addad et al. (2021). Inspired by this we propose a novel setting to study the performance of online GMMs.

Online k-means At time t the online algorithm proposes a set of k candidate cluster centres, $C_t = \{c_{t,1}, \ldots, c_{t,k}\}$ before observing the data point x_t with the goal of minimizing regret defined as:

$$regret_T = \sum_{t=1}^{T} l_t(C_t, x_t) - \min_{C:|C|=k} \sum_{t=1}^{T} \min_{c \in C} ||x_t - c||_2^2 \tag{3}$$

where the loss incurred by the algorithm at time step t is:

$$l_t(C_t, x_t) = \min_{c \in C_t} ||x_t - c||_2^2 \tag{4}$$

Online GMM At time t the online algorithm proposes a set of k Gaussians with parameters, $\theta_t = \{(\pi_{t,1}, \mu_{t,1}, \Sigma_{t,1}), \ldots, (\pi_{t,k}, \mu_{t,k}, \Sigma_{t,k})\}$ before observing the data point x_t with the goal of minimizing regret defined as:

$$regret_T = \sum_{t=1}^{T} -\log p(x_t|\theta_t) - \min_{\theta} \Big(\sum_{t=1}^{T} -\log p(x_t|\theta) \Big) \tag{5}$$

where the loss incurred by the algorithm at time step t is:

$$l_t(\theta_t, x_t) = -\log p(x_t|\theta_t) = -\log\{\sum_{k=1}^{K} \pi_{t,k} \mathcal{N}(x_t|\mu_{t,k}, \Sigma_{t,k})\} \tag{6}$$

5. FTL k-means and FTL GMM - Linear Regret Worst Case

In this section, we demonstrate a counter example that achieves linear regret for FTL k-means and FTL GMM. For FTL k-means, this comes from the following theorem.

Theorem 1. *(Cohen-Addad et al., 2021, Theorem 2.3.) FTL obtains $\Omega(T)$ regret in the worst case, for any fixed $k \geq 2$ and any dimension.*

To show this, they construct the following counterexample. Intuitively, the data stream consists of points that are placed in three locations $\delta, 0, (1-\delta)$ for some $\delta < \frac{1}{4}$. In each round we want to find $k = 2$ optimal clusters. As proven in Theorem 1 stated above, the optimal clusters are either

a $(-\delta)$ - clustering or a $(1-\delta)$ - clustering. Consider a $(-\delta)$ - clustering as the scenario where all points at $-\delta$ are assigned to one cluster also at $-\delta$, and the remaining points belong to the other cluster. Similarly, let $(1-\delta)$ clustering be the case if all the points at $(1-\delta)$ belong to one cluster at $1-\delta$, and all other points to the other cluster.

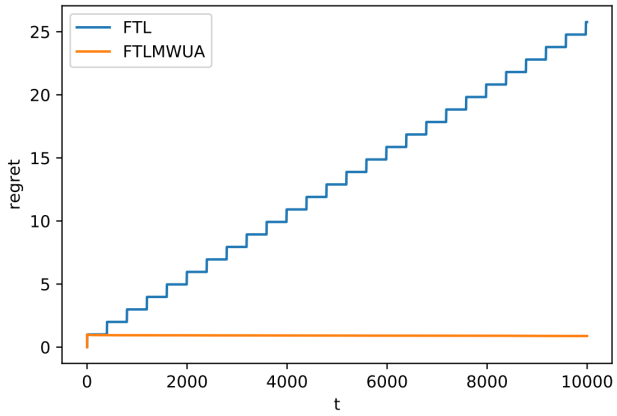

Figure 1. FTL k-means - Linear regret from (Cohen-Addad et al., 2021)

Now consider a sequence where the initial point is at $(1-\delta)$ and then we alternate between adding points between $-\delta$ and 0 (see Figure 2). In this case, while initially the optimal clustering will be at $(1-\delta)$ (see Figure 2) there exists a moment t_2+1 where the optimal clustering will be a $(1-\delta)$ clustering. However, since the algorithm Follow the Leader makes decision based on previously seen rounds, in round t_2+1 the algorithm will propose $-\delta$ clustering, leading to a significant loss in this round. This type of loss can be periodically repeated and which leads to linear regret incurred by Follow the Leader, graphically represented in a staircase pattern.

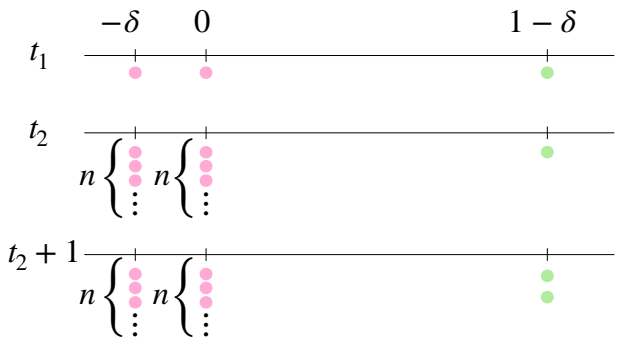

Figure 2. Constructed counter example

Note: While (Cohen-Addad et al., 2021) propose the counter example in Theorem 1, they do not derive the value t_2 in their provided counter example. However, from the Figure 1. we can conclude that the "jump" occurs at every 400 steps for a $\delta = 0.1$ as reported in Cohen-Addad et al. (2021). Our experiments, however, show a different value for this repeated interval (see Figure. 4.), which we confirm in our derivation.

Proposition 1. *Given the counter example in Theorem 1. the change in optimal clusters occurs every 324 steps for a value of* $\delta = 0.1$.

Proof Sketch. Since the optimal solution within the counter example is either a $-\delta$ clustering or a $1-\delta$ clustering, as previously proven in (Cohen-Addad et al., 2021) we want to compare the losses of these two clusterings. A switch from a $(1-\delta)$ clustering to a $(-\delta)$ clustering happens when the loss of a $(1-\delta)$ clustering $l(C_{(1-\delta)})$ is greater than loss of a $(-\delta)$ clustering $l(C_{-\delta})$.

$$l(C_{(1-\delta)}) = \frac{n\delta^2}{2} \tag{7}$$

$$l(C_{(-\delta)}) = \frac{n(1-\delta)^2}{n+1} \tag{8}$$

(Note: $l(C_{(-\delta)})$ is derived by finding the optimal point S(see Figure 3) which is $\frac{1-\delta}{n+1}$.) Then, it follows that the switch happens when $n > 2\frac{(1-\delta)^2}{\delta^2} - 1$ with $\delta = 0.1$ resulting in $n = 162$, which means the change in optimal clusters occurs every 324 steps. □

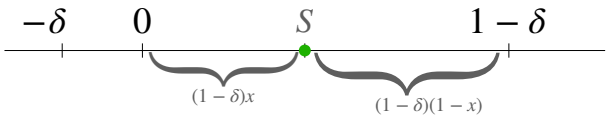

Figure 3. Optimal point S

Implementation Details (FTL k - means): When we run our experiments we ensure that we find the optimal solution of k - means using the following method. We exploit the property that the optimal solution within the counter example is either a $-\delta$ clustering or a $1-\delta$ clustering, as previously described and proven in (Cohen-Addad et al., 2021).

Therefore in our implementation of k-means—in order to make sure that we reach the global solution—we implement the algorithm such that it operates with $k=1$ and on data points that are not at $-\delta$ (or conversely points that are not at $1-\delta$). We compare losses for both these scenarios and choose the clustering which results in a lower loss. The second cluster is then the cluster where we assumed the

loss is zero, —i.e. either the $-\delta$ cluster or the $1-\delta$ cluster, respectively.

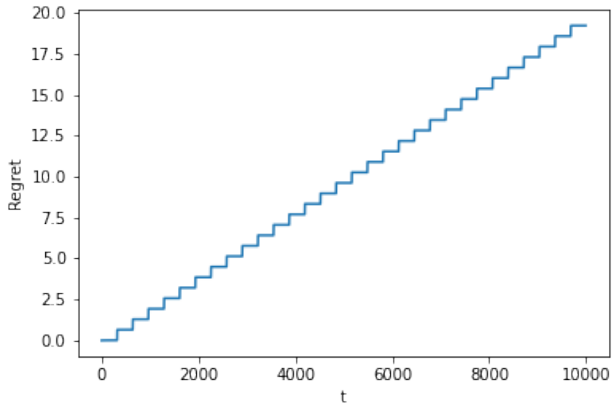

Figure 4. FTL k-means - Linear Regret (our results)

Implementation Details (FTL GMM): While we observed different time steps for our regret interval versus the regret interval reported in (Cohen-Addad et al., 2021) we can see that general staircase-like pattern is present. This observation, motivated us to try out the same counter-example but with different time steps on FTL GMM. In the experiments with Follow the Leader and GMM we use a time step of 18 steps.

To ensure that GMM finds an optimal solution (or close to one) we test it with multiple intializations. Specifically, we initialize GMM using all pairwise combinations among 30 starting means uniformly distributed in the interval $[-\delta, 1-\delta]$ for both clusters as the optimal solution will contain means between these two points.

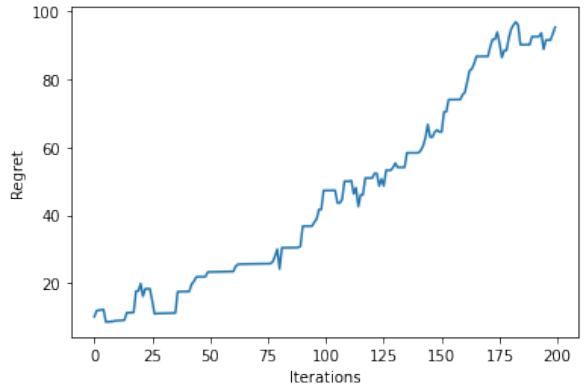

Figure 5. FTL GMM - Linear Regret

Empirically, we observe approximately linear regret. These experiments point to many interesting questions such as:

1. Why does the optimal solution switch every 18 time steps for FTL GMM while k-means switches every 324 time steps?
2. What role does variance play in this counter example?
3. Why is the regret approximately linear for FTL GMM but does not have the "clean" staircase look of FTL k-means.

Ideally, we would like to answer these interesting questions in the future and prove theoretically that the counter example yields linear regret for FTL GMM.

6. FTL k-means and FTL GMM - Natural Datasets

We begin by attempting to replicate the results of (Cohen-Addad et al., 2021) which experiments with FTL k-means on 4 datasets. The first 3 are generated sets of Gaussian clusters that are either well-separated (distance between means is at least 3 standard deviations) or poorly separated (distance between means is 0.7 standard deviations) and the last is MNIST (Deng, 2012). Their findings then indicate that despite FTL k-means poor worst-case performance, they still manage to get logarithmic regret on each of these datasets.

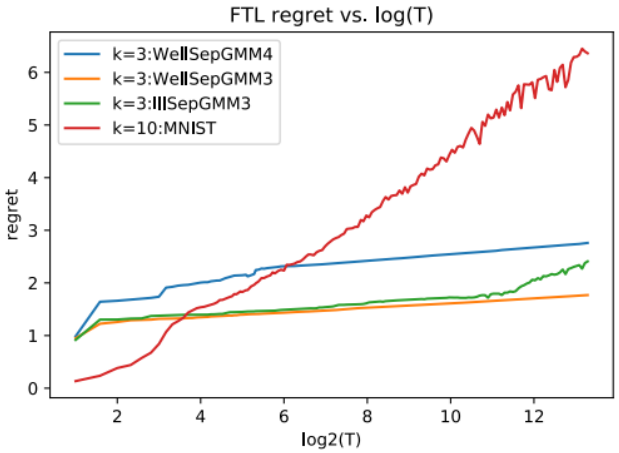

Figure 6. Original results from (Cohen-Addad et al., 2021)

6.1. FTL k-means Implementation

Unfortunately, the exact implementation details of (Cohen-Addad et al., 2021) are not included in the original paper and as a consequence we make reasonable assumptions when details are left unspecified (e.g. the normalization described below ensures the scale of regret values is roughly similar to their paper).

For our experiments, in addition to their 4 datasets, we add the well-known Iris (Fisher, 1936) and Wine (Forina,

1988) datasets to further examine the performance on low-dimensional, real-world datasets. The Iris dataset has 150 data points with 4 features belonging to 3 classes while the Wine has 178 data points with 13 features belonging to 3 classes. For the Gaussian clusters, we generate sets of 2-dimensional clusters that meet their specifications. Finally, for each dataset, we normalize the data feature-wise to be between 0 and 1 before dividing each value by $\sqrt{d}$, where d is the feature dimensionality (the L2 distance scales with dimensionality so this normalization helps ensure the scale of regret is somewhat comparable between datasets). Then, for each experiment on a dataset, a sequence of length $\min(datasetSize, 1000)$ is chosen using random sampling without replacement.

For our implementation of FTL k-means, we specify 2 solvers: the online solver and the optimal solver. Both make use of *Scikit-learn*'s implementation of k-means (Pedregosa et al., 2011) with k-means++ initialization with a number of clusters equal to the number of classes of the dataset (except for WellSepGMM4 where we only use 3 clusters). Due to computational constraints, we only use 10 random initializations for the online solver (though we tested with 300 initially and it had little to no effect). For the optimal solver, we use 300 random initializations as specified in (Cohen-Addad et al., 2021).

6.2. FTL k-means Results

Using the parameters described above, our attempt to replicate their exact results yields some minor differences in the exact regret values we achieve for all datasets as well as a major difference in the regret dynamics for the MNIST dataset.

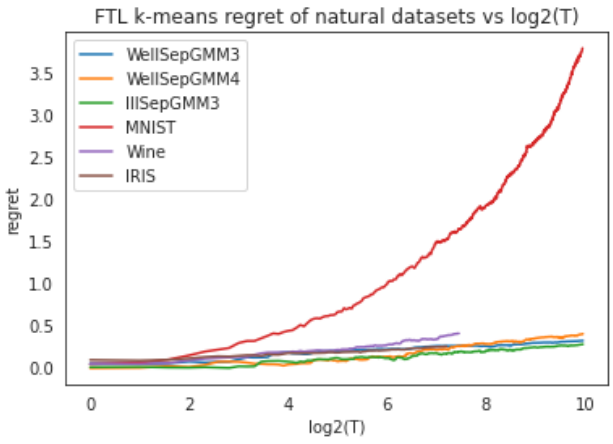

Figure 7. FTL k-means regret experiments on natural datasets

For the Gaussian clusters and the two datasets we added, we replicate the finding of roughly logarithmic regret. However, for MNIST, our experiments indicate that regret grows faster than $O(\log(t))$ but slower than $O(t)$ (i.e. we still haven't hit the worst case). As shown in the graph below, obtained by plotting the OLS solution to $regret_t = mx + b$ where x is either $\log(t)$, $\sqrt{t}$ or t, regret on MNIST is very close to $O(\sqrt{t})$.

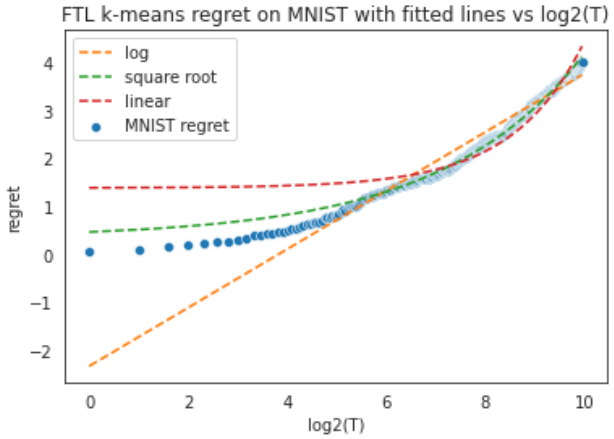

Figure 8. The OLS solution to $regret_t = m\sqrt{t} + b$ fits the actual regret very well.

6.3. FTL GMM Implementation

For our experiments on FTL GMM, we use the datasets as specified previously and once again create 2 solvers. Both use *Scikit-learn*'s implementation (Pedregosa et al., 2011) of a GMM with the same number of clusters as we used in k-means (each of which is initialized using k-means++).

We specify a maximum of 100 iterations of EM (helps with stability when there are only a few data points) and 5 initializations for the online solver and 100 for the optimal solver (once again, varying this number had little effects on the results). Finally, we use either a spherical covariance matrix ($\sigma\mathbf{I}$ where σ is a scalar) or a diagonal covariance matrix ($\boldsymbol{\sigma}\mathbf{I}$ where $\boldsymbol{\sigma}$ is a d-dimensional vector). When computing regret, we compare the online solver to the optimal solver with the same restrictions (i.e. we compare the online spherical GMM to an offline spherical GMM whereas we compare the online diagonal GMM to an offline diagonal GMM).

6.4. FTL GMM Results (General)

Using regret as specified in (6), for the non MNIST datasets, FTL GMM gets roughly logarithmic regret for both spherical and diagonal covariance matrices. Interestingly, this result imitates the regret dynamics for FTL k-means despite the higher capacity models and more sophisticated loss function (though this is perhaps unsurprising for the Gaussian clusters seeing as a GMM can perfectly fit them). The close link between GMM and k-means (especially for

a spherical covariance matrix) could potentially explain this phenomenon. More surprising, however, is the finding that these similarities do not extend completely to MNIST.

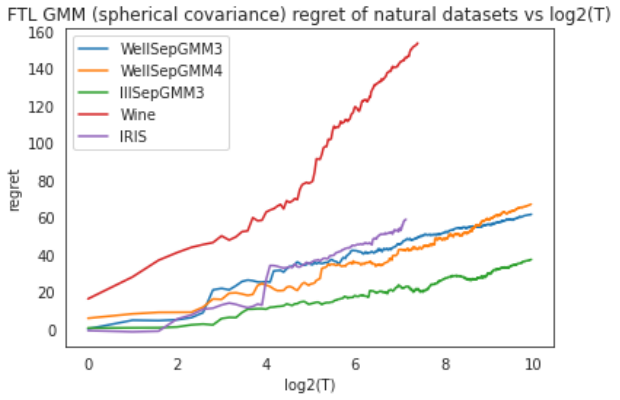

Figure 9. Results from running the same experiments with FTL GMM (spherical)

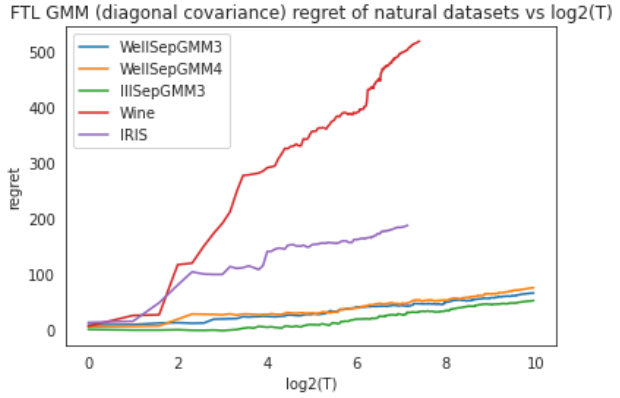

Figure 10. Results from running the same experiments with FTL GMM (diagonal)

6.5. FTL GMM Results (MNIST)

On MNIST, the spherical covariance matrix yields a regret curve very similar to the one observed in k-means. Once again, the resemblance of the result for the spherical covariance FTL GMM with the FTL k-means is unsurprising given their close connection (a GMM with a spherical covariance matrix converges to k-means as $\sigma \to 0$).

In fact, during experiments, the σ of each cluster was always between $\mathbf{0.00001}$ and $\mathbf{0.0001}$. While the small scale of σ is partially due to data normalization which lead most feature values to be small, the small values of σ also help explain the closeness in performance between the 2 models. Nonetheless, more work is needed to determine why the σ parameter in a spherical GMM tends towards 0 for the MNIST dataset.

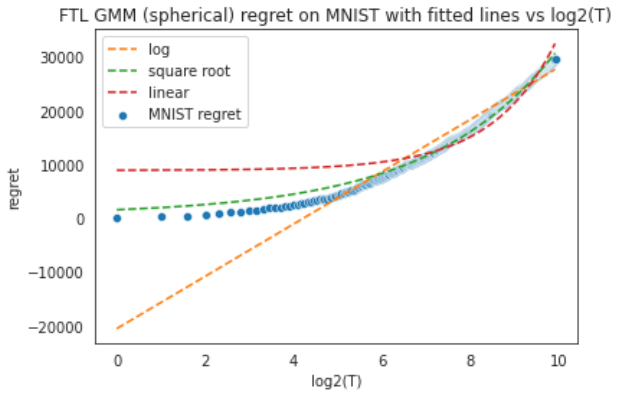

Figure 11. Spherical FTL GMM yields similar regret dynamics as the k-means algorithm on MNIST

The diagonal covariance matrix on the other hand has a regret curve that tapers off towards the end, approaching logarithmic regret, a novel and interesting finding. An initial assessment could indicate that this is due to the higher capacity model which can then better fit the data. However, in an online setting, adding capacity to a model has a dual effect. While it does allow the online solver to better fit the data, it also improves the quality of the the static offline solution we are comparing it to. This tradeoff, at least in this case, seems to favor the online solver as the improvements in its solution are larger as time goes on when compared to the improvement of the offline solution.

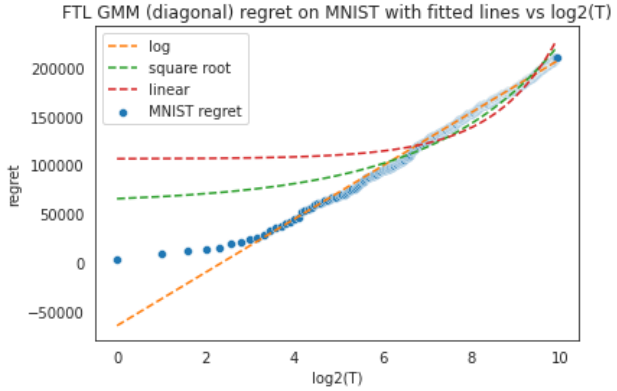

Figure 12. Diagonal FTL GMM approaches logarithmic regret on MNIST

7. Discussion

Beginning with the worst case analysis, we confirm the results of (Cohen-Addad et al., 2021) for FTL k-means and even derive analytically at which interval the regret steps should occur. Then, expanding to FTL GMM with a spherical covariance matrix, we demonstrate experimentally that

the counter example still holds and provide strong intuition as to why this should be the case (though future work is required to prove this). As such, our findings provide further evidence that online clustering algorithms based on vanilla FTL struggle to provide sublinear regret in the worst case. Further research on other forms of FTL (for example regularized FTL) is needed to investigate whether a better worst case bound is possible.

After, considering more natural datasets, we mostly replicate the findings of (Cohen-Addad et al., 2021) and show that each of these 3 models (k-means, spherical GMM and diagonal GMM) yield roughly logarithmic regret on most datasets. The results are more interesting when we examine MNIST. First of all, we fail to replicate their finding of logarithmic regret, instead getting regret that more closely resembles $O(\sqrt{t})$. We arrive at the same result when examining FTL GMM with a spherical covariance matrix.

These findings suggest that these FTL-based clustering algorithms do not perform as well as previously thought on all natural datasets. Further research is however required to determine whether this poor performance on MNIST is due to some inherent structure of the data or potentially from the high dimensionality of the dataset. These results also provide additional evidence of the close link between k-means and GMM with a spherical covariance matrix as in all cases they yielded very similar regret dynamics.

However, when switching to FTL GMM with a diagonal covariance matrix, we get regret dynamics that tend towards logarithmic regret. This finding exposes the interplay between model capacity and regret, indicating that higher capacity online models might perform better (even when they are being compared to a similarly higher capacity offline model). More work is needed to determine exactly why this is the case or perhaps if it is just some characteristic unique to diagonal GMM.

Finally, these direct translations from offline to online clustering yield much to be desired in terms of performance. As noted in the section on experiments on natural datasets, sacrifices often had to be made to perform these experiments in a reasonable amount of time (unsurprisingly, retraining the model each time a new data point comes in is rather inefficient). Future work into more computationally efficient online clustering methods is needed to see whether similar or better regret dynamics can be obtained while improving clustering speed.

Acknowledgements

This work was supervised by professor Gauthier Gidel.

References

Albers, S. *Competitive online algorithms*. BRICS, 1996.

Choromanska, A. and Monteleoni, C. Online clustering with experts. In *Artificial Intelligence and Statistics*, pp. 227–235. PMLR, 2012.

Cohen-Addad, V., Guedj, B., Kanade, V., and Rom, G. Online k-means clustering. In *International Conference on Artificial Intelligence and Statistics*, pp. 1126–1134. PMLR, 2021.

Declercq, A. and Piater, J. H. Online learning of gaussian mixture models-a two-level approach. In *VISAPP (1)*, pp. 605–611, 2008.

Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Fisher, R. Iris data set, 1936.

Forina, M. e. a. Wine data set, 1988.

Gentile, C., Li, S., and Zappella, G. Online clustering of bandits. In *International Conference on Machine Learning*, pp. 757–765. PMLR, 2014.

Hasan, B. A. S. and Gan, J. Q. Sequential em for unsupervised adaptive gaussian mixture model based classifier. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 96–106. Springer, 2009.

Higuchi, T., Ito, N., Araki, S., Yoshioka, T., Delcroix, M., and Nakatani, T. Online mvdr beamformer based on complex gaussian mixture model with spatial prior for noise robust asr. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(4):780–793, 2017. doi: 10.1109/TASLP.2017.2665341.

Miao, S., Chen, X., Chao, X., Liu, J., and Zhang, Y. Context-based dynamic pricing with online clustering. *arXiv preprint arXiv:1902.06199*, 2019.

Orabona, F. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

Song, M. and Wang, H. Highly efficient incremental estimation of gaussian mixture models for online data stream clustering. In *Intelligent Computing: Theory and Applications III*, volume 5803, pp. 174–183. International Society for Optics and Photonics, 2005.

Ueda, N., Nakano, R., Ghahramani, Z., and Hinton, G. E. Split and merge em algorithm for improving gaussian mixture density estimates. In *Neural Networks for Signal Processing VIII. Proceedings of the 1998 IEEE Signal Processing Society Workshop (Cat. No. 98TH8378)*, pp. 274–283. IEEE, 1998.

Vattani, A. The hardness of k-means clustering in the plane. *Manuscript, accessible at http://cseweb. ucsd. edu/avattani/papers/kmeans_hardness. pdf*, 617, 2009.

Yang, J., Yuan, X., Liao, X., Llull, P., Sapiro, G., Brady, D. J., and Carin, L. Gaussian mixture model for video compressive sensing. In *2013 IEEE International Conference on Image Processing*, pp. 19–23. IEEE, 2013.

Yang, J., Yuan, X., Liao, X., Llull, P., Brady, D. J., Sapiro, G., and Carin, L. Video compressive sensing using gaussian mixture models. *IEEE Transactions on Image Processing*, 23(11):4863–4878, 2014.