RREM - A New Algorithm for Robust EM

Archit Sharma
Indian Institute of Technology Kanpur
architsh@.iitk.ac.in

Amur Ghose
Indian Institute of Technology Kanpur
amur@iitk.ac.in

Abstract

We introduce RREM (Ridiculously Robust EM) - an algorithm for quick, robust EM for K-Gaussian mixture models, able to withstand poor initialization and noise difficulties. We show the difficulty of this problem via various illuminating counterexamples in the case of adversarial noise. In general, the adversarial setting leads to complete loss of guarantees. However, we provide some heuristics in this regard that remain applicable and analyze bounds by considering pathological cases. Our algorithm leads to markedly better empirical results in practice.

1 Introduction

As we have outlined in our reading project, EM tends to be a very widely used algorithm for non-convex optimization. However, the drawbacks with its convergence may be summarized thus:

- In general, most of the work for EM focuses on robustness to bad initialization. Very little theoretical work goes into making EM resistant to added noise, especially if the noise is adversarial. (YLL12; BWY$^+$17)
- Most of the proofs of EM's convergence focus on initialization as well. To add to the problem, the most thorough analysis of EM (Wu83) was done for the case of an unique global mode, something that is always violated for clustering since any permutation of labels produces the same likelihood.
- The approaches that do exist tend to be of a heuristic nature (YLL12) revolving around various notions of **likelihood weighing**. Essentially, points that have had too much noise are surely in some form outliers. Such outliers, if the model is performing correctly, ought to have low likelihood. If we weigh points by their likelihoods to get our updates, the algorithm should correctly filter the outliers.

Unfortunately, astute observers will note that any form of likelihood weighing is a chicken-and-egg problem: likelihood weighing only works if the model at time t is good enough to have any meaning to its likelihood values at all! This problem - of having to be in a close enough region to the optimum to have any shot at all - is, unfortunately, very much a feature of the problem in general and an initialization close to the optimum ends up being assumed in a lot of EM proofs.

2 The Absurd Difficulties of Adversarial Noise - An Example

Note: Through all our analyses, we will break the problem down into cases, but the assumption is that neither the algorithm nor the adversary knows what particular case they fall into, they are merely playing the game as follows: The algorithm specifies a count m to the adversary of points they are able to corrupt, the adversary sees data generated by some blackbox, and corrupts and then provides the data to the algorithm.

Consider the case where unlabeled data is served up to us without any variance from $\mathbb{R}$. That is, we are provided N real numbers, and we know that a cluster is simply a value - there is zero variance.

Under this scenario consider the recovery of top K clusters. Let the adversary be able to corrupt up to m points.

Clearly, our algorithm for recovery is to sort the values by frequency and output the top K ones. Suppose the generation initially creates exactly K unique values in the first place (this is not necessary as the generation process is stochastic as well, however, with sufficiently high number of samples, the probability of K clusters is very high). What should be the minimum value of m ? It is clearly enough to set $m \geq c/2$, where c is the smallest frequency count of a value. The adversary can simply shift points from the smallest cluster up to m points and ensure it never gets in.

The general case

The above is already restrictive, but it grows hopeless when we have to recover K clusters among $K' > K$. If we set as our goal the accurate recovery of the top K clusters, and let n_1 to $n_{K'}$ be the counts of clusters ranked from 1 to K', it clearly suffices to have $m \geq n_k - n_{k+1}$ to throw off the algorithm by at least one cluster.

The general algorithm proceeds as follows. Clearly, it's easier to "knock out" a smaller cluster than a larger one. If we can have a blackbox boolean function $KNOCK_m(k)$ that lets us know if we can knock out at least k clusters using a budget of m points then it is clearly valid to do the following :

- Consider the clusters ranked between $K/2$ to K. Evaluate $KNOCK_m(K/2)$ for knocking out these clusters. If true, search similarly in the range of clusters ranked between $K/2$ to 1, starting from $K/4$. Note that the argument for $KNOCK$ will become $3K/4$ at $K/4$, as we are looking at the indices in a reverse fashion.
- If not, search in the range between $K/2$ to K, starting from $3K/4$, i.e. the argument for $KNOCK$ being $K/4$.

We need to come up with an algorithm to evaluate $KNOCK_m(k)$ now. The following algorithm suffices with us as the adversary and with m points to "spend" :

- Let the gap between the largest cluster and the second largest be Δ. Remove Δ points off the largest cluster. If this is untenable, move on to the set excluding the largest cluster and recur.
- Redistribute the Δ points into k groups with Δ/k elements each and recur on the new set. As needed, equate successively the second, third, fourth cluster and so on by chopping.
- The chopped smaller groups should be merged with groups ranking outside the top K groups in ascending order of g_i where g_i denotes the minimum number of extra elements required to get into top K.

The above approach can lead to ridiculous scenarios. Consider, for example, a case where every set ranking above the top K is just 1 less than every group in the top K. K points suffice for the adversary to defeat us effortlessly.

3 Problem Setup

Given a data matrix $X \in \mathbb{R}^{N \times d}$, with $\leq \eta N$ adversarially corrupted points, we want to create K clusters by using Gaussian Mixture Modelling. In particular, we are interested in recovering the cluster means as accurately as possible. We will proceed to obtain them using maximum likelihood estimation, with some modification to the EM algorithm to make it robust.

Likelihood Weighing Methods

As discussed in the introduction, likelihood weighing works due to the outliers having low likelihoods in comparison to the "real" points. A natural algorithm from this is to apply some weighing function on the likelihood. For instance, Huber loss functions applied to the likelihood (TL00) perform well in practice, but have no theoretical guarantees. Another natural approach is to simply cut off points below a threshold likelihood, which can be thought of as analogous to other thresholding

algorithms like IHT (JTK14; BD09). It should be noted that these approaches have been extensively tested empirically, but they can in general only be shown to be robust to initialization, not to noise. Further, we have some approaches via MLqE methods which optimize MLE-like terms (FY$^+$10; QP13) but they too rely on robustness heuristics of being able to do this for an infinite variety of terms and thus cross-validate.

The Uniform Likelihood Model

Consider all outliers as belonging to an extra cluster, that is the count of clusters is not $K+1$. Let this cluster have only one constraint: it assigns equal probabilities to all its elements, that is, it has a uniform likelihood in case of continuous spaces. Why is this a valid assumption? Since, we do not have a model for adversary (in fact having any model for adversary, will allow the adversary to use it against us), it is fair to assume that the adversary can place the outlier anywhere uniformly in the smallest Euclidean ball centred at origin enclosing the points after adversarial corruption (one can choose the space of points differently as well).

Index the extra cluster by 0. The expected complete likelihood to be maximized is

$$\mathcal{L}(\pi, \mu, \sigma) = \sum_{i=1}^{N} \sum_{j=0}^{K} \mathbb{E}[z_{ij} = 1] \Big(\log p(x_i | \mu_j, \sigma_j) + \log p(z_{ij} = 1) \Big) \tag{1}$$

Here, $\log p(x_i|\mu_0, \sigma_0) = 0$ for all x_i. We have assumed the likelihood of all points in the cluster 0 to be 1, which is valid upto a propotionality constant. Now, the only additional parameter to be learnt is π_0. This will fail, unless we introduce some constraint on π_0. The reason is that all points can be assigned to cluster 0, and π_0 can be set to 1, with all other $\pi_j = 0$. This will be the global maximum and is a useless clustering from our perspective. Therefore, we need to constrain π_0. A natural constraint is $\pi_0 \leq \eta$, which arises from our assumption that η points have been corrupted (in addition to $\sum_{j=0}^{K} \pi_j = 1$). Note, we do not need to know η in practice, we can tune it as a hyperparameter.

The solution for all other parameters except π remain the same as those from standard EM. For π_0, we use the KKT condition to solve for every π in every step (note that bashing the Lagrangian is prohibitive). The solution to this results in one of two things:

- $\pi_0 = \eta$, indicating that we allocate to our extra cluster as much as we can. In this case, we will have to ensure that $\sum_{j=1}^{K} \pi_j = 1 - \eta$.
- $\pi_0 = 1 - \sum_{i \neq 0} \pi_i$, i.e. we give to the cluster the remainder whenever this remainder is less than η.

Intuitively, it seems clear that the algorithm would always result in setting $\pi_0 = \eta$, because that does not decrease the likelihood.

Proof in the Pudding - Empirical Results of our Approaches

If we let centroid recovery be the primary purpose of our algorithm, then let s be the set of centroids in the one-dimensional case and s' be the recovered set. Clearly to compare them one should sort them first. Indeed, consider any two vectors a, b. Let a' be any valid permutation of the indices of a and b' be one of b. Suppose we wish to minimize the quantity

$$d = ||a' - b'||^2 = ||a'||^2 + ||b'||^2 - 2\langle a', b' \rangle$$

But the first two quantities in the RHS are fixed and indeed, norms are not changed by permuting indices. The quantity $2\langle a', b' \rangle$ is maximized over all permutations of indices of a, b when both a', b' contain indices in sorted order (this is called the rearrangement inequality). Hence, we are assured that to compare different centroid recovery, all we need to do is to sort the vectors and compute their norms.

Our framework uses $K = 3$ GMMs with means of $1, 10, -5$. For all clusters we use the same σ. There is a modest improvement with likelihood thresholding in terms of Gaussian noise of unit

variance, that grows rapidly as we choose $\eta = 0.05$ fraction of points at random and corrupt them with uniform noise in a high range, such as $[16, 17]$. The robust algorithm is seen to discard these points correctly.

Table 1: L2 results on 3-GMM

Algo	Zero mean	Uniform noise	Non-zero-mean Gaussian
EM	6.78	18.17	14.72
Threshold	3.84	16.52	11.68
RREM	7.45	6.822	8.87

The mean for the Gaussian used above was 4. In general, we found that previous empirical approaches which were designed for Gaussian noise failed drastically as soon as the noise was allowed to assume a non-zero mean.

4 Analysis of the Algorithm for Soft-EM

Consider the entire region of the dataset as existing within a big normed ball of radius B. We can suppose that the uniform prior exerts an effective density of $\frac{1}{V_B}$ over this entire region, and that the GMMs are K ellipsoidal regions within the ball that "capture" points. In short, consider a model that is very much like K massive objects that capture elements gravitationally as a rough physical analogy.

Where then might the adversary choose to place his points? Suppose any Gaussian likelihood cluster exists. A point placed at a distance d shifts the centroid by an amount proportional to:

$$\frac{d}{\sqrt{2\pi\sigma^2}} \exp(\frac{-d^2}{2\sigma^2})$$

By differentiating the above expression, we obtain that for maximal shifting, the adversary ought to place his points at a distance σ. This happens to be a well known result (BDLS17). Intuitively, the points cannot look too off-for then any outlier filtering would throw them away, and neither can they be too close to the centroid. Plugging in the above value and noting that at most ηN points are placeable, we get that the shift is

$$\frac{\eta N}{c_{min}\sqrt{2\pi e}}$$

where c_{min} is the size of the smallest cluster. We thus get that the maximal shift in a centroid is **linear in** η. This is a very strong result, analogous to bounds for Gaussian recovery (LRV16) and similar to Huber's ϵ-class problems (H^+65). In fact, based on this, we may heuristically conjecture that for every i, j the clusters K_i, K_j ought to be separated by at least

$$\frac{\eta N}{\min(|K_i|, |K_j|))\sqrt{2\pi e}}$$

The above framework is explicitly based on a Soft-EM variant, where each point contributes to its cluster proportional to its likelihood. But effectively, this is often replaced by **Hard-EM**, where each point is assigned to its likeliest cluster. We now consider an attack on Hard-EM.

Analyzing a Stacking Attack on Hard-EM

We call an adversarial attack as a **stacking attack** if all the points are placed in one place, thereby creating a very highly weighed region in the space that centroids will converge to. From our previous analysis, we are interested in the case where :

- The Gaussians are K ellipsoidal regions
- The centroid initialization is reasonably close to the actual ellipsoid centres

- The adversary has, for some cluster c, put all his points at a distance of σ from its center. Assuming all the eigenvalues of the Gaussian covariance matrix to be roughly similar, this basically means that the distance is $O(\lambda_i)$, and so throughout the rest of the analysis, **we will take σ to mean a quantity that well approximates all the λ_i simultaneously**, and that in general, our analyses **will fail when the covariance matrix is pathological and has wide variance in λ_i**. One solution to this can be the pre-scaling of data to unit variance across dimensions.

It's clear to see this kind of setup defines a "dangerous" region within the ellipsoid, where, if the centroid is initialized, it will end up sticking to the adversarially placed points. Let the centroid at a timestep be at c_t. Then, the situation is thus :

- The centroid c_t is not at its correct position c_{true}. However, it still associates a likelihood over the points that belong in the correct cluster. This likelihood grows as c_t is brought closer to c_{true}.
- Opposing this, there is a likelihood that c_t includes all the outliers. This likelihood increases as c_t is brought closer to where all the ηN points have been put - at say P. If c_t starts getting too close to P, this likelihood will begin to dominate, drawing c_t ever closer. We seek to demarcate a safe zone.

The Acquisition and Spiral conditions

Let the extra cluster associate a likelihood L to all its points. If there are a total of N points then by definition, the GMM center i will begin acquiring the outliers if

$$\pi_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp(\frac{-d^2}{2\sigma^2}) \geq 1/N$$

The above is sufficient to start capturing points from the extra cluster and can be seen as an **acquisition condition**. It essentially says that we can reassign the outliers to c_t from the extra cluster. But this has to set off an increasing likelihood chain, a **spiral condition** that brings c_t ever closer to the outlier point-stack P. As this process goes on, c_t will begin losing its own cluster points (on the opposite side of the ellipsoid from P) to the $K+1$-th cluster (note that we assume that all other Gaussians are far enough to not capture). So the spiral condition will be such a condition that, if satisfied, will guarantee the following :

- Setting $d_t = ||c_t - P||$, we will find d_t as a nonincreasing sequence
- Decompose $LL(c_t) = LL_C(c_t) + LL_O(c_t)$, where the subscript C denotes the cluster points and O the outliers. We should find LL_C as a nonincreasing sequence and LL_O as a nondecreasing one.

Begin by solving the first equation. π_i is just c_i/N, where c_i is the number of elements in cluster i. It will simplify our assumptions greatly to assume that no matter where we initialize the centroid within the ellipsoid, c_i is not affected (other than by acquiring outliers), or to put it simply, all the Gaussian ellipsoids are far enough from each other to not capture each other's points. Upon solving, we get that :

$$d^* = ||c_t - P|| = \sigma\sqrt{2\log(\frac{c_i}{\sigma\sqrt{2\pi}})}$$

We already have some sort of a bound here, since this is a necessary condition to set off a spiral, and as long as our initialization is at least d^* away nothing too bad can happen by a stacking attack. Now, we need to see how bad it can get in case we end up within this zone. Clearly, one of two things can happen :

- The c_t sequences move towards P and stay around it
- The c_t sequences move **past** P (which is at a distance σ from c_{true}) and keep going away

The second scenario is clearly unfeasible. To demonstrate this, consider the straight line from c_{true} to P and take the projection of c_t on this line. Clearly replacement by the projection will not hurt the objective value i.e. LL. The rest follows by observing that on this line the LL function is monotonic decreasing past P. (Of course, the problem with this argument shows up as c_t starts losing membership of far-off points)

Spiraling Case 1 : Partial

Solving for a spiral condition is tricky, but, can be simplified by noting the following. In the case that $\eta N < c_i$, there is no way for c_t to completely become loose from its cluster elements, and so, unlike total spiraling, we can ignore the question of whether the whole thing is possible and instead seek to bound the worst case. Let $c_i - \eta N = r_i$ be the minimal number of points that must remain allocated to c_t from c_i. We now seek to obtain the quantity :

$$\min d_t = ||c_t - P||$$

subject to $c_t = \frac{\eta N P + \sum_{i=1}^{r_i}(p_i)}{\eta N + r_i}$ and $p_i \in$ cluster i. With high probability, this is equivalent to decomposing the problem into computing the centroid of a cut-off ellipsoid and adding ηN worth of weighted points at P. Flatten into an equivalent single dimension problem and we are left with computing the quantity :

$$\int_k^\infty \frac{z}{\sqrt{2\pi\sigma^2}} \exp(\frac{-z^2}{2\sigma^2}) dz$$

with k satisfying the equation

$$\int_{-\infty}^k \frac{1}{\sqrt{2\pi\sigma^2}} \exp(\frac{-z^2}{2\sigma^2}) dz = \frac{\eta N}{c_i}$$

Unfortunately k will become an inverse Φ function now, which greatly complicates things and makes the calculation infeasible for the generic case. But we can have one interesting result. Consider $\frac{\eta N}{c_i} = 0.5$, which readily yields $k = 0$. The one-sided Gaussian expected value is known to be :

$$\int_0^\infty \frac{z}{\sqrt{2\pi\sigma^2}} \exp(\frac{-z^2}{2\sigma^2}) dz = \sigma\sqrt{\frac{2}{\pi}}$$

This yields c_t as the mean of $\sigma\sqrt{\frac{2}{\pi}}$ and σ (since ηN is half of c_i, r_i will also become ηN). And so, we end up with, **in high probability**:

$$||c_{true} - c_t|| \approx \frac{\sigma}{2}(1 + \sqrt{\frac{2}{\pi}}), ||c_{true} - c_t|| \leq \sigma(1 + \sqrt{\frac{2}{\pi}})$$

What are our confidence bounds on this? Since everything we did above holds in expectation, we can use McDiarmid's (other inequalities seem less applicable here). At this point we have to invoke an effective "radius" of R for the ellipsoids to have any form of stability, in which case we have stability with $\frac{2B}{r_i}$ i.e. $\frac{4B}{c_i}$. This still gives us the $O(-c_i\epsilon^2)$ bound we seek. Finally, we can look back and observe that being able to corrupt upto half of cluster entries - a dealbreaker for zero variance - only costs us centroid movements up to $O(\sigma)$ here! Unfortunately, we had to invoke a radius term, but there seems to be no way without it.

Spiraling Case 2 : Pathological Cases of Total Spiraling

Total spiraling requires the c_t sequence to fall into P with $\eta N > c_i$. The initial points in the cluster go to cluster $K + 1$. Is such a case even possible? This question equally applies for partial spiraling, but there we did no such analysis and got bounds anyway. We argue that it is, and provide a pathological case where this happens.

Consider an ellipsoidal surface E for a Gaussian cluster i. Let us presuppose three steps :

- The cluster mean c_t is initialized close to P.
- The ηN points become part of c_t. c_t shifts closer to ηN. In the process, some points $\in$ cluster i begin to fulfill the condition that their likelihood under the uniform cluster $\geq$ their likelihood under c_t. These points are now lost.
- c_t loses the above points forming c_{t+1}. The effective cluster now centered close to P finds itself with a lower variance than at step t.
- Due to the lower variance, c_{t+1} begins discarding any points that are too far-off. This will, in pathological cases, include **all the remaining points of cluster** i.

Construct this pathological case as follows. Let the ellipsoid be a sphere, and let every point lie on the surface of the sphere, such that at the surface, we have that:

$$LL_c(p_i) = LL_{K+1}(p_i)$$

That is, every point on the surface is **just** held on to by the GMM cluster at a distance of σ. Let the cluster have c_i points. Put $\eta N = c_i$ points $\frac{\sigma}{2}$ away from the centroid. This causes a shift of $\frac{\sigma}{4}$ in the centroid. What is the new σ' ? Invoke the result that:

$$E(X^2) = E(X - E(X))^2 + (E(X))^2$$

We have shifted the centroid $E(X)$ by $\frac{\sigma}{4}$. With a little computation, we end up at :

$$Var_{new} = 0.5 * \frac{\sigma^2}{16} + 0.5 * \frac{17\sigma^2}{16} = \frac{9\sigma^2}{16}$$

$$\sigma' = \frac{3}{4}\sigma$$

Now, this instantly cuts off the entire sphere surface : the $\sigma' = \frac{3}{4}\sigma$ and the centroid is shifted by $\frac{\sigma}{4}$, so there is exactly one point on the sphere still at a distance σ' which is exactly how far the GMM can hold on to. Effectively, this gives us that the adversary can shift the c_t by $\sigma/2$ generally, and there seems to be little we can do about it. The next step will sink the σ to 0 and fix c_t on P permanently.

References

[BD09] Thomas Blumensath and Mike E Davies, *Iterative hard thresholding for compressed sensing*, Applied and computational harmonic analysis **27** (2009), no. 3, 265–274.

[BDLS17] Sivaraman Balakrishnan, Simon S Du, Jerry Li, and Aarti Singh, *Computationally efficient robust sparse estimation in high dimensions*, Conference on Learning Theory, 2017, pp. 169–212.

[BWY+17] Sivaraman Balakrishnan, Martin J Wainwright, Bin Yu, et al., *Statistical guarantees for the em algorithm: From population to sample-based analysis*, The Annals of Statistics **45** (2017), no. 1, 77–120.

[FY+10] Davide Ferrari, Yuhong Yang, et al., *Maximum lq-likelihood estimation*, The Annals of Statistics **38** (2010), no. 2, 753–783.

[H+65] Peter J Huber et al., *A robust version of the probability ratio test*, The Annals of Mathematical Statistics **36** (1965), no. 6, 1753–1758.

[JTK14] Prateek Jain, Ambuj Tewari, and Purushottam Kar, *On iterative hard thresholding methods for high-dimensional m-estimation*, Advances in Neural Information Processing Systems, 2014, pp. 685–693.

[LRV16] Kevin A Lai, Anup B Rao, and Santosh Vempala, *Agnostic estimation of mean and covariance*, Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on, IEEE, 2016, pp. 665–674.

[QP13] Yichen Qin and Carey E Priebe, *Maximum l q-likelihood estimation via the expectation-maximization algorithm: A robust estimation of mixture models*, Journal of the American Statistical Association **108** (2013), no. 503, 914–928.

[TL00] Saldju Tadjudin and David A Landgrebe, *Robust parameter estimation for mixture model*, IEEE Transactions on Geoscience and Remote Sensing **38** (2000), no. 1, 439–445.

[Wu83] CF Jeff Wu, *On the convergence properties of the em algorithm*, The Annals of statistics (1983), 95–103.

[YLL12] Miin-Shen Yang, Chien-Yo Lai, and Chih-Ying Lin, *A robust em clustering algorithm for gaussian mixture models*, Pattern Recognition **45** (2012), no. 11, 3950–3961.