Report on the reproduction of Gaussian Mixture experiments

Sergey Pavlov

2016

The original article with the experiments is:

https://papers.nips.cc/paper/5229-distributed-estimation-information-loss-and-exponential-families.pdf.

What is done

1. Gaussian mixture model is applied to MNIST dataset using 4 types of parameters estimators: Global MLE, Local MLE, KL-Average, Linear-Average. This is made for different dataset size;
2. Each model is evaluated on it's train dataset and on testing dataset;
3. Each model is evaluated on randomly chosen data and on the data ranged by MNIST-labels(0,1,2...);
4. Quality of each model is estimated by log-likelihood function applied to train/test dataset.

Next I will give my explications for each of these cases.

Data partition

In order to reduce the dimensionality of the data, I use PCA, that chooses 100 main components. PCA is applied to all dataset(train, test) and after is reshared into train and test.

MNIST dataset contains 60.000 entries in the training test and 10.000 entries in the testing set.

Data is not ranged by MNIST-labels. So in order to construct a randomly chosen dataset of the size N I take randomly N entries and **always** divide them into 10 groups.

To construct a label-wise partition, I first range all 60.000 entries in training dataset into 10 groups according to its' labels(0,1,2...) and after I randomly form 10 groups evenly(each by N/10 elements) according to this label-partition.

General Issues

As told in the article, GMM-model contains 10 mixture components. To avoid the noise caused by randomness, I average each result for each type of model over 10 trials(I don't make for 100 trials as in the article because it's too long).

I average each logloss-result over 10 local estimators (that's why all my graphs are divided by 10 as you will see later).

Global MLE

Model is simply trained on the all dataset of the needed size regardless data partition. It's made for random or label-wise choice of data.

Local MLE

This is not really a model of parameter estimation. This is just average of log-likelihoods of 10 local models. It's made for random/label-wise choice of data and computed for training and testing dataset as TESTING DATASET.

When counting for the training dataset, each model computes log-likelihood on ALL training data(containing N entries, but not only on it's training part of N/10 entries).

KL-Average estimator

To calculate the $\hat{\theta}^{KL}$ from 10 local models. To do this, I use the technique of bootstrap, described in the article. I take 500 random vectors from each of 10 models, I form from these entries a dataset Y containing 5000 entries. After I fit a non-supervised GMM model M. It can be proved, that model M converges to the model with parameter $\hat{\theta}^{KL}$ as number of entries falls to the infinity.

Linear Average(naive)

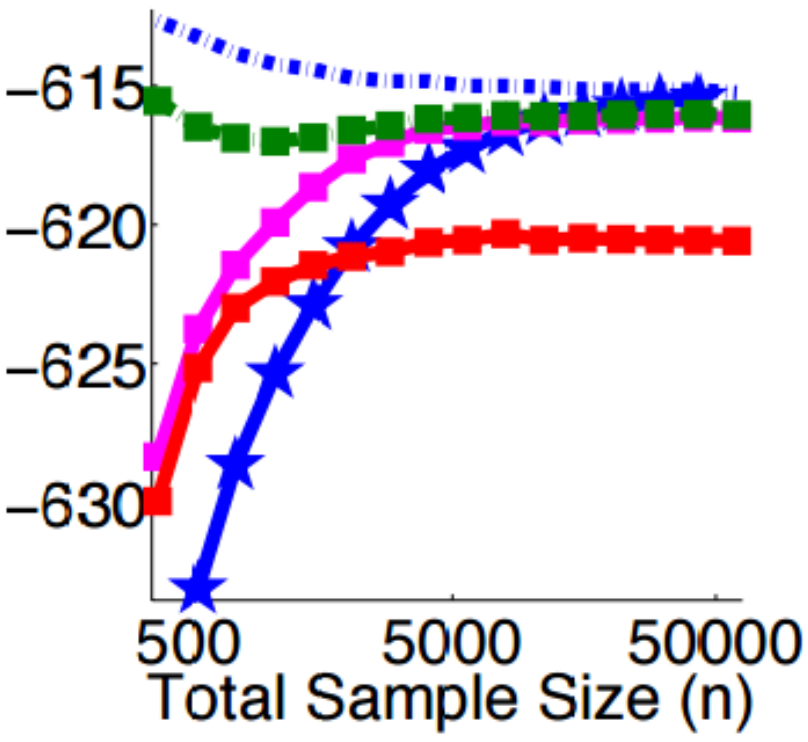
This model is to take an average of all the parameters of 10 local models. These averages form another model that is considered to be optimal. The problem here is that all components in GMM can be mixed(we don't know apriori the order of the components of components for each of local GMM models), so it's not very clever to naively average in this case.

In the article it's proposed to tackle this issue by "considering a matched linear average that first matches indices by minimizing the sum of the symmetric KL divergences of the different mixture components".
This part is not made by me, because I haven't precisely understood how this should be done.

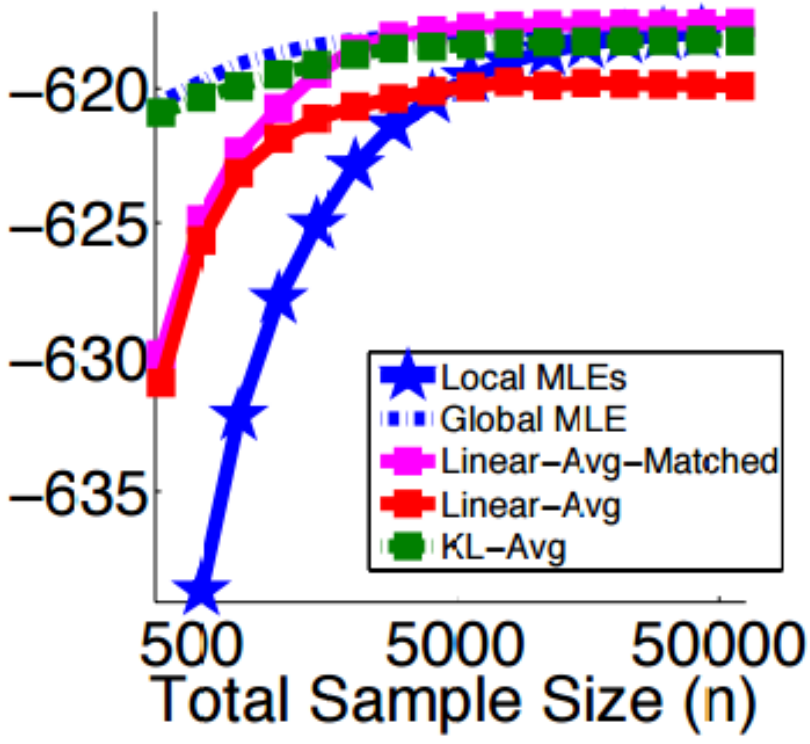
That's why you'll not see pink line on my graphs corresponding to this matched-Average estimator.

Results for random partition

These are results from the article:



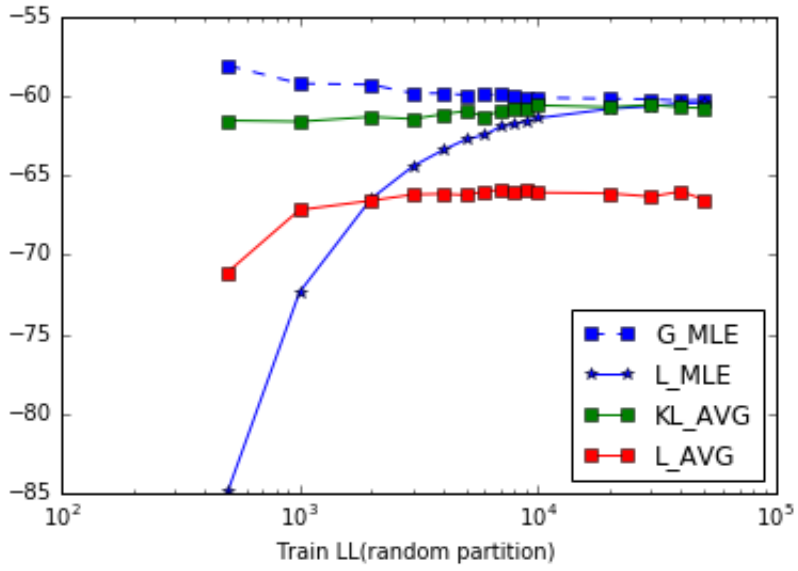
(a) Training LL
(*random partition*)

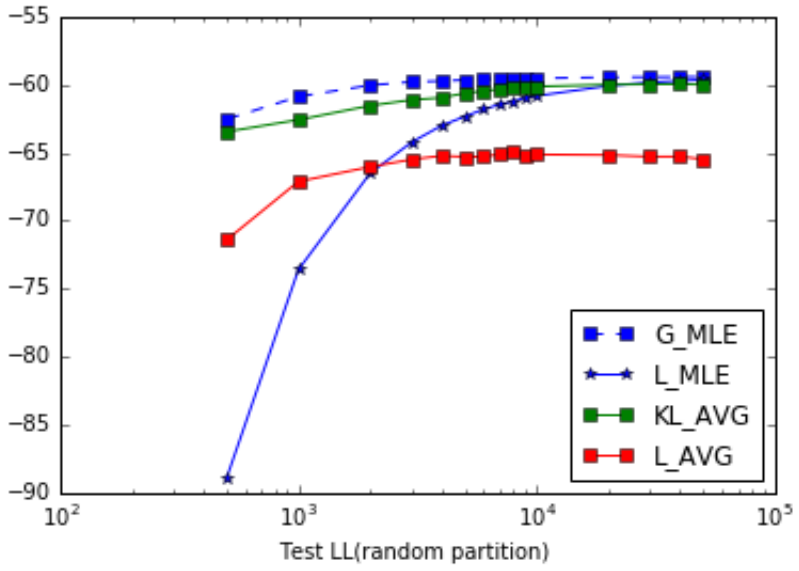


(b) Test LL
(*random partition*)

Results for random partition

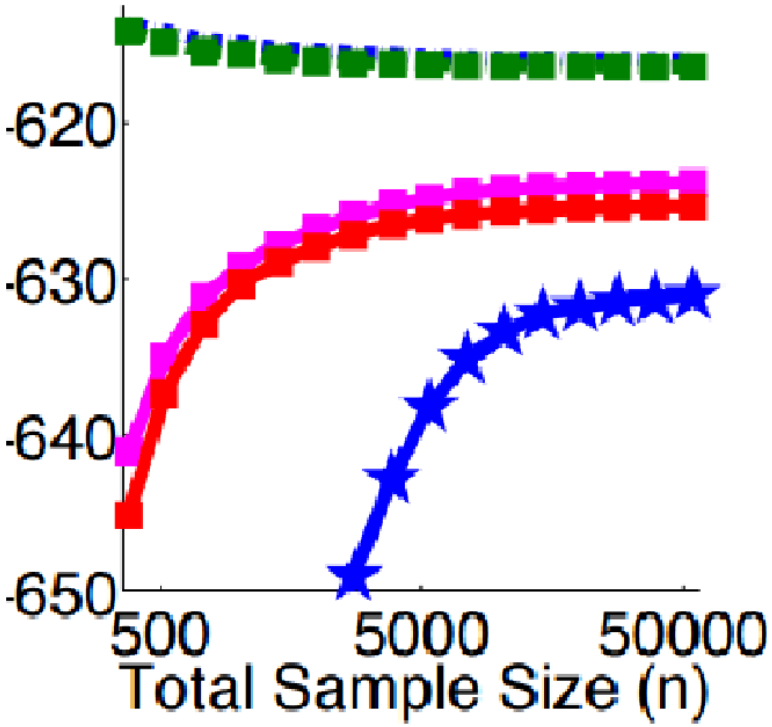
These are my results:



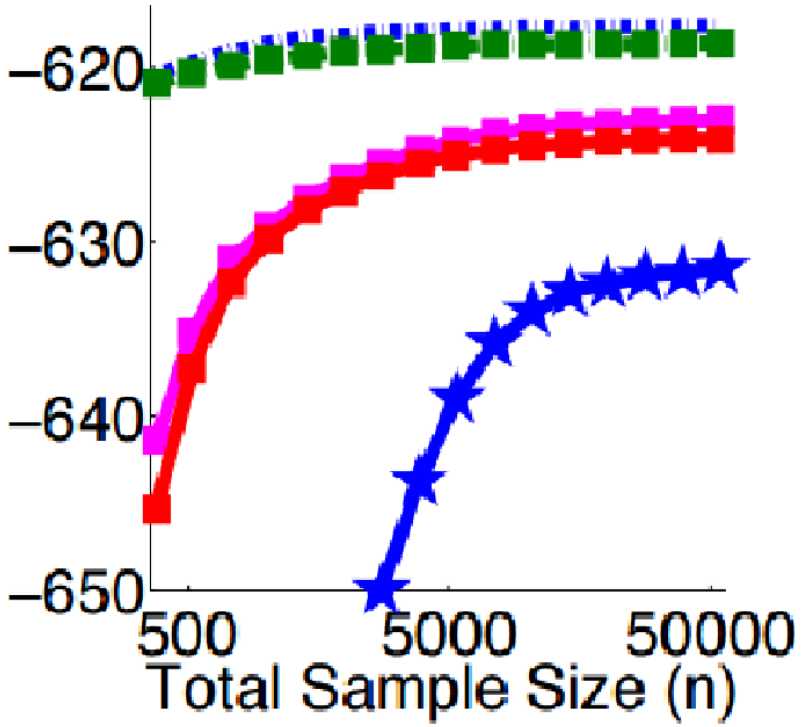


Results for label-wise partition

These are results from the article:



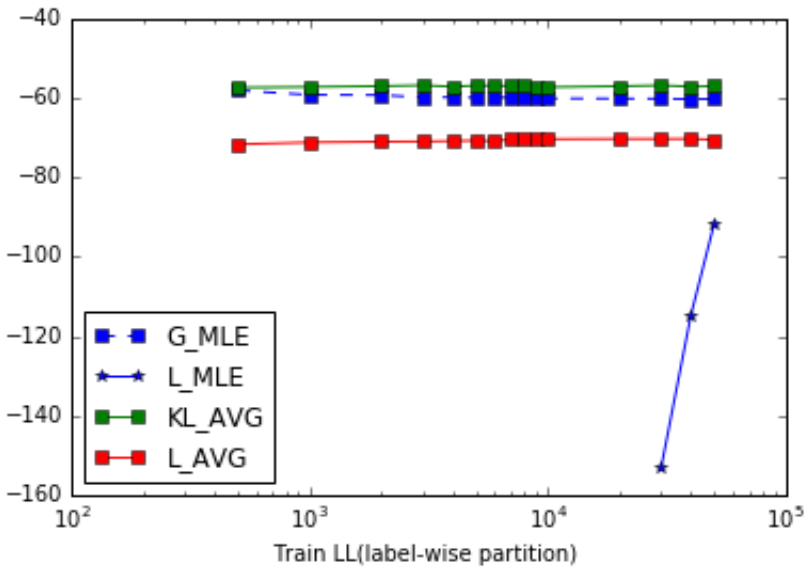
(c) Training LL
(*label-wise partition*)

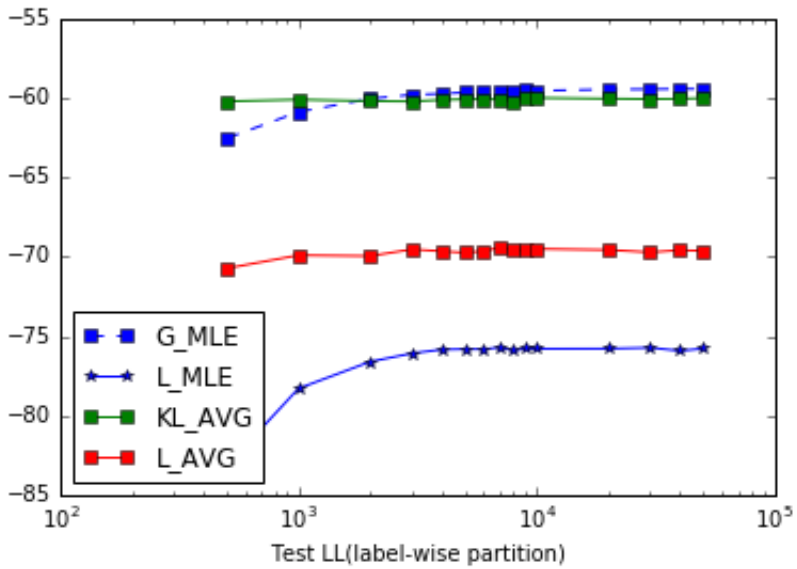


(d) Test LL
(*label-wise partition*)

Results for label-wise partition

These are my results:





We'll apply the α-integration to our problem. We'd like to replace KL by different divergencies generalised by α - divergency for different values of α. More precisely, we train 10 models as before independently. We need to solve the next problem:

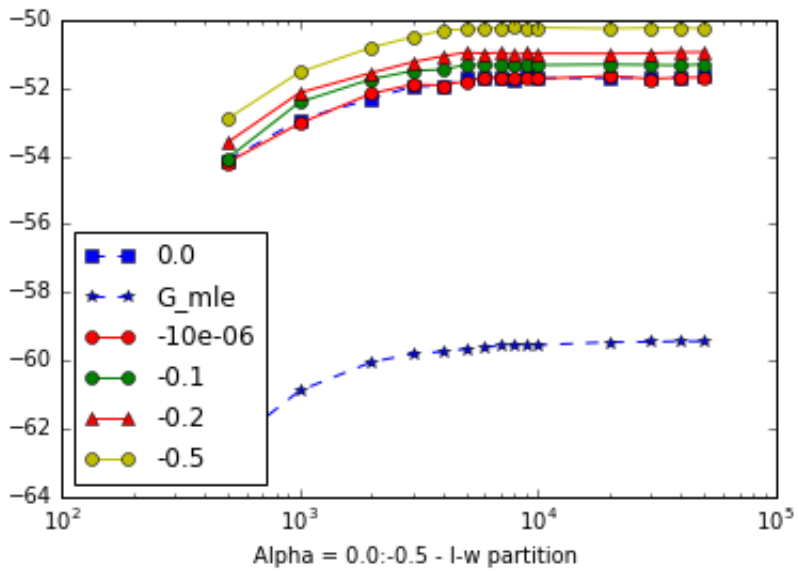
$$\min_{q(s)} R_\alpha[q(s)] = \min_{q(s)} \frac{1}{n} \sum_{i=1}^{10} D_\alpha(p_i(s)||q(s))$$

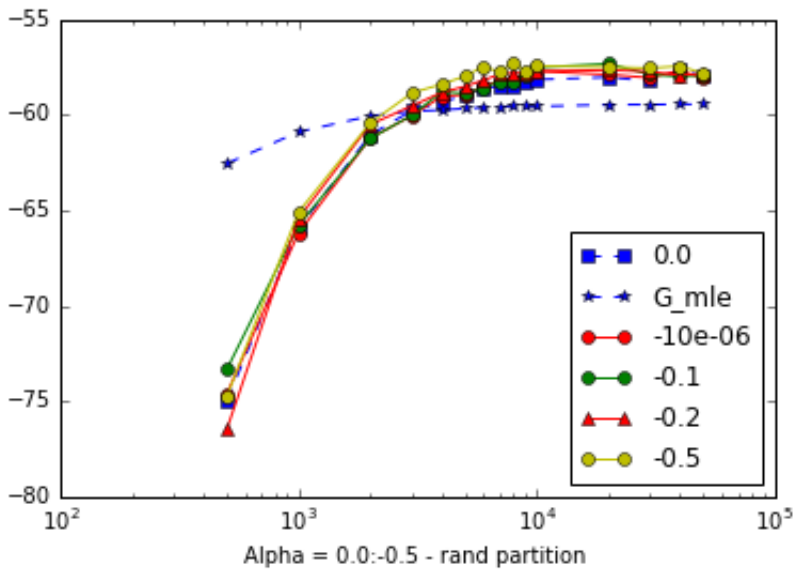
The solution is given by [1]:

$$q(s) = f_\alpha^{-1}\left(\frac{1}{n}\sum_{i=1}^{10} f_\alpha(p_i(s))\right)$$

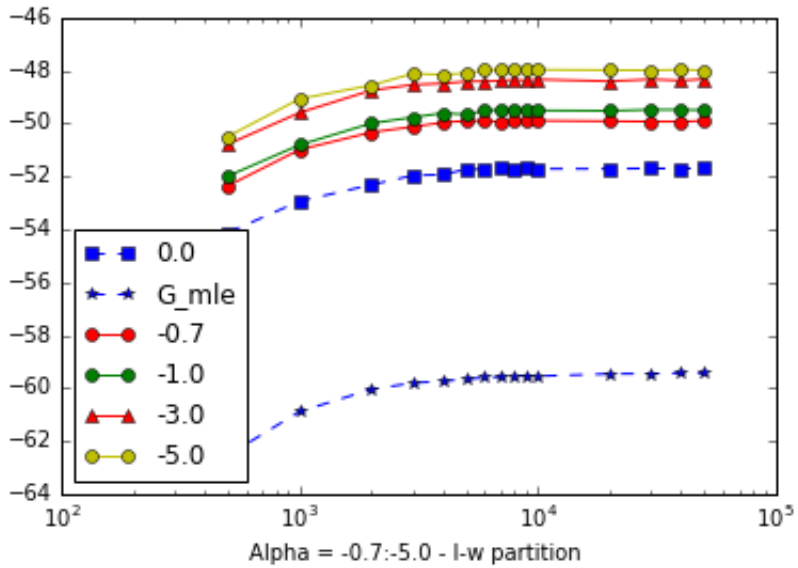
We understand that this can be non-normalised but we still need to calculate the log-loss of this model on the test set. So for each pair (X_i, y_i) in the test set we calculate the probabilities $p_j(X_i, y_i), 1 \leq j \leq 10$ and we use them to calculate $q(X_i, y_i)$. After that we calculate the log-loss of q.

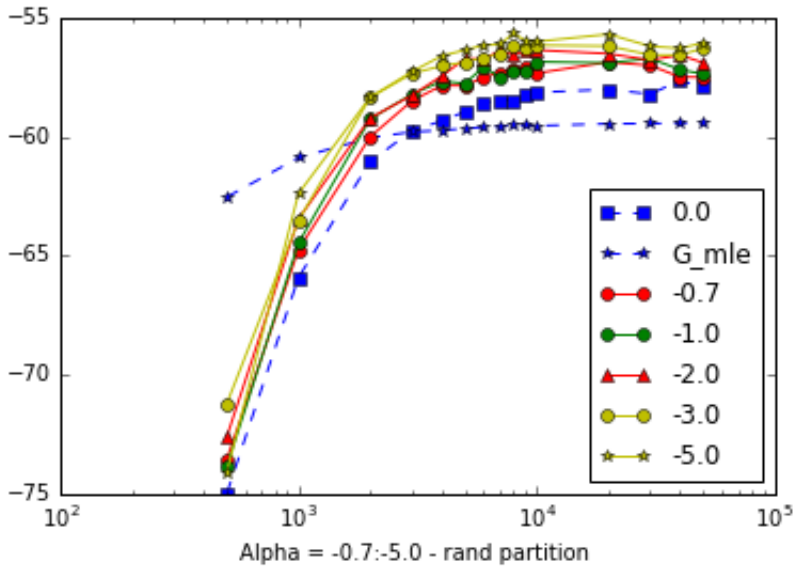
Results for the α - integration. We use random partition of data and label-wised(GMM number i is responsible for i-labeled instances). On the left we have the results for the label-wised partition and on the right - for the random partition.



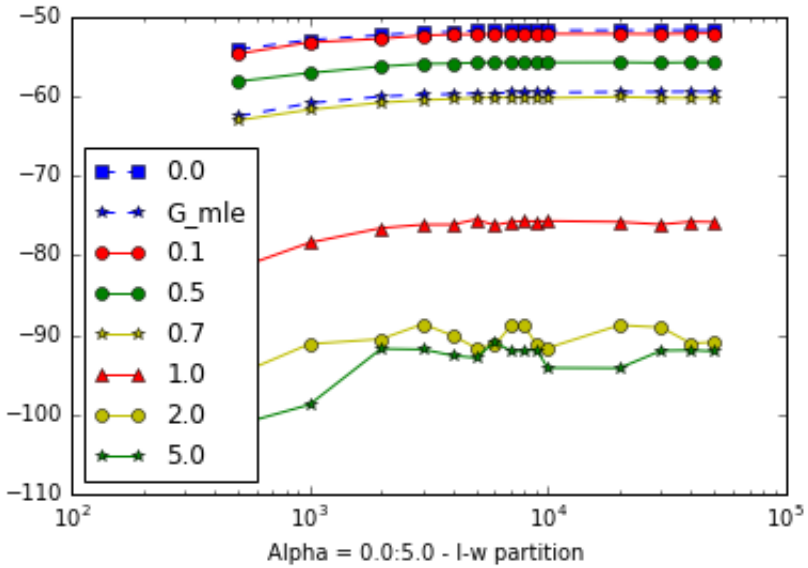


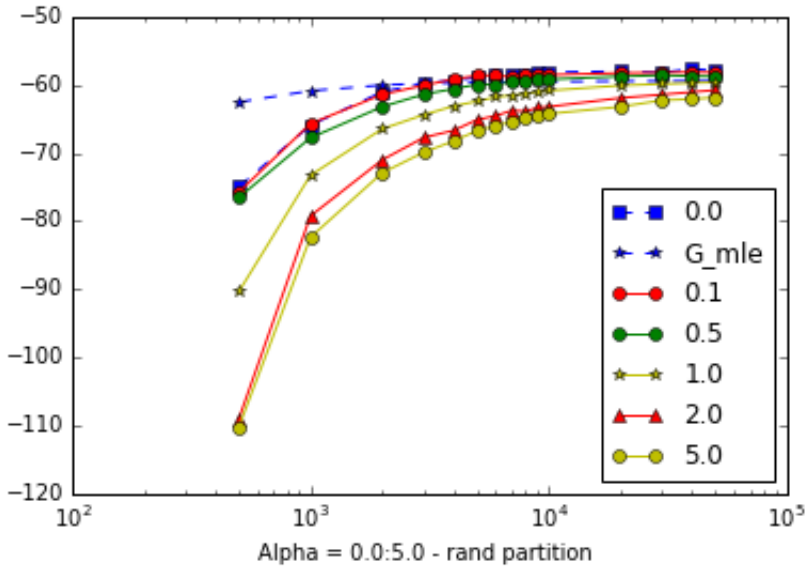
Here we can see that the results for $D_{-1}(p|q) = KL(p|q)$ (in terms of [1] !!!) are shifted up in comparison with the results for KL obtained by bootstrap(there KL-line was close to to Global-mle). And, probably, it's strange to have the results above the Global-mle. So we really need the normalisation.





Here we see that for label-wised partition the value $\alpha = 1.0$ is critical and changes the situation.






Amari Shun-ichi.
Integration of Stochastic Models by Minimizing α-Divergence.
2007.