

Semi-supervised Hidden Markov Random Fields (HMRF) Kmeans. Theory to Implementation

Dimitrios Pritsos and Efstathios Stamatatos

University of the Aegean
Karlovassi, Samos – 83200, Greece.
{dpritsos, stamatatos}@aegean.gr

1 Introduction

The objective of the Semi-supervised Clustering is to incorporate in the procedure of clusters discovery or assignment, the prior knowledge about the skeleton of the clusters schema. There are several efforts on Semi-supervised model inference in both Expectation Maximization (EM) clustering based models and in Agglomerating clustering based models. According to [3] there three main groups of EM based semi-supervised clustering methods:

1. *Constraints-based methods* are using the provided supervision for guiding the algorithm towards a data partitioning which is avoiding (but not preventing) the constraints violation.
2. *Distance-based approaches* in clustering method with a particular distance function; the distance function is parametrized and the parameters values are learned to satisfy the constraints.
3. *Semi-supervised clustering based on Hidden Markov Random Fields* where the constraint-based and distance-based approaches are combined into a *unified probabilistic model*.

In EM clustering based models there have been several efforts where the labeled data were provided in the clustering model in the form of data-labels pairs or in the initialization phase of the clustering model. In this work we present the Hidden Markov Random Fields Model (HMRF) Kmeans, where the prior knowledge about the structure or skeleton of the clusters schema has been embedded into the model in the form of constraints [1]. The HMRF Kmeans is a hard-clustering model due to the *hard* assignment of the data point to one of the a-priori fixed number of clusters. However, the same models can be transformed into a relatively easy soft-clustering model where of each data point only Maximum a-posteriori Probability (MAP) of the point to be a member of each cluster of the final schema.

The HMRF Kmeans Semi-supervised clustering method it has been implemented, in this work, for being tested on the Web Genre Identification (WGI) Information Retrieval (IR) taxonomy problem. Therefore, here we only present the model inference procedure where the distance measure, a.k.a distortion function/measure, is the cosine similarity because in the IR literature is the distortion measure where in most cases maximizes performance in problems where the

feature space is particularly large, as in this case. In case one would be interested in changing the model to a soft-clustering method then the probability density function of the final model should have been chosen to be some properly parametrized Von Mises Fisher distributions.

Since this work is focusing on the implementation of Semi-supervised HMRF-Kmeans in the IR domain framework, it has to be noted that this Semi-supervised model advantage is the interactive learning setting where this model can be used [3], since the constraints are provided to the model in two different sets the *Must-Link* and *Cannot-Link*. These sets are not necessarily the same in size or complementary one to the other.

What it follows is the model inference line of thought based on the there resources [1, 3, 2].

2 Model inference

The objective of a semi-supervised model like HMRF Kmeans is to drive the procedure of clustering schema taking into account the prior knowledge we have about the clusters in the form of *must-link* and *cannot-link* constraints. The main difference in the graphical model of a topical EM algorithm is the nodes of the observed labeled data over the hidden, a.k.a latent, variables as depicted in fig.1, with red colored arrows and gray shaded nodes.

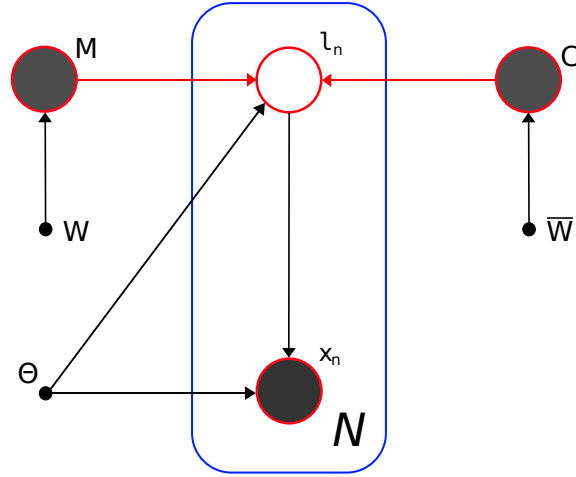


Fig. 1. Semi-supervised HMRF Expectation Maximization or Kmeans Clustering Graphical Model.

The goal of EM is to maximize the *log likelihood function* $P(X|\Theta)$ with respect of Θ as in eq.1.

$$\ln P(X|\Theta) = \ln \left\{ \sum_l P(X, L|\Theta) \right\} \quad (1)$$

Where $X = \{\mathbf{x}_i\}_{i=1}^N$ is the set of *observable random variables* given by the conditional probability $P(X|\Theta)$ and \mathbf{x}_i is a random vector (or data point) of the corpus under taxonomy. Note the boldface notation of the random vector in order not to be mixed with x_i which will a feature (or variable) of this vector. Moreover, N is the number of vector as depicted in the graphical model of fig.1.

Due to the relatively complex marginal distributions, like $P(X|\Theta)$, over observed data points where they are computationally intractable, there is a common practice to incorporate *latent or hidden variables* in order to express the conditional probability calculation more tractable over the expanded space of observed and latent variables. In eq.1 L is the set of hidden (or latent) variables over X observable data points [2].

Therefor a *hidden field* $L = \{\mathbf{l}_i\}_{i=1}^N$ random variables, *whose values are unobservable*. In the clustering framework, the set of hidden variables are the unobserved cluster labels on the points, indicating cluster assignments. Every hidden variable \mathbf{l}_i takes values from the set $1, \dots, N$, which are the labels of the clusters.

Now every random data point \mathbf{x} can be generated from a conditional probability distribution $P(x_i|\mathbf{l}_i)$ determined by the corresponding hidden variable \mathbf{l}_i . Note that we know a-priori that the random data points X are conditionally independent given the hidden variables L . Thus $P(X|L) = \prod_{\mathbf{x}_i \in X} P(\mathbf{x}_i|\mathbf{l}_i)$. Note that \mathbf{l}_i can be either a vector or a singleton depending on the algorithm setting, either for soft-clustering or for hard-clustering respectively. Thus value and vector for \mathbf{l}_i will be used interchangeably in this text.

The set of $\{X, Z\}$ is *the complete data set*, while the $\{X\}$ is the incomplete data set. The likelihood function for the complete data set simply takes the form $\ln p(X, Z|\Theta)$ as shown in eq.1, and theoretically that maximization of this complete data log likelihood function is straightforward. However, in practice we are not given the complete data set, but only the incomplete data points X . Thus, our only knowledge of for the values of the latent (hidden) variables in L is given by the posterior distribution $p(Z|X, \Theta)$. Because we don't have available the complete-data log likelihood, we consider instead its expected value under the posterior distribution of the latent (hidden) variables.

In eq.1 and in fig.1 there are some parameters Θ which are governing the initial and the final schema of the PDF's mixture. These are the parameters we have to find computationally and where in employee EM algorithm (alg.1) with the following general and distribution inexpedient (i.e. irrespectively where the PDF's are Gaussian, Von Mises Fisher etc). In particular with EM we are interactively finding the proper set of Θ parameters by calculating the expected posterior distribution $P(L|X, \Theta)$ of the latent (hidden) variables L .

Algorithm 1.1: The generic for of Expectation Maximization either for both Soft- and Hard-clustering

Data:

\mathbf{X} observable data points.

\mathbf{K} possible states clusters we expect to be existing in the data-set.

$\mathbf{L}_{i=1}^K$ the hidden field variables (or vectors) with values $\{1, \dots, \mathbf{K}\}$ or $(0, 1]$

Θ an initial state about the PDF mixture model.

Result:

$\mathbf{l} = \{\mathbf{p}_k\}_{k=1}^K$ where \mathbf{p} can be either $\{0, 1\}$ or $(0, 1]$ depending on the soft-clustering or hard-clustering set-up.

Θ the final set of parameters after the Probability Density Functions Mixture.

```

1 Choose an Initial setting for  $\Theta^{OLD}$ ;
2 while  $I$  iterations reached do
3   1. E-step Evaluate  $P(\mathbf{L}|\mathbf{X}, \Theta^{OLD})$ ;
4   2. M-step Evaluate  $\Theta^{NEW}$  given by (a) and (b);
5       a.  $\Theta^{NEW} = \arg_{\Theta} \max \Omega(\Theta, \Theta^{OLD})$ ;
6       b.  $\Omega(\Theta, \Theta^{OLD}) = \sum_{\mathbf{L}} P(\mathbf{L}|\mathbf{X}, \Theta^{OLD}) \ln P(\mathbf{X}, \mathbf{L}, \Theta)$ ;
7   if log likelihood convergence reached then
8     | Breaking the loop and Ending the Clustering;
9   else
10    |  $\Theta^{OLD} \leftarrow \Theta$ ;
11  end
12 end

```

The EM algorithm can also be used to find MAP (maximum a-posterior) solutions in case we have a good knowledge about the prior distribution $P(\Theta)$ over the parameters Θ . In this case the E-step remains the same as in the maximum likelihood case, while in the *M-step* (b) the $\Omega(\Theta, \Theta^{OLD}) + \ln P(\Theta)$.

In HMRF-Kmeans derivation process we are starting with the PDF mixture we would like to maximize by exploring the EM for the reasons explained above, as shown in eq.2.

$$P(\mathbf{X}, \mathbf{L}, \Theta | \mathbf{M}, \mathbf{C}) = P(\Theta | \mathbf{M}, \mathbf{C}) P(\mathbf{L} | \Theta, \mathbf{M}, \mathbf{L}) P(\mathbf{X} | \mathbf{L}, \Theta, \mathbf{M}, \mathbf{C}) \quad (2)$$

Where the set of vector are the same as in alg.1 but this time the constraints set of the fig.1 have been included, i.e. Mast-link and Cannot-link constraints set $\{\mathbf{M}, \mathbf{C}\}$. Moreover, as the graphical model is describing the constraints are independent from \mathbf{X} and parameters Θ are, also, independent from the constraints set. Thus:

$$P(\mathbf{L} | \Theta, \mathbf{M}, \mathbf{L}) P(\mathbf{X} | \mathbf{L}, \Theta, \mathbf{M}, \mathbf{C}) = P(\mathbf{X} | \mathbf{L}, \mathbf{M}, \mathbf{C}) \quad (3)$$

$$P(\Theta | \mathbf{M}, \mathbf{C}) = P(\Theta) \quad (4)$$

Considering \mathbf{X} observable data-set is convenient due to computational issues to simplify the algorithm by assuming the vectors \mathbf{x} are *mutually impediment*, thus:

$$P(\mathbf{X}|\mathbf{L}, \Theta) = \prod_{i=1}^N P(\mathbf{x}_i|\mathbf{l}_i, \mathbf{M}, \mathbf{L}) \quad (5)$$

Consequently, from equations 2, 3, 4 and 5 we are getting the following MAP which it should be maximize.

$$P(\mathbf{X}, \mathbf{L}, \Theta|\mathbf{M}, \mathbf{C}) = P(\Theta)P(\mathbf{L}|\Theta, \mathbf{M}, \mathbf{C}) \prod_{i=1}^N P(\mathbf{x}_i|\mathbf{l}_i, \mathbf{M}, \mathbf{L}) \quad (6)$$

At this step of the clustering method building process we have to decide whether the clustering would be soft or hard. This decision has two consequences. Firstly, is related to the \mathbf{L} latent variables type and range of values as shown in alg.1, i.e. whether \mathbf{l} will be a variable or a vector of variables and whether its values will be probability estimates or 0,1 depending whether or not a data-point is belonging to the cluster \mathbf{k}_i . Secondly, is related whether the algorithm will be probabilistic based or distance based, i.e. whether a MAP will be maximized or an objective function with a specific distance measure (a.k.a distortion function/measure) will be minimized.

In our case go for the *distortion function* path, as the Kmeans term of the algorithm implies. As explained above since we are focusing on IR domain problems we are going to show the complete algorithm building process for the cosine similarity as the distortion function of our choice.

Each hidden random variable \mathbf{l}_i has an associated set of neighbors $\Gamma_i \subset \mathbf{L}$. The must-link and cannot-link constraints define the neighborhood over the hidden labels, such that the neighbors of a point x_i are all points that must-linked and/or cannot-linked with. The *random field defined over the hidden variables* is a *Markov Random Field*, where the PDF of the hidden variables obeys the following Markov property:

$$\forall i, P(\mathbf{l}_i|\mathbf{L}-\{\mathbf{l}_i\}, \Theta, \mathbf{M}, \mathbf{C}) = P(\mathbf{l}_i|\{\Gamma_i, \Theta, \mathbf{M}, \mathbf{C}\}) \quad (7)$$

Therefore, the probability distribution of \mathbf{l} labels is only dependent on the must-link and cannot-link constraints on \mathbf{x} data-points. The above paragraph justifies the name for the algorithm as HMRF and letting us assume any a-piory PDF for an arbitrary label setup.

Since, by the *Hammersley-Clifford theorem*, the a-prior PDF of a particular label setup can be expressed as a Gibbs distribution eq.8, which conveniently belongs to the exponential distributions family, as we will see later.

$$P(\mathbf{L}|\Theta, \mathbf{M}, \mathbf{C}) = \frac{1}{Z_1} \exp(- \sum_{\Gamma_i \in \Gamma} V_{\Gamma_i}(\mathbf{L})) \quad (8)$$

Where V_{Γ_i} is the *potential function*, as it called when the clustering model is *hard* type (**NOTE: Potential Function also found this in the Bishops**

Book and I think this term is used in Kmeans case and not in the probabilistic EM), for each neighborhood of labels defined by the $\{M, C\}$ sets of constraints.

Since the constraints should probably been given as pairs then eq.8 will become eq.9. **(NOTE: A question here is constraint $A=(2,50)$ will be the same as $B=(50,2)$, will both be present etc. Should the constraints be a Graph, a Matrix or a set of pairs? (Implementation Language Performance VS Math/Computational Performance)).**

$$P(\mathbf{L}|\Theta, \mathbf{M}, \mathbf{C}) = \frac{1}{Z_1} \exp\left(-\sum_i \sum_j V(i, j)\right) \quad (9)$$

$$V(i, j) = \begin{cases} f_{\mathbf{M}}(x_i, x_j) & \text{if } (x_i, x_j) \in \mathbf{M} \\ f_{\mathbf{C}}(x_i, x_j) & \text{if } (x_i, x_j) \in \mathbf{C} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Where $f_{\mathbf{M}}$ and $f_{\mathbf{C}}$ are positive value cost functions for the must-link and cannot-link constraints violation. Clearly, eq.9 is discouraging the violation of the constraints by reducing the value of joint probability MAP in eq.6.

Maximizing the joint HMRF probability in eq.6 (left part of the equation), is equivalent to jointly maximizing the likelihood of generating data points from the model and the probability of label assignments that respect the constraints, while regularizing the model parameters. The essential part of the right side of the equation the conditional probability $P(\mathbf{X}|\mathbf{L}, \mathbf{M}, \mathbf{C})$, equivalent to eq.3. Firstly, because its PDF will define our prior assumption about the PDF mixture of the data-set. Secondly, because the same PDF should be used as the constraints potential function V , thus, penalty functions f . Finally, because its PDF will ultimately define the parametrized (by Θ parameters) *distortion measure*.

Using the convenience in the assumption of a regular exponential distributions for the observed data points X and a regular Bregman divergences, the PDF of the observed data would be in eq.11.

$$P(\mathbf{x}_i|\mathbf{l}_i, \Theta) = \frac{1}{Z_2} \exp(-D_{\mathbf{A}}(x_i, \mu_{\mathbf{l}_i})) \quad (11)$$

where $D_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_{\mathbf{l}_i})$ is the Bregman divergence, i.e. distortion measure, between \mathbf{x}_i and $\mu_{\mathbf{l}_i}$. Where \mathbf{l}_i is indicating that the distance is measured for an arbitrary \mathbf{x}_i is from the cluster centroid (or cluster's PDF expected value) when the point and the centroid are under the same cluster PDF **(NOTE: This is my conclusion and no one in literature give any specification related to the μ notation)**. Z_2 is the normalizing term (a.k.a *partition function*).

As explained above in general EM algorithm description, we are at the step where we have to assume the PDF mixture of the observable data. In the semi-supervised case the a-priori distribution assumption will be the same, over the observable data-set \mathbf{X} and the observable constraints sets \mathbf{M}, \mathbf{C} . Thus, in our case, for the IR problems based on the literature, we assume for both the data-set

and the constraints sets von-Mises Fisher (vMF) distribution with unit concentration parameter, which is equivalent to the spherical Gaussian distribution [1, 3]. In order to make PDF of eq.11 to be equivalent to vMF the distortion parameter $D_{\mathbf{A}}$ will be replaced with the *parametrized cosine distance* as defined in eq.12 by the parameters matrix \mathbf{A} .

$$D_{\cos \mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^T \mathbf{A} \mathbf{x}_j}{\|\mathbf{x}_i\|_{\mathbf{A}} \|\mathbf{x}_j\|_{\mathbf{A}}} \quad (12)$$

Where $\|\mathbf{x}_i\|_{\mathbf{A}} = \sqrt{\|\mathbf{x}_i^T \mathbf{A} \mathbf{x}_j\|}$, (NOTE: In [1] under square root there is no norm, but in practice the product sometimes drops under zero and breaks the code run.)

In the eq.6 (right side of the equation) we have defined the conditional probabilities for the observed X and the L , where the second is conditioned under the constraints M, C . The presence of the $P(\Theta)$ is occurring due to the explicit presence of the joint distribution in eq.6 (left side of the equation) and the graphical model in fig.1. This is enabling us to decide whether or not we want to apply our a-priori knowledge about the distribution of the parameters. This can be omitted when Θ parameters are not explicitly present, as in [1], thus no prior assumptions about the parameters distribution is required or can be applied.

Depending on the prior distribution, parameters Θ are separated in two sets $\Theta = \{\mathbf{A}, \mathbf{M}\}$, where the second one is the expected values of the mixture's PDF(s) and \mathbf{A} is the rest of the required parameters.

$$P(\Theta) = P(\mathbf{A})P(\mathbf{M}) \quad (13)$$

2.1 Putting all together

The objective of the semi-supervised HMRF Kmeans is to maximize the MAP as defined in eq.6 with the parts of this equation to be analysed in eq.8 eq.8 and eq.13 . Putting all these equations together we have eq.14.

$$P(\mathbf{X}, \mathbf{L}, \Theta | \mathbf{M}, \mathbf{C}) = P(\mathbf{A})P(\mathbf{M}) \left(\frac{1}{Z_1} \exp\left(-\sum_{\Gamma_i \in \Gamma} V_{\Gamma_i}(\mathbf{L})\right) \right) \prod_{i=1}^N \frac{1}{Z_2} \exp(-D_{\mathbf{A}}(x_i, \mu_{\mathbf{l}_i})) \quad (14)$$

Our early decision was the hard-clustering setup for the algorithm thus my explaining the logarithmic properties from eq.14 we are getting eq.15.

$$\ln P(\mathbf{X}, \mathbf{L}, \Theta | \mathbf{M}, \mathbf{C}) = \ln P(\mathbf{A}) + \ln P(\mathbf{M}) - \sum_{\Gamma_i \in \Gamma} V_{\Gamma_i}(\mathbf{L}) + \sum_{i=1}^N D_{\mathbf{A}}(x_i, \mu_{\mathbf{l}_i}) + \ln Z_1 + N \ln Z_2 \quad (15)$$

Equation eq.15 is now called the *Objective Function* and the goal of the algorithm is to minimize its output, which is equivalent to the MAP maximization.

Expressing it in words the minimization process of eq.15 is equivalent to the iterative process in order to find the proper clustering configuration where each arbitrary data-point is the closest to the mean value of the cluster it will belong into at the end. Moreover, the constraints violation will also be minimized.

Before we proceed in replacing the exact violation V and distortion measure D functions with the parameterized cosine similarity we are considering the rest of the equation starting with the normalizing components Z_1 and Z_2 . In additions to consider the distributions of the PDF mixture parameters A and Mu .

The normalizing components of observable data points and the hidden variables distributions are depended on the prior PDF assumption. In case of Gaussian distribution their value can be calculated in a *closed form* (NOTE:Surely exactly this is) while for all the other case we have to employ an *approximation method*. In our case where cosine similarity is must be used an approximate inference method is required which can be very expensive computationally. In this case we can assume that Z and Z are constant but not eq.15 will no longer be an joint probability. However, empirically it has been shown that this objective functions will work properly and the algorithm will manage to converge in a local minimum [1]. However, if in some applications it is important to preserve the semantics of the underlying joint probability model, then the normalizers Z and Z must be estimated by approximate inference methods.

References

1. Basu, S., Bilenko, M., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 59–68. ACM (2004)
2. Bishop, C.: Pattern Recognition and Machine Learning, pp. 439–447. Springer (2006)
3. Chapelle, O., Scholkopf, B., Zien, A., et al.: Semi-supervised learning. pp. 73–102. The MIT Press (2006)