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 mehtods* are using the provided supervision for guiding the algorithm towards a data partitioning which is avoiding (but not prevening) the constraints violation.
2. *Distance-based approaches* in clustering method with a particular distance funciton; 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 where provided in the clustering model in the form of data-labels pairs or in the initialization phase of theclustering 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 fixednumber of clusters. However, the same models can be transformed into a relatively easy soft-clustering model where of each data point only Maximum a-posteriry Probability (MAP) of the point to bea member of each cluster of th 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 distortionmeasure 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 asoft-clustering method then the probability destiny 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 interactivelearning 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 complimentary 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.

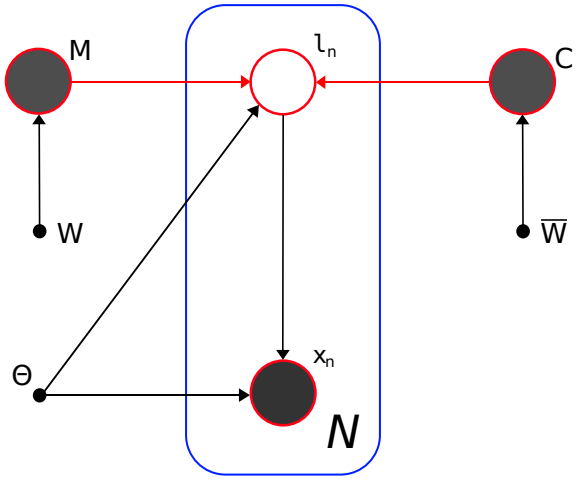

Fig. 1. Semi-supervised HMRF Expectation Maximization or Kmeans Clustering Graphical Model.

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

$$lnP(X|\Theta) = ln\left\{\sum_{l} P(X, L|\Theta)\right\} \tag{1}$$

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 $1, ..., N$, 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-priory 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 $lnp(X, Z|\Theta)$ 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 $\{1, ..., \mathbf{K}\}$ or $(0, 1]$

$\mathbf{\Theta}$ 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 of hard-clustering set-up.

$\mathbf{\Theta}$ the final set of parameters after the Probability Density Functions Mixture.

```
Choose an Initial setting for Θ^OLD;
while I iterations not reached do
    1. E-step Evaluate P(L|X, Θ^OLD);
    2. M-step Evaluate Θ^NEW given by (a) and (b);
            a. Θ^NEW = arg_Θ max Ω(Θ, Θ^OLD);
            b. Ω(Θ, Θ^OLD) = Σ_L P(L|X, Θ^OLD) ln P(X, L, Θ);
    if log likelihood convergence reached then
        Breaking the loop and Ending the Clustering;
    else
        Θ^OLD ← Θ;
    end
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 $P(\mathbf{\Theta})$ over the parameters $\mathbf{\Theta}$. In this case the E-step remains the same as in the maximum likelihood case, while in the *M-step (b)* the $\Omega(\mathbf{\Theta}, \mathbf{\Theta}^{OLD}) + lnP(\mathbf{\Theta})$.

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.

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

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 $\mathbf{\Theta}$ are, also, independent from the constraints set. Thus:

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

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

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

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

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

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

At this step of the clustering method building process we have to decide weather 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 of 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 $\mathbf{\Gamma}_i \subset \mathbf{L}$. 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-inked 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*:

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

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.

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

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? (Impementation Language Performance VS Math/Computational Performance))**.

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

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

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.

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

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 a 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-piori 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 $\mathbf{A}$.

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

Where $\|\mathbf{x}_i\|_{\mathbf{A}} = \sqrt{\mid \mathbf{x}_i^T \mathbf{A} \mathbf{x}_j \mid}$, **(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 $P(\mathbf{\Theta})$ 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-piori knowledge about the distribution of the parameters. This can be omitted when $\mathbf{\Theta}$ 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 $\mathbf{\Theta}$ are separated in two sets $\mathbf{\Theta} = \{\mathbf{A}, \mathbf{M}\}$, where the second one is the expected values of the mixture's PDF(s) and $\mathbf{A}$ is the rest of the required parameters.

$$P(\mathbf{\Theta}) = P(\mathbf{A})P(\mathbf{Mu}) \tag{13}$$

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 analyses in eq.8 eq.8 and eq.13 . Putting all these equations together we have eq.14.

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

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.

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

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 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. **(NOTE: Applying the logarithm on eq.14 seems to me that all the sights in the right part of eq.15 should have been exactly the opposite but in literature is not. Am I missing something here?)**

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 $\mathbf{A}$ and $\mathbf{Mu}$.

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:No sure what exactly that is)** while for all the other case we have to employee an *approximation method*. In our case where cosine similarity must be used an approximate inference method is required which can be very expensive computationally. In this case we can assume that Z_1 and Z_2 are constants. Now 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 joint probability model, then the normalizers should be estimated by approximate inference methods.

The distribution on cluster centroids $\{\mu_i\}_{i=1}^{K}$ can safely assumed to be uniformly distributed thus $P(\mathbf{Mu})$ can be assumed as constant, thus, $lnP(\mathbf{Mu}) = 0$. Distributions parameters $\mathbf{A}$ can lead to *diegenerate solutions* of the optimization problem. As an example, for squared Euclidean distance it can be $\mathbf{A} = 0$ as such solution. In order to prevent *degenerate solutions*, $P(\mathbf{A})$ is used to regularize the parameter values using a prior distribution. In addition, we mostly need non-negative solutions for several computational problems might occur otherwise, thus, one such a distributions can be Rayleigh distribution as described in eq.16, with the assumptions of mutually independent parameters values α.

$$P(\mathbf{A}) = \prod_{\alpha_{ij} \in \mathbf{A}} \frac{\alpha_{ij} \exp(-\frac{\alpha_{ij}^2}{s^2})}{s^2} \tag{16}$$

Where s is the width parameter of the distortion.

The last step for deriving the Semi-supervised HMRF Kmeans is to define exactly the f_M and f_C sub-functions of the constraint violation V functions of eq.10 based on the cosine similarity as explained above.

Starting with f_M, it should return the violation cost of must-link constrains between an (i, j) pair if only $\mathbf{l}_i \neq \mathbf{l}_j$, i.e. their labels are not equal which means the are not belonging to the same cluster as they should. Moreover, we want to use the constraint violations to learn the underlying distance measure (distortion

function), thus, the penalty for violating a must-link constraint between distant points should be higher than that between nearby points. Therefore, we have to adjust the distortion function to get there points closer as they should upon the must-link constraints. To do so we need the penalty scaling function to be a monotonically increasing function of the distance between x_i and x_j according to the current distortion measure. Consequently, we have the following equation eq.17 which satisfies the above properties for F_M [1, 3] **(NOTE: The reasoning is very close but note exactly and I think my reasoning is more clear and more sensible, because theirs is a bit odd for example related to W, in Chappell's is better explained but more complex in notation. Also my notation defers allot because theirs finally is equivalent to my notation.)**.

$$f_{\mathbf{M}}(x_i, x_j) = w_{ij} D_{cos_{\mathbf{A}}} \Psi(l_i \neq l_j) \tag{17}$$

The function f_C for the cannot-link constraint, same as must-link one, should entourage the distortion function parameters to be changes for putting as far as possible the x_i from x_j upon the cannot link violation. Since we again using the distance measure of the algorithm as the violation function this time we need to get the compliment value which the maximum value the distortion function can return minus the current distance of the points. Thus, the f_C is becoming as shown in eq.18.

$$f_{\mathbf{M}}(x_i, x_j) = \overline{w}_{ij} (D^{max}_{cos_{\mathbf{A}}} - D_{cos_{\mathbf{A}}} \Psi(l_i = l_j)) \tag{18}$$

Note that in both eq.17 and eq.18 the $\Psi()$ function is returning $0, 1$ depending whether of not the condition given is False or True respectively. As for w_{ij} and $\overline{w}_{ij}$ are weights for each must-link and cannot-link constraint significance if there is a need to be difference upon the problem setting, and they are also depicted in fig.1.

Putting all the above together equation eq.15 together with eq.17, eq.18 and Z_1, Z_2, $P(\mathbf{Mu})$ being constant values (where for 1 their logarithms to be equal to 0) we are getting the following Objective function eq.19. In addition we assume Rayleigh distribution eq.16 for the $\mathbf{A}$ parameters as explained above.

$$\begin{aligned} J_{obj} = \sum_{i=1}^{N} D_{\mathbf{A}}(x_i, \mu_{\mathbf{l}_i}) + \sum_{(x_i, x_j) \in \mathbf{M}} w_{ij} D_{cos_{\mathbf{A}}} \Psi(l_i \neq l_j) \\ + \sum_{(x_i, x_j) \in \mathbf{C}} \overline{w}_{ij} (D^{max}_{cos_{\mathbf{A}}} - D_{cos_{\mathbf{A}}} \Psi(l_i = l_j)) \\ + \sum_{\alpha_{ij} \in \mathbf{A}} \frac{a_{ij}}{s^2} - \sum_{\alpha_{ij} \in \mathbf{A}} \ln \alpha_{ij} + Na \ln s^2 \end{aligned} \tag{19}$$

Where Na is the amount of the model's parameters **(NOTE: the last 3 terms of the sum which are the lnP(A) was missing in the Basu et. all and I susspect that this was the reason I am having (0,0) centroids same times plus my Global Objective is not monotonically decreasing)**.

2.2 The HMRF Kmeans algorithm steps

Going back to the EM algorithm, we can properly adjust it in the semi-supervised form as expressed in eq.19. In particular in algorithm alg.2 we present the kmeans form of th EM with the goal to minimize the objective function, which concurrently is minimizing the distances of the data-points from their cluster centroid and the violation of the must-Link and cannot-link constraints.

Algorithm 1.2: The HMRF Kmeans.

Data:
X observable data points.
K possible states clusters we expect to be existing in the data-set.
$\mathbf{L}_{i=1}^{K}$ the hidden field variables with values $(0, 1]$
A a set of parameters for initializing the distortion measure.
Result:
$\mathbf{L} = \{l_k\}_{k=1}^{K}$ where $(0, 1]$ final label assignment.
$\mathbf{A}^0$ the final set of parameters after the Probability Density Functions Mixture.

```
Initializing the K clusters by calculating their initial centroids
M^0 = (μ_1^0, ..., μ_k^0);
while I iterations not reached do
    1. E-step
        Re-assigning cluster labels (l_1^{i+1}, ..., l_k^{i+1});
    2. M-step
        a. Re-calculating M^{i+1} = (μ_1^{i+1}, ..., μ_k^{i+1}) to minimize J_obj
    objective function;
        b. Re-estimating parameters A^{i+1};
    if J_obj min threshold has been reached then
        Breaking the loop and Ending the Clustering;
    else
        A^i ← A^{i+1};
    end
end
```

The E-Step of the alg.2 is calculating the cluster labels where the data-point are belonging into at the current i iterations of the EM. This step in actually a full function of its own where an other algorithm or strategy of assignment has to be selected. In particular the E-step can be one of the following algorithms:

- Iterated Conditional Modes (ICM).
- Belief Propagation.
- Linear Programming Relaxation.
- Mean-field approximation based algorithm of Lange et al. cited in [3].

In this implementation we have selected ICM because of its straightforward implementation. ICM performs a sequential clustering assignment *for all points* **in random order**. Each point $\mathbf{x}_i$ is assigned to cluster representative μ_k that minimized the point's contribution in such a manner that the eq.20 is minimized.

$$
\begin{aligned}
J_{obj} = D(x_i, \mu_{\mathbf{k}}) + & \sum_{(x_i,x_j)\in\mathbf{M}} w_{ij} D_{cos_{\mathbf{A}}} \Psi(k_j) \\
+ & \sum_{(x_i,x_j)\in\mathbf{C}} \overline{w}_{ij} (D^{max}_{cos_{\mathbf{A}}} - D_{cos_{\mathbf{A}}} \Psi(k = l_j)) \\
+ & \sum_{\alpha_{ij}\in\mathbf{A}} \frac{a_{ij}}{s^2} - \sum_{\alpha_{ij}\in\mathbf{A}} \ln \alpha_{ij} + Na \ln s^2
\end{aligned}
\tag{20}
$$

After all points are assigned the process repeated until *no change in cluster assignment occures* between two successive iterations. **(NOTE: In eq.20 the last line has been appended by my based on Chappells book which is missing from Basu respective formula.)**.

The M-Step of the alg.2 has two steps. The (a.) step, where centroids of the clusters (for the current) iteration are calculated, based on the parametrized cosine distance, as shown in equation eq.21.

$$
\mu_h^{cos_{\mathbf{A}}} = \frac{\sum_{x_i\in\mathbf{X_h}} x_i}{\| \sum_{\mathbf{x}_i\in\mathbf{X}_h} \mathbf{x}_i \|_{\mathbf{A}}} \tag{21}
$$

The (b.) step is calculating the parameters $\mathbf{A}$ in order to adopt the distortion measure to fit best the current state of the clustering. In the description of eq.12 we have seen that the parameters of the cosine distance is a $F \times F$ matrix where F is the size of the feature set, i.e. the number of dimensions of the data vectors. In order to reduce the computational overhead due to a potential large feature set, as this is the case is IR problems in general, we set $A = diag(Ap)$ $Ap = (a_1, ..., a_F)$. Thus A is a diagonal matrix and all we have to be calculated now is a vector of values same in size as the feature set.

In order to update the $\mathbf{A}$ all we have to do is to calculate the partial derivatives of its variables and update its values with eq.22, where η is the *learning/adaption rate*. In this way the distance measure will gradually adapted in order the vector space to be transformed for getting similar data-points closer and dissimilar farther apart.

$$
a_m = a_m + \eta \frac{\partial J_{obj}}{\partial a_m} \tag{22}
$$

Although for some distance measures such as Euclidean we can have *tractable closed-form solution* for the partial derivatives, in general the solution of the partial derivative is intractable. Gradient descent **(NOTE: As I can tell is some short of partial derivative approximation. However, I am not sure what exactly this is.)** is a alternative and general solution irrespectively of the distance measure we use in the algorithm. In equation eq.23 we show the *gradient descent* for the partial derivative for the parameters of the parametrized cosine similarity [1].

$$\frac{\partial D_{cos_{\mathbf{a}}}(\mathbf{x}_i,\mathbf{x}_j)}{\partial a_m} = \frac{x_{im}x_{jm}\|\mathbf{x}_i\|_{\mathbf{A}}\|\mathbf{x}_j\|_{\mathbf{A}} - \mathbf{x}_i^T\mathbf{A}\mathbf{x}_j\frac{x_{im}^2\|\mathbf{x}_j\|_{\mathbf{A}}^2 + x_{jm}^2\|\mathbf{x}_i\|_{\mathbf{A}}^2}{2\|\mathbf{x}_i\|_{\mathbf{A}}\|\mathbf{x}_j\|_{\mathbf{A}}}}{\|\mathbf{x}_i\|_{\mathbf{A}}^2\|\mathbf{x}_j\|_{\mathbf{A}}^2} \tag{23}$$

This equation will be used in order to calculate eq.24 which is the right most part of the eq.22. Its value for each variable a_{ij} or the parameters matrix $\mathbf{A}$ will change by a percentage of η in each iteration, as shown in eq.22.

$$\begin{aligned}\frac{\partial J_{obj}}{\partial a_m} &= \sum_{\mathbf{x}_i\in\mathbf{X}} \frac{\partial D_{cos_{\mathbf{a}}}(\mathbf{x}_i,\mu_{l_i})}{\partial a_m} \\ &+ \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathbf{M}} w_{ij}\frac{\partial D_{cos_{\mathbf{a}}}(\mathbf{x}_i,\mathbf{x}_j)}{\partial a_m}\Psi(l_i\neq l_j) \\ &+ \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathbf{M}} \overline{w}_{ij}\left[\frac{\partial D_{cos_{\mathbf{a}}}^{max}}{\partial a_m} - \frac{\partial D_{cos_{\mathbf{a}}}(\mathbf{x}_i,\mathbf{x}_j)}{\partial a_m}\right]\Psi(l_i = l_j) \\ &- \frac{\partial lnP(A)}{\partial a_m}\end{aligned} \tag{24}$$

Where the the last line of the eq.24 is the partial derivative of the prior distribution of the distortion measure parameters. When Rayleigh priors are used then the partial derivative is equivalent to eq.25.

$$\frac{\partial lnP(A)}{\partial a_m} = \frac{1}{a_m} - \frac{a_m}{s^2} \tag{25}$$

2.3 Prof of convergence

In general we can say that the HMRF-KMeans algorithm monotonically decreases its objective function, while alternates between updating the assignment of points to clusters, and updating the parameters. This is depicted in eq. 26 which we can considered as the Global Objective function compare to the eq.20, which is the objective for each data-point classification separability in the ICM algorithm.

$$\begin{aligned}G_{obj} = \sum_{h=1}^{K}\sum_{i=1}^{N} D_{\mathbf{A}}(x_i,\mu_{\mathbf{h}}) + \sum_{h=1}^{K}\sum_{(x_i,x_j)\in\mathbf{M}} w_{ij}D_{cos_{\mathbf{A}}}\Psi(h\neq l_j) \\ + \sum_{h=1}^{K}\sum_{(x_i,x_j)\in\mathbf{C}} \overline{w}_{ij}(D_{cos_{\mathbf{A}}}^{max} - D_{cos_{\mathbf{A}}}\Psi(h=l_j)) \\ + \sum_{\alpha_{ij}\in\mathbf{A}}\frac{a_{ij}}{s^2} - \sum_{\alpha_{ij}\in\mathbf{A}}\ln\alpha_{ij} + Na\ln s^2\end{aligned} \tag{26}$$

In the first line of equation eq.26 each cluster centroid $_h$ is re-estimated by taking the mean of the points in the neighborhood X_h, which minimizes the component $D_{cos_A}(\mathbf{x}_i, \mu_h)$. The constraint potential functions, at the second and third line, are not taking a part in centroid re-estimation, because they are not explicit functions of the centroid. Therefore, given the cluster assignments and the distortion parameters, J_{obj} *will decrease or remain the same.*

The parameter estimation step (b.) in the E-step, decreases J_{obj} or keeps it unchanged. Thus, the objective function decreases after every cluster assignment, centroid re-estimation, and parameter re-estimation step.

That the objective function is bounded below by a constant: *being the negative log likelihood* **(NOTE: Here is claimed to be the Negative Log while in other places in Basu or Chappel is not mentions. Here is from Chappel. If this is correct then the above NOTE does not need an asware)** of a probabilistic model with the normalizer terms, J_{obj} is bounded below by zero. Even without the normalizers, the objective function is bounded below by zero, since the *distortion and potential functions are non-negative* due to the fact that A is **positive definite** (NOTE: How they can claim this. How I assure that the A is positive definite. I think the Rayleight prior pdf should do this but is not event mentioned in Basu paper). Therefore, J_{obj} is bounded below, and HMRF-KMeans results in a decreasing sequence of objective function values, the value sequence must have a limit. The limit in this case will be a fixed point of J_{obj}, since neither updating the assignments nor the parameters can further decrease the value of the objective function [3].

"As a result, the HMRFK Means algorithm will converge to a fixed point of the objective. **In practice, convergence can be determined if subsequent iterations of HMRF-KMeans result in insignificant changes in**" J_{obj}.

2.4 Some quetions for Stathis

- What would be the best structure for must-link and cannot-link constraints. Should be given as pairs or as a graph. A pair (5,6) is equivalent to (6,5) and should both be present?
- Eq.26 is not exactly as in the [3], but I think mine is more descriptive. Is it?
- The $P(A)$ and Rayleight prior distribution for the parameters are not mentioned in [1] but are very important part in [3].
- In my implementation based mainly in [1] the 26 is not decreasing monotonically. On the contrary is decreasing up to a point and then starts increasing again? I think the omitted $P(A)$ might be the problem. I ll check it before you even read this line.
- Note that I've omitted the Z normalization factors assuming value 1 thus 0 for their logarithm. Are they important given the assumption that they are constant as explained above, something that making the objective not to be a log likelihood anymore?

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)