

---

# Hierarchical Affinity Propagation Supplementary Material

---

**Inmar E. Givoni, Clement Chung, Brendan J. Frey**

Probabilistic and Statistical Inference Group

University of Toronto

10 King's College Road, Toronto, Ontario, Canada, M5S 3G4

## A Hierarchical Facility Location

Facility location (FL) is a well-researched problem in operational-research (Şahin Güvenç & Haldun, 2007), and is closely related to EBC. Given a set of  $N$  customers and  $M$  potential facilities to open, as well as costs  $s_{ij}$  for customer  $i$  to use facility  $j$ , and a facility setup cost  $c_j$  associated with opening a facility  $j$ , we wish to find the set of facilities to open, as well as the assignments of each customer to exactly one of the open facilities so as to minimize the overall sum of costs. Similar to the *negative* preference in AP, the facility setup costs dictate the number of facilities that will be opened, and the problem can be viewed as that of finding an optimal trade-off between incurred customer-facility costs (negative pairwise similarities) and facility setup costs.

Let  $\{h_{ij}\}, (i = 1, \dots, N, j = 1, \dots, M)$  be a set of  $NM$  binary hidden variables where  $h_{ij} = 1$  indicates customer  $i$  has chosen facility  $j$  as its facility. Furthermore, let  $\{e_j\}, j = 1, \dots, M$  be a set of  $M$  binary hidden variables, where  $e_j = 1$  indicates facility  $j$  is open. Negating both pairwise and setup costs to align the notation with that of AP, we define the following functions:

$$C_j(e_j) = c_j e_j \quad (13)$$

$$S_{ij}(h_{ij}) = s_{ij} h_{ij} \quad (14)$$

$$I_i(h_{i\cdot}) = \begin{cases} 0 & \sum_j h_{ij} = 1, \\ -\infty & \text{otherwise.} \end{cases} \quad (15)$$

$$E_j(h_{\cdot j}, e_j) = \begin{cases} 0 & e_j = \max_i h_{ij}, \\ -\infty & \text{otherwise.} \end{cases} \quad (16)$$

Thus, there are only two differences between AP and the formulation for FL. First, in AP we require that  $N = M$ , while in FL,  $N$  in general is not equal to  $M$ . Second, since the exemplars in FL (facilities) are not members of the set of objects to be clustered (customers), there is no longer meaning for the exemplar consistency constraint (3). It is replaced by constraint

(16) enforcing that if any point  $i$  chooses facility  $j$  as an exemplar ( $h_{ij} = 1$ ), then the indicator variable for that exemplar  $e_j$  should be turned on. This objective function can again be stated in terms of the factor graph shown within the dotted square of Fig. 1(a) with the column indices running up to  $M$  and not  $N$ , and with the objective function (8).

As was shown in (Lazic et al., 2009), the message updates to be calculated iteratively until convergence are:

$$\alpha_{ij} = \min[0, c_j + \sum_{k \neq i} \max(0, \rho_{kj})] \quad (17)$$

$$\rho_{ij} = s_{ij} - \max_{k \neq j} (\alpha_{ik} + s_{ik}) \quad (18)$$

The generalization of standard FL to a hierarchical version is for the case where we have at the bottom layer a set of customers and potential facilities, and then at every additional layer a set of potential facilities serving the previous layer's facilities. It is described by the model in Fig. 1(a), where the column indices in the first layer go up to  $M$ , and the row indices in every layer  $l$  are equal to the column indices in the previous layer  $l-1$  where  $l > 1$ . The FL literature contains several different variants that are called hierarchical FL, and the one we describe here is equivalent to that stated in, *e.g.*, (Kantor & Peleg, 2009).

FL functions are as in (5) - (8) with (7) replaced by

$$E_j^l(h_{\cdot j}^l, e_j^l) = \begin{cases} 0 & e_j^l = \max_i h_{ij}^l, \\ -\infty & \text{otherwise.} \end{cases} \quad (19)$$

## B Alternative Objective Function

An alternative way to formulate a hierarchical exemplar based objective function concerns the case where at each layer we cluster *all* the points (not just the exemplars from the previous layer), but the set of exemplars chosen at each layer must be a subset of the exemplars from the previous layer. Once again, this is achieved by modifying the constraints  $I_i^{l>1}$ . The

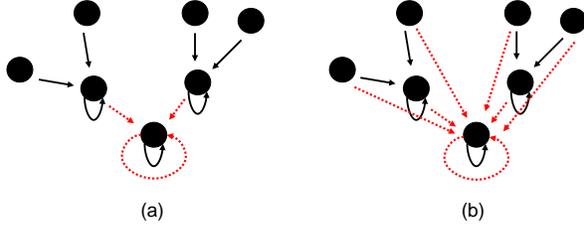


Figure 8: Two alternative objectives for hierarchical exemplar-based clustering. (a) HAP: At each layer, the goal is to cluster only exemplars from the previous layer. (b) HAP-B: At each layer, the goal is to cluster all points from the previous layer, but maintain nesting of exemplars.

modified function is given by:

$$I_i^{l>1}(h_{i:}^l, e_i^{l-1}) = \begin{cases} 0 & \sum_j h_{ij}^l = 1, h_{ii}^l \leq e_i^{l-1}, \\ -\infty & \text{otherwise.} \end{cases} \quad (20)$$

A valid solution for either objective function described for HAP results in a nested set of exemplars. In other words, the exemplars at each layer are a subset of the exemplars in the previous layer. From the solution using the first objective function we can also easily derive nested clusters, but we note that in the second, a particular datapoint may choose a different exemplar  $t$  at layer  $l$  from the exemplar  $j$  it chose in layer  $l-1$ , even if  $j$  is still an active exemplar at layer  $l$ .

The two alternative formulations are shown pictorially in Fig. 8. We will refer to the former, described in Sec. 3 and Fig. 8(a) as HAP and the alternative described here and in figure Fig. 8(b) as HAP-B.

Clearly, we can also mix and match layers of the two exemplar-based clustering objectives, and of facility location, defining somewhat novel tasks. The exact formulation for an arbitrary topology is recovered by using an appropriate  $E^l$  function at layer  $l$ , depending on whether it is an EBC layer, in which case (7) is used, or a FL layer, in which case (19) is used. The  $I^l$  functions remain as defined in (5) and (6).

### C Algorithms for HFL and HAP-B

For an FL layer, the messages originating from function node  $E_i^l$  (updates  $\tau_j^{l+1}$  and  $\alpha_{ij}^l$ ) are modified with respect to (9) and (11) above and we get:

$$\tau_j^{l+1} = c_j^l + \sum_k \max(0, \rho_{kj}^l) \quad (21)$$

$$\alpha_{ij}^l = \min[0, c_j^l + \phi_j^l + \sum_{k \neq i} \max(0, \rho_{kj}^l)] \quad (22)$$

If we choose the alternative objective function formulation given in (20), the messages originating from function node  $I_i^{l>1}$  (updates  $\phi_j^{l-1}$  and  $\rho_{ij}^l$ ) are modified and

(10) and (12) above are replaced by:

$$\begin{aligned} \phi_j^{l-1} &= \max[0, \alpha_{jj}^l - \max_{k \neq j}(\alpha_{jk}^l + s_{jk}^l)] \quad (23) \\ \rho_{ij}^{l>1} &= \begin{cases} \min(0, \tau_i^l) - \max_{k \neq i}(\alpha_{ik}^l + s_{ik}^l) & i = j \\ s_{ij}^l + \min \begin{cases} \max(0, -\tau_i^l) - \alpha_{ii}^l, \\ - \max_{k \notin \{i, j\}}(\alpha_{ik}^l + s_{ik}^l) \end{cases} & i \neq j \end{cases} \quad (24) \end{aligned}$$

## D Experimental setup and results

All experiments were run on a 3.2GHz Dell Precision T7500 workstation. Unless otherwise specified, the preference range for the data is chosen by using the heuristic preference range (publicly available for download from the Affinity Propagation authors at [www.psi.toronto.edu](http://www.psi.toronto.edu)) where depending on the number of layers, this range was divided in log-linear scale. Setting  $L$ , the number of layers, is a user specified quantity. However, we note that when the given  $L$  exceeds the natural number of layers in the data (when, *e.g.*) the data is synthetic, and the ground truth is known, the top layers will all include the same exemplar, or in other words, the nesting saturates, and therefore setting  $L$  to be too large will not have severe negative results on the quality of the clustering.

**BP:** As is the case for many loopy BP applications, HAP does not always converge, and monitoring the objective function 8 may yield an oscillating pattern. This was observed for some of the toy-data experiments, where we used a strategy of decoding a bottom layer after several rounds of message passing, and continuing clustering with the given set of exemplars. This can be seen as a quasi-greedy heuristic that does allow some information to propagate down before making a decision, and indeed proves to be useful in terms of obtaining better results.

**MPLP:** We have run experiments for the 2D toy dataset using MPLP instead of standard max-product, as the underlying inference mechanism for the same graphical model. We find that MPLP performs poorly in comparison to HAP and the greedy algorithm. The decoding scheme we use for MPLP is based on a similar one to that used for HAP. It is possible that further investigation into more involved schemes will lead to better results, and this is a direction we are currently pursuing, as well as looking at potentially tighter relaxations such as (Sontag et al., 2008).

## E Modified Rand Index

The modified Rand index measures the agreement between two clustering solutions  $C, \hat{C}$ , where one is the

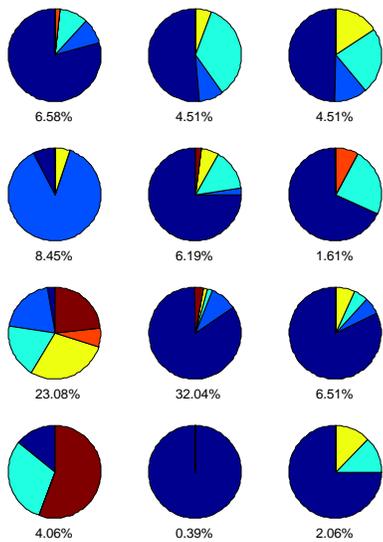


Figure 9: Continent composition of each second-layer cluster for the real HIV data (see Fig. 7(a) for color coding). Percentages indicate the number of datapoints associated with each cluster, of all datapoints clustered in this layer.

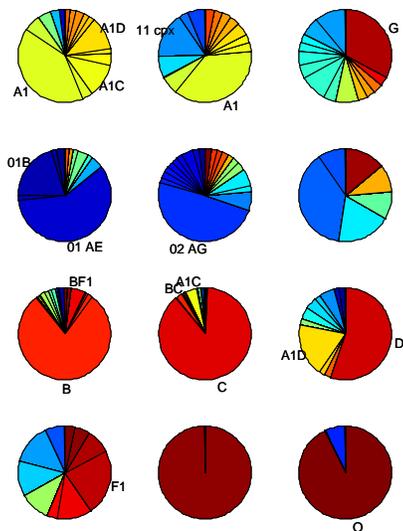


Figure 10: Subtype composition of each second-layer cluster for the real HIV data. Subtype labels are shown only if the number of datapoints associated with a subtype in a cluster is larger than 10.

ground truth ( $C$ ) and the other is obtained by one of the algorithms ( $\hat{C}$ ). The Rand index can be interpreted as the probability that the two clustering solutions agree on whether two randomly drawn points belong to the same cluster or to different clusters. It

is computed as follows:

$$R(C, \hat{C}) = \frac{\sum_{i>j} [c_i=c_j \wedge \hat{c}_i=\hat{c}_j]}{2 \sum_{i>j} [\hat{c}_i=\hat{c}_j]} + \frac{\sum_{i>j} [c_i \neq c_j \wedge \hat{c}_i \neq \hat{c}_j]}{2 \sum_{i>j} [\hat{c}_i \neq \hat{c}_j]}, \quad (25)$$

where  $c_i$  ( $\hat{c}_i$ ) indicates the cluster index of point  $i$  according to  $C$  ( $\hat{C}$ ).

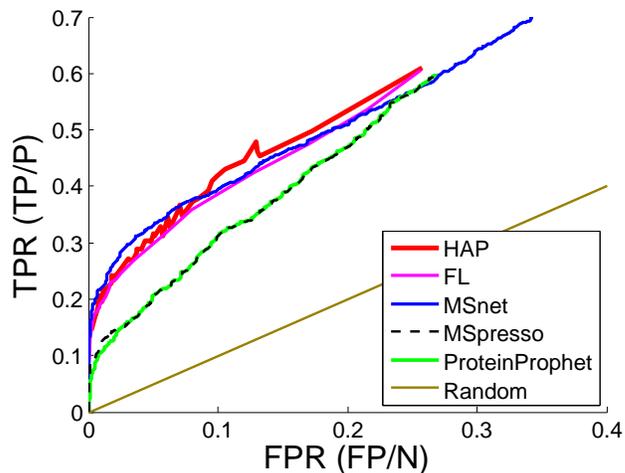


Figure 11: A plot of the true positive rate (TPR) *v.s.* false positive rate (FPR) for HAP, three current state-of-the-art protein identification algorithms, and a facility location-based formulation of the problem, on the problem of protein identification using mass spectrometry data derived from yeast wild-type grown in a rich medium.

## F Second Layer Analysis of Real HIV Data

We provide more details on the second layer of the clustering tree found by HAP. We find interesting relationships between HAP clusters and geographic location (Fig. 9), or subtype (Fig. 10). Higher-layer clusters are further broken down here so that they are typically mostly from one region, or of one dominant subtype, or both. Note that the first 6 clusters (counting across rows) are those associated with the third layer’s first cluster, the next 4 with the third layer’s second cluster, and the rest with the third layer’s third cluster. Clusters that have big geographical spread (such as the 7th pie chart in Fig. 9) are highly specific for the subtype (corresponding chart in Fig. 10). The ‘C’, ‘D’, and ‘E’ subtypes that were clustered together in the third layer are now separated by the second layer clustering. It is known that the ‘C’ group is predominantly from Asia and Africa, while ‘D’ is only seen in Africa (Goudsmit, 1997).

While it is encouraging to observe that known trends

about the HIV strains are captured in the HAP hierarchy, what we hope to do in future work is to evaluate hypotheses on strain migration by investigating outlier strains that cluster with unexpected groups. Those may indicate how the disease was spread, since the geographical location indicates where the strain was collected, but not necessarily the country of origin of the individual carrier.

### F.1 ROC Curve for the Protein Data

We provide the ROC curve for the protein data that was summarized in Sec. 5.4. In addition to the three state-of-the-art algorithms, ProteinProphet (Nesvizhskii et al., 2003), MSpresso (Ramakrishnan et al., 2009b) and MSnet (Ramakrishnan et al., 2009a), described in Sec. 5.4, we compared against HAP and a reformulation of the problem as a facility location problem.

## References

- Goudsmit, J. (1997). *Viral Sex; The Nature of AIDS*. Oxford University Press.
- Şahin Güvenç & Haldun, S. (2007). A review of hierarchical facility location models. *Computers & Operations Research*, *34(8)*, 2310–2331.
- Kantor, E. & Peleg, D. (2009). Approximate hierarchical facility location and applications to the bounded depth steiner tree and range assignment problems. *J. of Discrete Algorithms*, *7(3)*, 341–362.
- Lazic, N., Givoni, I. E., Aarabi, P. & Frey, B. J. (2009). FLoSS - facility location for subspace segmentation. In *ICCV*.
- Sontag, D., Meltzer, T., Globerson, A., Jaakkola, T. & Weiss, Y. (2008). Tightening lp relaxations for map using message passing. In *UAI*.