Chapter 5: Cross-partition Clustering

In this chapter, we move on from the elementary setting of the coupled-clustering approach described in the previous chapter to a more general approach, termed *cross-partition* (CP) clustering. Section 5.1 below details the differences between coupled clustering and the CP framework. After a detailed review in Section 5.2 of methods grounding our approach, Section 5.3 introduces the CP method. The method is demonstrated experimentally in Section 5.4 and is discussed further in the concluding Section 5.5.

5.1 Cross Partition versus Coupled Clustering

The CP setting generalizes some aspects of the problem treated by the coupled-clustering method. The aspects by which cross-partition clustering differs from the coupled-clustering method are:

- Cross-partition clustering is in general "soft". A data element can be assigned to several clusters, in varying assignment levels, at the same time. Specifically, the assignments are probabilistic, i.e., the assignment levels of any given element associating it with all clusters are all non-negative and sum up to 1 (see Chapter 2, Subsection 2.1.2.2).

- The coupled-clustering framework models analogies through producing clusters that contain elements of two distinct subsets of the data. However, other than the convention of thinking about analogies as involving two systems, there is no inherent reason for restricting the setting to two subsets. The cross-partition clustering setting allows pre-given partitioning of the data element set to more than two subsets, across which *correspondences* are to be drawn (talking about 'correspondence' might seem more appropriate than 'analogy' for this generalized setting).

- Formally, the cross-partition approach allows also 'soft' pre-partitions: elements can be probabilistically assigned to some or all of the pre-given subsets (which should not be confused with probabilistic assignments to clusters). Our experimental work, however, was restricted to the hard pre-partition setting.

- Another characteristic attribute of the coupled-clustering framework is that it assumes given pairwise similarity values that apply to pairs of elements of the two pre-given subsets. While in principle data might happen to be readily available in this form, our practical experience was with data consisting of co-occurrence counts of data elements with features. Co-occurrence counts can provide basis for calculating pairwise similarities (see Chapter 2, Subsection 2.1.3.4 and Chapter 4, Subsection 4.1.2), but they can be utilized also more directly. The cross-partition clustering

setting processes element-feature count distributions rather than pairwise similarities. Thus the mediating stage of computing similarities, which necessarily implies loss of information, is avoided.

- One last noticeable aspect of the coupled clustering method is that it ignores the whole collection of within-subset similarities altogether. The cross-partition method tackles the neutralization of within-subset regularities in a more principled manner.

5.2 Background: Information Theoretic Approaches

The CP method takes an information-theoretic approach to data clustering, which is related with the "communicative" aspect of data clustering (see Chapter 2, Subsection 2.1.1). Particularly, we elaborate on the *information bottleneck* data clustering method (IB, Tishby, Pereira & Bialek, 1999; Gilad-Bachrach, Navot & Tishby, 2003) and on an earlier variation on the same distributional-clustering theme (Pereira, Tishby & Lee, 1993), which has been recently studied further under the name *Information Distortion* clustering (ID, Gedeon, Parker & Dimitrov, 2003).

In Section 5.2.1, we discuss the ID method, which is taken as the basis for our elaboration, with an emphasis on the role of the maximum entropy principle (Jaynes, 1982) in this method. As the two methods are tightly related, some results obtained originally with regard to the IB method are cited as well. In Section 5.2.2, we refer more specifically to the IB method, which underlies additional variants of our CP algorithm. Both methods are described as aiming at minimization problems (rather than constrained minimization or other types of optimization). One further development around the IB theme, directly relevant to the CP task, is the method of *information bottleneck with side information* (IB-SI, Tishby & Chechik, 2003), reviewed in Section 5.2.3.

5.2.1 The Information Distortion Method

The ID method was introduced, under the name *distributional clustering*, by Pereira, Tishby & Lee (1993) and has recently been studied further by Gedeon, Parker & Dimitrov (2003).

5.2.1.1 Input and Output

As a probabilistic clustering method (see Chapter 2, Subsection 2.1.4.6), the ID method employs formal random variables X, Y and C with values ranging over all data elements, features and cluster labels, respectively. The relative frequency $p(x)$ of each data element x to occur in the given dataset and the conditional probabilities $p(y|x)$ of each feature y to occur in association with each element x are given as input. Based on this input, the ID method outputs a probability distribution $p(c|x)$ over the clusters for each element x. This distribution defines x's "assignment level" or "association level" with each cluster c. In addition, the ID method constructs conditional probability distributions, $p(y|c)$,

over all features for every cluster c. The $p(y|c)$ distributions can be considered as supplementary output, specifying a representative for each cluster c, a centroid in the feature space (Subsection 2.1.4.4).

5.2.1.2 Underlying Principles and Formulation

The ID method designates the relevance features as the sole basis for directing the formation of clusters: *clusters are formed so that they are optimally informative with regard to the feature distribution*. In order to determine the assignments of data elements into the formed clusters, the ID method applies also the maximum entropy principle (Jaynes, 1982) constrained by the first direction.

Many optimization problems (including data clustering, see Chapter 2, Subsection 2.1.4.2) are formulated so that they are solvable through minimization of a *cost term* or a *Lyapunov function* (often, the minimum practically obtained is local, implying sub-optimal solution). The ID method, as well, accomplishes the counterbalance between the two above principles, feature relevance and maximum entropy, through minimizing a single cost term – *the ID functional*:

$$F^{ID} \equiv -H(C|X) + \beta \hat{H}(Y|C). \qquad (5.1)$$

$\beta > 0$ is a parameter counterbalancing the relative impact of the two principles. $H(C|X)$ is the *entropy* of cluster distribution conditioned on the distribution of data elements

$$H(C|X) \equiv -\sum_{c,x} p(c,x) \log p(c|x) = -\sum_x p(x) \sum_c p(c|x) \log p(c|x), \qquad (5.2)$$

where x and c range over all possible values of the variables X and C, i.e., the sum runs over all cluster-element combinations. $\hat{H}(Y|C)$ is defined as

$$\hat{H}(Y|C) \equiv -\sum_{c,x} p(c,x) \sum_y p(y|x) \log p(y|c) = -\sum_x p(x) \sum_c p(c|x) \sum_y p(y|x) \log p(y|c). \qquad (5.3)$$

The conditional entropy $H(C|X)$ is the expected length of a transmission, communicating that C has the value c under the assumption that the value of X is known to be x. It quantifies the overall information that the data variable X leaves unexplained with regard to the cluster variable C, or in other words, the level of uncertainty regarding cluster distribution knowing the data distribution (see Thomas & Cover, 1991, p. 20)[1]. Following the maximum entropy principle, the goal of the ID method is to maximize this uncertainty expressed by $H(C|X)$ (subject to the constraint). Accordingly, the ID method seeks to *minimize* the F^{ID} term, which counterbalances *minus* $H(C|X)$ against $\hat{H}(Y|C)$.

---

[1] In general, the definition in Eq. (5.3) employs base 2 logarithm. However, as a change in the logarithm base adds a constant to the conditional entropy and other related values, we prefer to use throughout this chapter the natural log, which somewhat simplifies mathematical derivations.

The $\hat{H}(Y|C)$ term introduces the constraint of forming clusters that are informative with regard to the features ($\hat{H}(Y|C) = 0$, for instance, implies that the clusters completely determine the feature distribution). It incorporates an expected value of $p(y|c)$ distributions, averaged over the feature distribution $p(y|x)$ relatively to each data element x.

The ID method follows one further assumption that we term here *the ID conditional independence assumption* (also known as the *markovity assumption*, Gilad-Bachrach, Navot & Tishby, 2003), stating that clusters and features are assumed independent given the data:

$$p(c,y|x) \;=\; p(c|x)\,p(y|x) \tag{5.4}$$

for each x, c and y. (Equivalently, one may require that clusters and features would share zero mutual information given the data: $I(C;Y\,|X) = 0$.[2]) Taking the expected value over all x of both sides of Eq. 5.4, we get

$$p(c,y) = \textstyle\sum_x p(x)\,p(c,y|x) = \sum_x p(x)\,p(c|x)\,p(y|x)\,. \tag{5.5}$$

It follows that under the conditional independence assumption, $\hat{H}(Y|C)$ is exactly equal to the entropy $H(Y|C)$ of feature distribution conditioned on the clusters:

$$\begin{aligned} H(Y|C) \;\equiv\; & -\textstyle\sum_{c,y} p(c,y)\,\log p(y|c) \;= \\ & -\textstyle\sum_{c,y}\big(\sum_x p(x)\,p(c|x)\,p(y|x)\big)\,\log p(y|c) \;=\; \hat{H}(Y|C)\,, \end{aligned} \tag{5.6}$$

where c and y range over all possible values of the variables C and Y, i.e., the sum runs over all cluster-feature combinations. Therefore, if the independence assumption holds (which turns to be the case, as shown in the next subsection), we can rewrite F^{ID} (Eq. 5.1) as

$$L^{ID} \;=\; -H(C|X) + \beta H(Y|C)\,. \tag{5.7}$$

In conclusion, given that the independence assumption is satisfied, the ID method maintains a counterbalance between maximizing $H(C|X)$, to keep high uncertainty level regarding assignments into clusters, and minimizing $H(Y|C)$. $H(Y|C)$ measures the uncertainty about the feature distribution

---

[2] The explicit formula for the equivalent form of the conditional independence assumption is:

$$I(C;Y|X) \;=\; \sum_x p(x) \sum_{c,y} p(c,y\,|\,x) \log \frac{p(c,y\,|\,x)}{p(c\,|\,x)\,p(y\,|\,x)} \;=\; 0\,.$$

If for all c, x and y $p(c,y|x) = p(c|x)\,p(y|x)$ ($\neq 0$) then the arguments of all log terms in the sum are equal to 1 and hence $I(C;Y|X) = 0$. Suppose now $I(C;Y|X) = 0$. Mutual information amounts to a sum of *KL* divergences, each of which is non-negative and is equal to zero if and only if its arguments are identical distributions (Cover & Thomas, 1991 p. 19), i.e., $p(c,y|x) = p(c|x)\,p(y|x)$ for all c, x and y.

left after revealing cluster distribution and, therefore, minimizing $H(Y|C)$ realizes the principle with which this subsection opens: clusters are expected to be informative about feature distribution. As explained, the ID method follows this principle restrictedly: minimizing F^{ID} indeed decreases the above uncertainty so that formed clusters are informative with regard to the feature distribution, but only up to the level enabled by the maximum-entropy directed element assignments.

5.2.1.3 The ID Algorithm

The iterative ID algorithm was originally introduced by Pereira, Tishby & Lee (1993). The algorithm consists of two steps that update the $p(c|x)$ and $p(y|c)$ distributions, each in its turn, so that they accomplish the weighed balance between minimizing $H(Y|C)$ and maximizing $H(C|X)$, through consistent decrease of the F^{ID} value.

Set $t = 0$, and repeatedly iterate the two update-steps sequence below, till convergence (at time step $t = 0$, initialize $p_t(c|x)$ randomly or arbitrarily and skip step ID1):

ID1: $p_t(c|x) = \frac{1}{z_t(x,\beta)} e^{-\beta KL\left[p(y|x)\,\|\,p_{t-1}(y|c)\right]}$

where $z_t(x,\beta) = \sum_{c'} e^{-\beta KL\left[p(y|x)\,\|\,p_{t-1}(y|c')\right]}$

ID2: $p_t(y|c) = \frac{1}{p_t(c)} \sum_x p(x)\, p_t(c \mid x)\, p(y \mid x)$

where $p_t(c) = \sum_x p(x)\, p_t(c \mid x)$

$t = t + 1$

Figure 5.1: The iterative ID clustering algorithm (with fixed β, and $|C|$).

At the starting iterative cycle, when $t = 0$, the ID1 step just initializes the $p(c|x)$ distributions randomly or arbitrarily. At all later cycles (with $t > 0$), step ID1 updates all $p(c|x)$ values leaving $p(y|c)$ unchanged, so that the value of F^{ID} (Eq. 5.1) decreases as the following lemma shows. The second part of this lemma affirms that step ID2, which updates the $p(y|c)$ values leaving $p(c|x)$ fixed, decreases the value of F^{ID} as well.

Lemma 5.1:

(A) At any iterative cycle with $t > 0$, update step ID1 decreases the value of F^{ID} by

$$\Delta F^{ID1}{}_t = \sum_x p(x) KL[p_{t-1}(c|x) \| p_t(c|x)] . \quad (5.8)$$

(B) Update step ID2 decreases the value of F^{ID} by

$$\Delta F^{ID2}{}_t = \beta \sum_c p_t(c) KL[p_t(y|c) \| p_{t-1}(y|c)] . \quad (5.9)$$

Proof: see Appendix D (the proof of part (A) is original; (B) follows Gilad-Bachrach, Navot & Tishby, 2003)) □

Note that step ID2 affects only one of the components of F^{ID}, namely $\hat{H}(Y|C)$, as the other component of F^{ID}, $H(C|X)$, does not depend on $p(y|c)$. Step ID2 imposes the conditional independence assumption (Eq. 5.5). Therefore, at the end of each iterative cycle also the alternative formulation of F^{ID} (Eq. 5.7) is guaranteed to decrease (ID1 is only assured to decrease the value of F^{ID} as given in Eq. 5.1, as the independence assumption does not hold).

Lemma 5.2 (following Tishby, Pereira & Bialek, 1999)**:** Stable points of the ID algorithm (i.e., probability distributions that remain unchanged under the update steps, so that $p_{t+1}(c|x) = p_t(c|x)$ and $p_{t+1}(y|c) = p_t(y|c)$ for all c, x and y) are local extremum points of F^{ID} (Eq. 5.1).

Proof: see Appendix D □

Conclusion: The ID algorithm converges to a local minimum of F^{ID} (unless initialized to an extremal point of a different type).

Proof: From lemma 5.1, the value of F^{ID} decreases in each iterative cycle of the ID algorithm (with $t > 0$) by a non-negative quantity $\Delta F^{ID}{}_t = \Delta F^{ID1}{}_t + \Delta F^{ID2}{}_t$. As $-H(C|X) \geq -H(C) \geq -\log|C|$ and $\beta H(Y|C) \geq 0$, the value of F^{ID} is bounded from below and the algorithm converges to a locally minimal value (unless initialized to a stable value that is not minimal). From Lemma 5.2 it follows that those probability distributions, $p(c|x)$ for each x and $p(y|c)$ for each c, assigning to F^{ID} its stable value at the ID algorithm convergence point[3] define an extremal, hence (locally) minimal, point of F^{ID}. □

The ID algorithm is a version of the k-means scheme described in Chapter 2 (Subsection 2.1.4.4). Step ID1 assigns each element to each cluster in proportion to their similarity in the feature space as

---

[3] Gilad-Bachrach, Navot & Tishby (2003) show that, assuming the number of local minima is finite, the convergence is onto particular definite limit distributions (otherwise, it could have been the case that the sequence of distributions assigning the converging sequence of values to F^{ID} do not converge).

calculated in the previous update cycle. More concretely, each assignment $p_t(c|x)$ is in inverse proportion to the *KL* divergence between the feature vector representation of x ($p(y|x)$) and the centroid of c as calculated in the last iterative cycle ($p_{t-1}(y|c)$). The *KL* divergence is not an arbitrary dissimilarity measure, even though the general *k*-means scheme allows such arbitrariness. Rather, the *KL* divergence emerges from the ID cost term, so that it is particularly tailored to address the considerations underlying this cost. Step ID2 updates the $p_t(y|c)$ distributions so that they satisfy the conditional independence assumption and they are consistent with the input and the recently calculated $p_t(c|x)$ distributions.

5.2.1.4 Controlling the Number of Clusters by Modifying the Value of β

The value of the parameter β governs the tendency of the re-assignments performed by step ID1 to be probabilistic or deterministic. With $\beta = 0$, implying that only the $H(C|X)$ part of F^{ID} is articulated, assignments of each element x to all clusters c are in equal probability (so all clusters are in fact identical) as the unconstrained maximum entropy principle entails. For larger β values, assignments turn more discriminative. In the limit of $\beta \to \infty$, each element is assigned, with probability 1, to a distinct singleton (assuming the number of clusters $|C|$ is allowed to be as large as the number of data elements $|X|$ and unless there are elements with identical feature representations).

In between the above two extreme cases of zero and infinity, the value of β dictates the number of distinct clusters that can be formed in a manner resembling thermodynamics of physical systems, where β takes a role that is opposite to that of temperature. The higher β is, i.e., the stronger is the bias to construct clusters that convey detailed information regarding feature distribution, a larger number of distinct clusters is enabled. Specifically, for any given number of clusters $|C| = 2, 3, \ldots$, there is a minimal β value enabling the formation of $|C|$ distinct clusters. Setting β to be smaller than this critical value corresponding to the current $|C|$ would result in two or more duplications of the same cluster. Once β is raised just above the critical value, the same cluster would not duplicate any more: it splits, or *bifurcates*, to two distinct clusters due to the stronger emphasis on the requirement to convey feature information.

Based on the above dynamics, the iterative algorithm can be applied repeatedly within a gradual cooling-like, or *deterministic annealing*, scheme: starting with random initialization of the $p_0(c|x)$'s, generate two clusters, to be discovered empirically, with the critical β value for $|C| = 2$. Then, use a perturbation on the obtained two-cluster configuration to initialize the $p_0(c|x)$'s for a larger set of clusters and execute additional runs of the algorithm to identify the critical β value for the larger $|C|$. And so on: each output configuration is used as a basis for a more granular one. In our actual experiments, we always split one cluster – the largest one (of highest $p(c)$) – so each output

configuration includes one cluster more than its predecessor. The final outcome is a "soft hierarchy" of probabilistic clusters.

5.2.2 The Information Bottleneck Method

The IB method interprets clustering as a *distorted representation*, optimized for conveying the meaningful part of the information embodied within given data. In their presentation, Tishby, Pereira & Bialek (1999) base the IB method on the notion of *mutual information* rather than the conditional entropy we use. They define the IB cost term to be minimized (the *IB functional*) as

$$L^{IB} = I(C;X) - \beta I(Y;C). \qquad (5.10)$$

As $I(C;X) = H(C) - H(C|X)$ and $I(Y,C) = H(Y) - H(Y|C)$, it turns that F^{IB} closely resembles the ID cost term F^{ID} (Eq. 5.7). The two terms differ by subtraction of $\beta H(Y)$, which is a constant depending on β and the data, and by addition of $H(C)$ that is not a constant factor. Note that taking Eq. 5.10 as the IB cost term presumes an independence assumption, the same as in the ID method (Eqs. 5.4, 5.5). Gilad-Bachrach, Navot & Tishby (2003) explicate a Lyapunov function (corresponding to Eq. 5.1) that does not depend on this assumption.

As the IB and ID cost terms resemble each other, also the iterative IB algorithm that finds a local minimum for F^{IB} is similar to the ID algorithm (Figure 5.1):

Set $t = 0$, and repeatedly iterate the three update-steps sequence below, till convergence (at time step $t = 0$, initialize $p_t(c|x)$ randomly or arbitrarily and skip step ID1):

IB1: $$p_t(c|x) = \frac{1}{z_t(x,\beta)} p_{t-1}(c) e^{-\beta KL\left[p(y|x)\,\|\,p_{t-1}(y|c)\right]}$$

where $$z_t(x,\beta) = \sum_{c'} p_{t-1}(c') e^{-\beta KL\left[p(y|x)\,\|\,p_{t-1}(y|c')\right]}$$

IB2: $$p_t(c) = \sum_x p(x)\, p_t(c \mid x)$$

IB3: $$p_t(y|c) = \frac{1}{p_t(c)} \sum_x p(x)\, p_t(c \mid x)\, p(y \mid x)$$

$$t = t + 1$$

Figure 5.2: The iterative IB clustering algorithm (with fixed β, and $|C|$).

There are two differences between the IB algorithm and the ID algorithm. The IB algorithm includes a separate step for calculating $p_t(c)$. In the ID algorithm, the same calculation actually takes place but $p_t(c)$ has the mere role of a normalization factor. The other difference is that a prior of $p_{t-1}(c)$ is added to the term calculated in update step IB1. We hence occasionally refer to the IB algorithm as a

priored version of the ID algorithm, and to the ID algorithm as a *non-priored* version of the IB algorithm.

Lemma 5.3: The update cycle at time t decreases the value of L^{IB} (Eq. 5.10) by $\Delta F^{IB1}{}_t + \Delta F^{IB2}{}_t + \Delta F^{IB3}{}_t$, where

$$\begin{aligned} &\text{(A) } \Delta F^{IB1}{}_t &&= \sum_x p(x) KL[p_{t-1}(c|x) \| p_t(c|x)] , \\ &\text{(B) } \Delta F^{IB2}{}_t &&= KL[p_t(c) \| p_{t-1}(c)] , \\ &\text{(C) } \Delta F^{IB3}{}_t &&= \beta \sum_x p_t(c) KL[p_t(y|c) \| p_{t-1}(y|c)] . \end{aligned} \tag{5.11}$$

Proof: Minimizing L^{IB} is equivalent, under the appropriate independence assumption (Eq. 5.5), with minimizing the following term

$$F^{IB} \equiv H(C) - H(C|X) + \beta \hat{H}(Y|C) . \tag{5.12}$$

($\hat{H}$ is as in the definition in Eq. 5.3). It can be shown that step IB1 decreases F^{IB} by $\Delta F^{IB1}{}_t$ and step IB-3 decreases it by $\Delta F^{IB3}{}_t$, following the same argumentation as in Lemma 5.1 (A) and lemma 5.1 (B), respectively. A proof that step IB2 decreases F^{IB} by $\Delta F^{IB2}{}_t$, which relies on argumentation similar to that of the proof of lemma 5.1 (B), is given by Gilad-Bachrach, Navot & Tishby (2003). □

Lemma 5.4 (Tishby, Pereira & Bialek, 1999)**:** Stable points of the IB algorithm are locally extremal points of F^{IB} (Eq. 5.10).

Proof: The same idea as in the proof of Lemma 5.2 above (proved in Appendix D). □

From Lemmas 5.3 and 5.4, proof of convergence for the IB algorithm follows, as in the convergence proof of the ID algorithm (Subsection 5.2.1.3).[4]

5.2.2.1 The IB Method and Information Theory

The IB method draws an illuminative relation between data clustering and Claude Shannon's information theory. Rate-distortion theory (Thomas & Cover, 1991, ch. 13) shows that the average of number of bits needed for conveying a distorted (lossy) representation C of information X is the mutual information $I(C;X)$. (Minimizing this mutual information is equivalent with *maximizing* $H(X|C) = H(X) - I(C;X)$, which is the number of bits that are *saved* due to lossy encoding being employed, out of the $H(X)$ bits that are needed to represent X with no loss). The complementary

---

[4] In the same vein, for any cost term $-H(C|X) + \alpha H(C) + \beta H(Y|C)$, with positive α and β, there is an algorithm similar to the IB algorithm minimizing it. The modification required in order that the IB algorithm will work in this general case is replacing, in step IB1, the prior $p_{t-1}(c)$ with $p_{t-1}(c)^{\alpha}$. Then, Lemma 5.3 holds, with (B) replaced by $\Delta F^{B2}{}_t = \alpha KL[p_t(c) \| p_{t-1}(c)]$.

constraint of the IB method, maximizing $I(C;Y)$, which is completely equivalent with minimizing $H(Y|C)$ as done by the ID method, is related with another topic in information theory – the channel-capacity problem (Thomas & Cover, 1991, pp. 190–194). By combining these two classical problems into one doubly constrained problem, the IB method interprets data clustering as minimizing data representation size, liable to preserving the part that is most informative with regard to the features.

5.2.3 Information Bottleneck with Side Information

Recently, Chechik & Tishby (2003) introduced the method of information bottleneck with *side information* (IB-SI). Their approach emerged from recognizing that production of relevant clusters can be facilitated through considering attributes of the data that are *irrelevant* to the patterns to be revealed, in distinction from the standard relevance features. In order to incorporate the effect of these additional attributes, the IB-SI method introduces an additional set of *irrelevance features* represented by a new variable Y^-.

The IB-SI method, like the IB and ID methods, aims at minimizing a cost term. Specifically, the cost term to be minimized by the IB-SI method is:

$$L^{IB\text{-}SI} = I(C;X) - \beta I(Y^+;C) + \gamma I(Y^-;C)\,. \qquad (5.13)$$

This term incorporates the impact of the irrelevance features Y^- as if it symmetrically opposes the bias introduced by the relevance features (represented here by Y^+, rather than by Y). As in the derivation of the IB and ID algorithms, an iterative algorithm can be based on the IB-SI cost term:

Set $t = 0$, and repeatedly iterate the four update-steps sequence below, till convergence (at time step $t = 0$, initialize $p_t(c|x)$ randomly or arbitrarily and skip step IB-SI1):

IB-SI1: $p_t(c|x) = \frac{1}{z_t(x,\beta)} p_{t-1}(c) e^{-\beta\left(KL[p(y^+|x)\|p_{t-1}(y^+|c)] - KL[p(y^-|x)\|p_{t-1}(y^-|c)]\right)}$

where $z_t(x,\beta) = \sum_{c'} p_{t-1}(c') e^{-\beta\left(KL[p(y^+|x)\|p_{t-1}(y^+|c')] - KL[p(y^-|x)\|p_{t-1}(y^-|c')]\right)}$

IB-SI2: $p_t(c) = \sum_x p(x) p_t(c|x)$

IB-SI3: $p_t(y^+|c) = \frac{1}{p_t(c)} \sum_x p(x) p_t(c|x) p(y^+|x)$

IB-SI4: $p_t(y^-|c) = \frac{1}{p_t(c)} \sum_x p(x) p_t(c|x) p(y^-|x)$

$t = t + 1$

Figure 5.3: The IB-SI clustering iterative algorithm (with fixed β, γ and $|C|$).

The two lemmas below are the IB-SI equivalents of the ID and IB lemmas (Subsections 5.2.1.3, 5.2.2).

Lemma 5.5: The update cycle at time t subtracts from the value of $L^{IB\text{-}SI}$ (Eq. 5.13) $\Delta F^{SI1}{}_t + \Delta F^{SI2}{}_t + \Delta F^{SI3}{}_t - \Delta F^{SI4}{}_t$, where

$$\begin{aligned} &\text{(A) } \Delta F^{SI1}{}_t &&= \textstyle\sum_x p(x) KL[p_{t-1}(c|x) \| p_t(c|x)] , \\ &\text{(B) } \Delta F^{SI2}{}_t &&= KL[p_t(c) \| p_{t-1}(c)] , \\ &\text{(C) } \Delta F^{SI3}{}_t &&= \beta \textstyle\sum_x p_t(c) KL[p_t(y^+|c) \| p_{t-1}(y^+|c)] , \\ &\text{(D) } \Delta F^{SI4}{}_t &&= \gamma \textstyle\sum_x p_t(c) KL[p_t(y^-|c) \| p_{t-1}(y^-|c)] . \end{aligned} \tag{5.14}$$

Proof: Minimizing $F^{IB\text{-}SI}$ is equivalent, under the appropriate independence assumption (as in Eq. 5.5, incorporating Y^-, symmetrically to Y^+), to minimizing the following term

$$F^{IB\text{-}SI} \equiv H(C) - H(C|X) + \beta \hat{H}(Y^+|C) - \gamma \hat{H}(Y^-|C) . \tag{5.15}$$

($\hat{H}$ is as in the definition in Eq. 5.3). Following argumentation similar to that of Lemmas 5.1 and 5.3, it can be shown that step IB-SI1 decreases $L^{IB\text{-}SI}$ by $\Delta F^{SI1}{}_t$, step IB-SI2 decreases it by $\Delta F^{SI2}{}_t$, step IB-SI3 decreases it by $\Delta F^{SI3}{}_t$ and step IB-SI2 *increases* it by $\Delta F^{SI4}{}_t$. □

Lemma 5.6: Stable points of the IB-SI algorithm are extremal points of $F^{IB\text{-}SI}$ (Eq. 5.13).

Proof: Following the same argumentation as in Lemmas 5.2 and 5.4 above. □

However, the argumentation used for proving the convergence of the ID and IB algorithms (Subsections 5.2.1.3, 5.2.2) cannot be applied in the IB-SI case. Convergence of the IB-SI algorithm depends on the ratio between $\Delta F^{SI1}{}_t + \Delta F^{SI2}{}_t + \Delta F^{SI3}{}_t$ and $\Delta F^{SI4}{}_t$, thus cannot be proven for any arbitrary combination of β, γ, $|C|$ and a given dataset.

The IB-SI approach extends the IB method, thus it facilitates explaining clustering with side information in classical information theoretical terms. A slightly different approach to clustering with side information is based on the ID method. The underlying cost-term of this ID-based variant is

$$L^{ID\text{-}SI} = -H(C|X) + \beta H(Y^+|C) - \gamma H(Y^-|C) . \tag{5.16}$$

From this cost term the following iterative algorithm is derived:

Set $t = 0$, and repeatedly iterate the three update-steps sequence below, till convergence (at time step $t = 0$, initialize $p_t(c|x)$ randomly or arbitrarily and skip step ID-SI1):

ID-SI1: $$p_t(c|x) = \frac{1}{z_t(x,\beta)} e^{-\beta\left(KL[p(y^+|x)\|p_{t-1}(y^+|c)]-KL[p(y^-|x)\|p_{t-1}(y^-|c)]\right)}$$

where $$z_t(x,\beta) = \sum_{c'} e^{-\beta\left(KL[p(y^+|x)\|p_{t-1}(y^+|c')]-KL[p(y^-|x)\|p_{t-1}(y^-|c')]\right)}$$

ID-SI2: $$p_t(y^+|c) = \frac{1}{p_t(c)} \sum_x p(x) p_t(c \mid x) p(y^+ \mid x)$$

where $$p_t(c) = \sum_x p(x) p_t(c \mid x)$$

ID-SI3: $$p_t(y^-|c) = \frac{1}{p_t(c)} \sum_x p(x) p_t(c \mid x) p(y^- \mid x)$$

where $p_t(c)$ is as above

$$t = t + 1$$

Figure 5.4: The ID-SI clustering iterative algorithm (with fixed β, γ and $|C|$).

Observations equivalent to the ones made above with regard to the IB-SI algorithm (Lemmas 5.5, 5.6) similarly hold with regard to the ID-SI algorithm.

5.3 The Cross-partition Method

Cross-partition (CP) data clustering aims at identifying, through clusters of data elements, themes that are common to several subsets that together form the given dataset. To this end, the formed clusters should *cut across* the pre-given partition into subsets: each cluster is expected to contain elements from all subsets. As mentioned, the CP problem generalizes the coupled-clustering setting of the previous chapter (the noticeable differences between the methods are detailed in Section 5.1 above). The basic setting is described in Chapter 3.

In this section, we introduce a novel approach to the CP clustering task. This task is particularly challenging in cases where the given subsets are relatively homogenous, i.e., the elements within each subset are typically more similar to one another compared to their similarity to elements of other subsets. The suggested method is designed to overcome such cases. Lead by feature information that is shared across the subsets, it produces clusters that capture the commonalities while neutralizing possibly salient within-subset regularities. Our method is inspired by the ID and IB methods reviewed above. Below, we describe how the CP method extends the ID data clustering setting

(Subsection 5.3.1). Then, we characterize the desired form of solution to this task (Subsection 5.3.2) and present the algorithm that we propose in order to accomplish it (Subsection 5.3.3). Finally, we specify additional versions of our algorithm, motivated by the differences between the IB and ID methods (Subsection 5.3.4).

5.3.1 The Cross-partition Data Clustering Task

The CP method, which addresses the CP data clustering task introduced in Chapter 3, extends the standard probabilistic clustering setting in terms of both input accepted and output constructs being produced, as described below.

5.3.1.1 Input: The Pre-partitioning Variable

The identity of the pre-given subset to which a particular data element belongs is a source of information that plays a role in the CP clustering task, additional to the relevance feature distribution. In order to articulate this information, we introduce an additional formal variable W, the *pre-partitioning variable*. The values that W can get range over the labels of the subsets of the pre-given partition (two or more subsets). We denote that a data element x belongs to a subset w, by writing $p(w|x) = 1$. If x does not belong to w, we write $p(w|x) = 0$. In our experiments (Section 5.4), we have restricted each element to be uniquely associated with one subset. Allowing $p(w|x)$ values between 0 and 1 would enable probabilistic ("soft") pre-partitioning, which accords with our formalism but have not been empirically tested. The pre-partitioning information $p(w|x)$ is supplemented to $p(x)$ and $p(y|x)$ as input considered by the CP method.

We note that in order that the CP method produces meaningful results, the pre-partitioning variable W is expected to correlate to some extent with the feature variable Y (i.e., $I(Y;W) > 0$, or equivalently $H(Y) > H(Y|W)$), additionally to the correlation between X and Y that is essential also for standard data clustering.

5.3.1.2 Output: Re-association of Features and Clusters

A common way to convey the essence of a cluster c in the probabilistic setting is to specify those elements x with highest $p(c|x)$ scores. It is interesting, as well, to specify in addition the features that are most typical to a cluster. We note in Chapter 2 that the centroid of a cluster c – the $p(y|c)$ feature distribution – indicates the location of the cluster in the feature space. However, the features that are most characteristic for a cluster c are those features y with high $p(c|y)$ scores rather than high $p(y|c)$, as the latter might reflect the fact that the feature y appears frequently in all clusters and not discriminatively in c.

In the ID (or IB) setting, $p(c|y)$ can be straightforwardly calculated through Bayes rule: $p(c|y) = p(y|c)\,p(c)\,/\,p(y)$. A novel aspect of the CP method is that it quantifies differently the level of association of features with clusters. Hence, along with the probability distributions $p(c|x)$, the CP method outputs distributions that associate each feature y to each one of the clusters c. We denote these probabilities by $p^*(c|y)$, to emphasize the fact that they are different than the $p(c|y)$ distributions of the ID case.

As in the ID case, the CP method produces representative probability distributions of features for each cluster. These representative distributions are derived from the newly introduced $p^*(c|y)$, hence denoted $p^*(y|c)$. Finally, the CP method produces yet another type of supplementary output, which has no correspondence in the ID and IB methods: for each combination of a cluster c and a pre-given subset w, a probability distribution over the features $p(y|c,w)$. Such distributions form feature-based centroids of c restricted to the elements originated in w or, as we term them, *W-projected centroids*.

5.3.2 Underlying Principles Characterizing the Solution

As stated above, there are four types of probability distributions that together form the CP method output: $p(c|x)$, which can be considered as the main target of the method, and in addition $p(y|c,w)$, $p^*(c|y)$ and $p^*(y|c)$ (where c, x, y and w denote cluster labels, data elements, features and pre-given subset labels, respectively). These four types of distributions constitute the whole set of parameters that the CP method manipulates.

The core idea of the CP method lies in the step implementing feature-cluster re-association conveyed through the $p^*(c|y)$ distributions (see Subsection 5.3.2.3 below). The associations of the characterizing features with the formed clusters are biased so that these associations become *independent* of the given pre-partition of the data. The conception and formulation of this imposed independence, underlying the focusing on relevant cross-system information and the defocusing of irrelevant system-specific information, is the basis to our original interpretation of the notion of analogy.

Below, we characterize in detail how all four types of probability distributions link up together as a solution to the CP clustering task.

5.3.2.1 Assignments of Elements to Clusters

Similarly to the case with the ID and IB methods, the assignments of elements to clusters in the CP method follow a maximum entropy principle. This is formalized through the following term:

$$F^{CP1} \equiv -H(C|X) + \beta \hat{H}^*(Y|C), \tag{5.17}$$

where $\beta > 0$ is a counterbalancing parameter, $\hat{H}^*(Y|C)$ is defined to be

$$\hat{H}^*(Y|C) \equiv -\sum_{c,x} p(c,x) \sum_y p(y|x) \log p^*(y|c) = -\sum_x p(x) \sum_c p(c|x) \sum_y p(y|x) \log p^*(y|c) \quad (5.18)$$

and $H(C|X)$ is the entropy of cluster distribution conditioned on the data (Eq. 5.2), the constrained value of which the CP method seeks to maximize. The target of the CP method is thus to find $p(c|x)$ values, constrained to sum up to 1 for each x, which bring the value of F^{CP1} to minimum. Following this direction, the CP method maximizes $H(C|X)$, subject to a further constraint involving the probability distributions $p^*(y|c)$. For the purpose of assigning elements to clusters, the $p^*(y|c)$ distributions are considered as if they are given and fixed. We will refer later to how the CP method modifies these distributions (Subsection 5.3.2.4).

5.3.2.2 *W*-projected Centroids

As the case is with the IB and ID methods, the CP method aims at identifying a partition of the data that is optimally informative about the relevance features, represented by the variable Y. Such configuration may consider as an information source relevant to predicting feature distribution not only the partition to clusters, C, but also the pre-given partition W. Consequently, optimizing the feature information extractable from the two partitions together would be carried out through minimizing the conditional entropy term $H(Y|C,W)$. To be more precise, the CP method actually minimizes a related term, which is equivalent, under an appropriate independence assumption (explicated below), to $H(Y|C,W)$:

$$F^{CP2} \equiv \sum_x p(x) \sum_c p(c|x) \sum_y p(y|x) \sum_w p(w|x) \log p(y|c,w) . \quad (5.19)$$

The conditional independence assumption of the IB and ID methods (Eq. 5.4), is extended by the CP method to apply to W as well, namely C, Y and W are independent given X:

$$p(c,y,w|x) = p(c|x)\, p(y|x)\, p(w|x) \quad (5.20)$$

(for each c, x, y and w), or equivalently $I(C;Y;W|X) = 0$.

Summing up both sides of Eq. 5.20 over all x values, we obtain

$$p(c,y,w) = \sum_x p(x)\, p(c|x)\, p(y|x)\, p(w|x) . \quad (5.21)$$

Assuming Eq. 5.21 holds, we can re-write F^{CP2}:

$$F^{CP2} = \sum_{c,y,w} \log p(y|c,w) \sum_x p(x)\, p(c|x)\, p(y|x)\, p(w|x) = \quad (5.22)$$

$$\sum_{c,y,w} \log p(y|c,w)\; p(c,y,w) = H(Y|C,W) .$$

As we will see in Subsection 5.3.3, the independence assumption is indeed maintained by the CP method. Therefore, from Eq. 5.22 we conclude that by minimizing F^{CP2} the CP method minimizes

the conditional entropy term $H(Y|C,W)$ or, in other words, it optimizes the level of information about the features provided by the combination of the clusters C and the pre-given partition W.

5.3.2.3 Feature-cluster Re-association

As said above (in Subsection 5.3.1.2), an innovative aspect of the CP approach is that it re-associates features with clusters differently than what is straightforwardly expected from the assignments of data elements into clusters. Re-associating features with clusters is carried out so that the associations reflect the following fundamental principle:

> *The way features (Y) and clusters (C) are associated is not supposed to correlate with the pre-partition (W).*

Assuming a triply joint probability distribution $p^*(c,y,w)$ (where the asterisk comes to distinguish between this probability to the one in Eq. 5.21), the above principle would be formulated as:

$$P^*(c,y,w) \;=\; p^*(c,y)\,p(w) \tag{5.23}$$

(for each c, y and w), or equivalently $I^*(C,Y;W) = 0$.

Under the assumption that Eq. 5.23 holds, we formulate below a second maximum entropy principle that the solution to the CP problem is supposed to realize, which would direct the re-association of features with clusters. Specifically, the CP method aims at minimizing the following term:

$$F^{CP*1} \;\equiv\; -H^*(C|Y) + \eta \hat{H}(Y|C,W)\,, \tag{5.24}$$

where, $H^*(C|Y)$ is a conditional entropy term of cluster distribution conditioned on the distribution of features

$$H^*(C|Y) \;\equiv\; -\sum_{c,y} p^*(c,y) \log p^*(c|y) \;=\; -\sum_y p(y) \sum_c p^*(c|y) \log p^*(c|y) \tag{5.25}$$

(the sum runs over all cluster-feature combinations), and $\hat{H}(Y|C,W)$ is defined to be

$$\hat{H}(Y|C,W) \;\equiv\; \sum_{c,y,w} p^*(c,y,w) \log p(y|c,w) \;=\; \sum_w p(w) \sum_y p(y) \sum_c p^*(c|y) \log p(y|c,w)\,. \tag{5.26}$$

η is a parameter with a positive value, counterbalancing the relative impact of the two components of F^{CP*1}. $\hat{H}(Y|C,W)$ articulates the constraint on the maximum entropy principle posed by the w-projected centroids. The target of the CP method is to find feature-cluster probabilistic associations $p^*(c|y)$, minimizing F^{CP*1} while being constrained to sum up to 1 for each y. Thus, the CP method maximizes the conditional entropy $H^*(C|Y)$, subject to a further constraint posed by the probability distributions $p(y|c,w)$. Although we have already seen in the previous subsection how the $p(y|c,w)$ distributions are to be determined, for the purpose of re-associating features with clusters these constraining distributions are referred to as if they are given and fixed.

5.3.2.4 Centroids that Cut Across the Pre-partition

Finally, there are the centroids distributions used in characterizing the assignments of the solution to the CP problem (Subsection 5.3.2.1). These distributions are expected to minimize the following term:

$$F^{CP*2} = \sum_y p(y) \sum_c p^*(c|y) \log p^*(y|c), \qquad (5.27)$$

which is identical to the conditional entropy $H^*(Y|C)$, as $p(y)\,p^*(c|y) = p^*(c,y)$. However, the $p^*(y|c)$ values are referred by F^{CP*2} as variables, while $p^*(c|y)$ are treated as if they are given and fixed.

5.3.3 The CP Algorithm

Starting from a random or arbitrary clustering configuration, the CP algorithm updates iteratively the four types of probability distributions $p(c|x)$, $p(y|c,w)$, $p^*(c|y)$ and $p^*(y|c)$. The algorithm's iterative update cycle follows the four principles described in the previous subsection. Each step of the cycle optimizes one class of probability distributions relatively to one of the above principles, while the other distributions are held constant.

Set $t = 0$ and repeatedly iterate the following update steps sequence, till convergence (in the first iteration, when $t = 0$ randomly or arbitrarily initialize $p_t(c|x)$ and skip step CP1):

CP1: $$p_t(c|x) = \frac{1}{z_t(x,\beta)} e^{-\beta KL\left[p(y|x)\,\|\,p^*_{t-1}(y|c)\right]}$$

where $$z_t(x,\beta) = \sum_{c'} e^{-\beta KL\left[p(y|x)\,\|\,p^*_{t-1}(y|c')\right]}$$

CP2: $$p_t(y|c,w) = \frac{1}{p_t(c,w)} \sum_x p(x)\,p_t(c\,|\,x)\,p(y\,|\,x)\,p(w\,|\,x)$$

where $$p_t(c,w) = \sum_x p(x)\,p_t(c\,|\,x)\,p(w\,|\,x)$$

CP*1: $$p^*_t(c|y) = \frac{1}{z^*_t(y,\eta)} \prod_w p_t(y\,|\,c,w)^{\eta p(w)}$$

where $$z^*_t(y,\eta) = \sum_{c'} \prod_w p_t(y\,|\,c',w)^{\eta p(w)}$$

CP*2: $$p^*_t(y|c) = \frac{1}{p^*_t(c)} \sum_y p(y)\,p^*_t(c\,|\,y)$$

where $$p^*_t(c) = \sum_y p(y)\,p^*_t(c\,|\,y)$$

$$t = t + 1$$

Figure 5.5: The cross partition clustering iterative algorithm (with fixed β, η, and $|C|$).

The CP method probabilistically associates, or assigns, elements to clusters in proportion to the element-centroid similarity (step CP1 of the algorithm; in this respect, the CP method follows the *probabilistic representative-based clustering scheme*, Chapter 2, Subsection 2.1.4.6). More specifically, as in the IB and ID algorithms, an element x is assigned into a cluster c in proportion exponentially inverse to the *KL* divergence between the representative feature distributions $p(y|x)$ and $p^*_{t-1}(y|c)$. The *KL* divergence is not an arbitrary proximity measure but it rather emerges from the first maximum-entropy principle described in 5.3.2.1 above (see also Lemma 5.8 below).

Based on the assignments calculated by step CP1, the next step, CP2, calculates expected values of the current W-projected centroid distributions, $p_t(y|c,w)$. This step conforms to the information maximization direction of 5.3.2.2 above. Particularly, step CP2 imposes the extended conditional independence assumption (Eqs. 5.20, 5.21).

Step CP*1 re-associates features with clusters, by calculating for every feature y probability distribution over the clusters $p^*_t(c|y)$ proportional to biased ('flattened') geometric mean over all current $p_{t-1}(y|c,w)$ values. This step facilitates feature-cluster associations that cut across the pre-partitioned subsets: strong association of a feature y with a cluster c, i.e., a high $p^*_t(c|y)$ value, requires high $p_t(y|c,w)$ values across all subsets w (in contrast to some y' and c' for which $p_t(y'|c',w)$ is high on average but varies across the w's). The bias introduced within this weighted geometric-mean scheme is directed by a free parameter η. Low values of η underlie loss of information: $\eta = 0$ implies that all features are uniformly assigned to all clusters regardless of the $p_t(y|c,w)$ values. Higher η values preserve more of the information embodied within the W-projected centroids. Regardless of the value of η, step CP*1 integrates $p_t(y|c,w)$ values over all values of W, so the result is independent of any particular w. This scheme, which was motivated intuitively in Dagan, Marx & Shamir (2002), turns to realize the supplementary maximum-entropy direction introduced in Subsection 5.3.2.3.

Step CP*2 derives the $p^*_t(y|c)$ probability distributions, which are the centroids for the next update cycle, from the current $p^*_t(c|y)$ values and the input $p(y)$ distribution through Bayes rule. It realizes the information-maximization principle of Subsection 5.3.2.4.

In the case of the ID and IB algorithms, the existence of a cost term, that gets a smaller value at each update step, ensures the convergence to a configuration locally minimizing the value of the corresponding term. For the CP method, we cannot specify a cost term that is reduced by each update step, or by the whole update cycle. The algorithm, however, empirically converges in most examined test cases, particularly for all real-world and synthetic datasets where it has been reasonable to assume an underlying cross-partition structure (see Section 5.4 below). Whenever the algorithm converges,

the resulting stable-point probability distributions $p(c|x)$, $p(y|c,w)$, $p^*(c|y)$ and $p^*(y|c)$ necessarily maintain the relations between the distributions explicated in Subsection 5.3.2.

5.3.3.1 Further Observations

Below, we show that each one of the CP algorithm update steps "improves" the currently given configuration relatively to the principle corresponding to this step (unless the current configuration is a stable point of the algorithm). This is done by reducing the term associated with that step relatively to the particular distributions that the step updates (this, however, does not imply that the CP algorithm iterative cycle reduces the value of any *single* cost term).

Lemma 5.7: In the update cycle of time t, the four CP algorithm update steps CP1, CP2, CP*1, CP*2, decrease the values of F^{CP1} (Eq. 5.17), F^{CP2} (Eq. 5.19), $F^{\mathrm{CP*1}}$ (Eq. 5.24), $F^{\mathrm{CP*2}}$ (Eq. 5.27) by

$$\begin{aligned}
&\text{(A) } \Delta F^{CP1}{}_t &&= \sum_x p(x) KL[p_{t-1}(c|x) \| p_t(c|x)], \\
&\text{(B) } \Delta F^{CP2}{}_t &&= \sum_x p_t(c,w) KL[p_t(y|c,w) \| p_{t-1}(y|c,w)], \\
&\text{(C) } \Delta F^{\mathrm{CP*1}}{}_t &&= \sum_y p(y) KL[p^*_{t-1}(c|y) \| p^*_t(c|y)], \\
&\text{(D) } \Delta F^{\mathrm{CP*2}}{}_t &&= \sum_y p^*_t(c) KL[p^*_t(y|c) \| p^*_{t-1}(y|c)],
\end{aligned} \tag{5.28}$$

respectively.

Proof: see Appendix D □

Lemma 5.8: A set of probability distributions that form a stable point of the CP algorithm (i.e., ones satisfying $p_{t+1}(c|x) = p_t(c|x)$, $p_{t+1}(y|c,w) = p_t(y|c,w)$, $p^*_{t+1}(c|y) = p^*_t(c|y)$ and $p^*_{t+1}(y|c) = p^*_t(y|c)$, for all c, x, y and w) specifies locally extremal points for: F^{CP1} with respect to $p(c|x)$ ($p^*(y|c)$ held fixed), F^{CP2} with respect to $p(y|c,w)$ ($p(c|x)$ held fixed), $F^{\mathrm{CP*1}}$ with respect to $p^*(c|y)$ ($p(y|c,w)$ held fixed) and $F^{\mathrm{CP*2}}$ with respect $p^*(y|c)$ ($p^*(c|y)$ held fixed).

Proof: see Appendix D □

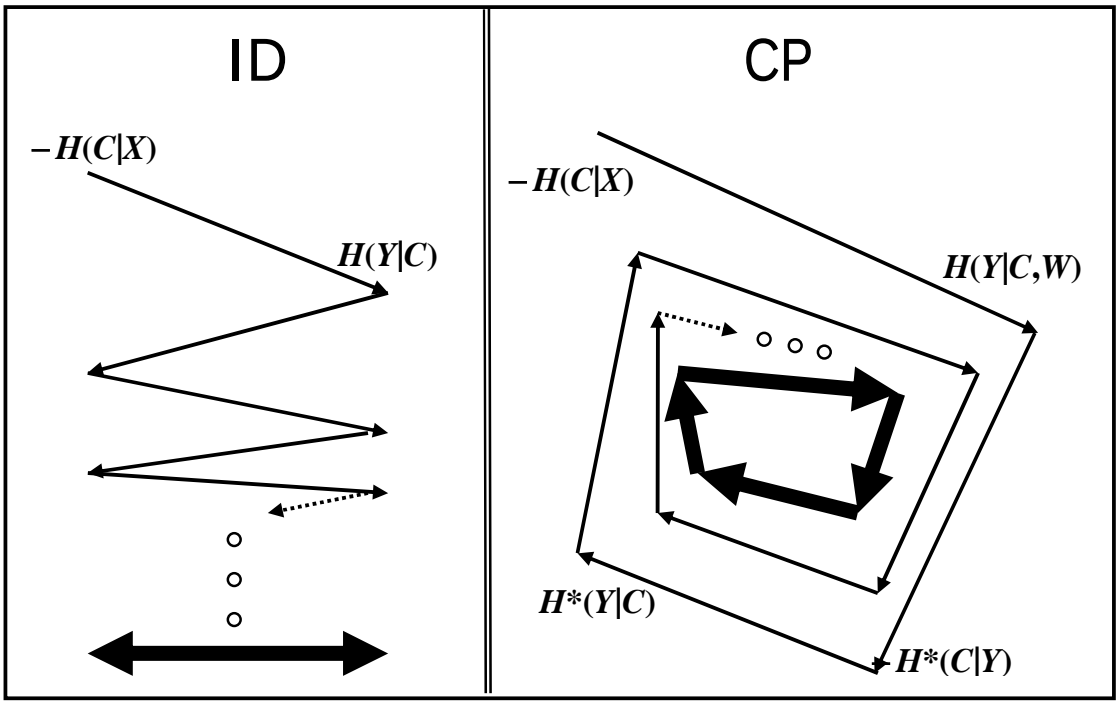

Figure 5.6: A schematic illustration of the dynamics of the ID algorithm versus that of the CP algorithm. In the ID algorithm, convergence is onto a configuration where the two systems of parameters complementarily balance one another, bringing a cost term to a locally minimal value. In the CP algorithm, stable configurations maintain balanced inter-dependencies (*equilibrium*) of four distinct systems of parameters.

During the execution of the CP algorithm, each of the four probability distribution types, $p(c|x)$, $p(c|y,w)$, $p^*(c|y)$ and $p^*(y|c)$, directs the formation of distributions of another type. In the resulting solution, the four types of conditional probability distributions take part in a closed cycle of dependencies, as described in Subsection 5.3.2. The dynamics characterizing the CP algorithm, in comparison to that of the ID algorithm, is illustrated in Figure 5.6.

Argumentation as used in the ID and IB cases cannot be used for proving convergence of the CP algorithm. Lemmas 5.7 and 5.8 do not provide information regarding how each of F^{CP1}, F^{CP2}, $F^{\mathrm{CP*1}}$ and $F^{\mathrm{CP*2}}$ is affected by the changes that occur in practice in factors that are considered to be fixed by the principle underlying its modification.

5.3.3.2 The Parameters β and η

Gradual increase of the value of β works in practice for the CP method much the same as it works for the IB and ID methods (Subsection 5.2.1.4): increasing β along subsequent runs enables the formation of configurations of growing numbers of clusters, each initialized based on a configuration of fewer clusters obtained previously. In general, we have experimented with η values that are fixed during a

whole cycle of runs, while only β is gradually incremented in order to produce increasing number of clusters. Understanding better the role of η and the inter-dependencies between the given data, η and β would be an interesting topic for future research.

There are two cases where the scheme of incrementing β gradually while η is held fixed, in order to produce a growing number of clusters, seems not to work. First, we encountered cases of synthetic datasets (randomly drawn with no underlying pre-tailored cross-partition structure) where the algorithm eventually did not converge but rather went through an endless oscillatory pattern. This behavior, characterized further in Subsection 5.4.1.3, took place for restricted ranges of η values. Convergence was always obtained for some η values outside these ranges. Further, for those datasets where underlying cross-partition either existed by construction or was expected to exist based on the content of the data (as in the experimental work Section 5.4) the algorithm converged with no exception.

The other potentially problematic scenario is where the CP algorithm converges to fewer clusters than initialized: some clusters are gradually vanished as update cycles keep being performed. Note that this never happens in the iterative IB and ID algorithms, where the formation of a centroid (step ID2/IB3) implies that there is some mass of data elements concentrated around it, ensuring that some points would be reassigned to the corresponding cluster in the next re-assignment step (ID1/IB1). In contrast, the formation of a CP centroid (step CP*2) does not guarantee that the centroid is backed with enough mass of data elements from all subsets. As a result, it might happen that the dominant features in a centroid formed in the previous update cycle are not sufficiently weighty in one or more of the subsets and hence the relative total weight of the cluster might tend to zero as the iterations are carried on.[5] This behavior was observed in a variety of cases. Particularly for very detailed pre-partitons (high $|W|$), we were not able to produce even small numbers of clusters. On the other hand, in the $|W| \leq 5$ cases to which the experimental part of this work was restricted, setting a lower η value whenever such behavior occurred consistently lead to the formation of the desired number of clusters. The effect of $|W|$ on the behavior of the algorithm (in interaction with other factors) should be studied further, both experimentally and theoretically.

[5] Somewhat related to this might be our empirical observation that the IB/ID iterative algorithms, although formally guaranteed to converge, often produce small clusters that do not capture significant themes in the data.

5.3.4 CP Algorithmic Variations Inspired by the IB Method

The IB method reviewed in Subsection 5.2.2 minimizes the cost term L^{IB} (Eq. 5.10), which, up to a constant factor, can be re-written as $H(C) - H(C|X) + \beta H(Y|C)$. As already noted, this term is reminiscent of the ID method cost term $L^{ID} = -H(C|X) + \beta H(Y|C)$ (Eq. 5.7). The difference lies in the non-constant term $H(C)$. In a like manner, it is possible to modify the two maximum-entropy-based terms of the equations underlying the CP method, namely F^{CP1} and F^{CP*1} (Eqs. 5.17 and 5.24). Thus, we may replace F^{CP1} by

$$F^{CP1'} \equiv H(C) - H(C|X) + \beta \hat{H}^*(Y|C) . \tag{5.29}$$

Regardless of the modification explicated by Eq. 5.29, we can also replace F^{CP*1} by

$$F^{CP*1'} \equiv H^*(C) - H^*(C|Y) + \eta \hat{H}(Y|C,W) , \tag{5.30}$$

where $H(C)$ and $H^*(C)$ are the entropy of C based on $p(c)$ and $p^*(c)$ respectively.

Set $t = 0$ and repeatedly iterate the following update steps sequence, till convergence (in the first iteration, when $t = 0$ randomly or arbitrarily initialize $p_t(c|x)$ and $p^*_t(c)$ and skip step CP1):

CP1': $p_t(c|x) = \frac{1}{z_t(x,\beta)} p_{t-1}(c) e^{-\beta KL[p(y|x) \| p^*_{t-1}(y|c)]}$

where $z_t(x,\beta) = \sum_{c'} p_{t-1}(c') e^{-\beta KL[p(y|x) \| p^*_{t-1}(y|c')]}$

CP2': $p_t(c) = \sum_x p(x) p_t(c|x)$

CP3': $p_t(y|c,w) = \frac{1}{p_t(c,w)} \sum_x p(x) p_t(c|x) p(y|x) p(w|x)$

where $p_t(c,w) = \sum_x p(x) p_t(c|x) p(w|x)$

CP*1': $p^*_t(c|y) = \frac{1}{z^*_t(y,\eta)} p_{t-1}{}^*(c) \prod_w p(y|c,w)^{\eta p(w)}$

where $z^*_t(y,\eta) = \sum_{c'} p_{t-1}{}^*(c') \prod_w p_t(y|c',w)^{\eta p(w)}$

CP*2': $p^*_t(c) = \sum_y p(y) p^*_t(c|y)$

CP*3': $p^*_t(y|c) = \frac{1}{p_t{}^*(c)} p(y) p^*_t(c|y)$

$t = t + 1$

Figure 5.7: The CP_{III} iterative algorithm (with fixed β, η, and $|C|$).

In case both above modifications take place, i.e. F^{CP1} is replaced by $F^{CP1'}$ and $F^{CP*1'}$ is replaced by $F^{CP*1'}$, a new algorithm, the CP_{III} algorithm (Figure 5.7), is derived in much the same way the CP algorithm (Figure 5.5) is derived. This algorithm was introduced, grounded on a different information-theoretic motivation, by Marx, Dagan & Shamir, 2004.

It is possible to replace only one of the two terms: either F^{CP1} by $F^{CP1'}$, or F^{CP*1} by $F^{CP*1'}$. If only F^{CP1} is replaced by $F^{CP1'}$, we derive the CP_{II} algorithm, which consists of update steps CP1', CP2', CP3' of the CP_{III} algorithm and steps CP*1, CP*2 of the original CP algorithm. The CP_{II} algorithm was introduced with intuitive motivation by Dagan, Marx & Shamir, 2002. If only the F^{CP*1} term is replaced by $F^{CP*1'}$, we derive the CP_I algorithm, with iterative cycle consisting of update steps CP1, CP2 of the original CP algorithm and CP*1', CP*2', CP*3' of the CP_{III} algorithm.

5.4 Experimental Work

In order to examine the capabilities of the algorithmic framework described above, we have conducted experiments on both artificial and real-world (textual) data.

The method of IB and ID with side information (IB-SI and ID-SI, described in Subsection 5.2.3) suggests a seemingly sensible alternative to our approach to CP clustering. As we aim at obtaining clusters that are not correlated with the given pre-partition, our setting is naturally mapped to the side information setting by considering the pre-partition W as the additional set of irrelevant features Y^-. Adapting this convention, our experimental results include comparison with IB-SI and ID-SI results and thus also with the plain IB and ID algorithms, which are equivalent to their corresponding SI version when the parameter γ is set to 0.

5.4.1 Experiments with Synthetic Data

In general, the CP method is designed to tackle cases where each one of the pre-given subsets is relatively homogenous and might be characterized by salient subset-specific structure. The target of the CP method is to neutralize such within-subset homogeneities and regularities and to reveal structure that is persistent across the pre-partition part, even if it is not as salient on average. The following setting aims at assessing the level by which the CP method reveals hidden cross-partition structure in the presence of more salient clustering configuration, with clusters that do not cut across the given pre-partition but are rather restricted to elements of one of the pre-given subsets.

5.4.1.1 Setting

Our synthetic setting consisted of 75 virtual elements, pre-partitioned into three 25-element subsets, corresponding to three admissible values of the variable W (in our formalism, for each element x in the w-th subset, $w = 1$, 2, or 3, $p(w|x) = 1$). On top of this pre-partition, we tailored together two independent (exhaustive) clustering configurations. One of them – the target configuration – will capture cross-subset correspondences, while the other – a masking configuration – will represent within dataset structure. We would like to see if and how the CP method reveals the target configuration, even in cases where the masking configuration is considerably more salient.

1. The target *cross-W* clusters: five clusters, each with representatives from all three pre-given subsets. In different experiments, we used three distinct cross-partition configurations, differing in the level of global balance (equal vs. diverging cluster sizes) and cross-partition balance (equal vs. diverging sizes of cluster-subset intersections). Figure 5.8 provide the details of the three different cross partition configurations that were used.

2. The Masking *within-w* clusters: six clusters, each consisting of either 13 or 12 of the 25 elements of a particular subset with no representatives from the other subsets.

The same set of features was used to direct formation of clusters. However, each cluster, of both target and masking configurations, was characterized by a designated subset of the features. Associating an element with the cross-W cluster and with the within-w cluster to which it is assigned by construction was carried out by specifying a count of co-occurrences with each one of the features designated as characteristic to both clusters. The masking within-w clusters were systematically designed to be more salient than the target cross-W clusters. The within-w clusters had more designated features than the cross-W clusters, per cluster (60 vs. 48) and in total ($6 \times 60 = 360$ vs. $5 \times 48 = 240$). In addition, the simulated co-occurrence counts associating elements with their within-w cluster (a base level of 900) were higher than the co-occurrence counts associating elements with cross-W cluster (700 in a *salient CP configuration* setting, 400 in a *non-salient CP configuration* setting). Noise (a random positive integer < 200) was added to all counts associating elements with the designated features of their within-w and cross-W clusters, as well as to approximately one quarter of the zero counts associating elements with features designated for other clusters.

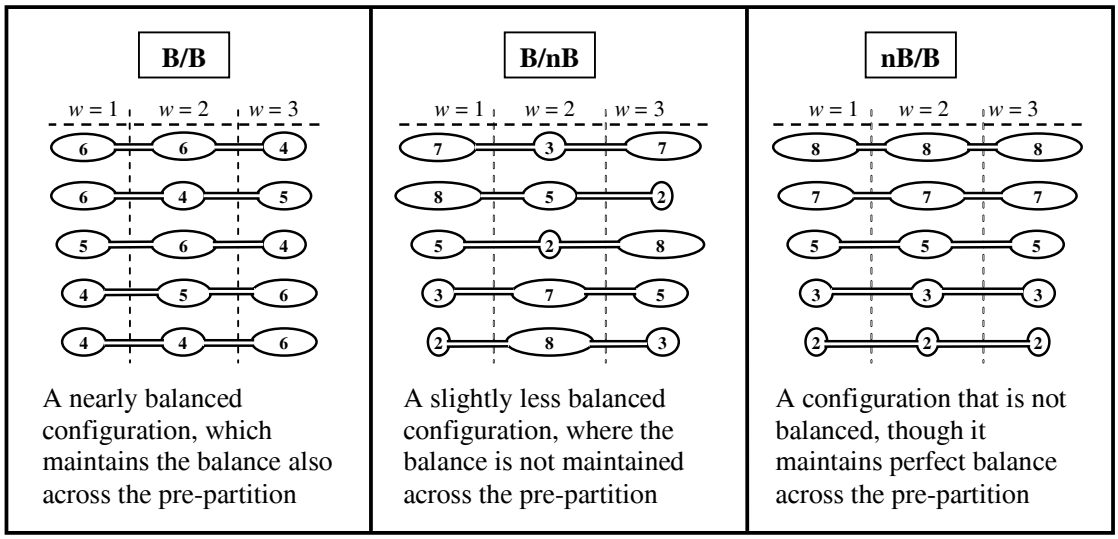

Figure 5.8: The three different cross-partition clustering configurations used in the synthetic data experiments. The numbers of elements in the intersections of each one of the five CP clusters and each one of the three pre-given subsets are indicated.

5.4.1.2 Results

Performance level in the synthetic data experiments was measured relatively to the target cross-*W* configuration – one of B/B, B/nB and nB/B (see Figure 5.8) – that was used in constructing the particular data being tested. Each one of the three cross-*W* configurations underlay two different types of datasets, distinguished by the saliency levels of the target relatively to the masking configuration (400 or 700 target co-occurrence counts versus 900 masking co-occurrence counts; see previous subsection). This gives a total of six experimental settings. The variance between the different test cases within each experimental setting was the result of two random factors: the random noise added and the overlap, i.e., the number of shared elements, between within-*w* clusters and cross-*W* clusters (partition of elements to clusters was random hence cluster overlap was random too).

In each one of the six experiment settings, we tested six different methods – the four CP method variants (5.3.3, 5.3.4) and the two SI variants (5.2.3) – each over a range of values of the parameters γ in the SI algorithms, and η in the CP algorithms. The values of γ and η were kept fixed throughout each run, while the β parameter was gradually incremented in order to produce the target five clusters (see Subsection 5.3.3.2). For values outside the tested parameter ranges, the majority of runs did not end with five clusters. Reasons for not obtaining the target number of clusters were that the run did not converge after a large number of iterative cycles or, in the CP case, it could also converge to too few clusters (Subsection 5.3.3.2). Each one of the reported results was averaged over 200 runs, differing by the noise and by within-*w* and cross-*W* cluster overlap.

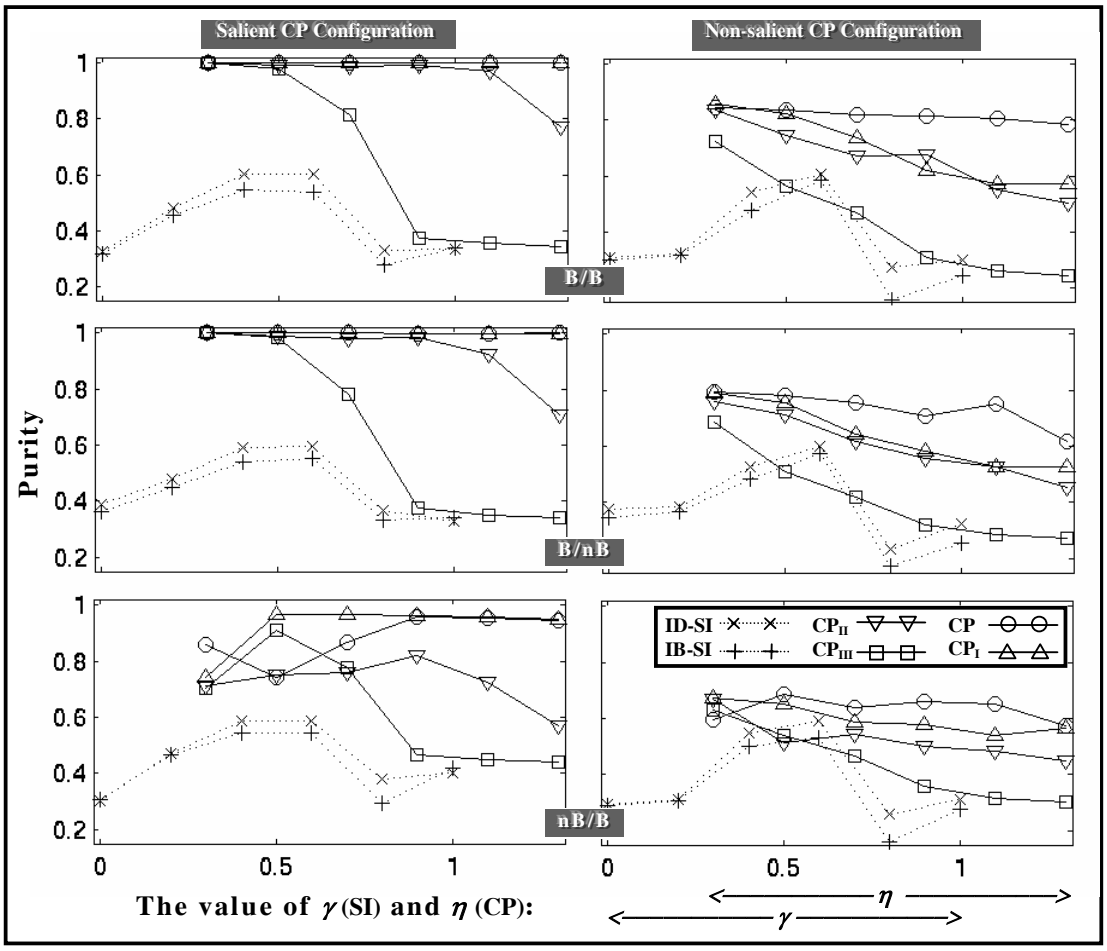

Figure 5.9: The results of the experiments with synthetic data by the six algorithms tested – CP, CP_I, CP_{II}, CP_{III}, ID-SI and IB-SI – in the six experimental settings. See the previous subsection for the description of the B/B, B/nB and nB/B cross-partition configurations and the difference between the salient and non-salient CP configuration settings.

As the target number of clusters was given by construction, we used the straightforward *purity* measure (the overall proportion of elements correctly assigned; Chapter 3, Subsection 3.3.1). Our measurements refer to the "hardened" version of the probabilistic output of the tested methods, i.e. the deterministic clustering configuration where each element x is considered a member in the cluster c with highest $p(c|x)$.

Figure 5.9 displays purity results produced by the six algorithms tested in the six experimental settings. The graphs displayed indicate that the various versions of the CP approach perform better than the SI approach in the majority of cases examined. The difference is most notable in the easier tasks, i.e., in the more balanced configurations and especially in the "salient" setting, where some of the CP variations consistently achieve almost perfect reconstruction of the target configuration, over a

large part of the tested η range. In addition, the SI algorithms tend to have a relatively narrow best-performance picks around certain γ value, while the CP performance is in general more stable across a large range of η values.

The "priorred" CP variations, especially CP_I and the CP_{III} which include the prior in their first step, tend to produce along with clusters that capture the target patterns small clusters that are often not part of the target configuration but rather seem to be the result of the added noise and interactions with the masking configuration. The plain CP algorithm was overall the best among the CP variations, while CP_{III} was the worst. The differences between the four versions are less noticeable in the lower η value range. Finally, there is a persistent advantage, though very small, to the ID-SI over the IB-SI. We will discuss possible reasons for the differences between the methods in the concluding section of this chapter.

5.4.1.3 Oscillatory Endless Loops

The previous subsection described the behavior of the CP algorithm (several variants) in cases where the data was drawn based on a prominent underlying cross partition structure and the algorithm converged in most cases. In the next section we will see that this nice behavior is indeed the case with the real world datasets we worked on. The current subsection shed some light on those cases where the underlying cross partition structure is not as prominent and consequently the algorithm is sometimes trapped in an endless loop.

We investigated this behavior experimentally through setting similar but simpler than the one described in Subsection 5.4.1.1 above. This simpler setting included eight virtual elements, pre-partitioned into two four-element subsets, with competing cross-*W* and within-*w* configurations. The two cross-*W* clusters included four elements each, two from each subset; the four within-*w* clusters, two within each subset, consisted of two elements each. The competing configurations were set to be in disagreement: the two elements of any within-*w* cluster were assigned to different cross-*W* clusters. Each cluster of both types was characterized by a single feature. As in the first setting, the within-*w* clusters were designed to be more salient with virtual element-feature co-occurrence count fixed on 100 (for the two elements of each cluster). The virtual cross-*W* co-occurrence counts varied in the different experiments between 0 and 100. Noise was added to all counts associating elements with the designated features of their within-*w* and cross-*W* clusters, as well as to approximately one quarter of the zero counts associating elements with features designated for other clusters.

On the eight-element dataset described above, we ran the (non-priorred) CP algorithm and applied the procedure of modifying the β value gradually in order to find the exact value inducing a split into two CP clusters. Table 5.1 shows the change in the proportion of times where the algorithm encountered

an endless oscillatory loop as a function of the saliency of the cross-*W* clusters. The more salient the cross-*W* clusters are, less cases of an endless loop are observed. These results are based on 200 runs for each tested value of cross-*W* count. In order to decide on an endless loop we counted 500,000 iterations (convergence whenever obtained typically occurred in tens or hundreds of iterations and never more than a few thousands). The η value was fixed on 1.0 in these experiments.

Table 5.1: The proportion of endless loop cases decreases as the relative weight of the Cross-*W* element-feature count increases. The Cross-*W* element-feature counts in this table are weighed comparatively to a fixed "within-*w* feature count" of 100.

Cross-*W* element-feature avg. count	0–55	60	65	70	75–85	90–100
Proportion of endless loop cases	40-44%	25%	9%	5%	2%	1%

As Table 5.1 shows, in this simple setting when the relative weight of the features associated with CP clusters is low, the CP algorithm is trapped in a loop in 40% or more of the cases. Table 5.2 brings one such example where the weight of the cross-*W* relative weight is fixed on zero (so its corresponding features are not present at all). In this example, the CP algorithm oscillated for β values between 1.858 and 2.738. For lower β values no split occurred in the data. In this example, but not always, two clusters were produced for β values higher than 2.738. Note that this behavior and parameter values can change due to exact values of initial assignments, split initialization, etc.

Table 5.2: An example for a setting where the CP algorithm does not converge (for particular β and η values). Each line in the table contains the co-occurrence count information for another one of the eight data elements. The underlying structure, as reflected by the feature counts, includes four within-subset clusters of two elements each, plus "noise" associating elements with clusters to which they are not assigned by construction.

Data Elements:		Features associated with …			
		Within-A cluster 1	Within-A cluster 2	Within-B cluster 1	Within-B cluster 2
Subset A	**A1**	10	—	—	—
	A2	9	1	—	—
	A3	1	9	—	—
	A4	—	8	—	2
Subset B	**B1**	—	—	10	—
	B2	—	2	8	—
	B3	—	—	—	10
	B4	1	—	1	8

5.4.2 Application to Religion Data

For testing our method on real world data, we used the religion-related datasets and the evaluation method (Jaccard coefficient scores) that were used in the previous chapter and were described in detail in the experimental part of the previous chapter (Chapter 4, Subsection 4.4.2.1; see also Appendix A). We note that the CP method, in difference from the coupled clustering method, can be used for identifying correspondences across more than two religions at a time, as demonstrated in the following subsection.

5.4.2.1 Results

We survey below some of the CP clustering output, exemplifying it through results produced by the plain CP algorithm with $\eta = 0.48$, applied to all five religions together ($|W| = 5$). We have found that even the most coarse two-cluster partition generated by the above method is highly informative and illuminating. It reveals two major aspects that seem to be equally fundamental in the religion domain, which we termed "spiritual" and "establishment" aspect. The cluster that corresponds to the "spiritual" aspect of religion incorporates terms related with theology, underlying concepts and religion-related personal experience. Many of the terms assigned to this cluster with highest probability, such as *heaven*, *hell*, *soul*, *god* and *existence*, are in common use of several religions, but there are religion-specific words such as *atman*, *liberation* and *rebirth*, which are key concepts of Hinduism. The "establishment" cluster contains names of schools, sects, clergical positions and other terms related with religious establishment, geographical locations and so on. Keywords assigned to this cluster with high probability are mainly religion specific: *protestant*, *vatican*, *university*, *council* in Christianity; *conservative*, *reconstructionism*, *sephardim*, *ashkenazim* in Judaism and so on (there are few keywords that are common to several religions, for instance *east* and *west*).

The same two-theme partition consistently repeats also when the CP method is applied to pairs and triplets of religion. As far as our corpora represent faithfully the domain and our method extracts well the relevant information, these two factors can be considered the two universal constituents upon which the very notion of religion is laid.

CLUSTER 1 "Schools"
Buddhism: america asia japan west east korea india china tibet
Christianity: orthodox protestant catholic west orthodoxy organization rome council america
Hinduism: west christian religious civilization buddhism aryan social founder shaiva
Islam: africa asia west east sunni shiah christian country civilization philosophy
Judaism: reform conservative reconstructionism zionism orthodox america europe sephardim ashkenazim

CLUSTER 2 "Divinity"
Buddhism: god brahma
Christianity: holy-spirit jesus-christ god father savior jesus baptize salvation reign
Hinduism: god brahma
Islam: god allah peace messenger jesus worship believing tawhid command
Judaism: god hashem bless commandment abraham

CLUSTER 3 "Religious Experience"
Buddhism: phenomenon perception consciousness human concentration mindfulness physical livelihood liberation
Christianity: moral human humanity spiritual relationship experience expression incarnation divinity
Hinduism: consciousness atman human existence liberation jnana purity sense moksha
Islam: spiritual human physical moral consciousness humanity exist justice life
Judaism: spiritual human existence physical expression humanity experience moral connect

CLUSTER 4 "Writings"
Buddhism: pali-canon sanskrit sutra pitaka english translate chapter abhidhamma book
Christianity: chapter hebrew translate greek new-testament book text old-testament luke
Hinduism: rigveda gita sanskrit upanishad sutra smriti brahma-sutra scripture mahabharata
Islam: chapter surah bible write translate hadith book language scripture
Judaism: tanakh scripture mishnah book oral talmud bible write letter

CLUSTER 5 "Festivals and Rite"
Buddhism: full-moon celebration stupa ceremony sakya abbot ajahn robe retreat
Christianity: easter tabernacle christmas sunday sabbath jerusalem pentecost city season
Hinduism: puja ganesh festival ceremony durga rama pilgrimage rite temple
Islam: kaabah id ramadan friday id-al-fitr haj mecah mosque salah
Judaism: sukoth festival shavuot temple passover jerusalem rosh-hashanah temple-mount rosh-hodesh

CLUSTER 6 "Sin, Suffering and Material Existence"
Buddhism: lamentation water grief kill eat hell animal death heaven
Christianity: fire punishment eat water animal lost hell perish lamb
Hinduism: animal heaven earth death water kill demon birth sun
Islam: water animal hell punishment paradise food pain sin earth
Judaism: animal water eat kosher sin heaven death food forbid

CLUSTER 7 "Community and Family"
Buddhism: child friend son people family question learn hear teacher
Christianity: friend family mother boy question woman problem learn child
Hinduism: child question son mother family learn people teacher teach
Islam: sister husband wife child family marriage mother woman brother
Judaism: child marriage wife mother father women question family people

Figure 5.10: A sample from a seven-cluster output CP configuration of the religion data: the first members – up to nine – of highest $p(c|x)$ within each religion in each cluster. Cluster titles were assigned by the author. See appendix E for the full configuration.

Partitions into clusters of finer granularity still seem to capture fundamental, though more focused, ingredients of religion. The partition into seven clusters reveals the following topics (our titles): "schools", "divinity", "religious experience", "writings", "festivals and rite", "material existence, sin, and suffering" and "community and family". The relation between this seven-cluster configuration to the coarser two-cluster configuration can be explained in soft-hierarchy terms: the "schools" cluster and, to some lesser extent "festivals" and "family", are related with the "establishment" aspect reflected in the partition to two, while "divinity", "religious experience" and "suffering" are clearly associated with the "spiritual" aspect of religion. The remaining topic, "writings", is equally associated with both. The probabilistic framework enabled the CP method to cope with these composite relationships between the coarse partition and the finer one. Figure 5.10 details the first members – up to nine – of highest $p(c|x)$ within each religion in each of the seven clusters. The whole two- and seven-cluster configurations produced by the CP method, including $p(c|x)$ and $p(c)$ values, are given in Appendix E.

It is interesting to have a notion of those features y with high $p^*(c|y)$ (Subsection 5.3.2.3). Many of those features are in fact identical with some of the corresponding cluster's terms, especially ones that are common to several religions but, occasionally, also ones that are specific to one religion but are mentioned in discussions regarding other religions. We exemplify those typical features, for each one of the seven clusters, through four of the highest $p^*(c|y)$ features that did *not* have a dual role of clustered keywords (more comprehensive lists are brought in Appendix E.):

- "schools" cluster: *central*, *dominant*, *mainstream*, *affiliate*;
- "divinity" cluster: *omnipotent*, *almighty*, *mercy*, *infinite*;
- "religious experience" cluster: *intrinsic*, *mental*, *realm*, *mature*;
- "writings" cluster: *commentary*, *manuscript*, *dictionary*, *grammar*;
- "festivals and rite" cluster: *annual*, *funeral*, *rebuild*, *feast*;
- "material existence, sin, and suffering" cluster: *vegetable*, *insect*, *penalty*, *quench*;
- "community and family" cluster: *parent*, *nursing*, *spouse*, *elderly*.

The above terms were not initially pre-marked but rather the CP clustering approach, through its feature-cluster re-association mechanism, has pointed each such feature as particularly informative with regard to the cluster with which it is associated.

The topic-based perspective on the religion domain, as obtained from the demonstrative results above, can be related with works in the field of the comparative study of religion. One notable source for drawing such relation is Ninian Smart's work, for instance his book *dimensions of the sacred* (1996).

Smart specifies six different dimensions spanning those essential aspects, in the light of which world religions can be understood and compared. These dimensions are the *ritual* dimension, the *mythic or narrative* dimension, the *experiential and emotional* dimension, the *ethical and legal* dimension, the *social* dimension and the *material* dimension. In addition, Smart separately mentions *political effects* of religion. It is obvious that this analysis is not geared by keyword counts, but leans on what appears to be abstract and deep considerations and knowledge. However, these dimensions fit rather nicely to the partition to "spiritual" versus "establishment" aspects suggested by the two-cluster partition of the keyword data produced by the CP method. Specifically, the *mythic or narrative* and the *experiential and emotional* dimensions are related with the "spiritual" aspect, while the other dimensions, including political effects have to do with the "establishment" aspect. In addition, some relations to our seven-cluster based topics can be observed. Two dimensions that are unambiguously mapped onto our clusters are the *ritual* dimension, which is mapped to the "festivals and rite" cluster, and the *experiential and emotional* dimension, which is mapped to the "religious experience" cluster. More associations, though less obvious, exist such as the ones relating the *mythic or narrative* dimension with the "writings" cluster and the *social* dimension with the "community and family" cluster.

Another example of a theoretical view on religion that is related with our empirical outcome is Kedar Nath Tiwari's book *Comparative Religion* (1992). This book systematically reviews several religions including the five religions we refer to, each religion in a separate chapter. Subsection titles are identical in all chapters. Thus, the repeating titles give a notion of what the author considers as the main factors common to all religions. The subsection titles are specified as follows (we indicate the ones that are unambiguously mapped to one of our seven clusters): *god* (mapped to our "divinity" cluster), *world*, *man*, *evil and suffering* (mapped to "material existence, sin, and suffering" cluster), *life after death*, *human destiny*, *discipline* and *sects* (mapped to "schools" cluster).

To summarize viewing our results in light of related studies of comparative religion, our findings cannot be said to capture the details of any particular theory. From the two examples above, we see that we can not expect such theories to largely overlap with one another. Interesting partial mapping between the clusters generated by the CP method and ingredients of existing theoretical views nevertheless exist and worth mentioning. In the next subsection, we relate our results further with knowledge from religion studies, this time more systematically and in quantitative terms.

5.4.2.2 Quantification of the Overlap with the Expert Data

We quantitatively evaluated results of the cross-partition clustering method applied to the religion data. Results by three different versions of the CP algorithm, produced with different fixed η values, were examined. As baselines, we used the basic Information Bottleneck (IB) method applied to the

union of the subsets, Information Bottleneck with Side Information (IB-SI) and our coupled clustering method (Chapter 4).

As in the previous chapter, we compared our results to classes of terms manually constructed by experts of comparative study of religion (as described in detail in Chapter 4, Subsection 4.4.2.3; see also Appendix C). The same 17 test cases involving pairs of religions were examined: all ten datasets made of pairs of subsets corresponding to all possible ten religion pairs were compared to the classes contributed by expert I. Four out of the same ten religion pairs could be further compared to classes contributed by expert II and three of the ten could be compared to the classes by expert III. Here as well, keywords not used by the expert were eliminated from the evaluated output *after* the completion of the clustering process. Again, we quantify the agreement between the output resulting from applying the CP method to a pair of religions at a time and the classes provided by the experts in terms of Jaccard coefficient (see Chapter 3, Subsection 3.3.2).

Given that the CP method produces probabilistic "soft" clustering, we had the option of using the "soft" Jaccard score variant. However, the scores produced by the soft version were similar to the standard Jaccard scores obtained from a "hardened" configuration (i.e., the "soft" scores did not reflect the potential added value of identifying real multi-assignments or ambiguities). We therefore used the standard version applied to the hard configuration resulting from assigning each element x to the cluster c with highest $p(c|x)$, but the expert data was considered probabilistic in cases of multi-assignment (as explained in more detail in Chapter 3, Subsection 3.3.2.1) similarly to the way the IB method was evaluated (as a baseline) in the previous chapter.

One further aspect regarding Jaccard scores, which is independent of the hard versus soft issue discussed above, refers to the adaptation of the scores to the cross-partition clustering setting. In the previous chapter, we used a version specifically adapted to coupled-clustering. This version counts only cross-subset pairs, while discarding the within-subset pairs altogether in a manner resembling the actual calculations conducted by the coupled clustering method, which relies solely on between-subset similarity values. Since the cross-partition approach of this chapter is essentially centroid-based and, as such, can be viewed as oriented towards clusters as wholes rather than towards the cross-subset associations set by the output configuration (Chapter 3, Subsection 3.3.2.1), we found it appropriate to apply here the standard version, which counts both within-subset and cross-subset pairs.[6]

---

[6] Without specifying the detailed scores, we denote that the Jaccard coefficient version adapted to coupled clustering produces in general higher scores in evaluating the CP method results, but the difference is not as noticeable as it is for the coupled clustering case described in Chapter 4.

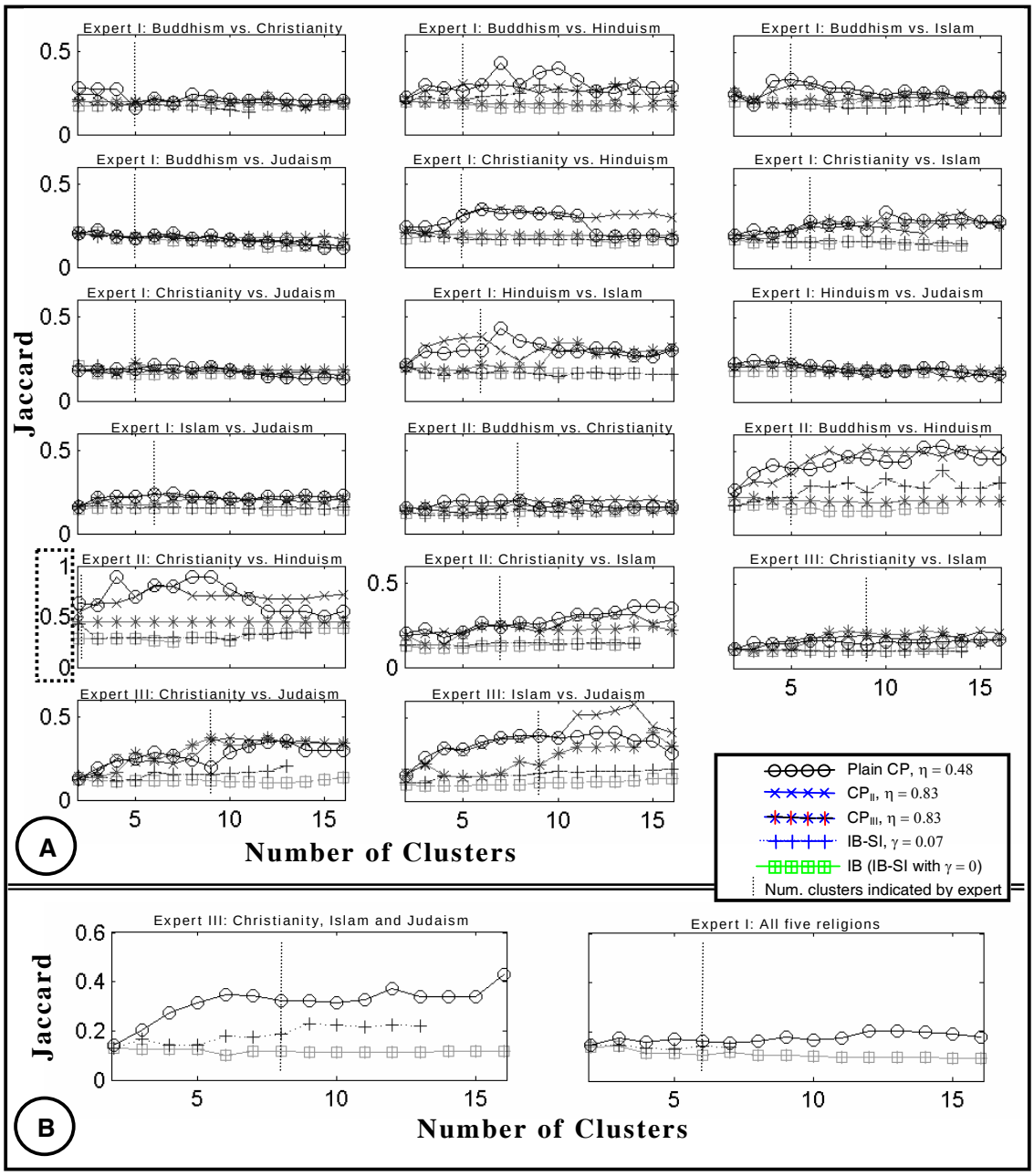

Figure 5.11: The religion keyword cross-partition clustering results evaluated relatively to the expert classes. Jaccard scores are shown for cluster numbers range from two to 16, for all 17 cases: ten by expert I, four by expert II and three by expert III. The algorithms in use: different versions of the CP clustering method with different η values (the plain version with $\eta = 0.48$, CP_{II} with $\eta = 0.83$ and CP_{III} with $\eta = 0.83$), the Information Bottleneck method and the Information Bottleneck with Side Information (with $\gamma = 0.07$). Note the different scale used for the "Expert II: Christianity vs. Hinduism" case, marked by a dotted frame.

The results of the experiments from all 17 test cases are displayed in Figure 5.11.A. The variations on the CP algorithm are demonstrated through three of the different versions: CP (with $\eta = 0.48$), CP_{II} (with $\eta = 0.83$) and CP_{III} (with $\eta = 0.83$). The other methods represented in Figure 5.11 are Information Bottleneck method and the Information Bottleneck with Side Information (with $\gamma = 0.07$). As already demonstrated, the CP method can be applied to data pre-divided into more than two subsets. Hence, apart from the 17 pairwise test cases, we also tailored a triple Christianity-Islam-Judaism keyword classification, based on the pairwise cross-religion comparisons provided by expert III. Similarly, we tailored also a configuration of term classes involving all five religions, based on the contribution of expert I. These configurations were used for evaluating the performance of the CP method on data pre-partition into three and five subsets. The results are displayed in Figure 5.11.B. As in the coupled clustering case (Chapter 4), the numbers of clusters indicated by each expert, denoted by dotted vertical lines in Figure 5.11, do not predict the actual number of clusters in the highest scored configuration. Thus, the target number of clusters was not assumed known, so that a whole range of output configurations of two to 16 clusters is scored. Figure 5.11 demonstrates that those CP versions using no prior or prior of one kind – CP (with $\eta = 0.48$) and CP_{II} (with $\eta = 0.83$) – perform better than the CP_{III} version (with relatively high $\eta = 0.83$) using both priors. All CP versions, however, perform better than the IB and IB-SI methods.

As discussed before (Section 5.3.3.2), parameter values outside a certain range prevent some of the examined algorithms from converging to sufficiently many clusters, or direct convergence to smaller number of clusters than the desired number. In our experiments, we used parameter values that allowed the formation of 16 clusters for all datasets. The existence of such parameter values is not obvious, as the datasets involving different pairs of religions differ from one another to much higher extent than the synthetic datasets (Subsection 5.4.1). However, it was not hard to find η values that worked well for all religion datasets.

Table 5.3 specifies, separately for each of several examined methods – CP, IB/IB-SI, and coupled clustering with the multiplicative cost function (Chapter 4) – an average over the 17 mean-values obtained by averaging over the range of examined numbers of clusters. As the table shows, the various η values that we tried yielded results that were similar on average, with the exception of slightly deteriorated performance by the CP_{III} version with the higher η value (which is in apparent agreement with the results of the synthetic experiments, Section 5.4.1). In contrast, the highest γ value that worked reasonably well for the IB-SI experiments, 0.07, was not sufficient for producing the desired number of clusters in the five religion case (note that this value is far lower than the optimal values in the synthetic experiments).

Table 5.3: Averages, over all 17 religion pair comparison cases, of means of 2–16 cluster Jaccard scores, recorded for the four CP method versions, each with four different η values.

	$\eta = 0.48$	$\eta = 0.56$	$\eta = 0.67$	$\eta = 0.83$
CP	0.2789	0.2778	0.2829	0.2816
CP_I	0. 2700	0.2716	0.2854	0.2954
CP_{II}	0.2701	0.2727	0.2820	0.2779
CP_{III}	0. 2664	0.2733	0.2656	0.2241
SI ($\gamma = 0.07$)	0.1812			
CC (multiplicative)	0.1806			
IB	0.1634			

Table 5.4: Average Jaccard scores over the 17 religion comparison evaluation cases. Each case is represented by the mean value (and, in parentheses, the best value) of all examined number of clusters. In parentheses: the average over the best score of each case. In the lower part of the table, difference values that were not found statistically significant (two-tailed *t*-test with 16 degrees of freedom, significance level 0.05) are marked with an asterisk.

Algorithm	Jaccard Score
Means ± standard deviations **of the 17 scores averaged over** (best of) **all examined numbers of clusters 2–16**	
CP ($\eta = 0.48$)	**0.2789**±0.1283 (0.3540±0.1692)
CP_{II} ($\eta = 0.83$)	**0.2779**±0.1319 (0.3452±0.1579)
CP_{III} ($\eta = 0.83$)	**0.2241**±0.0676 (0.2651±0.0809)
IB-SI ($\gamma = 0.07$)	**0.1812**±0.0525 (0.2214±0.0804)
CC (multiplicative)	**0.1806**±0.0514 (0.2475±0.0725)
IB	**0.1634**±0.0472 (0.1889±0.0601)
Means ± standard deviations **of the 17 coupled differences between scores averaged over** (best of) **2–16 clusters**	
CP – CP_{II}	**0.0009**±0.02298* (0.0088±0.0610*)
CP – CP_{III}	**0.0548**±0.0793 (0.0889±0.1282)
CP_{II} – CP_{III}	**0.0538**±0.0835 (0.0801±0.1055)
CP_{III} – IB-SI	**0.0429**±0.0583 (0.0437±0.0949*)
CP_{III} – CC	**0.0435**±0.0518 (0.0176±0.0858*)
IB-SI – CC	**0.0006**±0.0393* (–0.0261±0.0657*)
CC – IB	**0.0172**±0.0489* (0.0586±0.0861)
IB-SI – IB	**0.0177**±0.0342 (0.0325±0.0528)

Along with the same averages over 17 mean-values, Table 5.4 specifies (in parentheses) averages over 17 scores, each of which is the best of all examined numbers of clusters in a test case. In addition to the five methods exemplified in Figure 5.11, the table incorporates results of the coupled clustering (CC) method (Chapter 4) with the multiplicative cost function. The lower part of Table 5.4 confirms, based on the same data, the statistical significance of the differences between the CP versions and the IB, IB-SI and CC methods, which were recorded already in Figure 5.11 and Table 5.3.

5.4.2.3 Agreement between the Experts

Agreement between each two experts that contributed evaluation data for the same pair of religions is quantified through measuring the overlap between the classifications provided by the two experts, based on the commonly used terms. Together, there was a total of 16 cross-expert evaluation cases involving religion pairs: there was one religion pair (Christianity-Islam) to which all the three experts generated evaluation data and additional five religion pairs for each of which evaluation data was provided by two of the three experts. The average of the Jacccard scores quantifying agreement in the 16 cases is specified in the first line of Table 5.5 (same as in Chapter 4). In order to have common grounds for comparison with the common-term-based agreement between experts, terms not used by *either* expert were discarded also from the evaluated clusters (after the clusters were formed) leaving in the evaluated clusters only the terms used by both experts.

Table 5.5: The cross-expert agreement Jaccard score, along with the coupled differences of this score from means over of the 16 cross-expert religion pair evaluation cases (means over 2–16 cluster Jaccard scores, see text). The methods examined are CP, CP_{III}, SI and CC. The difference between the expert agreement and the plain CP method is marked with an asterisk to denote it is not statistically significant (two-tailed *t*-test with 16 degrees of freedom, significance level 0.05) in contrast to the other differences recorded.

	Jaccard Score
Means ± standard deviations **for cross-expert agreement scores**	
Cross-expert Agreement	**0.467**±0.2246
Means ± standard deviations **of the 17 coupled differences from expert agreements averaged over all 2–16 clusters**	
Expert agreement – CP ($\eta = 0.48$)	**0.405** (**0.0620**±0.1403*)
Expert agreement – CP_{III} ($\eta = 0.83$)	**0.293** (**0.1741**±0.1552)
Expert agreement – SI ($\gamma = 0.07$)	**0.201** (**0.2657**±0.2087)
Expert agreement – CC (multiplicative)	**0.202** (**0.2651**±0.2630)

Table 5.5 compares the agreement between experts to the clustering produced by the various methods examined, evaluated based on terms common to the two experts between which agreement is measured. It is not surprising that the Jaccard scores obtained based on those relatively few "consensual" terms are considerably better than the results in the previous subsection.

Table 5.5 explicates evidence that, on average, the results produced by the plain CP and the CP_{II} methods score closely to the cross expert agreement, up to a level where the difference is not statistically significant. Particularly, the CP and CP_{II} scores are noticeably closer to the expert agreement score than then the score of any of the other methods, including IB-SI and the CC methods and the CP_{III} variation as well.

5.5 Discussion

In this chapter, we have introduced and demonstrated the cross partition clustering method. In order to address the cross-partition clustering task, this method follows the regularities of the feature distribution in the data, in much the same manner as done by familiar standard probabilistic clustering methods, such as IB and ID methods (reviewed extensively above in Section 5.2). In difference from the standard clustering techniques, the CP method considers an additional source of information, namely a pre-partition of the data. It turns that the target regularities in the feature distribution – those cutting across the subsets of the given pre-partition – are not straightforwardly distinguishable from the subset-specific information that the CP method seeks to neutralize. Providing the means for distinguishing the cross-subset part of feature information from the subset-specific part is a key innovative aspect of the cross-partition method.

The initial motivation to developing the CP method was studying the notion of analogy (see Chapter 2, Section 2.2), with which it copes from a novel perspective. Each subset of the given pre-partition of the data represents one of the several systems between which analogy is drawn. Our method stresses those attributes that are common to the analogized systems within a framework similar to that of standard feature-based data clustering, which also realizes additional constraints related to the target of identifying a correspondence between the systems.

The maximum entropy principle plays in the CP method a key role in interleaving the principles that direct the CP task within one iterative loop. The iterative loop of the CP algorithm is divided to two parts, each taking care of one of two principles: forming clusters based on the relevance feature distribution and ensuring that formation of clusters is carried out independently of the pre-partition to subsets. Accordingly, the maximum entropy principle is also applied twice. The fact that the iterative loop of the cross partition algorithm realizes separately, through different update steps, two different directions has to do with our inability to specify a cost function that is minimized by the cross

partition algorithm as can be done for many other related methods (such as the information distortion, information bottleneck and information bottleneck with side information methods; all reviewed above Subsections 5.2.1, 5.2.2 and 5.2.3).

As mentioned, the core idea of the CP method's algorithmic mechanism lies in the step implementing feature-cluster re-association (Subsection 5.3.2.3). The associations of the characterizing features with the formed clusters are biased so that these associations become *independent* of the given pre-partition of the data.

The IB-SI approach (Chechik & Tishby, 2003; Subsection 5.2.3), which we consider a natural alternative to our method, can be understood in similar terms. In the way we implemented the IB-SI method (5.4), with the set of pre-given subset labels (our W variable) taken as the set of "negative features" (the IB-SI's Y^- variable), the IB-SI method aims at overall de-correlation of the formed clusters C from the information regarding the given pre-partition. As Y^+ and Y^- are correlated to some extent (otherwise there is initially no problem), such global treatment to C implies that de-correlating the unwanted C-Y^- association seems to affect undesirably the wanted C-Y^+ association and vice verse. In distinction, the CP method de-correlates the *associations* between features and the formed clusters, i.e., the detailed C-Y joint distribution, from the pre-partition. That way, the relation between C and Y is selectively focused on those regularities that cut across W, which fits our target more accurately as verified by the empirical results.

The IB-SI method, like the ID and IB methods (Section 5.2) and in distinction from the CP method, incorporates all its underlying considerations, including the target de-correlation between the set of "negative features" and the formed clusters as discussed above, within a single cost term (Eq. 5.13). This seems to be advantageous from the point of view of clarifying what the method aims at. The behavior of the CP method, namely convergence onto a steady state involving several equations is more complex and less intuitive to describe. The CP method, however, consistently outperforms the IB-SI and the other tested methods in the cross-partition clustering experiments we have conducted. This empirical superiority might suggest, for instance, a potential utility in expressing each one of the considerations underlying a composite task through a different term and seeking a solution that mutually constrains all terms relatively to one another rather than optimizes an all-encompassing cost term. Of course, such direction is yet to be formulated and examined in general terms – the CP method only exemplifies this option.

One further aspect in the comparison between the IB-SI to the CP approach is that both iterative algorithms are not guaranteed to converge (particularly, the iterative IB-SI algorithm is not guaranteed to minimize the IB-SI cost term). Nevertheless, the CP iterative algorithm have shown an empirical

advantage throughout our experiments in being more tolerant to changes in its parameter values, while the IB-SI requires tuning its parameter within a more restricted range for optimal performance.

The CP method improves significantly also relatively to the coupled clustering method introduced in the previous chapter. The coupled clustering method is a heuristics that is bound to some oversimplifications, most notably the assumption of a given similarity measure and the restriction to utilize only between subset similarities. The CP method not only utilizes the data more directly and thoroughly, but it does so in a more principled manner based on considerations of maximizing relevant information and the maximum entropy principle. Comparing the empirical results of the two chapters, we notice the difference in the Jaccard scores resulting from the two methods. Further, also the demonstrative examples show, to the best of our judgment, that while the CC results give an idea regarding a seemingly random selection of themes that are part of the religion domain, the CP outcome provides much more of an exhaustive and balanced sub-topical picture of the whole domain. The CP method reveals meaningful constituents that, to our understanding, indeed can be considered as the main building blocks of religion along various resolution levels.

We conclude this chapter with a remark regarding the priorred versus the non-priorred versions of the CP algorithm (Subsection 5.3.4). Including priors allow the formation of clusters of more diverging sizes, while the lack of priors poses a bias towards the limit of uniform distribution of elements over the clusters. On one hand, it can be argued that in some well-motivated information theoretic sense the methods that apply prior reveal what is "really" in the data. On the other hand, allowing small clusters along with large ones gives rise also to small clusters that are the result of "noise" rather than "true" information. Our conclusion is that it might be worth to include priors in cases where there is a reason to believe that the process is going to capture the "true" underlying structure very accurately. If this is the case, the utility of using a prior is intensified as far the ("true") distribution of elements over the clusters is from uniform distribution. It seems, however, that in many real data clustering applications high purity level cannot be granted. We believe that the superiority of the non-priorred version of the CP algorithm in our religion comparison task demonstrates well the widespread case where it is better not to apply any type of prior. Also in the synthetic experiments the non-priorred CP algorithm lost its superiority in settings of *both* imbalanced configuration and relatively low level of noise (see bottom left hand side of Figure 5.9).