

# An Analysis, Algorithm and Applications of Clustering and Co-occurrence Analysis

Kristiaan Pelckmans, David R. Hardoon

**Abstract**—This paper studies co-occurrence analysis from a theoretical as well as applied perspective. It is found that the analysis of this task fits nicely the recent PAC-Bayesian theorems, illuminating further differences between clustering, density estimation, co-clustering and the present co-occurrence analysis. The key difference of this last with respect to unsupervised learning paradigms in general is that it has a natural notion of ‘prediction loss’. This analysis motivates the PairWise Cluster Analysis (PWCA), which is essentially an extension of kernel Canonical Correlation Analysis (KCCA), and empirical evidence is presented to support the findings.

**Index Terms**—Clustering, Co-occurrence analysis, kernel methods, Machine Learning

## 1 INTRODUCTION

Consider the setup where individual observations come in two different representations  $(x, y)$ . This paper focuses on the questions: ‘If we observe a new  $x$ , what can be said about the corresponding  $y$ , and vice versa?’ While this abstract problem has obvious relations to classical supervised learning, its inherent symmetry relates it to unsupervised learning as well. This paper studies the above problem, specifying the properties to be predicted in terms of pre-specified membership functions.

Figure (1) gives a schematic representation of different (related) learning paradigms. Here the task is to learn to discriminate pictures of elephants from pictures of rhinos. If a picture is boxed, it indicates that labeling becomes available for learning, discriminating the supervised, unsupervised and semi-supervised paradigm. When two disjoint tasks (discriminate elephants from rhinos, and egrets from oxpeckers) is to be performed transfer learning and multi-task learning (see e.g. (?) and references) come into the picture. Self-taught learning as in ((1)) studies the task of (unsupervised) learning to discriminate oxpeckers/egrets, if labels for the elephant/rhino task are available. The last row then indicates then co-occurrence analysis, where all observations come in pairs. When observations (elephant, egret) and (rhino, oxpecker) are presented, the idea is that inference of the pairwise ‘symbiotic’ relation might help to form a notion of ‘elephant’, ‘rhino’, ‘oxpecker’ or ‘egret’. Specifically, if an oxpecker is seen, the task is to predict that a rhino might be close at hand. On the other hand, if an egret is observed, it can be expected that an elephant

might show up. It is exactly (abstractly) this notion of ‘clustering’ and ‘prediction’ we are after in this letter.

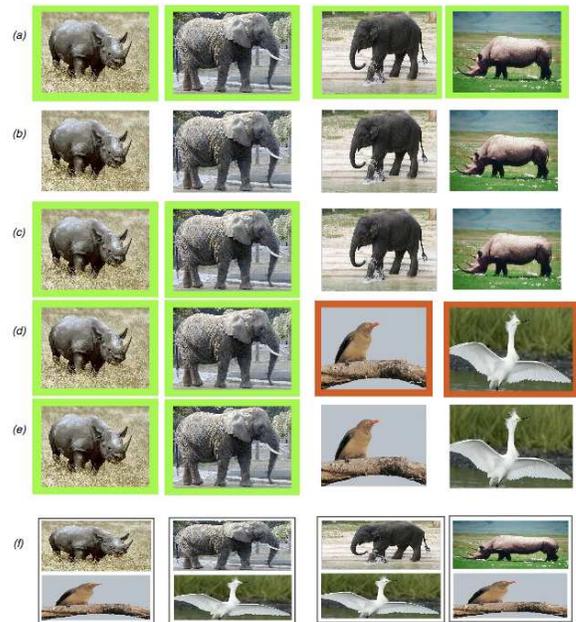


Fig. 1. Pictorial representation of different learning paradigms, extending the picture in (1). Suppose the aim is to discriminate elephants from rhinos. When a picture appears in a frame, a corresponding class-label is available. In cases: (a) supervised classification. (b) unsupervised learning. (c) semi-supervised learning. (d) transfer learning (the two different colors indicate two different cluster learning tasks). (e) self-taught learning, and (f) pairwise cluster analysis (PWCA). Note that in the last one we try not to find the class labels themselves, but to recover the symbiotic relation between elephant-egret, and rhino-oxpeckers. Specifically, the presence of oxpeckers might help us in predicting the presence of a rhino, and vice versa.

The next subsection explores the relationships of this learning paradigm with results in different results in the literature. The study of this precise learning setup was motivated by the works (2; 3).

• K. Pelckmans is with the Division of Systems and Control, Dept. of Information Technology, Uppsala University, Sweden e-mail: kristiaan.pelckmans@it.uu.se (see info <http://www.it.uu.se/katalog/kripe367>).

• D. R. Hardoon is with the Data Mining Dept., Institute for Infocomm Research (I<sup>2</sup>R), Singapore e-mail: davidrh@me.com (see info <http://www.davidroihardoon.com>).

## 1.1 Related Work

The analysis given in Section 2 phrases the learning problem in terms of the PAC-Bayesian theorem, much in the spirit of the recent (4). While the latter concerns density estimation for discrete variables, the presented ideas cover a spectrum of unsupervised learning (clustering). The analysis presented there concerns however (essentially) the same quantity  $E_Q[\mathcal{R}(h)]$  as in subsection 2.1, equation (8). The extension to pairwise clustering is fundamentally different - incorporating a notion of prediction ‘loss’ - while the relation of KL and the norm of an hypothesis establishes a relation with the learning algorithm presented next.

Section 3 (i) derives an effective learning algorithm, boiling down to a quadratic (or a generalized) eigenvalue problem. This learning machine is closely related to kernel Canonical Correlation Analysis (see e.g. (5) and references therein). Empirical (ii) evidence for this learning paradigm, and the proposed algorithm is then presented. Two simulation studies are conducted - at first a proof of concept is given based on artificial data. Secondly, the benchmark problem of learning structure from multi-lingual text-corpora as presented in (6) is used. Section 4 indicates a number of open questions.

## 2 A GENERIC ANALYSIS USING THE PAC-BAYES THEOREM

Consider a function  $h_r : \{x\} \rightarrow [0, 1]$  that verifies, for a given problem setting, how good a certain ‘rule’  $r$  performs on a sample  $x$ . The goal of a learning algorithm is to find the best rule  $r$  in a given set of plausible rules (the hypothesis set). Then, learning proceeds by collecting a dataset  $\{X_i\}_{i=1}^n$  of  $n$  observations assumed to be sampled independently from identical distributions (i.i.d)<sup>1</sup>. The empirical risk  $\mathcal{R}_n(h_r)$  and the actual risk  $\mathcal{R}(h_r)$  of an ‘hypothesis’  $h_r \in \mathcal{H}$  is defined as

$$\begin{cases} \mathcal{R}_n(h_r) = \frac{1}{n} \sum_{i=1}^n h_r(X_i) \\ \mathcal{R}(h_r) = \mathbb{E}[h_r(X)] \end{cases} \quad (1)$$

where the expectation  $\mathbb{E}[\cdot]$  concerns the fixed, unknown distribution underlying the  $n$  i.i.d observations. For supervised learning problems, (informally) an observation  $x$  consist typically of a couple  $(z, y)$  with a covariate  $z$  and an ‘output’  $y$ . Then  $h_r$  is often rephrased as  $h_r(x) = \ell(y - r(z))$ , where  $\ell : \mathbb{R} \rightarrow [0, 1]$  is the ‘prediction loss’ between the actual observation  $y$  and its prediction  $r(z)$ . In a Bayesian context, we assume that the hypothesis  $h_r \in \mathcal{H}$  are also ‘stochastic’ elements<sup>2</sup>, possessing some notion of likelihood, say  $Q : \mathcal{H} \rightarrow [0, 1]$  such that  $\int_{\mathcal{H}} Q(h_r) dh = 1$ . Consider at first the case where  $\mathcal{H}$  is

finite, we are interested in what happens on functions  $E_Q[h_r(x)]$ , which is defined as

$$E_Q[h_r(x)] = \sum_{h_r \in \mathcal{H}} h_r(x) Q(h_r). \quad (2)$$

If  $|\mathcal{H}|$  is infinite, then the sum can be replaced by an integral as usual, or  $E_Q[h_r(x)] = \int_{\mathcal{H}} h_r(x) Q(h_r) dh_r$ . In the analysis we will assume  $|\mathcal{H}| < \infty$  in order to avoid technical issues. Note that this is not quite a regular expectation  $\mathbb{E}[\cdot]$  as before. Now let the Kullback-Leibler distance be defined for each  $0 < p, q < 1$  as

$$\text{KL}(q, p) = q \log \frac{q}{p} + (1 - q) \log \frac{1 - q}{1 - p}, \quad (3)$$

where  $\log(\cdot)$  denote the natural logarithm. If  $|\mathcal{H}| < \infty$ , and we have functions  $Q : \mathcal{H} \rightarrow [0, 1]$  and  $P : \mathcal{H} \rightarrow [0, 1]$  that score these elements such that  $\sum_{h_r \in \mathcal{H}} Q(h_r) = \sum_{h_r \in \mathcal{H}} P(h_r) = 1$ , we extend the definition as

$$\text{KL}(Q, P) = \sum_{h_r \in \mathcal{H}} Q(h_r) \log \frac{Q(h_r)}{P(h_r)}. \quad (4)$$

If  $|\mathcal{H}| = \infty$ , we use the following definition

$$\text{KL}(Q|P) = \int_{\mathcal{H}} \log \frac{dQ(h)}{dP(h)} dQ(h), \quad (5)$$

where now  $Q : \mathcal{H} \rightarrow \mathbb{R}_+$  and  $P : \mathcal{H} \rightarrow \mathbb{R}_+$  are arbitrarily positive functions with  $\int_{\mathcal{H}} dQ(h) = 1$  and  $\int_{\mathcal{H}} dP(h) = 1$ , and the term  $\frac{dQ(h)}{dP(h)}$  is the Radon-Nikodym derivative of  $Q$  with respect to  $P$ , see e.g. (?) for details. We state the PAC-Bayes theorem as in (7):

**Theorem 1** For  $\delta > 0$  and for  $n \geq 8$ , we have that with probability exceeding  $1 - \delta$  we have that for all  $Q : \mathcal{H} \rightarrow [0, 1]$  the following inequality holds:

$$\begin{aligned} \text{KL} \left( E_Q[\mathcal{R}_n(h_r)], E_Q[\mathcal{R}(h_r)] \right) \\ \leq \frac{\text{KL}(Q, P) + \log \frac{1}{\delta} + \log(2\sqrt{n})}{n}. \end{aligned} \quad (6)$$

Specifically, this holds for a  $Q_n$  found by an algorithm based on the  $n$  i.i.d. observations. Note that this result is currently the most tight inequality, refining the ideas presented in (8). While till date most applications are found in the context of supervised learning, we will argue in the following that this theorem finds a ‘natural’ application towards unsupervised learning.

### 2.1 An Application of PAC-Bayes for Clustering

In what follows, assume that the  $n$  i.i.d. samples  $\{X_i\}_{i=1}^n$  take values in a bounded set in  $S \subset \mathbb{R}^d$  for a given  $d \in \mathbb{N}$ . In order to use the PAC-Bayes result to the generic application of clustering, we need to specify the loss function  $\ell : \mathbb{R}^d \rightarrow [0, 1]$  of interest. A ‘cluster’, represented as an indicator function  $h : \mathbb{R}^d \rightarrow \{0, 1\}$ , is understood here as a member of a user-specified set

1. We will use the convention to denote stochastic variables as capital letters, e.g.  $X, Y, \dots$ , while deterministic quantities are denoted in lower case, e.g.  $h, f, i, x, y, n, \dots$ .

2. In a PAC-Bayesian context, we will merely consider weighted sums of the elements in  $\mathcal{H}$ , rather than assuming a truly Bayesian setup.

of indicator functions  $\mathcal{H} = \{h : \mathbb{R}^d \rightarrow \{0, 1\}\}$ . Formally, one defines for a set  $c \subset \mathbb{R}^d$

$$h_c(x) = I(x \in c) = \begin{cases} 1 & x \in c \\ 0 & x \notin c. \end{cases} \quad (7)$$

Now, we look a bit closer what the term  $E_Q[\mathcal{R}(h_c)]$  represents in this context.

$$E_Q[\mathcal{R}(h_c)] = \sum_{h_c \in \mathcal{H}} \mathbb{P}(X \in c) Q(h_c) = \mathbb{E} \left[ \sum_{h_c \in \mathcal{H}} h_c(X) Q(h_c) \right], \quad (8)$$

where the second equality holds by linearity of the expectation, and where  $\mathbb{P}$  denotes the probability rules underlying the data. Consequently, the term  $E_Q[\mathcal{R}(h)]$  characterizes how well  $Q$  aligns with the distribution underlying the data, as filtered through the specific hypothesis space. Assume that the  $\mathcal{H}$  is designed such that all sets  $c$  corresponding to a  $h_c \in \mathcal{H}$  (i) cover the space  $S$  and (ii) are disjoint.

The function  $P : \mathcal{H} \rightarrow [0, 1]$  is the prior weighting function (think of it as a ‘prior distribution’ over  $\mathcal{H}$ ). In general, it is up to the user in a specific application to decide how to design  $(\mathcal{H}, P)$ : it is good practice to make it equally likely for each hypothesis  $h \in \mathcal{H}$  to explain the data by itself, - suggesting a uniform prior  $P$  over this set  $\mathcal{H}$ - while the result should be useful for the application in mind. Assume for example that all probability mass (underlying the samples) concentrates in the set corresponding with a single  $h_c$ , and  $Q(h_c) = I(i = j)$ , then this measure equals 1. On the other hand, if all samples are equally distributed over the  $|\mathcal{H}|$  sets  $h_c \in \mathcal{H}$ , the measure equals  $\frac{1}{|\mathcal{H}|}$ . This motivates the naming of  $E_Q[\mathcal{R}(h)]$  as the *explanatory power* of  $(\mathcal{H}, Q)$ . Specifically, if  $\mathcal{H} = \{I(x \in [-1, 1]^d)\}$ , the explanatory power of  $(\mathcal{H}, Q)$  is 1, but it however is not very useful, surprising nor *falsifiable*.

We argue that this PAC-Bayesian interpretation to clustering is often ‘natural’ because of three reasons.

- The present analysis does not need to recover the density function underlying the data, a feature which is highly desirable if working with high-dimensional data.
- The set of ‘underlying’ clusters is not recovered exactly, nor assumed to exist in reality. The actual stochastic rules underlying the observed data only say how well the hypothesis clustering ‘explains’ the data. When dealing with data arising from complex processes the assumption of a ‘true clustering’ is often an oversimplification.
- The characterization of performance of the found rule  $Q_n$  in terms of its deviation from the prior  $P$  is desirable if clustering is meant for looking for ‘consistent’ irregularities. Specifically, if the result  $Q_n$  is not what we (more or less) expected before seeing the data, substantial empirical evidence should be presented motivating this property.

Those reasons differentiate the approach substantially from approaches based on density estimation, or on mixtures of distributions. The following clustering algorithm is then motivated by application of the PAC-Bayesian theory. The theoretical objective becomes,

$$Q_* = \arg \max_Q E_Q[\mathcal{R}(h)] \text{ s.t. } \text{KL}(Q, P) \leq \omega, \quad (9)$$

and its empirical counterpart is given as

$$Q_n = \arg \max_Q E_Q[\mathcal{R}_n(h)] \text{ s.t. } \text{KL}(Q, P) \leq \omega, \quad (10)$$

where  $\omega > 0$ . Then the PAC-Bayes theorem establishes that  $Q_*$  is not too different from  $Q_n$  when  $\omega$  is not too large. This objective is also motivated from an information theoretical approach to clustering, as e.g. in (9). It turns out that the ‘regularization term’  $\text{KL}(Q, P)$  and  $\text{KL}(Q|P)$  can be bounded by more convenient quantities.

**Proposition 1 (Bound to K.-L. Divergence)** *If  $|\mathcal{H}| < \infty$  and  $\sum_{h \in \mathcal{H}} P(h) = 1$ , then*

$$\text{KL}(Q, P) = \sum_{h \in \mathcal{H}} Q(h) \log \frac{Q(h)}{P(h)} \leq \log \sum_{h \in \mathcal{H}} \frac{Q^2(h)}{P(h)}. \quad (11)$$

*Given  $\mathcal{H}$  with  $|\mathcal{H}| = \infty$  and  $P$  such that  $\int_{\mathcal{H}} dP(h) = 1$ , then*

$$\text{KL}(Q|P) = \int_{\mathcal{H}} \log \frac{Q(h)}{P(h)} dQ(h) \leq \log \int_{\mathcal{H}} \frac{Q^2(h)}{P(h)} dh. \quad (12)$$

Both expressions are a consequence of Jensen’s inequality.

Consider first the case where  $\mathcal{H}$  is finite. Let  $s_Q \in [0, 1]^{|\mathcal{H}|}$  be a vector representing the function  $Q$  where  $s_i^Q = Q(h_i)$  (enumerating the different elements  $h_i \in \mathcal{H}$ ), then

$$s_*^Q = \arg \min_{s^Q \geq 0, \sum_i s_i^Q = 1} \|s^Q\|_2 - E_Q[\mathcal{R}_n(h_c)]. \quad (13)$$

This learning problem tries to identify the clusters  $h$  which explain the data the most, while not be too specific towards the given sample. In practice, we implement this as follows

$$s_n^Q = \arg \min_{s^Q \geq 0, \sum_i s_i^Q = 1} \|s^Q\|_2 - E_Q[\mathcal{R}_n(h_c)]. \quad (14)$$

where the expectation  $\mathbb{E}[\cdot]$  is replaced by an empirical  $\frac{1}{n} \sum_{i=1}^n \cdot$ . Again, the PAC-Bayes theorem shows that the solutions  $s_*^Q$  and  $s_n^Q$  are not too different if the  $\text{KL}(Q|P)$  becomes not too large.

This formulation is close in spirit to the definition of ‘touchstone classes’, as given in (?), phrased in this context as

**Definition 2 (Touchstone classes)** *A touchstone class  $(\mathcal{H}, Q)$  for learning a probability density function on a measurable space  $\mathcal{X}$  is a class of real-valued functions  $\mathcal{H}$  on  $\mathcal{X}$ , with a positive weighting function  $Q : \mathcal{H} \rightarrow [0, 1]$  such that  $\int_{\mathcal{H}} Q(h) dh = 1$ . Let  $\ell : [0, 1] \times [0, 1] \rightarrow \mathbb{R}$  be a suitable*

loss function comparing two 'probability' functions  $Q, P$ . The error function  $\text{err}$  of such  $Q$  is defined as

$$\text{err}(Q) = E_Q[\ell(\mathcal{R}_n(h), \mathcal{R}(h))] \quad (15)$$

where  $\mathcal{R}_n(h)$  and  $\mathcal{R}(h)$  denotes the empirical density to the set indicated by  $h$ , and its actual counterpart.

The difference with the above description is threefold: (i) The function  $Q$  is to be fixed and to be set by the use. The notion of  $Q$  in our model is the target to be learned from data, and sys essentially which functions  $h$  are most useful for explaining the data (ii) The 'expectation' over  $Q$  is on the outside. It basically captures on which subset of functions  $h$  learning has to concentrate. (iii) The notion of error function is more general, as it holds for arbitrary loss functions, while we only look at the Kullback-Leibler difference. If  $Q$  represents the uniform distribution and  $\ell$  the  $\infty$ -norm for comparing densities, the notion of a touchstone having an error function tending to zero when  $n \rightarrow \infty$  would correspond with a Glivenko-Cantelli class, see e.g. (? ).

## 2.2 Density Estimation as Infinite Clustering

Consider the setup where  $|\mathcal{H}| = \infty$  consists of all Dirac functions, or

$$\mathcal{H}_\infty = \left\{ h_x = \delta_x \mid x \in \mathbb{R}^d \right\} \quad (16)$$

where the Dirac function  $\delta_x : \mathbb{R}^d \rightarrow \{0, 1\}$  is defined as  $\delta_x(x') = I(x = x')$ . Although such 'clusters' do not really amount to the general practice of clustering as used when performing vector quantization, it will turn out that this exercise is nevertheless useful and gives a possibly better explanation of techniques as kernel PCA and alike.

Now, we consider what would be required on the weighting functions  $Q : \mathcal{H} \rightarrow [0, 1]$  in order to deal with the case  $|\mathcal{H}| = \infty$ . Rather than dealing with such  $Q$  leading to outcomes typically be infinite small quantities - as  $\sum_{h \in \mathcal{H}} Q(h) = 1$  - we will represent this in terms of a function  $f_Q : \{\mathcal{H}\} \rightarrow \mathbb{R}$ . To do this, we need a formal structure of  $\mathcal{H}$  as a measure space permitting a collection of subsets, in turn following a sigma-algebra. Then for any such set  $S \subset \mathcal{H}$  we define

$$F_Q(S) = \int_{\mathcal{H}} 1_S dQ, \quad (17)$$

where the function  $1_S : \mathcal{H} \rightarrow \{0, 1\}$  is the indicator corresponding to  $S$ , and as such the right-hand side is a Lebesgue integral. Then, if  $F_Q$  is differentiable, we define the function  $f_Q$  as its Radon-Nikodym derivative

$$f_Q(h) = \frac{dF_Q}{dh}(h), \quad (18)$$

note that this definitions are paralleling the measure theoretic definition of probability distribution function and distribution density function. Although interesting in its own right and provoking different formal questions, we do not pursue this in this practical oriented

work (the margin is too small). As a result, we have that  $\int_{\mathcal{H}} f_Q(h) dh = 1$ . As a consequence, we modify the definition of the explanatory power to handle infinite sets  $\mathcal{H}$  as follows

$$\begin{aligned} E_Q[\mathcal{R}(h)] &= \int_{\mathcal{H}} \mathbb{E}[h(X)] dQ(h) \\ &= \mathbb{E} \left[ \int_{\mathcal{H}} h(X) dQ \right] = \mathbb{E} \left[ \int_{\mathcal{H}} h(X) f_Q(h) dh \right], \end{aligned} \quad (19)$$

by interchanging order of integration, and by definition of (18) and (17).

This definition of  $(\mathcal{H}, P)$  and  $f_Q$  permits us to relate the function  $E_Q[\mathcal{R}(h)]$  and the likelihood function. The expected likelihood function of a candidate probability density function  $f : \mathbb{R}^d \rightarrow \mathbb{R}_0$  is defined (in terms of the above notation) as

$$L(f) = \mathbb{E}[f(X)]. \quad (20)$$

In a similar vain, the relation of the above analysis to the log-likelihood criterion is explored. Here the latter - as forming the basis in statistical inference (see e.g. (? )) - is defined as

$$\mathcal{L}(f) = \mathbb{E}[\log f(X)]. \quad (21)$$

Now, assuming no ties occur in the data (or  $P(X \neq X') = 1$ ) the 'explanatory power' as defined in (8) becomes

$$E_Q[\mathcal{R}(h)] = \mathbb{E} \left[ \int_{\mathcal{H}} h(X) f_Q(h) dh \right] = \mathbb{E} [f'_Q(X)], \quad (22)$$

where the function  $f'_Q : \mathbb{R}^d \rightarrow \mathbb{R}_+$  is defined as

$$f'_Q(x) = Q(\delta_x) \quad (23)$$

Now, by interpreting  $f_Q$  as the likelihood function  $f$ , or the log-likelihood function  $\log f$ , one recovers  $L$  or  $\mathcal{L}$  as in (20) and (21) respectively. A first consequence of this observation is that the PAC-Bayes theorem can be used to bound the difference in empirical (log-) likelihood, and its theoretical counterpart in maximum likelihood based inference. Secondly, and perhaps more interestingly, it dictates that maximum likelihood based techniques can be used for inference of nonparametric models, as long as the solution is not too far off the 'prior'  $P$ . Once more, the 'prior' in this function is merely a (properly normalized) weighting of the hypothesis which is thought to be relevant before seeing empirical evidence. It gives as such a motivation for such techniques as the methods of sieves and penalized maximum likelihood, see e.g. (? ).

## 2.3 Towards PCA in an Hilbert Space

This subsection will connect the case of infinite hypothesis spaces to reproducing kernels, and will as such give a surprising motivation to the technique of kernel Principal Component Analysis. Earlier work on kernel PCA was mainly based on geometrical arguments (? ) or Maximum A-Posteriori inference (? ), while the motivation for such algorithms is often found in a relaxation

of the combinatorial min-cut problem (see e.g. (?) for a review and connections), or min-normalized cut problem (?).

TODO.

### 3 ANALYSIS OF CO-OCCURRENCE DATA

#### 3.1 An Application of PAC-Bayes for Co-Occurrence Analysis

Now we explain how the above insights lead to an analysis of Co-occurrence data. Let again  $\mathbb{Z}$  and  $\mathbb{Y}$  denote respectively the two domains of interest in which pairwise observations  $(x, y)$  are made. A first approach would be to rephrase the pairwise clustering problem as a standard clustering approach, where instead of the class of indicator functions  $\mathcal{H}_f \subset \{f : \mathbb{Z} \rightarrow [0, 1]\}$  in the first domain, one studies the cross-product of this class with the class of indicator functions in the other domain  $\mathcal{H}^{f,g} = \mathcal{H}_f \times \mathcal{H}_g$ , or

$$\mathcal{H}^{f,g} \subset \left\{ h = (f_h, g_h) \mid f_h : \mathbb{Z} \rightarrow [0, 1], g_h : \mathbb{Y} \rightarrow [0, 1] \right\}. \quad (24)$$

However, the reasoning in the introduction suggests another route. To see this, we formalize the intuition of a sample  $(x, y)$  being a target for prediction: (i) let  $z \in \mathbb{Z}$  represent the part of a sample  $x = (z, y)$  which might be used to predict (a property) of the (unobserved)  $y \in \mathbb{Y}$ ; and/or (ii) given  $y \in \mathbb{Y}$ , predict (a property) of the corresponding (unobserved)  $z \in \mathbb{Z}$ . Given a set  $\mathcal{H}^{f,g}$ : the knowledge of the ‘cluster’ to which  $X$  belongs, will be used to predict the cluster memberships of the corresponding  $y$ .

We will say that  $f_h$  explains  $z \in \mathbb{Z}$  if  $f_h(z) = 1$ , and similarly that  $g_h$  explains  $y \in \mathbb{Y}$  if  $g_h(y) = 1$ . In an ideal case, one would be able to associate exactly one different  $f_h \in \mathcal{H}_f$  to every  $g_h \in \mathcal{H}_g$  (i.e. describe a permutation). As such, one could predict the cluster  $g_h$  containing  $y$  corresponding to a given  $z$ . In the worst case, the choice of  $g$  that explains  $y$  is independent of  $z$  being explained by  $f$ . The pairwise clustering setup however differs from such a multi-class classification (structured output prediction) task as it is essentially symmetric: a given  $z$  is used to predict (cluster membership of) the corresponding  $y$ , and a given  $y$  is used to predict (cluster memberships of) the corresponding  $x$ . Now, a pairwise cluster  $h = (f, g) \in \mathcal{H}^{f,g}$  was useful for a sample  $(z, y) \in \mathbb{Z} \times \mathbb{Y}$  if  $f(z) = g(y)$ . Alternatively, a pairwise cluster  $c = (f, g)$  contradicts a sample if  $f(z) \neq g(y)$ . This motivates the following risk function

$$\begin{cases} \mathcal{R}_n(h) = \frac{1}{n} \sum_{i=1}^n I(f_h(Z_i) \neq g_h(Y_i)) \\ \mathcal{R}(h) = \mathbb{P}(f_h(Z) \neq g_h(Y)), \end{cases} \quad (25)$$

defined again in an ‘empirical’ and an ‘actual’ flavor. This definition measures how many (for how large a

probability mass) datapoints contradict a pairwise cluster  $h = (f_h, g_h)$ . Now the term  $E_Q[\mathcal{R}(h)]$  becomes

$$\begin{aligned} E_Q[\mathcal{R}(h)] &= \sum_{h \in \mathcal{H}^{f,g}} \mathbb{P}(f_h(Z) \neq g_h(Y)) Q(h) \\ &= \mathbb{E} \left[ \sum_{h_c \in \mathcal{H}} I(f_h(Z) \neq g_h(Y)) Q(h_c) \right], \end{aligned} \quad (26)$$

which basically captures how many mistakes are made when focussing on the subset of  $\mathcal{H}^{f,g}$  as directed by  $Q$ . This motivates the following practical approach: (i) given a dataset  $\{X_i = (Z_i, Y_i)\}_{i=1}^n$ , with the elements taking values in  $\mathbb{Z} \times \mathbb{Y}$ , and (ii) a set  $\mathcal{H}^{f,g}$  of pairwise clusters represented as  $h = (f, g)$ , and a ‘prior’ weighting function  $P : \mathcal{H}^{f,g} \rightarrow [0, 1]$ , then we aim to find a new weighting function  $Q_n : \mathcal{H}^{f,g} \rightarrow [0, 1]$  which is not too different from  $P$ , and which aligns well with the probability rules underlying the data as

$$Q_* = \arg \min_Q E_Q(\mathcal{R}(h_c)) \text{ s.t. } \text{KL}(Q, P) \leq \omega, \quad (27)$$

where  $\omega > 0$ . The PAC-Bayes theorem now guarantees that this problem is approximatively solved based on the data as

$$Q'_n = \arg \min_Q E_Q(\mathcal{R}_n(h_c)) \text{ s.t. } \text{KL}(Q, P) \leq \omega, \quad (28)$$

where  $\omega > 0$ . The resulting  $Q'_n$  will emphasize the pairwise clusters which are most often consistent with the data. Here we have a natural trade-off between specificity and accuracy, regulated by  $\omega_n$ . If  $\omega_n$  were small, the solution  $Q'_n$  cannot deviate from the uniform distributions over all pairwise clusters in  $\mathcal{H}^{f,g}$ , but then many different pairwise clusters will contradict on different samples, leading in turn to low explanatory power. On the other hand, allowing for arbitrary  $Q'_n$  will explain the individual samples fairly well (allowing a single pairwise cluster per sample), but the PAC-Bayesian result will not guarantee accuracy of the result anymore.

Let  $s_Q \in [0, 1]^{|\mathcal{H}|}$  be a vector representing the function  $Q$  where  $s_i^Q = Q(h_i)$  (enumerating the different elements  $h_i \in \mathcal{H}$ ), then

$$s_n^Q = \arg \min_{s^Q \geq 0_n, \sum_i s_i^Q = 1} \|s^Q\|_2 \text{ s.t. } E_Q[\mathcal{R}_n(h)] = 0. \quad (29)$$

implementing the so-called *realizable* case (as in the theory of Support Vector Machines). The optimal solution  $Q_n$  will try to find as many pairwise clusters as possible which are not contradicting the given data. We illustrate this notion in figure 2. In the ideal case, all observations are explained. In more realistic cases, merely a few pairwise clusters are found (i.e., the set  $\{h \in \mathcal{H} : Q(h) > 0\}$  contains only a few elements).

We extend this model to account for infinite  $\mathcal{H}$ , defined as  $h = (\delta_z, \delta_y)$  for each  $(z, y) \in \mathbb{Z} \times \mathbb{Y}$ , and where  $\delta_x$  denotes the Dirac delta. When extending the formulation in order to deal with infinite hypothesis spaces  $\mathcal{H}^{f,g}$ , we

replace vectors  $s_Q$  by functions  $Q : \mathcal{H} \rightarrow [0, 1]$ , which (for convenience) are assumed to be elements of a Hilbert spaces  $\mathbf{H}$ . This space is equipped with a corresponding inner-product (reproducing kernel)  $k : \mathbf{H} \times \mathbf{H} \rightarrow \mathbb{R}$ , implicitly defining  $\mathcal{H}$  and  $P$ . Note that  $0 \leq Q(h) \leq 1$  for all  $h \in \mathcal{H}$ , and  $\int_{\mathcal{H}} Q(h) dh = 1$ . This motivates the replacement of the term  $\text{KL}(Q, P)$  by  $\|Q\|_{\mathbf{H}}$ . As such (28) is equivalent (up to normalization) to

$$Q_n'' = \arg \min_Q \|Q\|_{\mathbf{H}} \text{ s.t. } E_Q[\mathcal{R}_n(h)] = 0. \quad (30)$$

where  $Q_n''(h) \geq 0$  for all  $h \in \mathcal{H}$ , and  $\int_{\mathcal{H}} Q_n''(h) dh = 1$ . Note that for the majority of pairwise clusters no data is sampled contradicting the cluster, and a smooth transition of  $Q$  inbetween the sample becomes possible. In the remainder we will assume the relevant Hilbert space  $\mathbf{H}$  can be decomposed additively uniquely as  $\mathbf{H}_{\mathbb{Z}} \otimes \mathbf{H}_{\mathbb{Y}}$ , and the norm of a function  $Q$  can then be written as  $\|Q\|_{\mathbf{H}}^2 = \|F\|_{\mathbf{H}_{\mathbb{Z}}}^2 + \|G\|_{\mathbf{H}_{\mathbb{Y}}}^2$ . Assume  $\mathcal{H}^{f,g}$  contains all pairwise clusters  $h = (\delta_z, \delta_y)$  for all  $(z, y) \in \mathbb{Z} \times \mathbb{Y}$  and  $\delta$  the Dirac delta. Under the assumption no ties occur in the data, problem (31) is

$$(F_n, G_n) = \arg \min_{F, G} \|F\|_{\mathbf{H}_{\mathbb{Z}}}^2 + \|G\|_{\mathbf{H}_{\mathbb{Y}}}^2 \quad (31)$$

s.t.  $F_i = G_i, \forall i = 1, \dots, n.$

enforcing that  $F(h) = G(h)$  for all  $h \in \mathcal{H}^{f,g}$ , enforcing again that  $F(h) \geq 0$  for all  $h \in \mathcal{H}^{f,g}$  and  $\int_{\mathcal{H}^{f,g}} F(h) dh = 1$ . Here  $F_i = F(\delta_{Z_i})$  and  $G_i(\delta_{Y_i}) = Q((\delta_{Z_i}, \delta_{Y_i}))$  for all  $i = 1, \dots, n$ . The next section shows how to solves this problem, relaxing the (in)equality constraints.

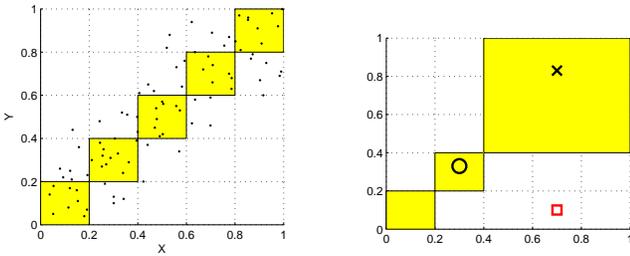


Fig. 2. Schematic representation of all pairwise clusters in a hypothesis space  $\mathcal{H}$  based on the 5 disjunct intervals  $d + [0, 0.2]$  in either domain (dotted lines). The dots  $(X, Y) \in \mathbb{R} \times \mathbb{R}$  represent samples from an underlying distribution. Suppose the different hypothesis can be factorized as  $h_c = (f, g)$ , where  $f : \mathbb{R} \rightarrow [0, 1]$  and  $g : \mathbb{R} \rightarrow [0, 1]$ , being the corresponding indicator functions in either domain. This means that there are 25 possible different pairwise clusters  $h_c$  (dotted squares), or  $|\mathcal{H}^{f,g}| = 25$ , (a) about 70% of the observations (dots) do not contradict the 5 pairwise clusters (yellow squares) simultaneously; (b) Only one sample (' $\square$ ') contradicts the shown pairwise cluster  $h_c$  (yellow squares), while the other two (' $\circ$ ' and ' $\times$ ') are consistent with  $h_c$ .

### 3.2 Methodology

This section studies how the learning problem (31) is solved (approximatively) by an efficient algorithm. Let  $X^a = (X_1^T, \dots, X_n^T)^T \in \mathbb{R}^{\ell \times m}$  and  $Y^b =$

$(Y_1^T, \dots, Y_n^T)^T \in \mathbb{R}^{\ell \times n}$  be matrices where  $\ell$  is the number of samples and  $m, n$  are the number of attributes/features for the first and second representation respectively. The functions  $Q$  are parametrised as  $F_{\mathbf{v}_c}(z) = \mathbf{v}_c^T z$  and  $G_{\mathbf{w}_c}(y) = \mathbf{w}_c^T y$ . The inequalities  $Q(h) \geq 0$  are enforced by representing this as  $Q(h) = F^2(f) = G^2(h)$  for all  $h \in \mathcal{H}^{f,g}$ . This is imposed by enforcing  $c_i = \sqrt{Q((\delta_{Z_i}, \delta_{Y_i}))} = F(\delta_{Y_i}) = G(\delta_{Y_i})$ , then  $\int_{\mathcal{H}} Q(h) dh = 1$  is enforced by imposing the constraint  $\mathbf{c}'\mathbf{c} = 1$  (similarly, maximizing  $\mathbf{c}'\mathbf{c}$ ). As such (28) becomes

$$\max_{\mathbf{c} \in \mathbb{R}^{\ell}, \mathbf{v}_c \in \mathbb{R}^m, \mathbf{w}_c \in \mathbb{R}^n} \mathbf{c}'\mathbf{c} - \gamma(\mathbf{w}_c' \mathbf{w}_c + \mathbf{v}_c' \mathbf{v}_c), \quad (32)$$

where  $A'$  is the transpose of matrix, or vector,  $A$  and such that  $\mathbf{c}_i = X_{a,i} \mathbf{w}_c = Y_{b,i} \mathbf{v}_c$ , for  $i = 1, \dots, \ell$ . Associating Lagrange multipliers  $\alpha_i, \beta_i$  to each of the  $\ell$  constraints gives the following Lagrangian

$$\mathcal{L} = \frac{1}{2} \mathbf{c}'\mathbf{c} - \frac{\gamma}{2} (\mathbf{w}_c' \mathbf{w}_c + \mathbf{v}_c' \mathbf{v}_c) - \alpha'(\mathbf{c} - X_a \mathbf{w}_c) - \beta'(\mathbf{c} - Y_b \mathbf{v}_c). \quad (33)$$

Taking derivatives of equation (33) with respect to  $\mathbf{w}_c, \mathbf{v}_c, \mathbf{c}$  and setting to zero give the following conditions for optimality as

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}_c} = \mathbf{0} &\rightarrow \mathbf{w}_c = \frac{1}{\gamma} X_a' \alpha \\ \frac{\partial \mathcal{L}}{\partial \mathbf{v}_c} = \mathbf{0} &\rightarrow \mathbf{v}_c = \frac{1}{\gamma} Y_b' \beta \\ \frac{\partial \mathcal{L}}{\partial \mathbf{c}} = \mathbf{0} &\rightarrow \mathbf{c} = (\alpha + \beta). \end{aligned}$$

Setting back into the optimisation in equation (32) gives the following dual problem

$$\max_{\alpha \in \mathbb{R}^{\ell}, \beta \in \mathbb{R}^{\ell}} \mathcal{J} = \frac{1}{2} (\alpha + \beta)' (\alpha + \beta) - \frac{1}{2\gamma} (\alpha' K_a \alpha + \beta' K_b \beta),$$

where  $K_a = X_a X_a'$  and  $K_b = Y_b Y_b'$  are the kernel matrices. Taking derivatives and setting to zero shows that  $\mathcal{J}$  achieves a (local) optimum when

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \alpha} = \mathbf{0} &\rightarrow \gamma(\alpha + \beta) = K_a \alpha \\ \frac{\partial \mathcal{J}}{\partial \beta} = \mathbf{0} &\rightarrow \gamma(\alpha + \beta) = K_b \beta, \end{aligned} \quad (34)$$

which can be rewritten as

$$\begin{bmatrix} K_a & 0_{\ell} \\ 0_{\ell} & K_b \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \gamma \begin{bmatrix} I_{\ell} & I_{\ell} \\ I_{\ell} & I_{\ell} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \quad (35)$$

where  $I_{\ell}$  is the identity matrix and  $0_{\ell}$  is a matrix of zeros, both of size  $\ell \times \ell$ . This equation may be solved as a generalized eigenvalue problem in the form of  $Ax = \lambda Bx$ . Alternatively, we observe that by setting

$$\beta = \left( \frac{1}{\gamma} K_a - I \right) \alpha,$$

we can express

$$\frac{1}{\gamma} K_a \alpha = \frac{1}{\gamma^2} K_b K_a \alpha - \frac{1}{\gamma} K_b \alpha,$$

which results in the following generalized eigenvalue problem for  $\alpha$

$$K_b K_a \alpha = \gamma (K_a + K_b) \alpha, \quad (36)$$

and by setting  $R$  to be the Cholesky decomposition of  $K_b K_a$  such that  $K_b K_a = R R'$  we obtain the following symmetric eigenvalue problem

$$I_\ell \alpha = \gamma R^{-1} (K_a + K_b) R^{-1'} \alpha.$$

Alternatively, under the assumption that the kernel matrices are invertible, we observe that it is also possible to express equation (36) by taking the inverse of the kernel matrices such that we obtain the following symmetric eigenvalue problem

$$\begin{aligned} I_\ell \alpha &= \gamma K_b^{-1} (K_a + K_b) K_a^{-1} \alpha \\ I_\ell \alpha &= \gamma (K_b^{-1} K_a K_a^{-1} + K_b^{-1} K_b K_a^{-1}) \alpha \\ I_\ell \alpha &= \gamma (K_b^{-1} + K_a^{-1}) \alpha. \end{aligned}$$

It may be necessary to regularize equation (35) with some small value  $\tau$  on the diagonal. This will result in our optimisation being rewritten as

$$\begin{bmatrix} K_a & 0_\ell \\ 0_\ell & K_b \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \gamma \begin{bmatrix} I_\ell(1+\tau) & I_\ell \\ I_\ell & I_\ell(1+\tau) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Furthermore, the above eigenvalue problem can be written as

$$\beta = \left( \frac{1}{\gamma} K_a - \tau I_\ell \right) \alpha$$

and

$$K_b K_a \alpha = \gamma^2 (I_\ell - \tau^2 I_\ell) \alpha + \gamma (\tau I_\ell K_a + \tau I_\ell K_b) \alpha,$$

which can be solved as a quadratic eigenvalue problem. It follows from the conditions for optimality that a new sample  $(\mathbf{x}_a, \mathbf{y}_b)$  can be assigned to a cluster by evaluating the functions

$$\begin{cases} F(\mathbf{x}_a) = |\mathbf{w}'_c \mathbf{x}_a| = \left| \frac{1}{\gamma} \sum_{i=1}^{\ell} K_a(\mathbf{x}_a, i, \mathbf{x}_a) \alpha_i \right|, \\ G(\mathbf{y}_b) = |\mathbf{v}'_c \mathbf{y}_b| = \left| \frac{1}{\gamma} \sum_{i=1}^{\ell} K_b(\mathbf{y}_b, i, \mathbf{y}_b) \beta_i \right|. \end{cases}$$

Then it is reasonable to assign the sample  $(\mathbf{x}_a, \mathbf{y}_b)$  to the cluster which has highest (absolute) factors  $F(\mathbf{x}_a)$  and  $G(\mathbf{y}_b)$  respectively.

### 3.3 Relationship to CCA

The proposed PWCA optimisation formulated in equation 32 has an interesting relationship to Canonical Correlation Analysis (CCA). We begin by quoting the CCA formulation from (5) given as

$$\max_{\mathbf{w}_c, \mathbf{v}_c} \rho = \frac{\mathbf{w}'_c X_a X'_a \mathbf{v}_c}{\sqrt{\mathbf{w}'_c X_a X'_a \mathbf{w}_c \mathbf{v}_c X'_b X'_b \mathbf{v}_c}}, \quad (37)$$

where  $\rho$  is the correlation value. Equation 37 is equivalent to

$$\min_{\mathbf{w}, \mathbf{e}} \|X'_a \mathbf{w}_c - X'_b \mathbf{v}_c\|^2. \quad (38)$$

This intuition of the above equivalence is formulated in the following theorem,

**Theorem 3** Vectors  $\mathbf{w}_c, \mathbf{v}_c$  are an optimal solution of equation (37) if and only if there exist  $\mu, \lambda$  such that  $\mu \mathbf{w}_c, \lambda \mathbf{v}_c$  are an optimal solution of equation (38).

Theorem 3 is well known in the statistics community and corresponds to the equivalence between one form of Alternating Conditional Expectation (ACE) and CCA (10; 11). Therefore, the CCA optimises an approximated equality between  $X'_a \mathbf{w}_c \approx X'_b \mathbf{v}_c$  (or  $K_a \alpha \approx K_b \beta$  in dual representation). We are able to observe that despite PWCA having a different optimisation formulation (equation (32)) at point of optimality (equation (34)) the resulting directions also maximally correlated the two sources as  $K_a \alpha = K_b \beta$ .

Secondly, another similarity emerges when we notice that  $\frac{1}{\gamma} (\alpha' K_a \alpha + \beta' K_b \beta)$  is in fact equivalent to maximising the CCA objective subject to  $\|X_a \mathbf{w}_c\|^2 = \|Y_b \mathbf{v}_c\|^2 = p$ , where  $p$  is some constant. For simplicity we can set  $p = 1$  and have it within a unit sphere. This, in turn, equates to forcing the projected direction to have equal length.

We are able to verify this by re-normalising the eigenvectors to a unit norm sphere (the eigenvalue solution are invariant to variations in the scale) such that  $\tilde{\alpha} = \frac{\alpha}{\sqrt{\alpha' K_a \alpha}}, \tilde{\beta} = \frac{\beta}{\sqrt{\beta' K_b \beta}}$ , and compute the resulting correlation value as

$$\rho = \tilde{\alpha}' K_a K_b \tilde{\beta} \quad (39)$$

for each cluster.

### 3.4 Primal Representation Approximation

We observe that it is not straightforward to solve the primal formation of our proposed optimization in equation (32). Therefore, for the sake of completeness, we give a primal approximation by replacing  $\|\mathbf{c}\|^2$  in equation (32) with a term indicating the co-clusters between the two views. This is a relaxation of the condition  $\mathbf{c} = X_a \mathbf{w}_c = Y_b \mathbf{v}_c$  such that we express  $\mathbf{c} = \frac{1}{2} (X_a \mathbf{w}_c + Y_b \mathbf{v}_c)$ , thus allowing us to rewrite the optimization as

$$\max_{\mathbf{w}_c \in \mathbb{R}^m, \mathbf{v}_c \in \mathbb{R}^n} \bar{\mathcal{J}} = \frac{1}{4} (X_a \mathbf{w}_c + Y_b \mathbf{v}_c)' (X_a \mathbf{w}_c + Y_b \mathbf{v}_c) - \frac{\gamma}{2} (\mathbf{w}'_c \mathbf{w}_c + \mathbf{v}'_c \mathbf{v}_c).$$

Taking derivatives and setting to zero gives

$$\begin{aligned} \frac{\partial \bar{\mathcal{J}}}{\partial \mathbf{w}_c} = \mathbf{0} &\rightarrow \gamma \mathbf{w}_c = X'_a X_a \mathbf{w}_c + X'_a Y_b \mathbf{v}_c \\ \frac{\partial \bar{\mathcal{J}}}{\partial \mathbf{v}_c} = \mathbf{0} &\rightarrow \gamma \mathbf{v}_c = Y'_b Y_b \mathbf{v}_c + Y'_b X_a \mathbf{w}_c, \end{aligned}$$

which can be rewritten as

$$\begin{bmatrix} X'_a X_a & X'_a Y_b \\ Y'_b X_a & Y'_b Y_b \end{bmatrix} \begin{bmatrix} \mathbf{w}_c \\ \mathbf{v}_c \end{bmatrix} = \gamma \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{w}_c \\ \mathbf{v}_c \end{bmatrix}$$

where  $\mathbf{I}$  is the identity matrix and  $\mathbf{0}$  is a matrix of zeros. This equation may be solved as a generalized eigenvalue problem in the form of  $A \mathbf{x} = \lambda B \mathbf{x}$ .

## 4 EXTENSION TO MULTI-VIEW CLUSTERING

In this section we generalize our methodology to multiple views. Expressing optimization in equation (32) for three sources gives

$$\max_{\mathbf{c} \in \mathbb{R}^\ell, \mathbf{w}_c \in \mathbb{R}^m, \mathbf{v}_c \in \mathbb{R}^n, \mathbf{z}_c \in \mathbb{R}^s} \frac{1}{2} \mathbf{c}' \mathbf{c} - \frac{\gamma}{2} (\mathbf{w}_c' \mathbf{w}_c + \mathbf{v}_c' \mathbf{v}_c + \mathbf{z}_c' \mathbf{z}_c), \quad (40)$$

such that  $c_i = X_{a,i} \mathbf{w}_c = X_{b,i} \mathbf{v}_c = X_{c,i} \mathbf{z}_c$ , for  $i = 1, \dots, \ell$ . Taking derivatives of equation (40) with respect to  $\mathbf{w}_c, \mathbf{v}_c, \mathbf{z}_c, \mathbf{c}$  and setting to zero will give the conditions for optimality. Substituting these conditions back into equation (40) gives the following dual problem

$$\max_{\alpha \in \mathbb{R}^\ell, \beta \in \mathbb{R}^\ell, \nu \in \mathbb{R}^\ell} \mathcal{J} = \frac{1}{2} (\alpha + \beta + \nu)' (\alpha + \beta + \nu) - \frac{1}{2\gamma} (\alpha' K_a \alpha + \beta' K_b \beta + \nu' K_c \nu),$$

where  $K_a = X_a X_a'$ ,  $K_b = X_b X_b'$  and  $K_c = X_c X_c'$  are the kernel matrices. Taking derivatives and setting to zero shows that  $\mathcal{J}$  achieves a (local) optimum when

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \alpha} = 0 &\rightarrow \gamma(\alpha + \beta + \nu) = K_a \alpha \\ \frac{\partial \mathcal{J}}{\partial \beta} = 0 &\rightarrow \gamma(\alpha + \beta + \nu) = K_b \beta \\ \frac{\partial \mathcal{J}}{\partial \nu} = 0 &\rightarrow \gamma(\alpha + \beta + \nu) = K_c \nu. \end{aligned}$$

which can be rewritten as

$$\begin{bmatrix} K_a & 0_\ell & 0_\ell \\ 0_\ell & K_b & 0_\ell \\ 0_\ell & 0_\ell & K_c \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \nu \end{bmatrix} = \gamma \begin{bmatrix} I_\ell & I_\ell & I_\ell \\ I_\ell & I_\ell & I_\ell \\ I_\ell & I_\ell & I_\ell \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \nu \end{bmatrix},$$

where again  $I_\ell$  is the identity matrix and  $0_\ell$  is a matrix of zeros, both of size  $\ell \times \ell$ .

Therefore, without loss of generality, we can extend PWCA to multiple  $\mathbf{i} = 1, \dots, s$  views<sup>3</sup>, where  $s \geq 2$ , such that

$$\begin{bmatrix} K_1 & \dots & 0_\ell \\ \vdots & \ddots & \vdots \\ 0_\ell & \dots & K_s \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_s \end{bmatrix} = \gamma \begin{bmatrix} I_\ell & \dots & I_\ell \\ \vdots & \ddots & \vdots \\ I_\ell & \dots & I_\ell \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_s \end{bmatrix}.$$

This equation may be solved as a generalized eigenvalue problem in the form of  $Ax = \lambda Bx$ .

## 5 EXPERIMENTS

In the following sections we illustrate our proposed method on a simulated problem as well as on a multi-lingual mate-retrieval experiment. In our experiments we use the CCA formulation as proposed by (12) for pair-wise and multiple views.

3. An extension of a similar nature had been previously proposed by (12) for CCA.

### 5.1 Synthetic data

We construct our simulated setting as follows

- Generate a 2 dimensional background cluster (background noise) constituting of 200 samples drawn independently from a uniform distribution over a  $10 \times 10$  cube centered at the origin.
- We then proceed to generated three paired 2 dimensional clusters, each constituting of 20 samples drawn independently from Gaussian distributions centered around  $\{(2, 2), (-4, -4), (8, 8)\}$  and  $\{(-7, -7), (2, 2), (7, 7)\}$  for the two views respectively.
- We compare our approach (PWCA) to kernel Principle Component Analysis (KPCA) and kernel CCA (KCCA).

To illustrate the difference between the three methods we plot the contour-plot of the scores  $F(\mathbf{x}_a), G(\mathbf{y}_b)$  for the top three eigenvectors corresponding to the largest three eigenvalues as well as their respective correlation values, which were computed using equation (39). We use a Gaussian kernels with the smoothing parameter set to  $\sigma = 2$  and regularization parameter set to  $\tau = 0.01$ . Both parameters were arbitrarily set.

We are able to visually observe in figure 3 that KPCA, computed independently for each view, is able to uncover the clusters with a decreasing value of correlation. As the analysis does not take into account the two views simultaneously it is not surprising that the correlation values rapidly decrease. In figure 4 we are able to observe the opposite effect with KCCA, i.e. taking into account the correlation between the two views indeed maximizes, as expected, the correlation value for each direction found, although the internal cluster structure is lost. Finally, we are able to observe in figure 5 that PCWA is able to both maximize the correlation between the two views and uncover (retain) their internal structure (clusters).

### 5.2 Mate-Based Retrieval

Mate-based retrieval is the information retrieval task where a user provides an item in one representation (*a query*), and the task is to come up with an item in the system's database which matches this item exactly (its *mate*) (13). This task is more particular than the general retrieval problem as each queries' mate can be assumed to exist. Specifically, we will look towards a text mining task where a query is given in the form of a text in a language  $x$ , and the task is to retrieve the translation of this document from a database of language  $y$ . This problem is phrased as a pairwise clustering task as follows. Suppose we have given a function  $h : \mathcal{Q}_x \times \mathcal{D}_y \rightarrow \mathbb{R}$ , where  $\mathcal{Q}_x = \{q_x\}$  denotes the set of queries in language  $x$ , and  $\mathcal{D}_y = \{d_y\}$  the set of documents in language  $y$ . The function  $h(q_x, \cdot)$  can be thought as giving the relative relevance for each document in  $\mathcal{D}_y$  for a query  $q_x$ . In particular, we aim to find a function  $h$  such that the Average Precision over  $\ell$  queries with its true

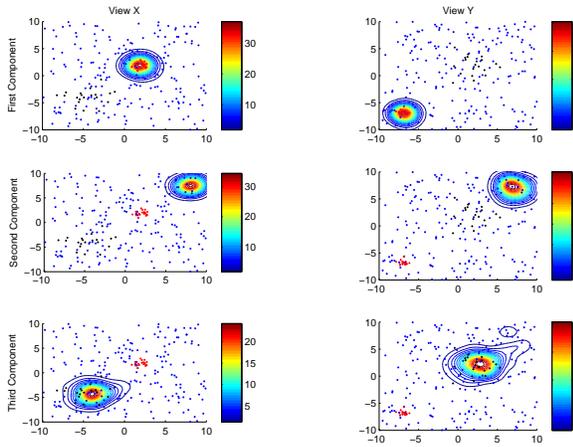


Fig. 3. KPCA on simulated data, the correlation values for the three top eigenvectors are top: 0.9123, middle: 0.6162, bottom: 0.3467. We are able to observe that despite the low correlation between the views (in second and third direction) the clusters are uncovered.

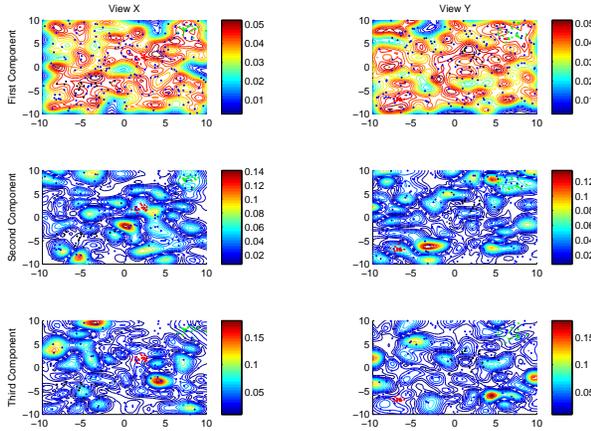


Fig. 4. KCCA on simulated data, the correlation values for the three top eigenvectors are top: 1.0000, middle: 0.9992, bottom: 0.9991. Despite high correlation between the two views, the internal cluster structure is lost.

(observed) matching document  $d_{y,q_x}$  is maximal. Given  $\ell$  pairs  $\{(q_{x,i}, d_{y,q_{x,i}})\}_{i=1}^{\ell}$

$$AP(h) = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{1}{|\{d_y \in \mathcal{D}_y : h(q_{x,i}, d_y) \geq h(q_{x,i}, d_{y,q_{x,i}})\}|}, \quad (41)$$

where  $|\cdot|$  denotes the cardinality of a set. The larger this measure is for  $h$ , the better predictions we will have when using the prediction rule

$$d_y^*(q_*) = \arg \max_{d_y \in \mathcal{D}_y} \{h(q_*, d_y)\}, \quad (42)$$

giving the document achieving the highest score on a query  $q_*$  using the function  $h$  (AP example illustrated in figure 6). Now since in pairwise clustering we factorize

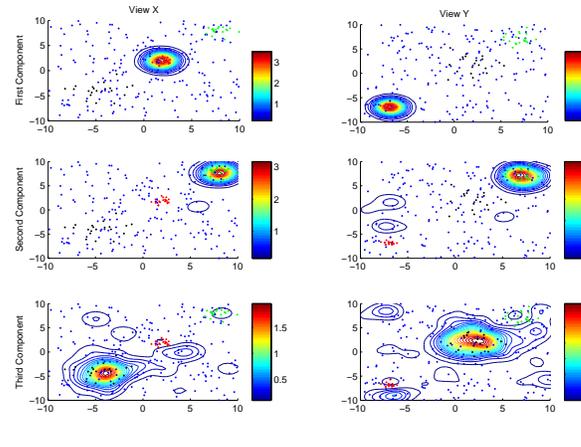


Fig. 5. PWCA on simulated data, the correlation values for the three top eigenvectors are top: 0.9322, middle: 0.7213, bottom: 0.6882. The methodology is able to maintain high correlation between the views while uncovering the clusters.

the function  $h$  in  $(f_h, g_h)$ , we write

$$AP(\{(f_h, g_h)\}) = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{1}{|\{(f_h, g_h) : g_h(d_{y,q_{x,i}}) \geq \tilde{g}_h(d_{y,q_{x,i}})\}|}, \quad (43)$$

where  $(\tilde{f}_{h,i}, \tilde{g}_{h,i}) = \arg \max_{(f_h, g_h)} f_h(q_{x,i})$ . This measure denotes for a sample  $(q_{x,i}, d_{y,q_{x,i}})$  how many clusters are better suited for capturing the document  $d_{y,q_{x,i}}$  than the one predicted by the corresponding query  $q_{x,i}$ , and the set of pairwise clusters  $\{(f_h, g_h)\}$ . Having a set of pairwise clusters  $(f_h, g_h)$  making this measure maximal, one can use this to predict the 'mate'  $d_y(q_x)$  to a given 'query'  $q_x$  as follows

$$d_y^*(q_*) = \arg \max_{d_y \in \mathcal{D}_y} \{\tilde{g}_h(d_y)\}, \quad (44)$$

where again  $(\tilde{f}_{h,i}, \tilde{g}_{h,i}) = \arg \max_{(f_h, g_h)} f_h(q_{x,i})$ . The following three sections illustrate the working of the algorithm on a practical case.

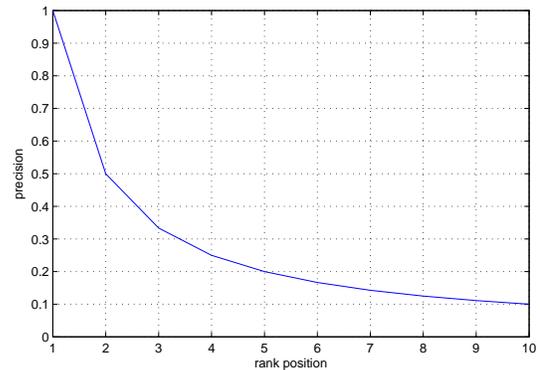


Fig. 6. Illustrative example of average precision values for  $\ell = 10$  across all possible rank locations achieved.

### 5.2.1 Bi-lingual Mate-based retrieval

In our mate retrieval experiments we use eight languages, as detailed in table 1, from the multi-lingual Europarl dataset (6)<sup>4</sup>, which has a total of 11968 aligned documents.

TABLE 1

We use the following eight languages in our mate retrieval experiments. We list the abbreviation of the language, the language name and its respective number of features.

Abbreviation	Language	Number of features (words)
da	Danish	78720
de	German	153499
en	English	60369
es	Spanish	171821
it	Italian	66548
nl	Dutch	105318
pt	Portuguese	66922
sv	Swedish	51116

Linear kernels are used throughout and we arbitrarily set the regularization parameter to  $\tau = 0.01$  for both methods.

In the first of our three experiments, for each pairing combination of languages, we randomly select 500 paired-documents for training and 5000 for testing. The analysis has been repeated 10 times. The results given in table 2 are the AP averaged across of all possible language-pair combinations for the languages indicated in the column. We are able to observe that PWCA is able to perform, on average, on a par with KCCA. The mean AP across all languages for KCCA is 0.4435 whereas for PWCA it is 0.4459. The performance of a random ranking is the average of all possible rank position, which as we have 5000 documents will be  $AP = 0.0019$ .

### 5.2.2 Tri-lingual Mate-based retrieval

In the second experiment we extend the previous analysis to a trilingual mate-retrieval task, i.e. we train on an aligned document corpus from three languages whereas during testing we compute the mean average precision of all the individual pair-wise mate-retrieval tasks (of the three languages). In other words we train on da-de-en and test on all possible pair-wise retrievals of da-de, da-en and de-en.

In this experiment we randomly select 500 tripartite-documents for training and 2000 for testing. Due to complexity we only repeat the analysis, for each 3 language combination, once. The results given in table 3 are the mean average precision for the language stated in the column and all its possible tripartite combinations (without repetition, i.e. for example; da-da-en is not be allowed). We are clearly able to see the improvement gained by PWCA over KCCA. We hypothesis that the PWCA performance improvement is due to the methods ability in retaining structure across multiple views whereas KCCA begins to over generalize.

### 5.2.3 Quad-lingual Mate-based retrieval

Finally, in our third experiment we further extend the previous tripartite framework to a quadlingual mate-retrieval task, i.e. we train on an aligned document corpus from four languages whereas during testing we compute the mean average precision of all the individual pair-wise mate retrieval tasks. The results given in table 4 are the mean average precision for the language stated in the column and all its possible quadripartite combinations (without repetition, i.e. for example; da-da-en-en is not be allowed). We are again able to observe significant improvement of PCWA over CCA.

We hypothesized that by creating a number of multiple co-clusters based on semantic information (assume  $K$  cluster each covering a different number of documents) we would be able to project a new query document into the cluster space (and respective co-paired clusters) and find the closest "real" document near to the projected query. The retrieved and query document should have the same (or very close) meaning. Since we actually do know that the exact same document exists in the different languages we should get the true document to be very close to our query in the projected semantic space.

Observing our aforementioned results we find that when increasing the number of co-training languages CCA losses its ability to maintain the quality of the found semantics, whereas PCWA maintains the underlying structure across multiple languages which allows for a richer semantic space. In fact, we are able to further observe that increasing the number of languages marginally improves on the average precision. Finally, our results demonstrate that on average our exact paired document is matched as the second ranked document (a document ranking at second location is equal to  $AP = 0.5$ ).

## 6 DISCUSSION

We had proposed a novel

## REFERENCES

- [1] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, p. 766.
- [2] K. Pelckmans, S. V. Vooren, B. Coessens, J. Suykens, and B. D. Moor, "Mutual spectral clustering: Microarray experiments versus text corpus," in *Proc. of the workshop on Probabilistic Modeling and Machine Learning in Structural and Systems Biology*. Helsinki, Finland: Helsinki University Printing House, 2006, pp. 55–58.
- [3] K. Sim, V. Gopalkrishnan, H. N. Chua, and S. K. Ng, "Macs: Multi-attribute co-clusters with high correlation information," in *Proceedings of The European Conference on Machine Learning and Principles and*

4. <http://people.csail.mit.edu/~koehn/publications/europarl.ps>

TABLE 2

We compare KCCA and PWCA on a bilingual mate-retrieval task (see table 1 for language abbreviation). The reported results are the AP for retrieving the exact paired document in another language, averaged across all possible language-pair combination for the language indicated in the column. The results are averaged over 10 repeats of the analysis.

CCA				
	da	de	en	es
da	-	0.3452±0.0163	0.4245±0.0275	0.3539±0.0230
de	0.3349±0.0614	-	0.3821±0.0145	0.3434±0.0352
en	0.3725±0.0231	0.4379±0.0088	-	0.3846±0.0183
es	0.3345±0.0259	0.3106±0.0269	0.4329±0.0146	-
it	0.3066±0.0545	0.2630±0.0353	0.4442±0.0131	0.3643±0.0296
nl	0.3395±0.0439	0.3054±0.0286	0.3861±0.0518	0.3347±0.0328
pt	0.3669±0.0177	0.3039±0.0224	0.4717±0.0182	0.4085±0.0255
sv	0.4500±0.0198	0.3377±0.0335	0.4465±0.0322	0.3565±0.0289
	0.4174±0.0462	0.3839±0.0549	<b>0.4979±0.0327</b>	0.4243±0.0253
	it	nl	pt	sv
da	0.3676±0.0249	0.3502±0.0372	0.4067±0.0141	0.4679±0.0334
de	0.3223±0.0174	0.3514±0.0273	0.3683±0.0180	0.3907±0.0355
en	0.4338±0.0191	0.3463±0.0273	0.4560±0.0208	0.4168±0.0468
es	0.4193±0.0211	0.3388±0.0218	0.4634±0.0191	0.3881±0.0313
it	-	0.3179±0.0213	0.4664±0.0139	0.3588±0.0234
nl	0.3599±0.0153	-	0.3791±0.0155	0.3965±0.0418
pt	0.4640±0.0117	0.3443±0.0101	-	0.4096±0.0129
sv	0.3766±0.0219	0.3654±0.0322	0.4238±0.0206	-
	<b>0.4572±0.0490</b>	0.4023±0.0145	<b>0.4939±0.0404</b>	0.4714±0.0337
PWCA				
	da	de	en	es
da	-	0.3762±0.0156	0.4118±0.0150	0.3666±0.0130
de	0.3947±0.0129	-	0.3780±0.0150	0.3765±0.0204
en	0.3474±0.0156	0.2986±0.0189	-	0.3704±0.0177
es	0.3534±0.0106	0.3361±0.0156	0.4268±0.0170	-
it	0.3267±0.0138	0.2865±0.0129	0.4002±0.0175	0.3629±0.0180
nl	0.3641±0.0108	0.3452±0.0171	0.3886±0.0173	0.3663±0.0266
pt	0.3441±0.0155	0.3007±0.0175	0.4137±0.0116	0.3963±0.0179
sv	0.4456±0.0178	0.3602±0.0145	0.4291±0.0125	0.3677±0.0168
	<b>0.4294±0.0401</b>	<b>0.4416±0.0343</b>	0.4747±0.0190	<b>0.4344±0.0114</b>
	it	nl	pt	sv
da	0.3644±0.0139	0.3736±0.0084	0.3943±0.0167	0.4742±0.0175
de	0.3414±0.0117	0.3837±0.0168	0.3688±0.0132	0.4194±0.0124
en	0.3841±0.0160	0.3336±0.0171	0.4021±0.0148	0.3988±0.0109
es	0.3995±0.0155	0.3591±0.0185	0.4429±0.0186	0.3905±0.0164
it	-	0.3137±0.0157	0.4238±0.0216	0.3572±0.0128
nl	0.3521±0.0187	-	0.3705±0.0165	0.3988±0.0139
pt	0.4142±0.0209	0.3302±0.0170	-	0.3908±0.0068
sv	0.3653±0.0086	0.3732±0.0129	0.4052±0.0077	-
	0.4368±0.0261	<b>0.4111±0.0266</b>	0.4679±0.0268	<b>0.4716±0.0360</b>

*Practice of Knowledge Discovery in Databases, Part II*, 2009, pp. 398–413.

- [4] Y. Seldin and N. Tishby, "PAC-Bayesian Generalization Bound for Density Estimation with Application to Co-clustering."
- [5] D. Hardoon, S.Szedmak, and J.Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16(12), pp. 2639–2664, 2004.
- [6] P. Koehn, "Europarl: A multilingual corpus for evaluation of machine translation," *Draft, unpublished*, unpublished.
- [7] A. Maurer, "A note on the PAC-Bayesian theorem," *Arxiv preprint cs/0411099*, 2004.
- [8] D. McAllester, "PAC-Bayesian model averaging," in *Proceedings of the twelfth annual conference on Computational learning theory*. ACM New York, NY, USA, 1999, pp. 164–170.
- [9] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *The Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.
- [10] L. Breiman and L. H. Friedman, "Estimating optimal transformations for multiple regression and correlation," *Journal of the American Statistical Association*, vol. 80, pp. 580–598, 1985.
- [11] T. J. Hastie and R. J. Tibshirani, *Generalized Additive Models*. Chapman & Hall/CRC, 1990.
- [12] F. Bach and M. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.
- [13] E. N. Efthimiadis and S. E. Robertson, *Perspectives in Information Management*. London: Butterworth, 1989, ch. Feedback and Interaction in Information Retrieval.

TABLE 3

We compare KCCA and PWCA on a trilingual mate-retrieval task (see table 1 for language abbreviation). The reported results are the mean average precision (and standard deviation) for retrieving the exact paired document in another language for all possible tripartite combinations of the language stated in the column (without repetition).

	da	de	en	es
CCA	0.3687±0.0412	0.3290±0.0325	0.3930±0.0521	0.3742±0.0512
PWCA	<b>0.5407±0.0245</b>	<b>0.5155±0.0252</b>	<b>0.5427±0.0251</b>	<b>0.5394±0.0251</b>
	it	nl	pt	sv
CCA	0.3792±0.0604	0.3501±0.0379	0.3917±0.0504	0.3909±0.0488
PWCA	<b>0.5310±0.0292</b>	<b>0.5246±0.0216</b>	<b>0.5406±0.0276</b>	<b>0.5504±0.0230</b>

TABLE 4

We compare KCCA and PWCA on a quadlingual mate-retrieval task (see table 1 for language abbreviation). The reported results are the mean average precision (and standard deviation) for retrieving the exact paired document in another language for all possible quadripartite combinations of the language stated in the column (without repetition).

	da	de	en	es
CCA	0.0831±0.0149	0.0797±0.0124	0.0846±0.0160	0.0833±0.0139
PWCA	<b>0.5403±0.0207</b>	<b>0.5295±0.0202</b>	<b>0.5446±0.0229</b>	<b>0.5415±0.0220</b>
	it	nl	pt	sv
CCA	0.0859±0.0160	0.0821±0.0137	0.0854±0.0153	0.0829±0.0152
PWCA	<b>0.5371±0.0234</b>	<b>0.5344±0.0210</b>	<b>0.5457±0.0229</b>	<b>0.5468±0.0198</b>