Semi-supervised discriminative clustering with graph regularization

Marek Śmieja[a,*], Oleksandr Myronov[b], Jacek Tabor[a]

[a]*Faculty of Mathematics and Computer Science*
Jagiellonian University
Łojasiewicza 6, 30-348 Kraków, Poland
[b]*Ardigen S.A.*
Bobrzyńskiego 14, 30-348 Kraków, Poland

Abstract

Pairwise constraints are a typical form of class information used in semi-supervised clustering. Although various methods were proposed to combine unlabeled data with pairwise constraints, most of them rely on adapting existing clustering frameworks, such as GMM or k-means, to semi-supervised setting. In consequence, pairwise relations have to be transferred into particular clustering model, which is often contradictory with expert knowledge.

In this paper we propose a novel semi-supervised method, **d-graph**, which does not assume any predefined structure of clusters. We follow a discriminative approach and use logistic function to directly model posterior probabilities $p(k|x)$ that point x belongs to k-th cluster. Making use of these posterior probabilities we maximize the expected probability that pairwise constraints are preserved. To include unlabeled data in our clustering objective function, we introduce additional pairwise constraints so that nearby points are more likely to appear in the same cluster. The proposed model can be easily optimized with the use of gradient techniques and kernelized, which allows to discover arbitrary shapes and structures in data. The experimental results performed on various types of data demonstrate that **d-graph** obtains better clustering results than comparative state-of-the-art methods.

*Corresponding author

Email addresses: marek.smieja@ii.uj.edu.pl (Marek Śmieja), alexander.myronov@gmail.com (Oleksandr Myronov), jacek.tabor@uj.edu.pl (Jacek Tabor)

Keywords: semi-supervised clustering, discriminative model, pairwise constraints, graph clustering

1. Introduction

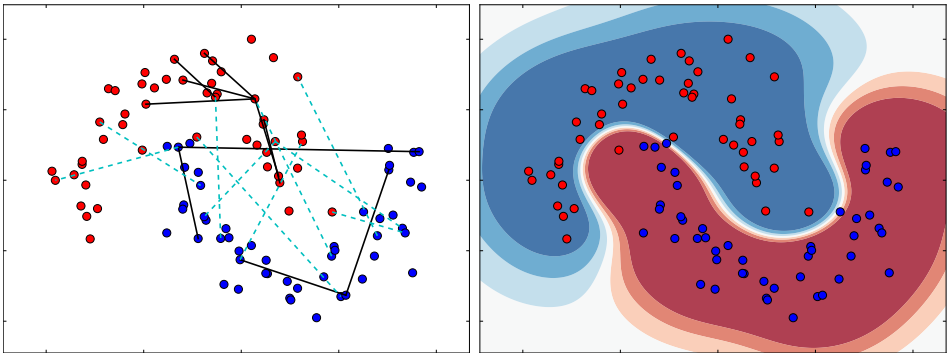
Cluster analysis, one of the core branches of machine learning, aims at splitting data into homogeneous groups. Since clustering does not use any external information about class labels, most algorithms use predefined models of clusters (or similarity measures) to define optimal shape or structure of groups [9, 12, 28, 29]. Unfortunately, there are no clear rules on how to select a criterion for a particular problem. In semi-supervised clustering, the expert indicates partial information about true class labels, which allows to clarify the underlying clustering problem [4]. Semi-supervised clustering found its applications in text processing [40], image annotation [41], social networks mining [20], etc.

Pairwise constraints (relations) are a typical form of additional class information used in semi-supervised clustering. They indicate whether two points belong to the same (must-link) or different groups (cannot-link). Most of semi-supervised approaches focus on adapting existing clustering frameworks to semi-supervised setting. In consequence, pairwise relations have to be transferred into a particular clustering model. This is not natural, because unsupervised models do not optimize classification error, but focus on finding regular clustering structures, which may be in contradiction with expert knowledge.

In this paper we follow an idea that pairwise constraints are the main source of information and they should guide the algorithm in the construction of the optimal clustering structure. To meet the user expectations given by pairwise constraints, we focus on the following question:

How to construct a clustering model, which maximizes the number of correct pairwise relations?

To deal with the above problem we follow a discriminative approach, commonly applied in classification, but rarely used in clustering. Discriminative model is more natural and effective for semi-supervised tasks than typical generative approaches, such as k-means or GMM (Gaussian mixture model), because it directly focuses on underlying classification problem. We assume



(a) (b)

Figure 1: Sample results of **d-graph** with RBF kernel on two moons data set (b). Pairwise constraints (a) covered 20 randomly selected pairs of points: must-link is marked with solid black line while dashed cyan line is used for cannot-link relations.

that the probability that point x belongs to k-th cluster is given by a logistic function

$$p_k(x) \propto \exp(\langle v_k, x \rangle + b_k),$$

where v_k, b_k are model parameters. We show that the maximization of the expected number (probability) of correct pairwise relations restricted to the sets of must-link $\mathcal{M}$ and cannot-link constraints $\mathcal{C}$ coincides with the maximization of

$$\sum_{(x,y)\in\mathcal{M}} \sum_k p_k(x)p_k(y) - \sum_{(x,y)\in\mathcal{C}} \sum_k p_k(x)p_k(y).$$

We extend the above formula to the set of unlabeled data by introducing additional pairwise constraints so that nearby points appear in the same cluster, which is a basis of the proposed objective function. On one hand, our model is theoretically well-motivated while, on the other hand, it obtains impressive experimental results.

We summarize the main contributions and the outline of our paper:

1. We formulate a clustering model, **d-graph**, which aims at maximizing the expected probability that pairwise relations are preserved (section 3.1).
2. Since pairwise constraints cover only small sample of examples, we use a modified graph approach to use internal similarities between data points (section 3.2).

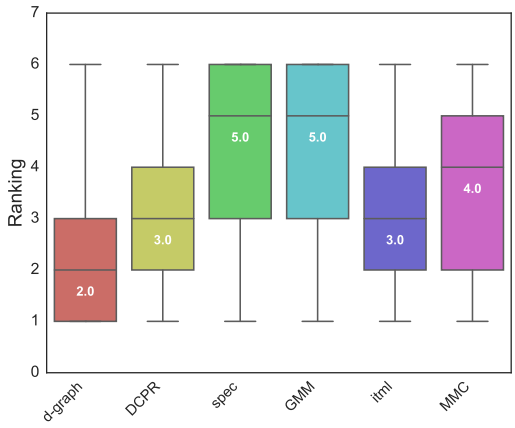


Figure 2: Box plot of ranks of examined methods summarizing the experimental study of the paper (the lower the better). This plot was constructed by ranking the methods on each data set including UCI examples (section 4.2), image data (section 4.4) and chemical data set (section 4.5). For a comparison we used semi-supervised versions of discriminative clustering (DCPR) [23], graph-based algorithm (spec) [24], model-based method (GMM) [25], metric learning approach (itml) [8] and max-margin technique (MMC) [38]

3. Our method is easy to implement and can be optimized with the use of gradient approach (section 3.3). Moreover, **d-graph** can be kernelized, see Figure 1 for the illustration.
4. We perform extensive experimental evaluation of the proposed method performed on various data sets, including UCI examples, image data as well as binary representations of chemical compounds (section 4). It is shown that **d-graph** outperforms existing state-of-the-art algorithms, see Figure 2 for the summary of the experiments.

A demo implementation of **d-graph** is available at `https://github.com/mareksmieja/d-graph`.

2. Related work

Pairwise constraints are a typical form of auxiliary information used in semi-supervised clustering. Its significance stems from the fact that in many cases it is much easier to decide whether two points belong to the same class than assign particular labels to data points. However, it is worth mentioning that there are also some works which apply relative relations [23, 2], partial labeling [39, 26] or initial decision boundary [27] as side information.

First works on applying pairwise relations into clustering focused on preserving all constraints. This idea was used by Wagstaff et al. in k-means algorithm [31] and by Shental et al. in Gaussian mixture model (GMM) [25]. Since some pairs of points may be mislabeled by human workers, these models were later modified so that the constructed partition does not have to meet all relations. It was implemented in k-means method by adding a penalty for an assignment inconsistent with side information [5] or by using hidden Markov random fields in GMM [19, 3]. Although this strategy was more flexible than the previous one, optimization of these models was computationally more expensive [22]. In recent years, combining spectral methods with pairwise constraints received a considerable attention. Generalized eigenvalue problem was defined by adding the subjection condition covering pairwise constraints to spectral objective function [33, 13]. Qian et al. [24] developed a framework for spectral clustering that allows using side information in the form of pairwise constraints, partial labeling, and grouping. Clustering techniques based on non-negative matrix or concept factorization can incorporate pairwise constraints as regularizers [18].

All the aforementioned methods incorporate pairwise constraints into a clustering objective function. Another line of research focuses on modifying distance or similarity measure based on side information. Xing et al. proposed learning a Mahalanobis distance by solving a constrained optimization problem [35]. The authors of [8] translated the problem of learning an optimal Mahalanobis distance to that of learning the optimal Gaussian with respect to an entropic objective. Chang et al. proposed to learn the semantic information in the manifold structure, and then integrate with supervised intentional knowledge in a local way [6]. The authors of [1] suggested reducing distances between data points with a must-link constraint and adding a dimension for each cannot-link constraint. After updating all other distances to, e.g., satisfy the triangle inequality, the thus obtained pairwise distance matrix can be used for unsupervised learning. Kamvar et al. [11] considered a similar procedure, taking the pairwise affinity matrix and setting must-links and cannot-links to predefined maximum and minimum values, respectively. Instead of clustering, they applied eigenvector-based classification taking the labeled data as a training set. Learning optimal Mahalanobis distance combined with entropy regularization for fuzzy c-means was proposed in [37]. Wang et al. used pairwise constraints to find lower dimensional representation followed by affinity propagation clustering algorithm [32]. Other works concern learning a kernel function from pairwise relations [36].

In contrast to the referred works, we derive our objective function directly from pairwise relations. Our model is not a modification of any existing clustering method, but it is designed so as to incorporate side information in the most natural way: it focuses on maximizing the expected number of correctly labeled pairs. The most closely related approach is a discriminative clustering model proposed by Pei et al.[23]. It combines maximum likelihood method for handling pairwise relations with information maximization (RIM) [15] to use unlabeled data. Although our technique is also based on discriminative approach, our objective function is significantly different. The experimental studies confirm that our approach leads to better clustering results.

3. Theoretical model

In this section, we introduce our clustering model and discuss its optimization. First, we show how to use pairwise constraints in a discriminative clustering framework. Next, we generalize obtained formula to handle unlabeled data. Finally, we combine both expressions and define **d-graph** objective function. We show that it can be easily kernelized, and optimized via gradient approach.

3.1. Pairwise constraints

We consider a data set $X \subset \mathbb{R}^D$, such that $N = |X|$, where every element $x \in X$ belongs to one of K unknown classes. By $\mathcal{X} = \{(x, y) \in X \times X : x \neq y\}$ we denote the set of pairs in X. In semi-supervised clustering partial information about class labels is revealed in the form of pairwise constraints, which cover selected pairs of data points $\mathcal{L} \subset \mathcal{X}$. Pairwise constraints indicate whether two points originate from the same or different classes, thus $\mathcal{L}$ can be split into the sets of must-link and cannot-link constraints given by [4]:

$$\mathcal{M} = \{(x, y) \in \mathcal{L} : x \text{ and } y \text{ belong to the same class}\},$$
$$\mathcal{C} = \{(x, y) \in \mathcal{L} : x \text{ and } y \text{ belong to the different classes}\}.$$

Clearly, if $(x, y) \in \mathcal{M}$ then $(y, x) \in \mathcal{M}$ (the same holds for cannot-link constraints).

In this paper we are motivated by the following problem:

Motivation. Clustering methods are usually evaluated using external cluster validity indices, such as Rand index (RI). Most of them measure the agreement between constructed partition $\mathcal{P}$ and reference grouping $\mathcal{R}$ by

counting the number of correctly classified pairs. More precisely, if TP and TN denote the sets of pairs, which are classified to the same or different clusters, respectively, in $\mathcal{P}$ and $\mathcal{R}$, then:

$$RI = \frac{1}{Z}(|TP| + |TN|),$$

where Z is a total number of pairs. Although the maximization of RI in a classical clustering process is impossible, because a ground-truth partition is unknown, one can use pairwise constraints to approximate its value. In this paper, we focus on defining a clustering model, which optimizes RI based on available information contained in pairwise constraints and unlabeled data.

Although RI suggests a reasonable objective in semi-supervised clustering, it might be difficult to directly maximize this quantity due to its discontinuity. Thus, we relax this problem to a continuous domain and, instead of maximizing the number of correctly classified pairs, we focus on maximizing the expected probability that pairs of data points are classified to correct clusters.

Our clustering model follows a discriminative approach, in which the assignments of data points to clusters are directly modeled by posterior probabilities. Let $p_k(x) = p(k|x)$ be a posterior probability that a data point $x \in X$ is assigned to k-th group, where $k = 1, \ldots, K$. Once these conditional probabilities are defined, we get a partition of X, in which a point $x \in X$ is assigned to this group that maximizes its posterior probability. Throughout this paper we assume that posterior probabilities are given by a logistic function:

$$p_k(x) = p_k(x; \mathcal{V}) = \frac{\exp(\langle v_k, x\rangle + b_k)}{\sum_{l=1}^{K} \exp(\langle v_l, x\rangle + b_l)},$$

where the set of parameters $\mathcal{V} = (v, b)$ consists of weight vectors $v = (v_1, \ldots, v_K)$ and bias values $b = (b_1, \ldots, b_K)$. More precisely, $v_k \in \mathbb{R}^D$ and $b_k \in \mathbb{R}$, for every $k = 1, \ldots, K$. To simplify the notation we omit model parameters $\mathcal{V}$ when they can be deduced from the context.

We are going to calculate the expected probability that the model defined by $p_k(\cdot)$ correctly classifies pairs of data points. Let $q : \mathcal{X} \to \{0, 1\}$ denote the indicator function (random variable) of ground-truth assignments, i.e:

$$q(x, y) = \begin{cases} 1, \text{ } x \text{ and } y \text{ originate from the same class,} \\ 0, \text{ otherwise.} \end{cases}$$

At training time, the values of q are unknown for most pairs except of must-link and cannot-link pairs: we have $q(x, y) = 1$, for $(x, y) \in \mathcal{M}$ and $q(x, y) = 0$, for $(x, y) \in \mathcal{C}$. However, the form of q has to be used to evaluate the expected value. Let us observe that the probability that a clustering model assigns two points $x, y \in X$ to the same cluster equals:

$$p_{\mathcal{M}}(x, y) = p_{\mathcal{M}}(x, y; \mathcal{V}) = \sum_{k=1}^{K} p_k(x) p_k(y).$$

Consequently, the probability that $x, y \in X$ are classified to different groups is given by:

$$p_{\mathcal{C}}(x, y) = p_{\mathcal{C}}(x, y; \mathcal{V}) = 1 - p_{\mathcal{M}}(x, y).$$

Observation 3.1. *Maximization of the expected probability of correctly classified pairs by a clustering model $p_k(\cdot)$ is equivalent to the maximization of*

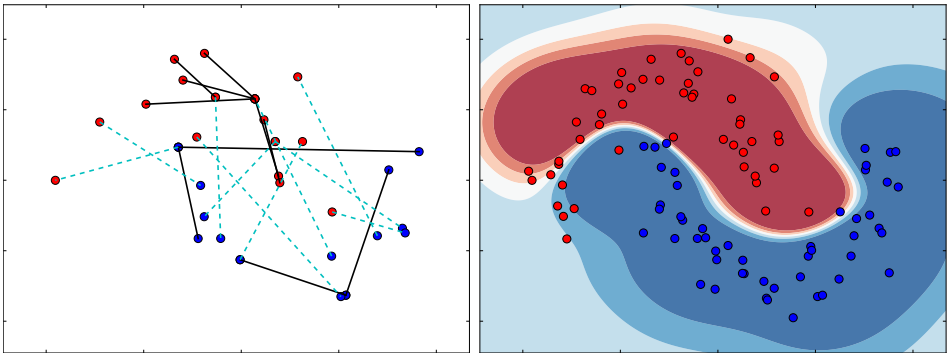
$$\frac{1}{|\mathcal{X}|} \sum_{(x,y)\in\mathcal{X}} (2q(x, y) - 1) p_{\mathcal{M}}(x, y). \tag{1}$$

Proof. Let us recall that $p_{\mathcal{M}}$ and $p_{\mathcal{C}}$ define probabilities that a clustering model (characterized by $p_k(\cdot)$) assigns data points to correct clusters. To calculate the expected value we go through all pairs of points, which gives:

$$\begin{aligned}
&\frac{1}{|\mathcal{X}|} \sum_{(x,y)\in\mathcal{X}} [q(x, y) p_{\mathcal{M}}(x, y) + (1 - q(x, y)) p_{\mathcal{C}}(x, y)] \\
&= \frac{1}{|\mathcal{X}|} \sum_{(x,y)\in\mathcal{X}} [2q(x, y) p_{\mathcal{M}}(x, y) - p_{\mathcal{M}}(x, y) - q(x, y) + 1] \\
&= \frac{1}{|\mathcal{X}|} \left[\sum_{(x,y)\in\mathcal{X}} (2q(x, y) - 1) p_{\mathcal{M}}(x, y) - \sum_{(x,y)\in\mathcal{X}} q(x, y) + |\mathcal{X}| \right] \\
&= \frac{1}{|\mathcal{X}|} \sum_{(x,y)\in\mathcal{X}} (2q(x, y) - 1) p_{\mathcal{M}}(x, y) - \frac{1}{|\mathcal{X}|} \sum_{(x,y)\in\mathcal{X}} q(x, y) + 1.
\end{aligned}$$

Since the last two terms are independent from the model selection, they can be discarded in the optimization problem, which completes the proof. □

Given the sets of must-link and cannot-link constraints one can optimize a clustering model by maximizing (1) restricted to the imposed pairwise



(a) Pairwise constraints (b) **d-graph** with pairwise constraints

Figure 3: Sample effects of **d-graph** with RBF kernel on two moons data set (b), using 20 pairwise constraints (a). Unlabeled data were not used for learning a model. Must-link constraint is marked with a solid black line while dashed cyan line is used for cannot-link relations.

relations. Since $q(x, y) = 1$, for $(x, y) \in \mathcal{M}$, and $q(x, y) = 0$, for $(x, y) \in \mathcal{C}$, then we aim at finding model parameters $\mathcal{V}$ which maximize:

$$\frac{1}{|\mathcal{L}|}\left[\sum_{(x,y)\in\mathcal{M}} p_{\mathcal{M}}(x, y) - \sum_{(x,y)\in\mathcal{C}} p_{\mathcal{M}}(x, y)\right]. \tag{2}$$

Figure 3 illustrates the effect of applying the above objective function on two moons data set. Observe that small number of pairwise constraints may be insufficient to correctly cluster the data. In the next section, we show how to approximate the values of q based on data points similarities.

3.2. Graph regularization

In addition to labeled pairs, a large number of unlabeled data is available. We show that a clustering framework defined in the previous subsection is able to handle unlabeled data as well. Our basic idea is to introduce additional constraints so that nearby points appear in the same cluster.

Let us first generalize formula (1) and define:

$$E(\mathcal{X}, w) = E(\mathcal{X}, w; \mathcal{V}) := \frac{1}{|\mathcal{X}|}\sum_{(x,y)\in\mathcal{X}} w(x, y)p_{\mathcal{M}}(x, y), \tag{3}$$

where $-1 \leq w(x, y) \leq 1$ are user defined weights, which replace the indicator function q. Positive values of $w(x, y)$ mean that x and y are more likely to

originate from the same class, while negative values indicate that they should be classified separately. Note, that $E(\mathcal{L}, l)$ with

$$l(x,y) = \begin{cases} 1, \text{ for } (x,y) \in \mathcal{M}, \\ -1, \text{ for } (x,y) \in \mathcal{C} \end{cases} \tag{4}$$

directly coincides with (2). Thus the maximization of (3) is equivalent to the maximization of the expected probability of correctly labeled pairwise relations over a given set of pairs $\mathcal{X}$ with a weight function w. Since a weight function w plays a role of true cluster indicator, its choice will be crucial for clustering performance. We show how to construct a weight function w for unlabeled data based on a graph approach and a semi-supervised learning paradigm.

Remark 3.1. Typical graph clustering aims at finding a matrix $F = (f_{ik})_{ik} \subset \mathbb{R}^{N\times K}$ which minimizes [30]:

$$\frac{1}{2}\sum_{i,j} a_{ij} \sum_{k} (f_{ik} - f_{jk})^2,$$

subject to $F^T F = I$, where $W = (w_{ij})_{ij} \subset \mathbb{R}^{N\times N}$ is a user-defined affinity matrix $W = (w_{ij})_{ij} \subset \mathbb{R}^{N\times N}$. Each column of F induces a discrimination function f_k, which defines a chance that a given data point x belongs to k-th cluster. One can observe that (3) replaces $\mathcal{L}_2$ norm $\sum_k (f_{ik} - f_{jk})^2$ of discrimination function f_k by its probabilistic counterpart $p_{\mathcal{M}}(x,y) = \sum_k p_k(x)p_k(y)$. In contrast to graph approaches, our posterior probabilities have a parametric form, which allows to classify new, previously unseen data points. Moreover, it has an intuitive probabilistic interpretation.

To set the weighs for unlabeled examples, we follow a paradigm of semi-supervised learning, which states that posterior probabilities change smoothly over nearby points. Therefore, if x is localized close to y, the weight $w(x,y)$ should be proportional to the similarity between x and y. To realize this idea, we construct a directed weighted ε-neighborhood graph on X as follows. Let $0 \leq s(x,y) \leq 1$ be any similarity measure between $(x,y) \in \mathcal{X}$. Although, we use a similarity measure given by a radial basis function (RBF), i.e.:

$$s_\gamma(x,y) = \exp(-\gamma \|x - y\|^2), \text{ for } \gamma > 0,$$

other choices are also possible. For each data point x, we consider its ε-neighborhood defined by:

$$N_\varepsilon(x) = \{y \in X \setminus \{x\} : s(x,y) \geq \varepsilon\}, \text{ for a fixed } \varepsilon \in [0,1].$$

If $y \in N_\varepsilon(x)$, then we link vertices x and y by a directed edge with a weight $w(x,y) = 2s(x,y) - 1$.

The above procedure will link nearby points by edges with positive weights, which could encourage the clustering algorithm to include all data points into the same cluster. To balance the graph and avoid such degenerate solutions, we add negative weights to edges related to all other pairs of points (analogical reasoning was used in the case of RIM clustering framework [15]). If we assume that all clusters in ground-truth partition are equally-sized, then probability that a randomly chosen pair of points originates from the same cluster equals $\frac{1}{K}$. We treat this quantity as a rough approximation of similarity. In consequence, the corresponding weights are given by:

$$w(x,y) = 2\frac{1}{K} - 1 = -\frac{K-2}{K}, \text{ for } y \notin N_\varepsilon(x), y \neq x.$$

Although setting non-zero weights for all pairs of points results in a dense graph, its optimization will be performed efficiently since most weights are equal (see the next subsection for details).

Finally, the weighing function w for unlabeled part $\mathcal{X}$ is defined as:

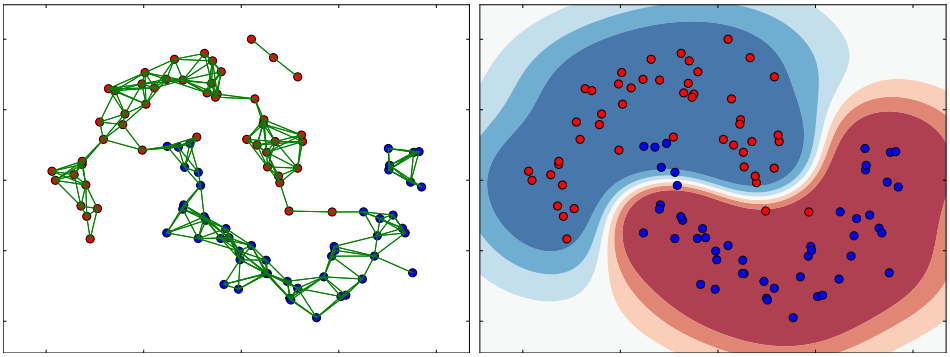
$$w(x,y) = \begin{cases} 2s_\gamma(x,y) - 1, & y \in N_\varepsilon(x), \\ -\frac{K-2}{K}, & y \notin N_\varepsilon(x), y \neq x. \end{cases} \tag{5}$$

Thus,

$$E(\mathcal{X}, w) = \frac{1}{|\mathcal{X}|} \sum_{(x,y)\in\mathcal{X}} w(x,y) p_{\mathcal{M}}(x,y) =$$

$$\frac{1}{|\mathcal{X}|} \left[\sum_{x\in X} \sum_{y\in N_\varepsilon(x)} (2s(x,y) - 1) p_{\mathcal{M}}(x,y) - \sum_{x\in X} \sum_{\substack{y\notin N_\varepsilon(x) \\ y\neq x}} \frac{K-2}{K} p_{\mathcal{M}}(x,y) \right]. \tag{6}$$

Graph construction for a sample data set is presented in Figure 4(a). Clustering results using only graph information (without pairwise constraints), see Figure 4(b), show that cluster memberships of some points cannot be easily deduced without expert knowledge. In consequence, unsupervised information about data should be combined with supervised knowledge to produce best results.



(a) Graph from unlabeled data (b) **d-graph** with unlabeled data

Figure 4: Sample results of **d-graph** with RBF kernel using only unlabeled data. Top $7 \cdot |X|$ nearest pairs were connected with vertices with positive weights (marked with solid green line), while other pairs had negative weights attached to corresponding vertices (they are not marked in the figure).

3.3. Objective function

We demonstrated that (2) and (6) express the expected probability of correct clustering based on information contained in pairwise constraints and data similarity. To define the objective function of **d-graph** we combine both terms by a trade-off factor $\tau \geq 0$, which allows to control the balance between supervised and unsupervised counterparts. By default, we put $\tau = 1$ to equally take into account both factors. Moreover, to prevent from model overfitting we add $\mathcal{L}_2$ regularization to the model parameters $v = (v_1, \ldots, v_K)$:

$$R(v) = \sum_{k=1}^{K} v_k^T v_k.$$

Note that biases are not penalized.

Concluding, a complete form of the objective function of **d-graph** is defined as follows:

Definition 3.1. Let $X \subset \mathbb{R}^N$ be a data set, where selected pairs form pairwise constraints $\mathcal{L} \subset \mathcal{X}$. We consider posterior probabilities $p_k(x); V$ parametrized by $\mathcal{V}$. Given model hyper-parameters $\tau \geq 0, \lambda > 0$, the clus-

tering objective functions of **d-graph** is given by

$$
\begin{aligned}
&E(\mathcal{L}, \mathcal{X}, l, w; \mathcal{V}) = E(\mathcal{L}, l) + \tau E(\mathcal{X}, w) - \lambda R(v), \\
&= \frac{1}{|\mathcal{L}|} \sum_{(x,y)\in\mathcal{L}} l(x,y) p_{\mathcal{M}}(x,y) + \tau \frac{1}{|\mathcal{X}|} \sum_{(x,y)\in\mathcal{X}} w(x,y) p_{\mathcal{M}}(x,y) - \lambda \sum_{k=1}^{K} v_k^T v_k,
\end{aligned} \tag{7}
$$

where wights function l and w are given by (4), (5):

$$
\begin{aligned}
l(x,y) &= \begin{cases} 1, \text{ for } (x,y) \in \mathcal{M}, \\ -1, \text{ for } (x,y) \in \mathcal{C} \end{cases} \\
w(x,y) &= \begin{cases} 2s_\gamma(x,y) - 1, & y \in N_\varepsilon(x), \\ -\frac{K-2}{K}, & y \notin N_\varepsilon(x), y \neq x. \end{cases}
\end{aligned}
$$

In order to maximize (7), we apply a gradient approach, for more details see Appendix A. We plug the gradient to L-BFGS quasi Newton-optimization algorithm[1] [17]. Since our objective function is non-concave, this procedure is only guaranteed to find one of its local maximums. To deal with this issue we restart L-BFGS multiple times with different initial coefficients[2] $\mathcal{V}$. Basically, we draw model parameters from normal distribution $N(0, 1)$.

Presented model builds linear clusters boundaries. To obtain more flexible clusters boundaries one can combine it with kernel functions, such as RBF or Tanimoto kernel, in a similar manner to [15]. If we set partial derivatives of our objective function to zero, then we get that:

$$
v_k = \alpha_{k0} + \sum_{x_i \in X} \alpha_{ki} x_i,
$$

for some $\alpha_k = (\alpha_{k0}, \alpha_{k1}, \ldots, \alpha_{kN})$. It means that at stationary points the weights are spanned over input data points. One can use this observation

[1] We used Mark Schmidt's implementation at `https://www.cs.ubc.ca/~schmidtm/Software`.

[2] Alternatively, one could first split data into initial clusters with the use of any clustering algorithm and next train classical (supervised) logistic regression classifier using obtained clusters labels, see [15] for details. On one hand, this procedure finds a unique solution given initial clustering, because the logistic regression classifier is concave; on the other hand, it is highly dependent on selected clustering algorithm.

to replace the inner products $v_k x$ with $\sum_{x_i \in X} \alpha_{ki} K(x_i, x)$, where $K(\cdot, \cdot)$ is a positive definite kernel mapping. Kernel functions can also be applied to a regularization term $R(v)$, where we use $\sum_k \sum_{i,j} \alpha_{ki} \alpha_{kj} K(x_i, x_j)$. Thus, it is sufficient to apply a gradient method to such a modified objective function parametrized by $\alpha_k \in \mathbb{R}^{N+1}$.

4. Experiments

In this section, we experimentally analyze the performance of proposed method and compare it with related clustering techniques. We use examples retrieved from UCI repository, images with labeled segments as well as a real data set of chemical compounds.

4.1. Experimental setup

We considered classification data sets, where each feature was standardized to have zero mean and unit variance. To generate pairwise constraints, we randomly selected a pair of points (x, y) and labeled it as must-link if both x, y belonged to the same cluster or as cannot-link, otherwise. We varied the number of constraints from $0.05N$ to $0.2N$ with a $0.05N$ increment.

The results were evaluated using adjusted rand index (ARI) [10]. ARI attains a maximum value 1 for a partition identical with a ground-truth, while for a random grouping it gives score 0. To reduce the effect of randomness, we generated 10 different samples of pairwise relations and averaged ARI scores. All hyperparameters of considered methods were tuned using 5-fold cross-validation procedure. We divided a set of pairwise constraints into 5 equally-sized parts. Four of them were used for training a model and its prediction rate was verified on the last one. A combination of parameters maximizing ARI averaged over 5 validation sets of constraints was chosen to train a final model.

Our method is parametrized by a trade-off factor τ, regularization parameter λ and graph parameters ε and γ. To reduce training effort we fixed all parameters, except λ, which was selected using a grid search. We put $\tau = 1$ to equally balance the importance between labeled and unlabeled data. Generally, higher values of τ can be used when there are no errors in pairwise constraints, but in the case of noisy side information its value should be smaller. Experimental analysis of the choice of τ is performed in Appendix C. Let us also observe that high values of τ increase the weights of labeled examples and, in consequence, may change data distribution. Thus,

if pairwise constraints are not a representative sample of data points, then τ value should not be too large. To construct ε-neighborhood graph we fixed ε so as to take into account $7|X|$ nearest neighbors in total. In consequence, similarity measure was used only for top $7|X|$ nearest pairs, while other pairs were attached weights equal $-\frac{1}{K}$. We fixed $\gamma = 1$ for RBF similarity function. We observed that its choice does not influence the results heavily. In section Appendix B, we demonstrate how the number of neighbors influences the clustering results. Parameter λ is a penalty for model complexity, so it has to be tuned individually for a particular data set. We optimized λ using a grid search over the range $\{\frac{1}{D \cdot 2^k}, k = 4, 6, 8, 10, 12\}$, where D is data dimension, which provided satisfactory results in most cases.

We compared **d-graph** with five state-of-the-art techniques, each one realizing different clustering paradigm:

- *Graph-based technique:* The first comparative method is a recent modification of spectral clustering to semi-supervised setting [24], which will be referred to as **spec**. Following the original paper, we chose RBF for calculating affinity matrix, where γ defines its radius. Additionally, the method is parametrized by a trade-off factor η combining unlabeled data with labeled examples. We explored the following parameters ranges $\eta \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ and $\gamma \in \{0.25, 0.5, 1, 2\}$.

- *Discriminative clustering:* Next, we considered a discriminative clustering model proposed by Pei et al. [23], referred to as **DCPR**. It incorporates pairwise relations using maximum likelihood approach and includes unlabeled data applying RIM framework [15]. It is parametrized by trade-off factor τ combining these two terms, and parameter λ, which defines a penalty for model overfitting. These parameters play analogical role to the ones used in our method. Thus, we also put $\tau = 1$ and select λ from the range $\{\frac{1}{D \cdot 2^k}, k = 2, 4, 6, 8, 10\}$.

- *Model-based method:* We used an extension of GMM to the case of pairwise constraints proposed by Shental et al. [25], which will be referred to as **GMM**. It applies transitivity of must-links and hidden Markov random fields to include pairwise constraints to the model. It does not require any user-defined parameters (except the number of clusters).

- *Metric learning:* We considered an information-theoretic metric learning approach, **itml**, [8]. It is parametrized by a slack parameter γ,

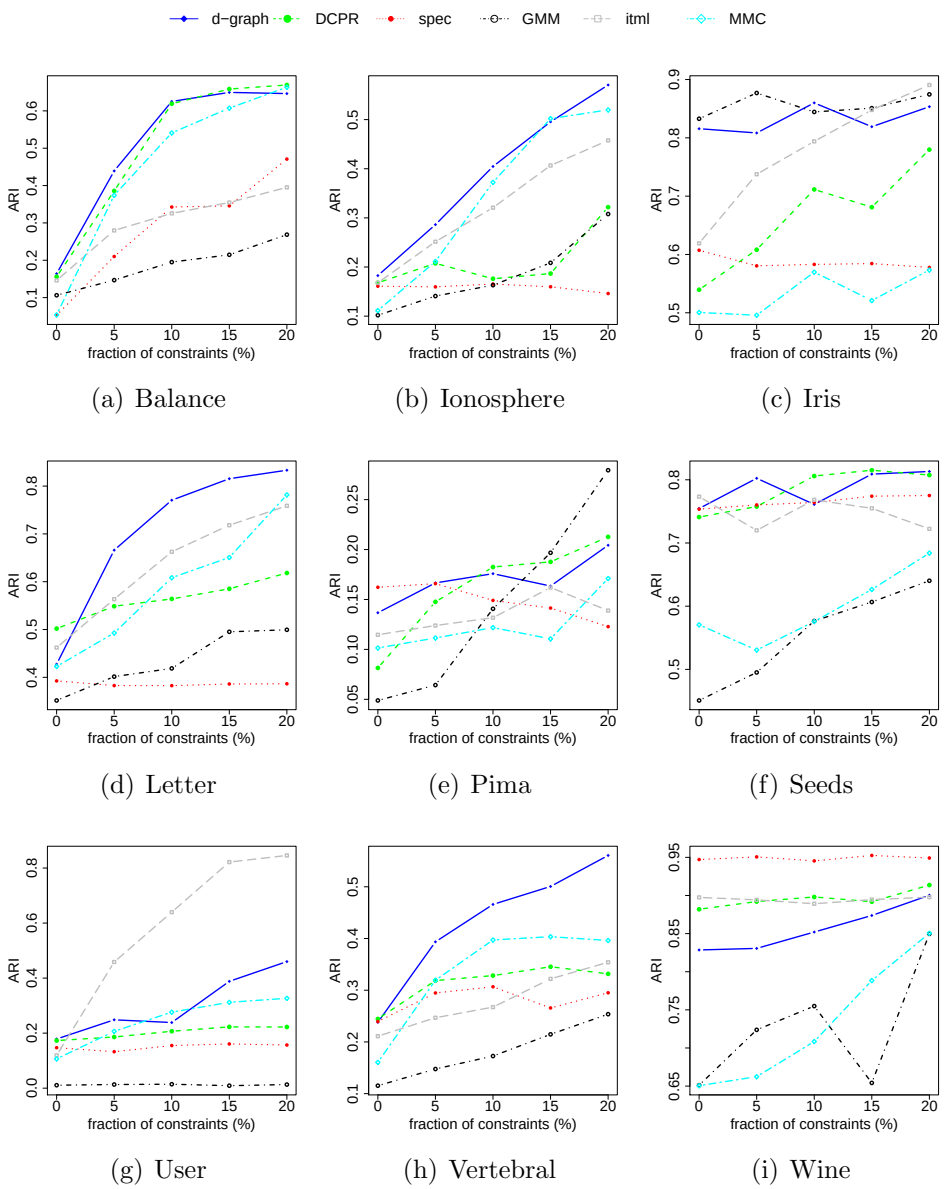
Table 1: Summary of UCI datasets used in the experiments.

Data set	**# Instances**	**# Features**	**# Classes**
Balance	625	4	3
Ionosphere	351	34	2
Iris	150	4	3
Letter^{+}	1000	16	5
Pima	768	8	2
Seeds	210	7	3
User Modeling	403	5	4
Vertebral	310	6	3
Wine	178	13	3

$^{+}$: We took a subset of letters: "A,B,C,D,E".

which was chosen from $\{0.01, 0.1, 1, 10\}$. We applied k-means with the learned metric to form final clusters assignments.

- *Maximum-margin clustering:* We applied a semi-supervised technique, which realizes a maximum-margin paradigm [38], referred to as **MMC**. Its trade-off factor δ was set to 1 (as suggested by the authors) while parameter λ, responsible for model regularization, was selected from the range $\{0.001, 0.01, 0.1, 1, 10, 100\}$.



(a) Balance (b) Ionosphere (c) Iris

(d) Letter (e) Pima (f) Seeds

(g) User (h) Vertebral (i) Wine

Figure 5: Adjusted rand index of examined methods evaluated on UCI data sets.

4.2. UCI data sets

In the first experiment, we considered 9 UCI classification data sets [16] summarized in Table 1 and investigated the influence of the number of constraints on the clustering performance.

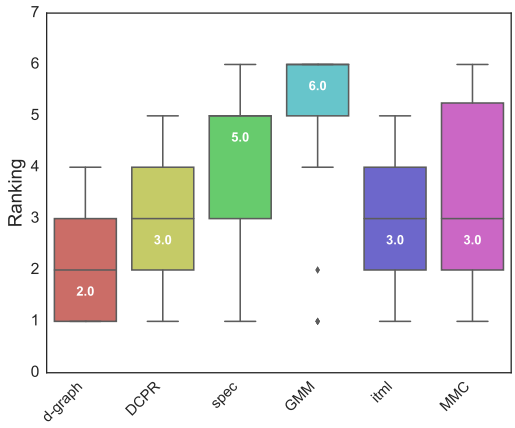


Figure 6: Box plot of ranks of examined methods evaluated on UCI data sets (the lower the better).

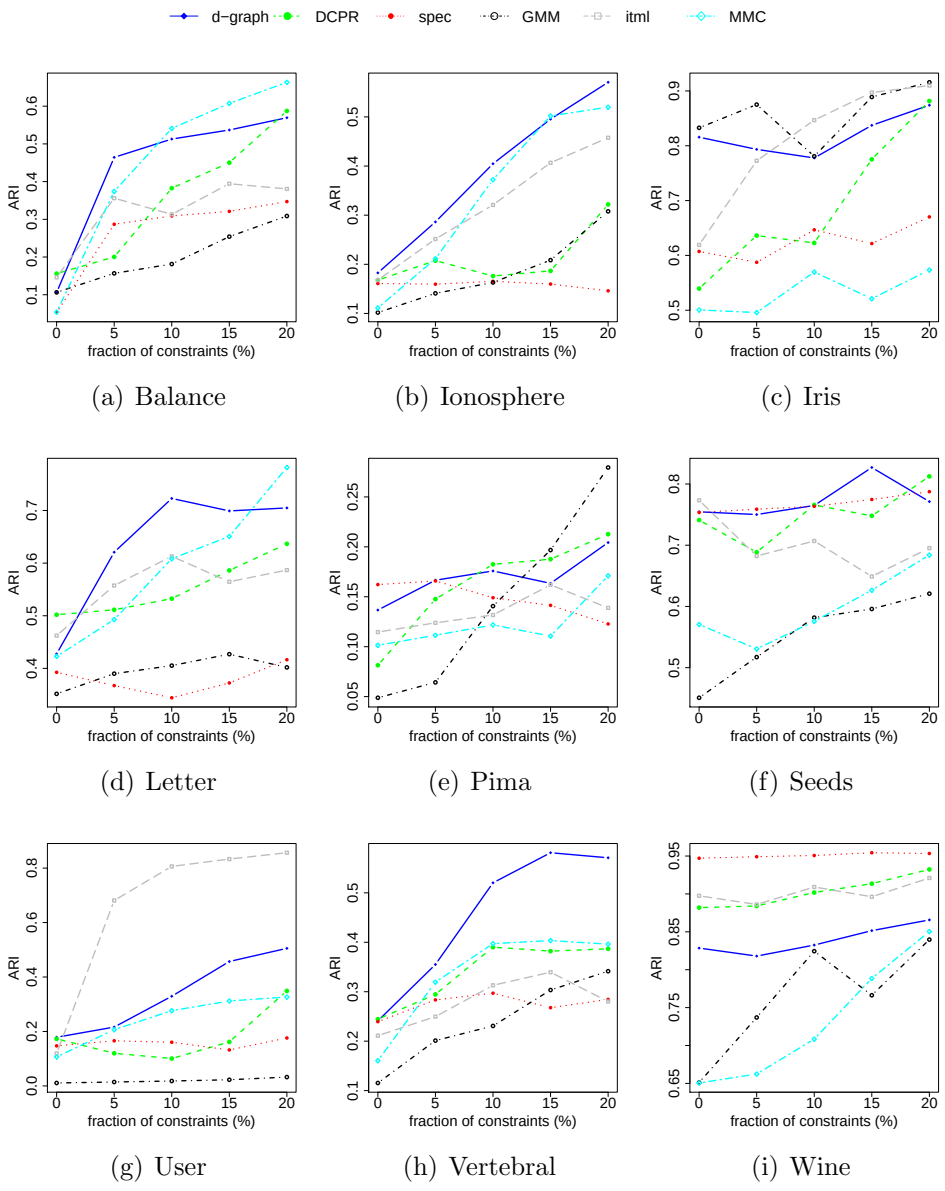
The results presented in Figure 5 show that **d-graph** usually obtained one of the best scores. The performance of **d-graph** gradually increases as the number of constraints grows. It significantly outperformed comparative algorithms on Vertebral and Letter data sets and was slightly better on Ionosphere and Balance. Its worse results on User and Wine data might follow from the linear model applied in **d-graph**. More flexible nonlinear methods such as **itml** and **spec** were able to obtain higher performance in these two cases. The ability of using kernel functions by **d-graph** to create nonlinear cluster boundaries will be investigated in Section 4.4.

To further analyze the results, we ranked the methods on each data set (the best performing method got rank 1, second best got rank 2, etc.). Figure 6 presents a box plot of ranks averaged over all data sets and all numbers of constraints. The vertical lines show the range of the ranks, while the horizontal line in the middle denotes the median. This summary confirms that **d-graph** was able to use additional knowledge in the most appropriate way. It can be observed that **itml**, **DCPR** and **MMC** also gave high resemblance with reference grouping in most cases. On the other hand, **GMM** and **spec** were not able to effectively use knowledge contained in pairwise constraints in most cases.

4.3. Few labeled classes

Pairwise constraints represent the expert knowledge and are assigned to a small number of pairs of points. However, they do not have to cover all

classes. In this experiment, we verify whether **d-graph** is able to detect classes that are not covered by any pairwise constraints. For this purpose, we generated pairwise relations only from two fixed reference classes of each data set.



(a) Balance (b) Ionosphere (c) Iris

(d) Letter (e) Pima (f) Seeds

(g) User (h) Vertebral (i) Wine

Figure 7: Adjusted rand index of examined methods evaluated on UCI data sets (2 classes labeled).

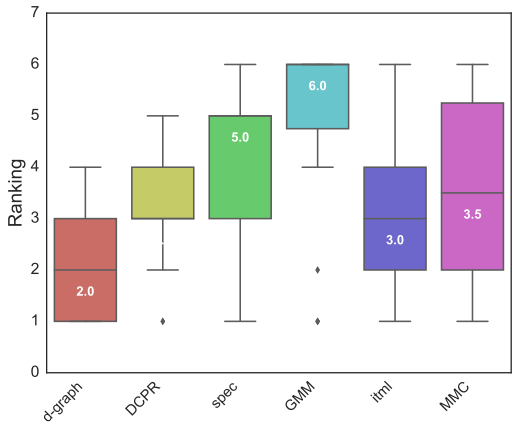


Figure 8: Box plot of ranks of examined methods evaluated on UCI data sets (2 classes labeled).

It is evident from the results presented in Figure 7 that **d-graph** gives comparable scores as in the previous experiment[3]. Moreover, a box plot of ranks, Figure 8, shows that **d-graph** outperformed other methods. This confirms that our method is also able to detect clusters which were not presented in side information.

4.4. Image segmentation

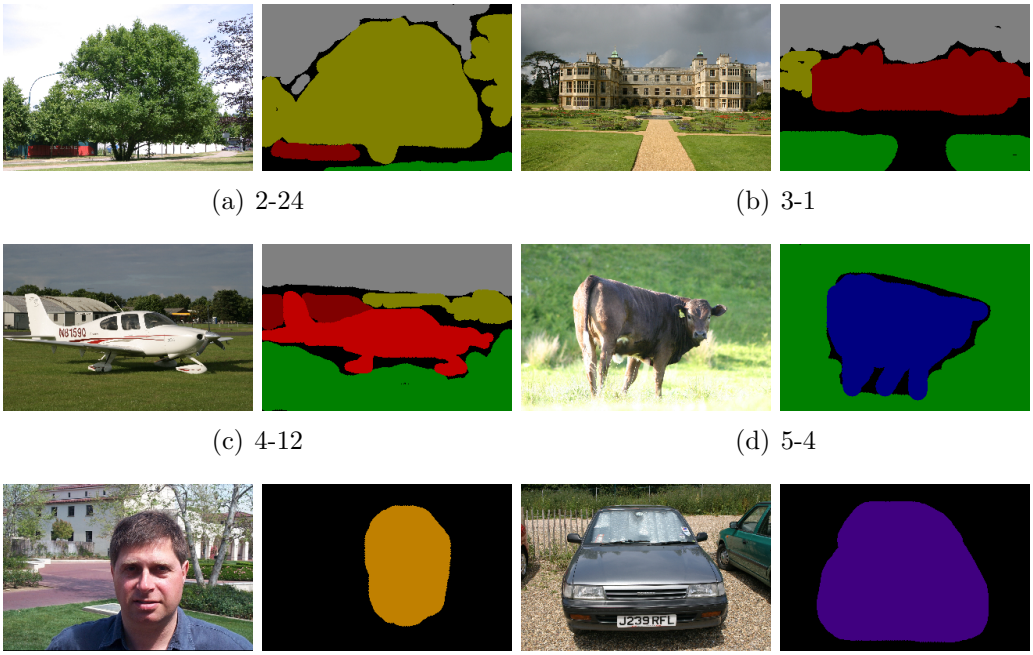
In this experiment, we considered image data retrieved from Object Class Recognition Database [21]. It contains manually segmented and labeled photographs. We selected 6 images, which together with their ground truth partitions are presented in Figure 9. Last three pictures can be seen as identification tasks, where only 2 or 3 classes are distinguished, while the others contain landscapes divided into more segments.

To allow for a higher flexibility of **d-graph**, RBF kernel function was used:

$$K(x, y) = \exp(-\gamma \|x - y\|^2).$$

We selected kernel width γ from the range $\{0.25, 0.5, 1, 2, 4\}$ using cross-validation procedure. RBF kernel was also combined with **DCPR** while other methods were run in standard settings.

[3]We excluded Ionosphere and Pima because they contain only two classes and, in consequence, the results are exactly the same as in Figure 5



(a) 2-24 (b) 3-1

(c) 4-12 (d) 5-4

(e) 6-2 (f) 7-18

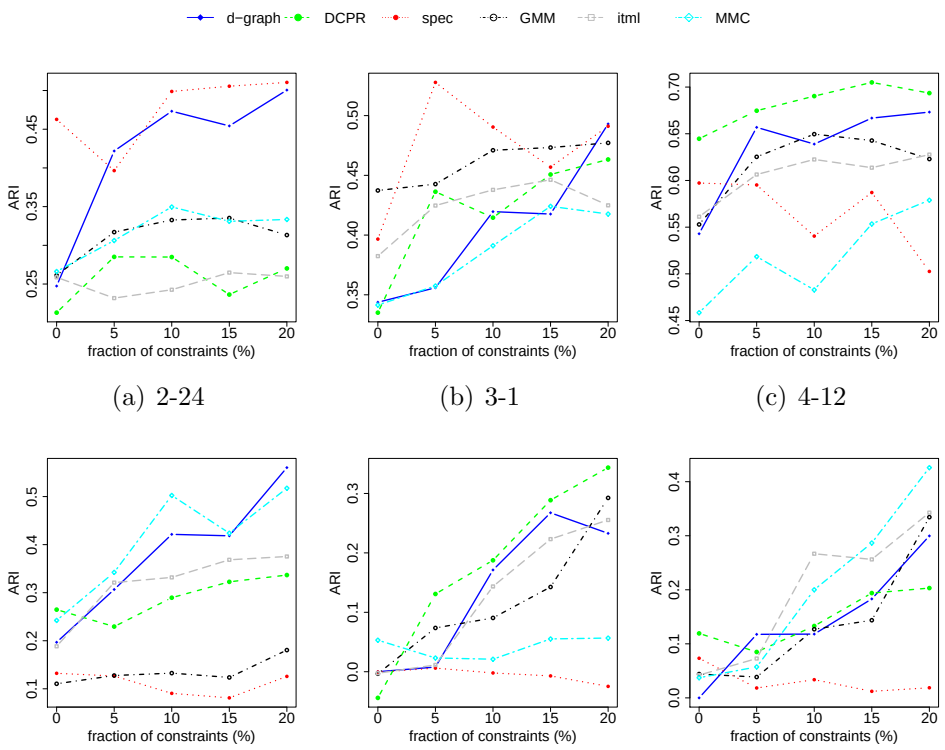
Figure 9: Image data sets and their reference segmentations.

The results presented in Figure 10 do not demonstrate as much advantage of **d-graph** as in previous experiments. However, the increase of its performance is still stable and its results do not differ much from the best ones. Summary of the results, Figure 10, shows that the performance of all methods, except **spec**, is comparable. Although the median rank of **d-graph** is better than the other algorithms, the range of its ranks is wide.

4.5. Detection of chemical classes

In the final experiment, we used a family of chemical compounds, which was manually divided into classes by the expert in the field [34]. The classification has a hierarchical structure as shown in Figure 12. The task undertaken in this section is to discover 8 classes from the bottom of hierarchy.

We used Klekota-Roth fingerprint to represent chemical compounds [14]. It describes every instance by a binary vector, where "1" means presence and "0" denotes absence of a predefined chemical pattern. This fingerprint takes into account 4860 chemical features, which results in 4860 dimensional input space. Due to binary type of data, we decided to use Tanimoto kernel [7] to



(a) 2-24 (b) 3-1 (c) 4-12

(d) 5-4 (e) 6-2 (f) 7-18

Figure 10: Adjusted rand index of examined methods evaluated on image data sets.

transform data into another space. Tanimoto kernel, defined by:

$$K(x, y) = \frac{\langle x, y \rangle}{\langle x, x \rangle + \langle y, y \rangle - \langle x, y \rangle}, \text{ for } x, y \in \{0, 1\}^D,$$

is one of the basic kernel functions applied to binary data. It was impossible to run **GMM** on such high dimensional space[4] – therefore, we decided to use PCA to reduce dimension of data to 15 principal components.

As can be seen from Figure 13(a), **d-graph** obtained the highest ARI score in almost all cases. The performance of **itml** and **DCPR** was slightly lower. Other methods gave significantly worse results. Despite the advantage of **d-graph** over comparative methods, its ARI scores just under one third are not satisfactory. Such low results might be caused by a relatively large

[4]covariance matrices of clusters were singular

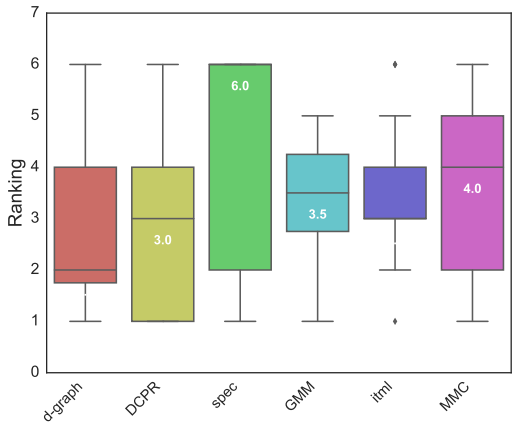


Figure 11: Box plot of ranks of examined methods on image data sets.

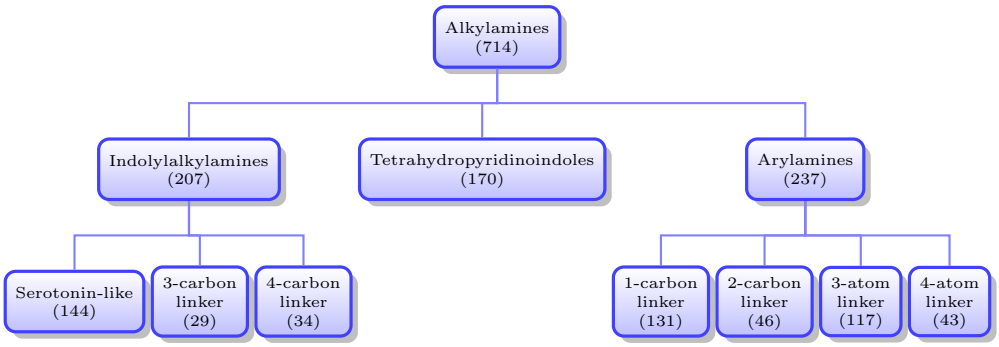


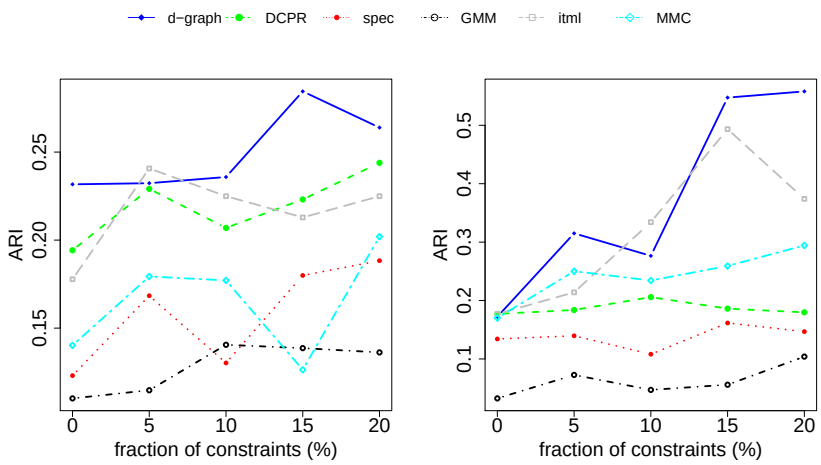
Figure 12: Hierarchy tree of chemical compounds classes. The numbers in brackets indicate the number of times the corresponding class appeared in the data set.

number of classes or by high similarity between some of them.

We considered a second task, in which the goal was to detect 3 classes from the top of hierarchy (Indolylalkylamines, Tetrahydropyridinoindoles and Arylamines). The results presented in Figure 13(b) show very good clustering effects of **itml** and **d-graph**. Both methods obtained ARI scores of above 50% after adding 15% of constraints. Other methods were not able to increase their performance and remained at the same level as in the case of 8 chemical classes.

5. Conclusion

In this paper we introduced a new discriminative model, **d-graph**, for handling pairwise constraints and unlabeled data in semi-supervised cluster-



(a) Alkylamines (8 classes) (b) Alkylamines (3 classes)

Figure 13: Adjusted rand index of examined methods evaluated on chemical data sets.

ing. Its optimization leads to the maximization of the number of correctly labeled pairs of data points, which is a typical goal in clustering. Experimental results confirmed that the constructed method can be successfully applied in semi-supervised clustering problems and allows to obtain better results than comparative methods. The strength of **d-graph** stems from its discriminative form, which focuses on a supervised problem. Moreover, its kernelization allows to discover arbitrary shapes of classes.

Appendix A. Gradient calculation

We show how to compute the gradient of our objective function (7). We begin with initial manipulations of (6) and then calculate the gradient.

Let us rewrite the formula (6) as:

$$E(\mathcal{X},w)=\frac{1}{|\mathcal{X}|}\left[\sum_{x\in X}\sum_{y\in N_\varepsilon(x)}(2s(x,y)-1)p_{\mathcal{M}}(x,y)-\sum_{x\in X}\sum_{y\notin N_\varepsilon(x),y\neq x}\frac{K-2}{K}p_{\mathcal{M}}(x,y)\right]$$
$$=\frac{1}{|\mathcal{X}|}\left[\sum_{x\in X}\sum_{y\in N_\varepsilon(x)}(2s(x,y)-\frac{2}{K})p_{\mathcal{M}}(x,y)+\sum_{x\in X}\frac{K-2}{K}p_{\mathcal{M}}(x,x)\right]$$
$$-\frac{1}{|\mathcal{X}|}\sum_{x\in X}\sum_{y\in Y}\frac{K-2}{K}p_{\mathcal{M}}(x,y).$$

If we denote an extended set of ε-neighborhood pairs by $X_\varepsilon=\bigcup\limits_{x\in X}\{(x,N_\varepsilon(x)\}\cup\{(x,x)\}$ and corresponding weights by

$$t(x,y)=\begin{cases}2s(x,y)-\frac{2}{K}, & y\in N_\varepsilon(x),\\ \frac{K-2}{K}, & y=x,\end{cases}$$

then the above function equals:

$$E(\mathcal{X},w)=\frac{1}{|\mathcal{X}|}\left[\sum_{x\in X}\sum_{y\in N_\varepsilon(x)\cup\{x\}}t(x,y)p_{\mathcal{M}}(x,y)-\frac{K-2}{K}\sum_{x\in X}\sum_{y\in X}p_{\mathcal{M}}(x,y)\right]$$
$$=\frac{|\mathcal{X}_\varepsilon|}{|\mathcal{X}|}E(\mathcal{X}_\varepsilon,t)-\frac{K-2}{K}\sum_k\left(\frac{1}{N}\sum_{x\in X}p_k(x)\right)^2.$$

For further calculations, by

$$p_k=\frac{1}{N}\sum_{x\in X}p_k(x)$$

we denote an empirical probability of k-th cluster, which can be substituted in the previous expression.

Let us start with computing partial derivatives of posterior probabilities $p_k(x)$,

$$\frac{\partial p_k(x)}{\partial v_{cd}}=(\delta_{kc}-p_k(x))p_c(x)x_d,$$

where δ_{kc} equals 1 when indices k and c are identical, and 0 otherwise. To obtain a derivative with respect to bias, it is sufficient to use the above formula with $x_{D+1} = 1$. For a regularization term R we have,

$$\frac{\partial R}{\partial v_{cd}} = -2v_{cd}. \tag{A.1}$$

The partial derivatives of function $E(\mathcal{X}, w)$ equal:

$$\frac{\partial E(\mathcal{X}, w)}{\partial v_{cd}} = \frac{|X_\varepsilon|}{|\mathcal{X}|}\frac{\partial E(\mathcal{X}_\varepsilon, t)}{\partial v_{cd}} - \frac{K-2}{K}\sum_k 2p_k\frac{\partial p_k}{\partial v_{cd}}. \tag{A.2}$$

We have:

$$\begin{aligned}\frac{\partial E(\mathcal{X}_\varepsilon, t)}{\partial v_{cd}} &= \frac{1}{|X_\varepsilon|}\sum_{(x,y)\in\mathcal{X}_\varepsilon} t(x,y)\sum_k\left[\frac{\partial p_k(x)}{\partial v_{cd}}p_k(y) + \frac{\partial p_k(y)}{\partial v_{cd}}p_k(x)\right] \\ &= \frac{1}{|X_\varepsilon|}\sum_{(x,y)\in\mathcal{X}_\varepsilon} t(x,y)\left[p_c(x)p_c(y)(x_d+y_d) - p_{\mathcal{M}}(x,y)(p_c(x)x_d + p_c(y)y_d)\right]\end{aligned} \tag{A.3}$$

and

$$\begin{aligned}\sum_k p_k\frac{\partial p_k}{\partial v_{cd}} &= \frac{2}{N}\sum_k p_k\sum_{x\in X}\frac{\partial p_k(x)}{\partial v_{cd}} \\ &= \frac{1}{N}\sum_{x\in X}\sum_k(\delta_{kc} - p_k(x))p_c(x)x_d p_k \\ &= \frac{1}{N}\sum_{x\in X}\left[(1-p_k(x))p_c(x)x_d p_k - \sum_{k\neq c}p_k(x)p_c(x)x_d p_k\right] \\ &= \frac{1}{N}\sum_{x\in X}p_c(x)x_d\left[p_c - \sum_k p_k(x)p_k\right].\end{aligned}$$

If we denote by $S(x) = \sum_k p_k(x)p_k$ an auxiliary function, then formula (A.2)

simplifies to:

$$\frac{\partial E(\mathcal{X}, w)}{\partial v_{cd}} = \frac{1}{N^2} \sum_{(x,y)\in\mathcal{X}_\varepsilon} t(x,y) \left[p_c(x)p_c(y)(x_d + y_d) - p_{\mathcal{M}}(x,y)(p_c(x)x_d + p_c(y)y_d)\right] - \frac{2(K-2)}{NK} \sum_{x\in X} p_c(x)x_d \left[p_c - S(x)\right]. \quad \text{(A.4)}$$

Making use of (A.3), we also calculate a derivative of $E(\mathcal{L}, l)$,

$$\begin{aligned}\frac{\partial E(\mathcal{L}, l)}{\partial v_{cd}} &= \frac{1}{|\mathcal{L}|} \sum_{(x,y)\in\mathcal{L}} l(x,y) \left[p_c(x)p_c(y)(x_d + y_d) - p_{\mathcal{M}}(x,y)(p_c(x)x_d + p_c(y)y_d)\right] \\ &= \frac{1}{|\mathcal{L}|} \left[\sum_{(x,y)\in\mathcal{M}} (p_c(x)p_c(y)(x_d + y_d) - p_{\mathcal{M}}(x,y)(p_c(x)x_d + p_c(y)y_d)) \right. \\ &\quad \left. - \sum_{(x,y)\in\mathcal{C}} (p_c(x)p_c(y)(x_d + y_d) - p_{\mathcal{M}}(x,y)(p_c(x)x_d + p_c(y)y_d)) \right]. \end{aligned} \quad \text{(A.5)}$$

Concluding, the gradient of our objective function is a combination of formulas (A.1), (A.4) and (A.5). The computational complexity of evaluation (A.5) is $O(|\mathcal{L}|KD)$. Note that, if one remembers $S(x)$ for every data point, then the cost of calculation (A.4) equals $O(|X_\varepsilon|KD)$. It is possible to reduce its computational complexity by taking higher number ε to construct a graph. In particular, $\varepsilon = 1$ leads to the linear complexity with respect to the number of data points.

Appendix B. Impact of graph construction

The performance of our method is influenced by the way we construct a similarity graph. The following experiment shows how the clustering results vary when we change the value of ε in ε-neighborhood graph. We selected 6 levels of ε so as to assign positive weights to $m|X|$ pairs, where $m \in \{1, 3, 5, 7, 9, 11\}$. Then, we individually calculated mean ranks of these 6

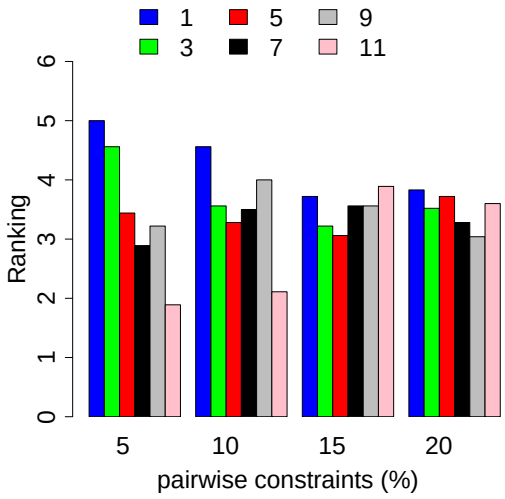


Figure B.14: Median ranks of **d-graph** with different number of neighbors used in regularization term calculated on UCI data sets.

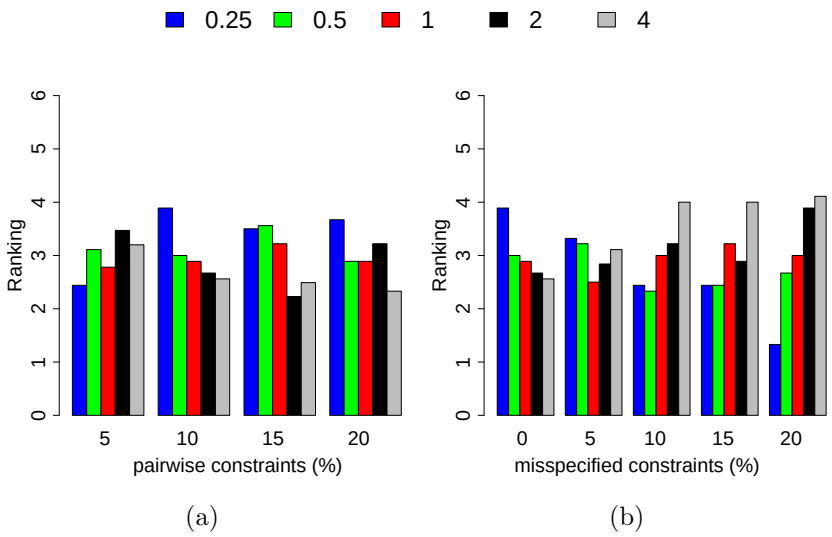
variants of our method for each level of pairwise constraints. We used UCI data sets.

As can be seen from Figure 6, higher values of m positively influenced the performance of **d-graph** when small number of constraints was given. In the case of 15% and 20% of constraints, there are no significant differences between the results. Intuitively, there is no need to use raw data, when we have an access to a large number of labeled examples. However, when our knowledge is limited, then any information (including pairwise distances) is of great importance.

Appendix C. Selection of trade-off parameter τ

To investigate the influence of parameter τ on the clustering results, we considered two scenarios. In the first one, we generated a given fraction of correct pairwise constraints (from $0.05N$ to $0.2N$), while in the second case we considered erroneous constraints. More precisely, in the second scenario we randomly selected $0.1N$ of pairwise constraints and assigned incorrect labels to a fixed percentage of them (0%, 5%, 10%, 15%, 20% of misspecified constraints). We ran **d-graph** with $\tau \in \{0.25, 0.5, 1, 2, 4\}$.

Figure C.15 presents mean ranks using different values of τ calculated in each scenario. At first glance, higher values of τ are more profitable in



(a) (b)

Figure C.15: Mean ranks of **d-graph** with different weight τ attached to labeled data calculated on UCI data sets. Figure 15(a) presents a noiseless case, while in Figure 15(b) a fixed fraction of constraints were mislabeled.

the noiseless case, Figure 15(a). However, this is not the case when only 5% of constraints are given. This effect may follow from the fact that such a small number of constraints is not representative for the whole data set and high values of τ make data imbalanced (small unrepresentative subset of data has too high weight compared to the remaining data). An opposite situation can be observed for erroneous constraints, see Figure 15(b). More precisely, high weight attached to labeled data positively influences the clustering results using correct labels. On the other hand, when at least 10% of constraints are mislabeled, then higher values of τ lead to higher errors. In consequence, the choice of τ is a double-edged sword – it allows to better exploit the information about labeled data, but may be risky in the case of noisy constraints.

Acknowledgement

The authors thank Yunali Pei, Pengjiang Qian and Hong Zeng for sharing their codes implementing semi-supervised versions of discriminative clustering, spectral clustering and maximum margin clustering.

This work was partially supported by the National Science Centre (Poland) grant no. 2016/21/D/ST6/00980 and grant no. 2015/19/B/ST6/01819.

References

[1] Asafi S, Cohen-Or D. Constraints as features. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Portland, OR; 2013. p. 1634–41.

[2] Bade K, Nürnberger A. Hierarchical constraints. Machine learning 2014;94(3):371–99.

[3] Basu S, Bilenko M, Mooney RJ. A probabilistic framework for semi-supervised clustering. In: Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD). Seattle, WA; 2004. p. 59–68.

[4] Basu S, Davidson I, Wagstaff K. Constrained clustering: Advances in algorithms, theory, and applications. CRC Press, 2008.

[5] Bilenko M, Basu S, Mooney RJ. Integrating constraints and metric learning in semi-supervised clustering. In: Proceedings of the twenty-first international conference on Machine learning. ACM; 2004. p. 11.

[6] Chang S, Aggarwal CC, Huang TS. Learning local semantic distances with limited supervision. In: Data Mining (ICDM), 2014 IEEE International Conference on. IEEE; 2014. p. 70–9.

[7] Czarnecki WM. Weighted tanimoto extreme learning machine with case study in drug discovery. IEEE Computational Intelligence Magazine 2015;10(3):19–29.

[8] Davis JV, Kulis B, Jain P, Sra S, Dhillon IS. Information-theoretic metric learning. In: Proceedings of the 24th international conference on Machine learning. ACM; 2007. p. 209–16.

[9] Fraley C, Raftery AE. Model-based clustering, discriminant analysis, and density estimation. Journal of the American statistical Association 2002;97(458):611–31.

[10] Hubert L, Arabie P. Comparing partitions. Journal of Classification 1985;2(1):193–218.

[11] Kamvar S, Klein D, Manning C. Spectral learning. In: Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI). Acapulco, Mexico; 2003. p. 561–6.

[12] Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY. An efficient k-means clustering algorithm: Analysis and implementation. IEEE transactions on pattern analysis and machine intelligence 2002;24(7):881–92.

[13] Kawale J, Boley D. Constrained spectral clustering using l1 regularization. In: Proceedings of the 2013 SIAM International Conference on Data Mining. SIAM; 2013. p. 103–11.

[14] Klekota J, Roth FP. Chemical substructures that enrich for biological activity. Bioinformatics 2008;24(21):2518–25.

[15] Krause A, Perona P, Gomes RG. Discriminative clustering by regularized information maximization. In: Advances in neural information processing systems. 2010. p. 775–83.

[16] Lichman M. UCI machine learning repository. 2013. URL: `http://archive.ics.uci.edu/ml`.

[17] Liu DC, Nocedal J. On the limited memory bfgs method for large scale optimization. Mathematical programming 1989;45(1):503–28.

[18] Lu M, Zhao XJ, Zhang L, Li FZ. Semi-supervised concept factorization for document clustering. Information Sciences 2016;331:86–98.

[19] Lu Z, Leen TK. Semi-supervised learning with penalized probabilistic clustering. In: NIPS. 2004. p. 849–56.

[20] Ma X, Gao L, Yong X, Fu L. Semi-supervised clustering algorithm for community structure detection in complex networks. Physica A: Statistical Mechanics and its Applications 2010;389(1):187–97.

[21] MicrosoftResearch . Object class recognition image database. 2005. URL: `http://research.microsoft.com/en-us/projects/ObjectClassRecognition/`.

[22] Nelson B, Cohen I. Revisiting probabilistic models for clustering with pair-wise constraints. In: Proceedings of the 24th international conference on Machine learning. ACM; 2007. p. 673–80.

[23] Pei Y, Fern XZ, Tjahja TV, Rosales R. Comparing clustering with pairwise and relative constraints: A unified framework. ACM Transactions on Knowledge Discovery from Data (TKDD) 2016;11(2):22.

[24] Qian P, Jiang Y, Wang S, Su KH, Wang J, Hu L, Muzic RF. Affinity and penalty jointly constrained spectral clustering with all-compatibility, flexibility, and robustness. IEEE transactions on neural networks and learning systems 2017;28(5):1123–38.

[25] Shental N, Bar-hillel A, Hertz T, Weinshall D. Computing Gaussian mixture models with EM using equivalence constraints. In: Advances in Neural Information Processing Systems (NIPS). Vancouver, British Columbia, Canada; 2004. p. 465–72.

[26] Śmieja M, Geiger BC. Semi-supervised cross-entropy clustering with information bottleneck constraint. Information Sciences 2017;421:254–71.

[27] Śmieja M, Struski L, Tabor J. Semi-supervised model-based clustering with controlled clusters leakage. Expert Systems with Applications 2017;85:146–57.

[28] Spurek P. General split gaussian cross–entropy clustering. Expert Systems with Applications 2017;68:58–68.

[29] Spurek P, Tabor J, Byrski K. Active function cross-entropy clustering. Expert Systems with Applications 2017;72:49–66.

[30] Von Luxburg U. A tutorial on spectral clustering. Statistics and computing 2007;17(4):395–416.

[31] Wagstaff K, Cardie C, Rogers S, Schrödl S, et al. Constrained k-means clustering with background knowledge. In: ICML. volume 1; 2001. p. 577–84.

[32] Wang H, Nie R, Liu X, Li T. Constraint projections for semi-supervised affinity propagation. Knowledge-Based Systems 2012;36:315–21.

[33] Wang Z, Davidson I. Flexible constrained spectral clustering. In: Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD). Washington, DC; 2010. p. 563–72.

[34] Warszycki D, Mordalski S, Kristiansen K, Kafel R, Sylte I, Chilmonczyk Z, Bojarski AJ. A linear combination of pharmacophore hypotheses as a new tool in search of new active compounds–an application for 5-HT1A receptor ligands. PloS ONE 2013;8(12):e84510.

[35] Xing EP, Jordan MI, Russell SJ, Ng AY. Distance metric learning with application to clustering with side-information. In: Advances in neural information processing systems. 2003. p. 521–8.

[36] Yin X, Chen S, Hu E, Zhang D. Semi-supervised clustering with metric learning: An adaptive kernel method. Pattern Recognition 2010;43(4):1320–33.

[37] Yin X, Shu T, Huang Q. Semi-supervised fuzzy clustering with metric learning and entropy regularization. Knowledge-Based Systems 2012;35:304–11.

[38] Zeng H, Cheung Ym. Semi-supervised maximum margin clustering with pairwise constraints. IEEE Transactions on Knowledge and Data Engineering 2012;24(5):926–39.

[39] Zhang H, Lu J. Semi-supervised fuzzy clustering: A kernel-based approach. Knowledge-Based Systems 2009;22(6):477–81.

[40] Zhang W, Tang X, Yoshida T. Tesc: An approach to text classification using semi-supervised clustering. Knowledge-Based Systems 2015;75:152–60.

[41] Zhao M, Chow TW, Zhang Z, Li B. Automatic image annotation via compact graph based semi-supervised learning. Knowledge-Based Systems 2015;76:148–65.