Semi-Supervised Affinity Propagation with Soft Instance-Level Constraints

Natalia M. Arzeno and Haris Vikalo, *Member, IEEE*

Abstract—Soft-constraint semi-supervised affinity propagation (SCSSAP) adds supervision to the affinity propagation (AP) clustering algorithm without strictly enforcing instance-level constraints. Constraint violations lead to an adjustment of the AP similarity matrix at every iteration of the proposed algorithm and to addition of a penalty to the objective function. This formulation is particularly advantageous in the presence of noisy labels or noisy constraints since the penalty parameter of SCSSAP can be tuned to express our confidence in instance-level constraints. When the constraints are noiseless, SCSSAP outperforms unsupervised AP and performs at least as well as the previously proposed semi-supervised AP and constrained expectation maximization. In the presence of label and constraint noise, SCSSAP results in a more accurate clustering than either of the aforementioned established algorithms. Finally, we present an extension of SCSSAP which incorporates metric learning in the optimization objective and can further improve the performance of clustering.

Index Terms—Clustering algorithms, graph algorithms, affinity propagation, semi-supervised learning, noisy pairwise constraints

1 Introduction

Affinity propagation (AP) is a frequently encountered clustering technique that uses similarities between data points to select the best representatives (*exemplars*) among them and assign each data point to its most suitable exemplar [1]. The algorithm automatically detects the number of exemplars (and, hence, the number of clusters), does not require that the similarities between data points are metric, and can take advantage of sparse similarities. AP is efficiently implemented as a message-passing scheme on a graph representing the data. More specifically, the AP message updates are obtained by applying the max-sum algorithm in a factor graph [2]. By adding nodes or modifying the factor node definitions, AP can be expanded to enforce an upper limit on the number of data points in a cluster [2], allow for uncertain or varying similarities [3], perform hierarchical clustering [4], enable finding subclasses within a category by allowing assignment of each exemplar to a super-exemplar [5], and more. To introduce semi-supervision in the AP clustering, Givoni and Frey [6] include additional variable nodes in the factor graph (so-called *meta-points (MTPs)*) and appropriately revise the similarity function.

In semi-supervised clustering, a subset of data labels or pairs of similar and dissimilar points are known. Knowledge of instance-level constraints—i.e., sets of pairs of data points that are similar (must-link (ML)) or dissimilar (cannot-link (CL))—is especially valuable in settings where data labels are expensive, such as those obtained by performing expensive or invasive medical procedures, or slow to obtain, such as in the case of enormous datasets. Note that partial labels can always be converted to instance-level constraints, whereas the reverse is not true. The inclusion of instance-level constraints in the formulation of the clustering problems results in higher accuracy of k-means [7], expectation-maximization (EM) [8], and AP [6] algorithms. Supervision can also be added to a clustering algorithm by modifying the similarity measure before or during the clustering [9], [10], [11], [12].

Algorithms that strictly enforce instance-level constraints assume that the provided labels or pairs of similar and dissimilar instances are correct. However, noisy labels or instance-level constraints can arise in a variety of situations including those that involve a certain level of subjectivity, scenarios where the information used for labeling is incomplete or inadequate, or when data entry errors exist [13]. In the medical domain, for instance, noisy labels can arise from subjectivity in situations where finding labels entails a qualitative ranking, such as in the case of determining disease severity or a disability outcome score. Furthermore, noisy labels may arise from incomplete information when established using a diagnosis where not all of the informative tests have been performed [13]. Web data, such as user-provided or search-based image and video tags, is also often plagued by noisy labels [14], [15]. The effect of noisy labels can be attenuated by using filtering techniques, optimization over both predicted soft labels and given hard labels, label normalization or tuning, noise modeling and feature extraction [13], [15], [16], [17], [18]. However, these methods assume that the labels for at least some of the data are known and do not consider the scenario where unlabeled data is related only by instance-level constraints.

Previous work on semi-supervised clustering with affinity propagation includes algorithms that strictly enforce instance-level constraints by insisting that data points with must-link constraints belong to the same cluster [6], [19].

• *The authors are with the Department of Electrical and Computer Engineering, the University of Texas at Austin, Austin, TX 78712. E-mail: narzeno@utexas.edu, hvikalo@ece.utexas.edu.*

Manuscript received 6 Dec. 2013; revised 20 June 2014; accepted 24 Aug. 2014. Date of publication 18 Sept. 2014; date of current version 4 Apr. 2015. Recommended for acceptance by M. A. Carreira-Perpinan.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2014.2359454

Some of the AP hard constraints have been removed in [19], [20], by means of bypassing the AP requirement that a point acting as an exemplar (cluster center) for others also acts as its own exemplar. However, the algorithms in [19], [20] are not designed to be used in the presence of noisy constraints and, in fact, still strictly enforce instance-level constraints. In the current paper, we describe a novel semi-supervised AP clustering algorithm where a confidence is assigned to the set of instance-level constraints; the constraints then need not be strictly enforced during the clustering. The paper is organized as follows. The affinity propagation algorithm with extensions including softening constraints or added supervision are presented in Section 2. The new algorithm, soft-constraint semi-supervised affinity propagation (SCSSAP), is derived in Section 3. The clustering datasets, evaluation metrics and results are described in Section 4. An extension of SCSSAP that employs metric learning is presented in Section 5. Section 6 concludes the work.

2 Prior Work: Affinity Propagation and Extensions

In affinity propagation [1], exemplars (i.e., representatives of the clusters) are selected after iterative exchange of messages between the nodes of a graph that represents data points (instances) being clustered. The chosen exemplars are data instances themselves, and each instance is assigned to one of the exemplars. Hence, in the graphical representation, clusters correspond to subgraphs that are spanned by edges connecting instances and their exemplars. The nodes of the graph are related by a pre-defined similarity measure such as the negative euclidean distance or the Pearson correlation coefficient. The similarities between the nodes, conveniently organized in a matrix, do not need to satisfy symmetry or the triangle inequality. The diagonal components of the similarity matrix, referred to as preferences, are typically set to the median value of the similarity between instances. In AP, the number of clusters need not be prespecified but increasing (decreasing) the preferences will result in a higher (lower) number of clusters. Once the similarity matrix has been defined, two messages are exchanged between the nodes of the graph: responsibility and availability. The responsibility ρ_{ij} indicates how well-suited node j is to be the exemplar for node i, while the availability α_{ij} reflects how appropriate it would be for node i to choose node j as its exemplar. These messages, derived from a max-sum algorithm in a factor graph [2], aim to maximize the sum of similarities between nodes and their exemplars. In a factor graph, binary variable nodes indicate whether one node is an exemplar of another. The factor nodes enforce two sets of clustering constraints: (a) each node must have exactly one exemplar (single membership), and (b) if a node serves as an exemplar to another node then it must serve as an exemplar to itself (self-selection).

In order to avoid oscillating solutions, a damping parameter μ is often incorporated in the message updates such that the new message is μ times the old message plus $1-\mu$ times the prescribed message update. Affinity propagation has several advantages over other clustering algorithms since it does not require a pre-specified number of clusters, can be formulated to take advantage of sparsity in the similarities, and does not require multiple initializations with varying initial cluster centers to find the clustering solution.

2.1 AP Extensions

The previously mentioned self-selection constraint may be relaxed by either introducing a penalty to the objective if the constraint is violated [20] or preventing each instance from choosing itself as an exemplar [19]. The soft-constraint AP can be extended to semi-supervised clustering [19] when a subset of data labels is available. Each labeled instance is assigned to a macro-node for its class while the similarity between an unlabeled instance and a macro-node is defined as the maximum similarity between the unlabeled instance and all instances assigned to the macro-node.

A natural extension of the semi-supervised AP algorithm from labeled data to instance-level constraints is obtained by introducing meta-points. The semi-supervised AP formulation in [6] enforces additional cannot-link constraints using factor nodes between meta-points that drive the objective to negative infinity when the cannot-link constraints are violated. More specifically, one meta-point is introduced for each must-link group and for each instance in a cannot-link constraint that is not also in a must-link group. The similarity of the ith instance and the mth meta-point MTP_m is given by

$$s(i, MTP_m) = \begin{cases} 0 & \text{if } i \in P_m, \\ \max\limits_{j \in P_m} s(i,j) & \text{otherwise,} \end{cases} \tag{1}$$

where P_m denotes the set of data points associated with MTP_m and $s(i,j) \leq 0$. Note that instances can choose a meta-point as an exemplar but a meta-point cannot choose other meta-points as exemplars. In fact, each instance in a must-link group will necessarily choose the meta-point associated with it as its exemplar. The cannot-link constraints are enforced by adding factor nodes connected to the meta-point such that if x has MTP_i as an exemplar, y has MTP_j as an exemplar, and (x,y) have a cannot-link constraint between them, then the exemplar for MTP_i cannot be the same as the exemplar for MTP_j. In this example, the cannot-link factor node between MTP_i and MTP_j is $-\infty$ if the two meta-points have the same exemplar, and is 0 otherwise. Following the addition of the cannot-link factor nodes, updates for the responsibilities need to be modified. The new updates can be interpreted as a change in the similarity for the meta-points that are connected by the cannot-link constraints. Semi-supervised AP outperformed both AP and constrained expectation maximization in image segmentation tasks [6].

More recently, semi-supervised AP methods where the clustering is preceded by supervision have been proposed. Wang et al. [21] use instance-level constraints to guide the search for a lower-dimensional projection in which AP is performed. Zhu et al. [22] propose a semi-supervised AP algorithm for networks where the supervision is facilitated by an appropriate construction of the similarity measure. In particular, the similarity between instances is set to 1 for must-link pairs and 0 for cannot-link pairs, while the similarity of pairs without constraints is determined based on

their relationships with other objects [23]. A fast AP clustering is then performed using the pre-defined similarities. Incremental AP clustering (I-APC) [24] and incremental and decremental AP (ID-AP) [25] incorporate supervision into AP via an iterative procedure that runs AP until convergence and updates the labeled data set based on associations between labeled and unlabeled instances. The similarity measure is updated once the new labeled data set is determined. In ID-AP, the negative euclidean distance is used as the basic similarity measure. The similarity between data instances in must-link constraints is set to zero before starting AP iterations, thus forcing similarly-labeled instances to be in the same cluster. Cannot-link constraints are enforced by setting the similarity between points in the constraint set to the smallest value from the set of similarities.

As implied by the discussion in this section, there exist no prior AP-based clustering scheme that directly imposes soft constraints on pairs of data instances. Since the use of meta-points and macro-nodes strictly enforces must-link constraints, semi-supervised AP which relies on those concepts [6], [19] cannot be readily modified to allow for softening of the instance-level constraints. Moreover, since the instance level constraints in [22] affect only the similarity metric while the actual AP algorithm remains unsupervised, any softening of the constraints in that scheme would have to be facilitated by modifying the initial similarity metric (which then remains unaltered in the AP iterations). Finally, while the semi-supervised AP algorithm in [21] does not strictly enforce instance-level constraints, imposing "softness" of the constraints in that framework appears to be challenging.

3 Soft-Constraint Semi-Supervised Affinity Propagation Algorithm

In this section we derive a new algorithm, soft-constraint semi-supervised affinity propagation, that incorporates pairwise constraints into the AP framework. As in AP, data points are related by a pre-determined similarity measure such as the negative euclidean distance. The instance-level constraints, pairs of points that are similar or dissimilar, are assumed to be known whether from partial labels or known relationships between pairs of points. The derivation of SCSSAP follows the AP message-passing framework on a factor graph [2], where pairwise constraints are incorporated into the AP framework by introducing factor nodes for each must-link and cannot-link constraint. A penalty is incurred when a constraint is violated, with potentially different penalties imposed on must-link and cannot-link constraints. Since clusterings that violate constraints are not prohibited, the penalty can express a confidence in the constraints. In the binary AP framework, variable node $c_{ij} = 1$ if the jth data instance is the exemplar for the ith one and 0 otherwise, and factor nodes enforce the constraints.

Fig. 1 illustrates an example of the connections between variable nodes and factor nodes in a segment of the factor graph. In this example, the pair of points (i, k) is in the cannot-link constraint set $\mathcal{C}$ and the pair (i, m) is in the must-link constraint set $\mathcal{M}$.

Factor nodes impose constraints that naturally arise in the clustering problems. For instance, $I_i(c_{i1,\dots,iN})$ restrict

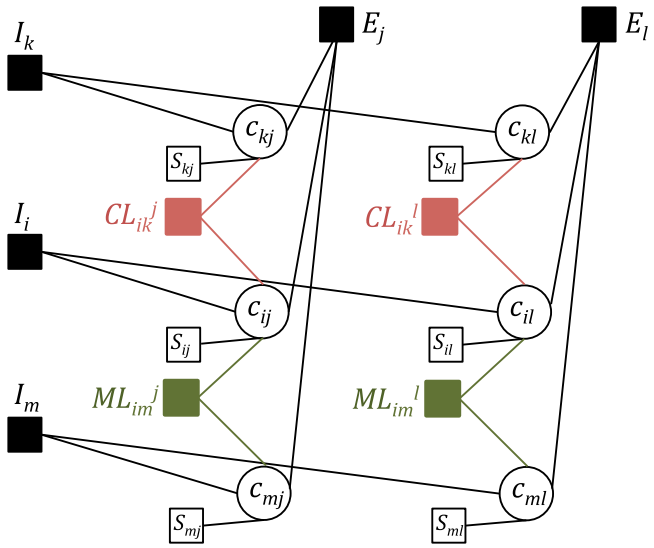

Fig. 1. Soft-constraint semi-supervised affinity propagation. The cannot-link and must-link factor nodes are not present in the classical AP formulation. In this graph, the pair (i, k) is in the set of cannot-link constraints and the pair (i, m) is in the set of must-link constraints, $i, j, k, l, m \in \{1, 2, \dots, N\}$, where N denotes the number of instances. Here S_{ij} is the similarity between ith and jth instance, and c_{ij} indicates whether the jth instance is the exemplar for the ith instance.

nodes to only have one exemplar,

$$I_i(c_{i1,\dots,iN}) = \begin{cases} -\infty & \text{if } \sum_j c_{ij} \neq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Furthermore, $E_j(c_{1j,\dots,Nj})$ enforces self-selection, i.e., if instance j is an exemplar for any other instance then it must be its own exemplar,

$$E_j(c_{1j,\dots,Nj}) = \begin{cases} -\infty & \text{if } c_{jj} = 0 \text{ and } \sum_i c_{ij} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The must-link and cannot-link factor nodes introduce penalties whenever the instance-level constraints are violated,

$$ML_{im}^{j}(c_{ij}, c_{mj}) = \begin{cases} -q_m & \text{if } c_{ij} \neq c_{mj} \quad \text{for } (i,m) \in \mathcal{M} \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

$$CL_{ik}^{j}(c_{ij}, c_{kj}) = \begin{cases} -q_c & \text{if } c_{ij} = 1 \text{ and } c_{ij} = c_{kj} \text{ for} (i,k) \in \mathcal{C} \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where $q_m \geq 0$ and $q_c \geq 0$. In particular, for a clustering specified by the set of variables $\mathbf{c} = \{c_{11}, c_{12}, \dots, c_{NN}\}$, the max-sum objective of SCSSAP is

$$\begin{aligned} \arg\max_{\mathbf{c}} & \sum_{i,j} S_{ij}(c_{ij}) + \sum_i I_i(c_{i1}, \dots, c_{iN}) \\ & + \sum_j E_j(c_{1j}, \dots, c_{Nj}) + \sum_{(i,k)\in\mathcal{C}} \sum_j CL_{ik}^{j} \\ & + \sum_{(i,m)\in\mathcal{M}} \sum_j ML_{im}^{j}, \end{aligned} \tag{6}$$

where $\mathcal{C}$ denotes the set of instance pairs with cannot-link constraints and $\mathcal{M}$ is the set of instance pairs with must-link constraints. The similarity factor nodes S_{ij} ensure that

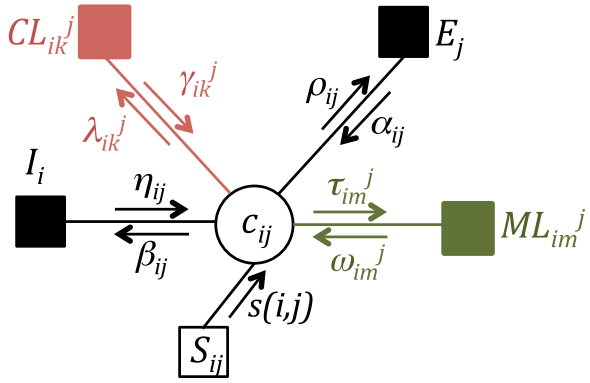

Fig. 2. Messages in soft-constraint semi-supervised affinity propagation. The α, ρ, η, and β messages are as same as those in the classical AP derived for the binary factor graph. Here $i, j, k, l, m \in \{1, 2, \ldots, N\}$, where N is the number of instances.

similarities contributing to the objective function (6) are only those between an instance and its exemplar,

$$S_{ij}(c_{ij}) = \begin{cases} s(i,j) & \text{if } c_{ij} = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Note that the probability of a given clustering $\mathbf{c}$ is given by

$$P[\mathbf{c}] = \frac{1}{Z} \prod_{i,j} \exp\big(S_{ij}(c_{ij})\big) \prod_{i} \exp(I_i[\mathbf{c}]) \prod_{j} \exp\big(E_j[\mathbf{c}]\big) \prod_{j,(i,k)\in\mathcal{C}} \bar{CL}_{ik}^{j} \prod_{j,(i,m)\in\mathcal{M}} \bar{ML}_{im}^{j}, \tag{8}$$

where $1/Z$ is a normalizing term, $\bar{CL}_{ik}^{j} = \exp(CL_{ik}^{j})$ and $\bar{ML}_{im}^{j} = \exp(ML_{im}^{j})$. Clearly, both $\bar{CL}_{ik}^{j}, \bar{ML}_{im}^{j} \in [0, 1]$ and hence q_m, q_c should be adjusted on a log scale in order to have a significant effect on the objective. The proposed SCSSAP framework reduces to the classical AP by setting $q_m = q_c = 0$, while the pairwise constraints can be strictly enforced by setting $q_m, q_c \to \infty$ such that a clustering configuration that violates pairwise constraints will have probability close to zero.

The SCSSAP messages between factor and variable nodes are labeled in Fig. 2. Update rules for these scalar messages are derived following the max-sum update rules [26],

$$\mathrm{m}_{f\to x}(x) = \max_{x_1,\ldots,x_M} \left[f(x, x_1, \ldots, x_M) + \sum_{i\in \mathrm{ne}(f)\setminus x} \mathrm{m}_{x_i\to f}(x_i) \right]$$
$$\mathrm{m}_{x\to f}(x) = \sum_{l\in \mathrm{ne}(x)\setminus x} \mathrm{m}_{f_l\to x}(x),$$

where $\mathrm{m}_{f\to x}$ and $\mathrm{m}_{x\to f}$ are the messages being passed between the factor node f and the variable node x, and $\mathrm{ne}(f)\setminus x$ indicates the neighborhood of node f excluding node x.

As in the factor graph derivation of unsupervised AP [2], each message is derived for both values of the variable node (e.g., $\alpha_{ij}(b), b \in \{0, 1\}$), with the difference defined as $\alpha_{ij} = \alpha_{ij}(1) - \alpha_{ij}(0)$. The messages from the variable nodes to the factor nodes are updated as

$$\beta_{ij} = s(i,j) + \alpha_{ij} + \sum_{m:(i,m)\in\mathcal{M}} \omega_{im}^{j} + \sum_{k:(i,k)\in\mathcal{C}} \gamma_{ik}^{j}, \tag{9}$$

$$\rho_{ij} = s(i,j) + \eta_{ij} + \sum_{m:(i,m)\in\mathcal{M}} \omega_{im}^{j} + \sum_{k:(i,k)\in\mathcal{C}} \gamma_{ik}^{j}, \tag{10}$$

$$\tau_{im}^{j} = s(i,j) + \alpha_{ij} + \eta_{ij} + \sum_{\substack{l:(i,l)\in\mathcal{M} \\ l\neq m}} \omega_{il}^{j} + \sum_{\substack{k: \\ (i,k)\in\mathcal{C}}} \gamma_{ik}^{j}, \tag{11}$$

$$\lambda_{ik}^{j} = s(i,j) + \alpha_{ij} + \eta_{ij} + \sum_{\substack{m: \\ (i,m)\in\mathcal{M}}} \omega_{im}^{j} + \sum_{\substack{l:(i,l)\in\mathcal{C} \\ l\neq k}} \gamma_{il}^{j}. \tag{12}$$

Since messages from the factor nodes to the variable node only depend on the value of the factor node and the messages to the factor node, the update equations for η_{ij} and α_{ij} remain the same as in [2],

$$\eta_{ij} = -\max_{k\neq j} \beta_{ik}, \tag{13}$$

$$\alpha_{ij} = \begin{cases} \min\left[0, \rho_{jj} + \sum_{k\notin\{i,j\}} \max[\rho_{kj}, 0]\right] & \text{if } i \neq j, \\ \sum_{k\neq j} \max[\rho_{kj}, 0] & \text{if } i = j. \end{cases} \tag{14}$$

The messages from the cannot-link factor nodes to the variable nodes are

$$\begin{aligned} \gamma_{ik}^{j}(1) &= \max_{c_{kj}} \left[CL_{ik}^{j}(c_{kj}, c_{ij} = 1) + \lambda_{ki}^{j}(c_{kj})\right] \\ &= \max_{c_{kj}} \left[-q_c \mathbb{1}(c_{kj} = 1) + \lambda_{ki}^{j}(c_{kj})\right], \end{aligned} \tag{15}$$

$$\begin{aligned} \gamma_{ik}^{j}(0) &= \max_{c_{kj}} \left[CL_{ik}^{j}(c_{kj}, c_{ij} = 0) + \lambda_{ki}^{j}(c_{kj})\right] \\ &= \max_{c_{kj}} \left[\lambda_{ki}^{j}(c_{kj})\right], \end{aligned} \tag{16}$$

where $\mathbb{1}$ is an indicator function, and $CL_{ik}^{j}(c_{kj}, c_{ij} = 0) = 0$ since cannot-link constraints only penalize the objective when an instance pair in the cannot-link set shares the exemplar (i.e., $(i,k) \in \mathcal{C}$ and $c_{ij} = c_{kj} = 1$). The final message $\gamma_{ik}^{j} = \gamma_{ik}^{j}(1) - \gamma_{ik}^{j}(0)$ is then

$$\begin{aligned} \gamma_{ik}^{j} &= \max_{c_{kj}} \left\{ -q_c \mathbb{1}(c_{kj} = 1) + \lambda_{ki}^{j}(c_{kj}) - \max_{c_{kj}} \left\{\lambda_{ki}^{j}(c_{kj})\right\}\right\} \\ &= \max\big\{ -q_c + \lambda_{ki}^{j}(1) - \max_{c_{kj}} \lambda_{ki}^{j}(c_{kj}), \\ &\qquad \lambda_{ki}^{j}(0) - \max_{c_{kj}} \lambda_{ki}^{j}(c_{kj})\big\} \\ &= -\min\big\{q_c + \max\big\{0, -\lambda_{ki}^{j}\big\},\ \max\big\{0, \lambda_{ki}^{j}\big\}\big\}. \end{aligned} \tag{17}$$

Following similar steps, we derive the message updates from the must-link factor nodes to the variable nodes as

$$\begin{aligned} \omega_{ik}^{j}(1) &= \max_{c_{kj}} \left\{ML_{ik}^{j}(c_{kj}, c_{ij} = 1) + \tau_{ki}^{j}(c_{kj})\right\} \\ &= \max_{c_{kj}} \left\{ -q_m \mathbb{1}(c_{kj} = 0) + \tau_{ki}^{j}(c_{kj})\right\}, \end{aligned} \tag{18}$$

$$\begin{aligned}\omega_{ik}^{j}(0) &= \max_{c_{kj}}\left\{ML_{ik}^{j}(c_{kj}, c_{ij}=0)+\tau_{ki}^{j}(c_{kj})\right\}\\ &= \max_{c_{kj}}\left\{-q_m\mathbb{1}(c_{kj}=1)+\tau_{ki}^{j}(c_{kj})\right\},\end{aligned} \tag{19}$$

$$\begin{aligned}\omega_{ik}^{j} &= \max_{c_{kj}}\left\{-q_m\mathbb{1}(c_{kj}=0)+\tau_{ki}^{j}(c_{kj})\right\}\\ &\quad-\max_{c_{kj}}\left\{-q_m\mathbb{1}(c_{kj}=1)+\tau_{ki}^{j}(c_{kj})\right\}\\ &= \max\left\{\min\left\{-\tau_{ki}^{j}, -q_m\right\},\ \min\left\{\tau_{ki}^{j}, q_m\right\}\right\}.\end{aligned} \tag{20}$$

Note that the confidence in each of the pairwise constraints may additionally be tuned by multiplying q_c in eq. (17) and q_m in eq. (20) with a constraint-specific value $\in [0,1]$.

The dependence of ρ_{ij} on η_{ij} can be eliminated using eqs. (9) and (13). The new update for ρ_{ij} becomes

$$\begin{aligned}\rho_{ij} = s(i,j) &+ \sum_{m:(i,m)\in\mathcal{M}}\omega_{im}^{j}+\sum_{k:(i,k)\in\mathcal{C}}\gamma_{ik}^{j}\\ &-\max_{l\neq j}\left\{s(i,l)+\alpha_{il}+\sum_{\substack{m:\\(i,m)\in\mathcal{M}}}\omega_{im}^{l}+\sum_{\substack{k:\\(i,k)\in\mathcal{C}}}\gamma_{ik}^{l}\right\}.\end{aligned} \tag{21}$$

From eq. (10), we can also write η_{ij} in terms of ρ, s, ω, γ. Substituting this definition of η_{ij} into the equations for τ_{im}^{j} and λ_{ik}^{j} (eqs. (11) and (12)), we can simplify these messages:

$$\tau_{im}^{j} = \alpha_{ij}+\rho_{ij}-\omega_{im,}^{j} \tag{22}$$

$$\lambda_{ik}^{j} = \alpha_{ij}+\rho_{ij}-\gamma_{ik}^{j}. \tag{23}$$

The update for ρ_{ij} can further be simplified by modifying the similarity metric and introducing

$$\hat{s}(i,j) = s(i,j)+\sum_{m:(i,m)\in\mathcal{M}}\omega_{im}^{j}+\sum_{k:(i,k)\in\mathcal{C}}\gamma_{ik}^{j}. \tag{24}$$

Then the update for ρ_{ij} can be rewritten as

$$\rho_{ij} = \hat{s}(i,j)-\max_{k\neq j}\{\hat{s}(i,k)+\alpha_{ik}\}, \tag{25}$$

which is of the same form as the responsibility update in the unsupervised AP with a modified similarity metric.

Note that $q_c > 0$ (by construction) and hence, as evident from the update for γ_{ik}^{j} in eq. (17), it holds that $\gamma_{ik}^{j} \leq 0$, i.e., as expected, the cannot-link constraints can only decrease the modified similarity function.

The SCSSAP algorithm summarized as Algorithm 1 replaces the τ_{ki}^{j} and λ_{ki}^{j} messages in the updates of γ_{ik}^{j} (eq. (17)) and ω_{ik}^{j} (eq. (20)) by their definitions in eqs. (22) and (23). Therefore, the algorithm only requires update and storage of two messages in addition to the availabilities α and responsibilities ρ. The damping parameter $\mu \in [0.5, 1)$ is also explicitly employed in Algorithm 1. The damping parameter aids the convergence of AP in the case of oscillating solutions.

Algorithm 1. Soft-Constraint Semi-Supervised Affinity Propagation

Initialize: $\mathbf{t=1}$, $\alpha_{ij}^{(0)}=0, \rho_{ij}^{(0)}=0, \gamma_{ik}^{j(0)}=0, \omega_{ik}^{j(0)}=0$ **for** $i,j,k \in \{1,2,\ldots,N\}$

while termination criteria not met do

$\gamma_{ik}^{j(t)} = -\min\{q_c+\max\ \{0, -(\alpha_{kj}^{(t-1)}+\rho_{kj}^{(t-1)}-\gamma_{ki}^{j(t-1)})\}, \max\{0, \alpha_{kj}^{(t-1)}+\rho_{kj}^{(t-1)}-\gamma_{ki}^{j(t-1)}\}\}$

$\omega_{ik}^{j(t)} = \max\Big\{\min\{-q_m, -(\alpha_{kj}^{(t-1)}+\rho_{kj}^{(t-1)}-\omega_{ki}^{j(t-1)})\}\ ,\ \min\{q_m, \alpha_{kj}^{(t-1)}+\rho_{kj}^{(t-1)}-\omega_{ki}^{j(t-1)}\}\}$

$\hat{s}(i,j)^{(t)} = s(i,j)+\sum_{m:(i,m)\in\mathcal{M}}\omega_{im}^{j(t)}+\sum_{k:(i,k)\in\mathcal{C}}\gamma_{ik}^{j(t)}$

$\alpha_{ij}^{(t)} = \mu\alpha_{ij}^{(t-1)}+(1-\mu)\min\Big\{0, \rho_{jj}^{(t-1)}+\sum_{k\notin\{i,j\}}\max\{\rho_{kj}^{(t-1)},0\}\Big\}$ if $i\neq j$

$\alpha_{jj}^{(t)} = \mu\alpha_{jj}^{(t-1)}+(1-\mu)\sum_{k\neq j}\max\{\rho_{kj}^{(t-1)},0\}$

$\rho_{ij}^{(t)} = \mu\rho_{ij}^{(t-1)}+(1-\mu)\big(\hat{s}(i,j)^{(t)}-\max_{k\neq j}\{\hat{s}(i,k)^{(t)}+\alpha_{ik}^{(t)}\}\big)$

$\mathbf{t=t+1}$,

end while

Identify exemplars c_i:

$\Psi = \{k : \alpha_{kk}+\rho_{kk} > 0\}$

$c_i = \arg\max_{k:k\in\Psi}\alpha_{ik}+\rho_{ik}$

While the availability α and responsibility ρ need to be calculated for all instance pairs, the messages γ_{ik}^{j} and ω_{lm}^{j} only need to be calculated for $(i,k)\in\mathcal{C}$, $(l,m)\in\mathcal{M}$ and $j\in\{1,\ldots,N\}$. Note that if $(i,k)\in\mathcal{C}$ then $(k,i)\in\mathcal{C}$; a similar statement can be made for the set $\mathcal{M}$ of must-link pairs. Therefore, compared to the classical AP, SCSSAP requires an additional $2N(|\mathcal{C}|+|\mathcal{M}|)$ message updates in each iteration, where $|\cdot|$ denotes the number of non-ordered pairs in the set. We may choose to terminate the iterations once a change in message values falls below a certain threshold, or after we obtain a consistent set of exemplars for a predetermined number of iterations, or upon reaching a maximum number of iterations. Instance j is identified as a self-exemplar if $\alpha_{jj}+\rho_{jj}>0$ after the iterations terminate. The exemplar for instance i is identified as $\arg\max_k \alpha_{ik}+\rho_{ik}$, where k is in the set of self-exemplars.

4 SCSSAP Evaluation

4.1 Evaluation Metrics and Datasets

To evaluate the performance of the proposed SCSSAP algorithm and compare it with existing schemes, we have used the negative squared euclidean distance as the similarity measure, $s(i,j) = -\|x_i - x_j\|^2$, and the affinity propagation damping parameter $\mu = 0.75$. The damping parameter was chosen to be the middle of the typical range $\mu \in [0.5, 1)$. The algorithms were also tested with a larger damping parameter, $\mu = 0.9$ (results not shown), however, the higher damping parameter did not change convergence properties and gave very similar results to $\mu = 0.75$. All the preferences (i.e., diagonal elements of the similarity matrix) were set to the median value of the similarities between data instances. Eight datasets from the UCI Machine Learning Repository [27] were examined: iris, wine, parkinsons [28], SPECTF heart, ionosphere, breast cancer, balance, and diabetes. We performed clustering with SCSSAP using several penalty parameters $\exp(-q_c) = \exp(-q_m) \in \{0, 0.00005, 0.1, 0.5, 0.9\}$, where $\exp(-q) = 0$ imposes an infinite penalty for violating

constraints and $\exp(-q) = 1$ is equivalent to the unsupervised affinity propagation.

For benchmarking purposes, SCSSAP is compared to the unsupervised AP, Givoni and Frey's semi-supervised AP (SSAP-G) [6], and Leone et al.'s soft-constraint AP with semi-supervision (SSAP-L) [19]. Note that the softness of the constraints in [19] is only for the AP self-selection constraint and not the instance-level constraints. Additionally, the latter algorithm only considers must-link constraints.

To extract instance-level constraints from the data in the noiseless setting, pairs of points are randomly selected to be part of the instance-level constraints. These pairs are assigned to the must-link set $\mathcal{M}$ if they share the same label or the cannot-link set $\mathcal{C}$ if they have different labels. The algorithms were tested in the presence of two types of noise. In the first type of noise, after the constraint sets $\mathcal{C}$ and $\mathcal{M}$ were selected, 5 or 10 percent of the constraint pairs in $\mathcal{C} \cup \mathcal{M}$ were moved from one set to the other. This type of noise occurs in the situation were the constraints are derived subjectively, such as when different physicians determine similarity of hospital patients, and can result in relationships such as $(i,j) \in \mathcal{M}, (i,k) \in \mathcal{M}, (j,k) \in \mathcal{C}$ when the constraints are propagated. The second type of noise examined simulates mislabeled data and does not result in contradicting pairwise constraints. The labels of 10 or 20 percent of the data instances are randomly changed before randomly selecting the pairwise constraints, simulating questionable labeling such as scoring of different stages of a progressive disease or in user-provided image tags. Note that data entry errors can result in either type of noise.

As an example of a semi-supervised clustering algorithm that is not based on AP, we also tested the performance of the constrained expectation-maximization [8] algorithm. Since constrained EM assumes prior knowledge of the number of clusters, we used the number of clusters identified by SCSSAP with an infinite penalty as an input to the constrained EM algorithm. Note that, in constrained EM, must-link constraints are enforced by introducing so-called chunklets in the formulation of the clustering problem. The chunklets are defined as the transitive closure of the must-link constraints; data instances absent from the constraint set are in a chunklet of size one. The chunklets are then treated as data points weighed by their cardinality. To incorporate the cannot-link constraints, the joint distribution of data instances and labels conditioned on the constraints is described using a Markov network.

Performance of the algorithms is quantified using the modified Rand index. The original Rand index, a measure of overall clustering accuracy, gives the percentage of instance pairs that are correctly classified as being in either the same cluster or different clusters. More specifically, let c_i be the label of instance i and $\hat{c}_i$ be the exemplar or a cluster assigned to instance i by the clustering algorithm. Then,

$$\text{Rand} = \frac{\sum_{i>j} \mathbb{1}\big(\mathbb{1}(c_i = c_j) = \mathbb{1}(\hat{c}_i = \hat{c}_j)\big)}{\text{total number of instance pairs}}. \tag{26}$$

For n data instances, there are $0.5n(n-1)$ instance pairs. The modified Rand index [6] weighs point pairs that are in the same cluster and those that are in different clusters equally,

$$\begin{aligned}\text{modRand} = &\frac{\sum_{i>j} \mathbb{1}(c_i = c_j)\mathbb{1}(\hat{c}_i = \hat{c}_j)}{2\sum_{i>j} \mathbb{1}(\hat{c}_i = \hat{c}_j)} \\ &+ \frac{\sum_{i>j} \mathbb{1}(c_i \neq c_j)\mathbb{1}(\hat{c}_i \neq \hat{c}_j)}{2\sum_{i>j} \mathbb{1}(\hat{c}_i \neq \hat{c}_j)}.\end{aligned} \tag{27}$$

Typically, clustering algorithms correctly separate the majority of instance pairs that should indeed belong to different clusters. This may result in a misleadingly inflated Rand index, especially when the number of clusters is large. In the modified Rand index, accurate predictions that pairs of points should be in different clusters contribute to no more than half of the accuracy measure, while the correct predictions that pairs of points belong to the same cluster account for the rest (this means that the latter carries a higher weight than in the Rand index as soon as there are more than two clusters). It should be noted that, as with the Rand index, the modified Rand index is inflated when applied to assess algorithms that produce a high number of clusters.

4.2 SCSSAP Results

Overall, the results for various data sets consistently demonstrate that the proposed SCSSAP algorithm performs at least as well as SSAP-G and SSAP-L (jointly referred to as SSAP), with significant improvements in the clustering accuracy when the constraint noise is present. Note that each point in the graphs shown in this section corresponds to the average of 20 random sets of constraints.

In the noiseless case (the results shown in Fig. 3), a very large SCSSAP penalty parameter ($\exp(-q) = 0$) leads to the most accurate clustering performance. Depending on the dataset, SCSSAP either performs comparably to SSAP (iris, parkinsons, breast cancer, diabetes) or outperforms SSAP over a wide range of explored constraints. As expected, a small SCSSAP penalty ($\exp(-q) = 0.9$) typically results in clustering accuracy similar to that of unsupervised AP, although a notable improvement in clustering is evident in the balance and iris datasets even when the SCSSAP penalty is small.

The advantage of allowing the constraints to be violated (and penalizing the clustering objective when such violations take place) instead of strictly enforcing the constraints become evident in the presence of constraint and label noise (see Figs. 4 and 5). By constructing metapoints, SSAP ensures that must-link constraints are satisfied in the final cluster assignments. In the case of contradicting constraints, some of the cannot-link constraints are ignored because a cannot-link factor node cannot point to a single metapoint. Noisy constraints greatly decrease the accuracy of SSAP and, in fact, often lead to a worse clustering solutions than unsupervised AP. The performance of SCSSAP may also deteriorate in the presence of noisy constraints and may result in inferior performance compared to unsupervised AP in scenarios where the penalty parameter is very large and the noisy constraints are numerous. However, in all of the datasets we studied, SCSSAP provides more accurate clustering than AP for some penalty parameter. Moreover, for most datasets, SCSSAP can overcome noisy constraints and lead to clustering with a modified

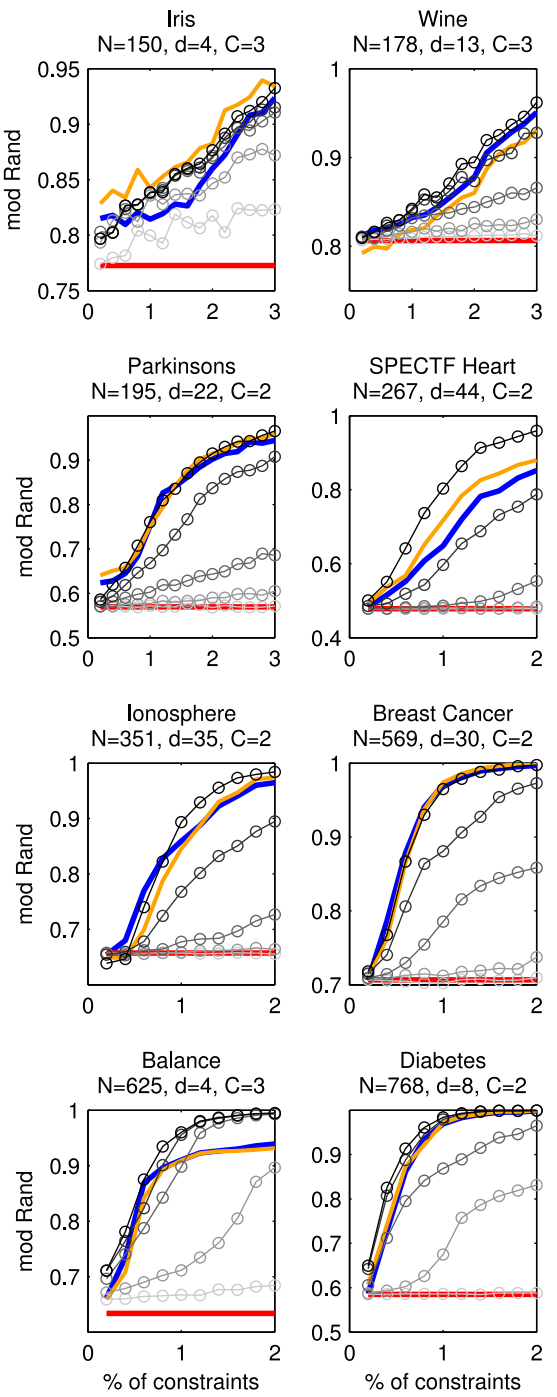

Fig. 3. Modified Rand index for SCSSAP (o), Givoni and Frey's SSAP (blue), Leone et al.'s SSAP (orange), and unsupervised AP (red). The darkness of the SCSSAP curve indicates magnitude of the penalty parameter, where the darkest curve is for $\exp(-q) = 0$.

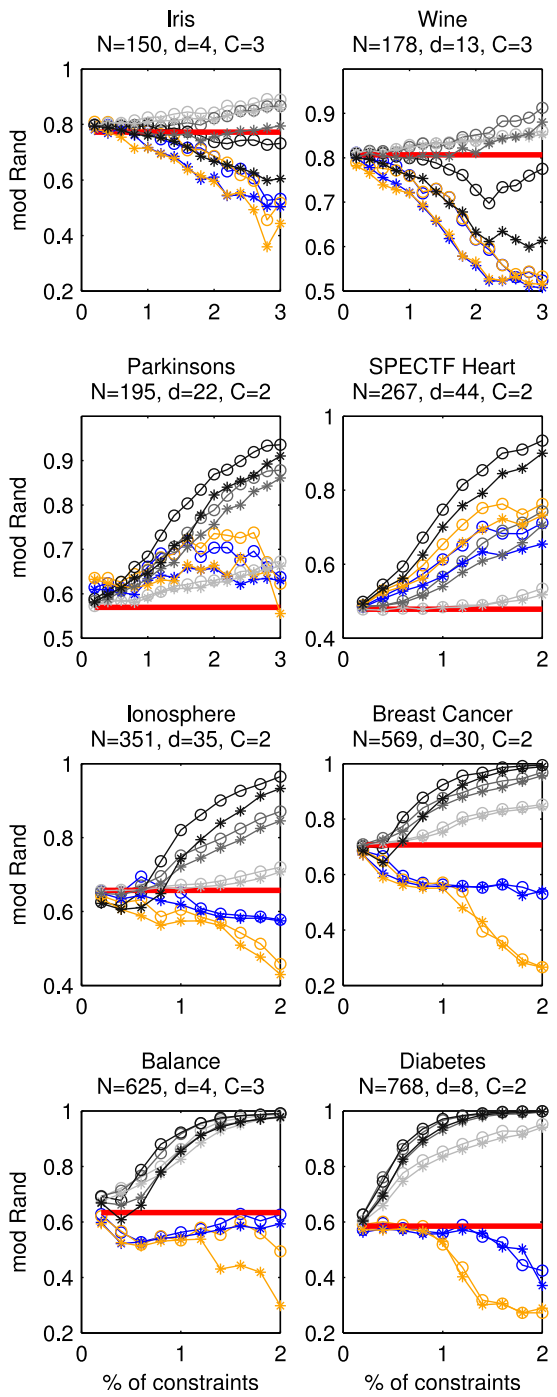

Fig. 4. Modified Rand index for SCSSAP, Givoni and Frey's SSAP (blue), Leone et al.'s SSAP (orange), and unsupervised AP (red) in the presence of 5 percent (o) and 10 percent ($*$) noisy constraints. The darkness of the SCSSAP curve indicates the magnitude of the penalty parameter, where the darkest curve is for $\exp(-q) = 0$.

Rand index closer to noiseless SCSSAP than to unsupervised AP.

Unlike the scenario where the noise is added directly to constraints, the constraints derived from noisy labels will not create contradictions in the transitive closures of the must-link and cannot-link sets. In situations where strong supervision (imposed by choosing very large values of penalty parameters) is beneficial to the clustering results, performances of SCSSAP and SSAP-G are comparable. This is reflected in the results for parkinsons, ionosphere, balance, and diabetes datasets shown in Fig. 5. However, if strict supervision is detrimental for the clustering performance (i.e., if one should use small values of penalty parameters), SCSSAP outperforms both SSAP and unsupervised AP. This is illustrated with the results for iris and wine datasets, and breast cancer with 20 percent label noise shown in Fig. 5.

Not only does SCSSAP outperform SSAP and unsupervised AP, it also clusters more accurately than constrained expectation-maximization (EM, see Fig. 6). Although the performances of constrained EM and SCSSAP are comparable

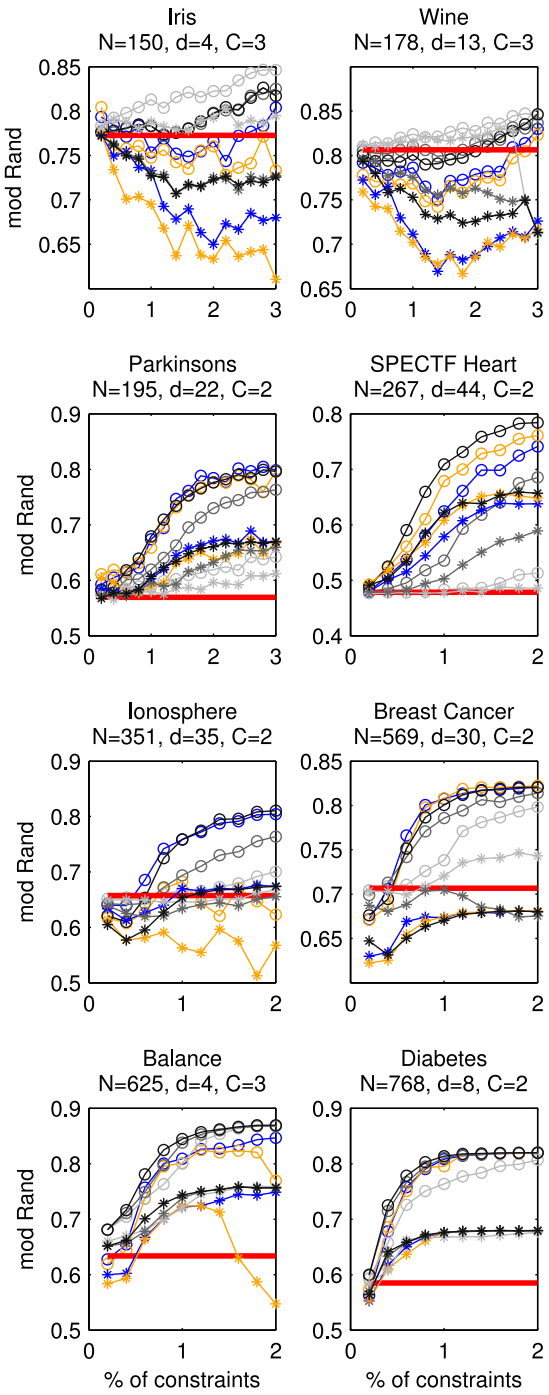

Fig. 5. Modified Rand index for SCSSAP, Givoni and Frey's SSAP (blue), Leone et al.'s SSAP (orange), and unsupervised AP (red) in the presence of 10% (o) and 20% (∗) noisy labels. The darkness of the SCSSAP curve indicates the magnitude of the penalty parameter, where the darkest curve is for $\exp(-q) = 0$.

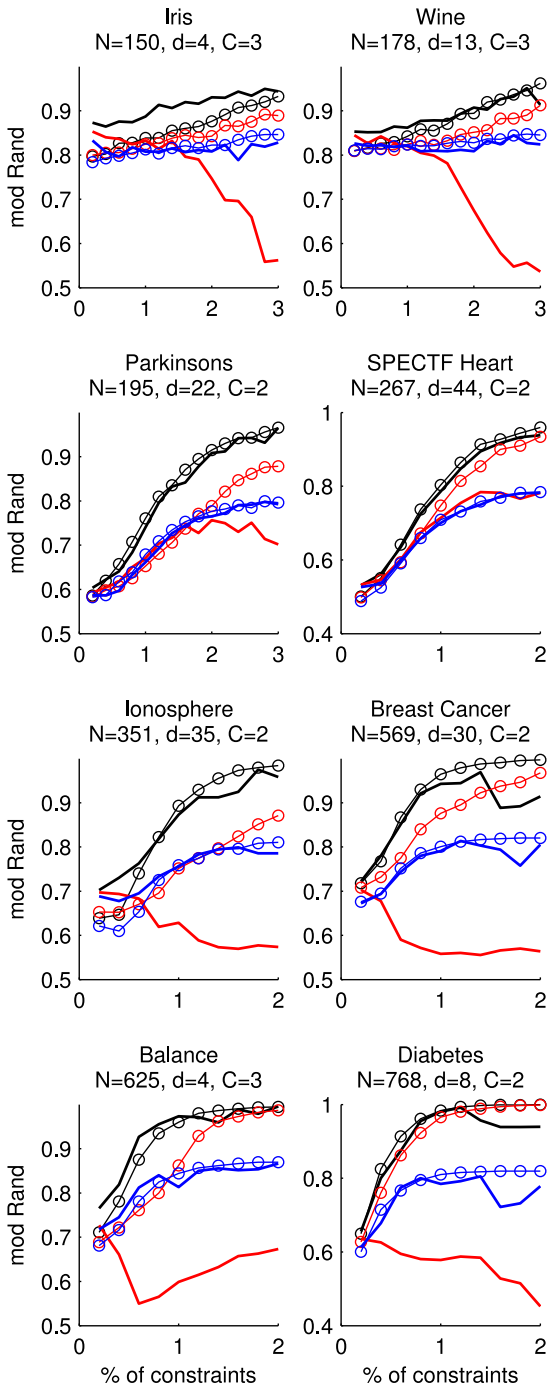

Fig. 6. SCSSAP (circles) and constrained EM (lines) modified Rand index for datasets without noise (black), 5 percent constraint noise (red), and 10 percent label noise (blue).

for noise-free constraints, SCSSAP is more accurate for all the tested datasets in the scenario where noise is added to the constraints.

5 SCSSAP with Metric Learning

Learning a global or local (cluster-specific) pseudometric prior to or during clustering can greatly increase clustering accuracy [9], [10], [11], [29] (note that, as in much of the metric learning literature, we will use the terms metric and pseudometric interchangeably). A pseudometric $d(x, y)$ satisfies the following: $d(x, y) \geq 0$, $d(x, y) = d(y, x)$, and $d(x, y) + d(y, z) \geq d(x, z)$. In this section, we focus on the Mahalanobis distance, where $d_A(x, y) = \|x - y\|_A = \sqrt{(x - y)^T A(x - y)}$. For $d_A(x, y)$ to be a metric, A must be positive semidefinite. Most of the work described in this section aims to find the Mahalanobis matrix A, where the feature selection or feature weighing problems constrain the potential matrices to those of diagonal form.

5.1 Related Work: Learning a Similarity Metric

In the early work on semi-supervised metric learning, Xing et al. [9] infer a global Mahalanobis metric by minimizing the squared distance over similar points subject to a minimum distance over dissimilar points. While [9] relies on must-link and cannot-link pairs, relevant components analysis (RCA) [11] learns the Mahalanobis matrix by examining the covariance of chunklets, groups of instances that are known to belong to the same class. This method yields similar clustering accuracy to [9], while finding the solution in a single step instead of requiring gradient descent [11]. Discriminant component analysis (DCA) [30] generalizes RCA by including dissimilarity constraints and aims to both minimize the variance of the data within the chunklets and maximize the variance of the data between chunklets that contain dissimilar instances. In [31], the DCA objective is changed from a ratio of determinants to a ratio of distances, expressed as traces, subject to an orthogonality constraint which prevents degenerate solutions.

The pseudo-metric online learning algorithm (POLA) [12] incrementally learns a metric and a threshold, and decides that a pair of points is dissimilar if the distance between them exceeds the threshold. The metric and threshold are updated using the feedback about correct classifications by minimizing the loss function and ensuring a positive semidefinite metric. In an information theoretic approach to metric learning [32], [33], the LogDet divergence between a preselected Mahalanobis distance function evaluated using prior information and the Mahalanobis matrix being learned is minimized subject to the constraints on distances between pairs of instances. In a Bayesian approach to metric learning [34], the Mahalanobis matrix is assumed to be a weighted combination of the top eigenvectors of the data, and the posterior distribution of the weights is learned by a variational method through EM-like iterations.

Other algorithms simultaneously learn a metric while clustering in either unsupervised or supervised settings [10], [35], [36], [37]. Bilenko et al. [10] learn cluster-specific Mahalanobis matrices in a k-means clustering framework through an EM procedure by alternating between assigning clusters (E-step) and updating centroids and performing metric learning (M-step). There, constraint violations incur a penalty weighted by the distance between points violating the constraints. In [35], the metric is updated using instance-level constraints but the clustering steps are unsupervised. The similarity metric in AP, the clustering method used, is gradually adjusted by modifying the weights for each pair of points in the constraint set based on how correctly they were classified. Adaptive metric learning (AML) [36], an unsupervised algorithm, is formulated as the maximization of the distance between clusters in a lower dimensional embedding. Locally adaptive clustering (LAC) [37], assigns a weight vector to each cluster, which is similar to learning a diagonal Mahalanobis matrix with unit trace. LAC alternates updating weights and centroids until convergence, which is achieved quickly due to using an exponential weighting scheme [37]. In [38], a data partitioning matrix and the cluster-specific classifiers are alternately optimized. The constrained optimization problem has a log loss function as the objective and constraints that regulate the cluster sizes and enforce the given instance-level constraints.

Sparse metrics are desirable in a multitude of applications. Metric learning algorithms that enforce sparsity often include an ℓ_1 penalty in the objective. Roth and Lange [39] use an EM framework where fuzzy labels are estimated in the E-step while the M-step employs a linear discriminant analysis (LDA) with feature selection by ℓ_1 penalty. In [40], a regression problem with constraints promoting supervised clustering and feature selection is solved. In sparse distance metric learning (SDML) [41], a sparse Mahalanobis matrix is learned by minimizing the sum of the LogDet divergence between a given matrix with the a priori distribution, the ℓ_1 norm of the off-diagonal elements of the matrix, and a loss function defined over the instance-level constraints. The semi-supervised sparse metric learning algorithm (S^3ML) [42] aims to minimize the LogDet divergence between a given matrix and the desired matrix with an added ℓ_1 penalty term that promotes a sparse metric.

By focusing on the nearest neighbors, metric learning algorithms designed for kNN classifiers [29], [43], [44], [45], [46], [47] often make no parametric assumptions about the data structure [29], [44], [46]. Neighborhood components analysis (NCA) [29] seeks to maximize the expected number of correctly classified points in a leave-one-out framework which is then solved with a gradient-based optimizer. Large margin nearest neighbor metric learning (LMNN) [44] minimizes the distance between points and the desired neighbors while maximizing the margin with points belonging to other classes. As in NCA and LMNN, local distance metric learning (LDM) [45] does not assume unimodal classes. The LDM objective aims to maximize the log-likelihood of correctly predicting the classes, where the probability of a correct prediction is derived from kernel-based kNN. The maximally collapsing metric learning algorithm (MCML) [46] objective aims to collapse instances from the same class into a single point and make distance between points in different classes infinite by minimizing the KL divergence. The Laplacian regularized metric learning (LRML) [47] objective minimizes a regularization term equal to the sum of distances between instances and their designated nearest neighbors along with loss terms corresponding the instance-level constraints.

Distance metrics can also be learned when sets of similar or dissimilar points are not readily available but are described in a rather qualitative form such as "x_i is more similar to x_j than it is to x_k" [48], [49].

Depending on the formulation of the objective function, the metric learning problem can be solved by means of convex optimization [9], [35], [40], [41], [44], [46], [47], eigendecomposition [12], [30], [31], [46], iterative schemes [32], [33], [36], [38], [42], [45], or admit closed form solutions [10], [11], [37].

5.2 SCSSAP with Metric Learning Algorithm

In this section, we add feature weighings to the clustering problem by learning a pseudometric for each cluster. In order to maintain the AP objective for clustering, we focus on metric learning objectives that include the Mahalanobis distance between data instances and their corresponding cluster centers. By defining the similarity function in SCSSAP as a negative squared Mahalanobis distance and adding a regularizing function g to the SCSSAP objective in

eq. (6), we can potentially improve clustering by learning a new metric for instances in the same cluster. The objective then becomes

$$\begin{aligned}\arg \max_{\mathbf{c},A_{l_1}\ldots A_{l_N}} & \sum_{i,j} S'_{ij}(c_{ij}) + \sum_i I_i(c_{i1},\ldots,c_{iN}) \\ & + \sum_j E_j(c_{1j},\ldots,c_{Nj}) + \sum_{k:(i,k)\in\mathcal{C}} \sum_j CL^j_{ik} \\ & + \sum_{m:(i,m)\in\mathcal{M}} \sum_j ML^j_{im} - \sum_i g(A_{l_i}),\end{aligned} \tag{28}$$

where l_i is the cluster of instance i. For data vectors x_i and x_j, $S'_{ij}(c_{ij}) = s'(i,j)$ if j is the exemplar of i and 0 otherwise. We define $s'(i,j) = -0.5\|x_i - x_j\|^2_{A_{l_i}} - 0.5\|x_i - x_j\|^2_{A_{l_j}}$. We only consider diagonal matrices A and add the constraint $\mathrm{trace}(A) = 1$ to be able to interpret the resulting matrix as feature weights.

We consider two regularizing functions $\overline{g}$ and $\hat{g}$ for the objective $\arg\max_{A_{l_i}} \sum_{i,j} S'_{ij}(c_{ij}) - \sum_i g(A_{l_i}) = \arg\ \min_{A_{l_i}} \sum_i (\|x_i - c_i\|^2_{A_{l_i}} + g(A_{l_i}))$,

$$\overline{g} = -\log\det(A_{l_i}) \tag{29}$$

$$\hat{g} = h\sum_d a^{l_i}_{dd}\log\big(a^{l_i}_{dd}\big), \tag{30}$$

where $a^{l_i}_{dd}$ is the dth component of the diagonal of A_{l_i} and $h \geq 0$. The metric learning is a convex problem if the regularizing function is convex and the matrix A is positive semidefinite. Both regularizing functions satisfy the convexity constraint. Moreover, since we are only considering diagonal matrices A, the regularizing functions force the values on the diagonal to be ≥ 0 thus ensuring positive semidefiniteness. The function $\overline{g}$, which enforces the positive semidefinite constraint on A, is derived from the solution to the maximum likelihood problem where the l_i^{th} cluster is Gaussian with covariance matrix $A^{-1}_{l_i}$ [10], [11]. The function $\hat{g}$, the negative entropy of the feature weight distribution, penalizes clusters that just use a single feature. The parameter h controls how much the distribution of the weights deviates from a uniform distribution [37].

These above functions are of interest since they have closed-form solutions [10], [37]. Let $\mathcal{U}$ be the set of clusters and $u \in \mathcal{U}$ be the set of data instances in cluster u, and c_u be the exemplar for cluster u. Our metric learning objective can be rewritten as

$$\arg\min_{A_u, u\in\mathcal{U}} \sum_{u\in\mathcal{U}}\sum_{i\in u} \big(\|x_i - c_i\|^2_{A_u} + g(A_u)\big). \tag{31}$$

Let $\overline{a}^{(u)}_{dd}$ be the dth component of the diagonal of A_u when $g = \overline{g}$, $\hat{a}^{(u)}_{dd}$ be the dth component of the diagonal of A_u when $g = \hat{g}$, $|\mathcal{X}_u|$ be the number of data instances in cluster u, and x_{id} and c_{id} be the dth element of x_i and c_i respectively. Then, the closed-form expressions for these entries can be found as

$$\overline{a}^{(u)}_{dd} = |\mathcal{X}_u|\left(\sum_{i\in u}(x_{id} - c_{id})^2\right)^{-1}, \tag{32}$$

$$\hat{a}^{(u)}_{dd} = \frac{\exp\Big(-\sum_{i\in u}(x_{id} - c_{id})^2/(h|\mathcal{X}_u|)\Big)}{\sum_d \exp\Big(-\sum_{i\in u}(x_{id} - c_{id})^2/(h|\mathcal{X}_u|)\Big)}. \tag{33}$$

The closed-form solution in eq. (32) does not enforce the constraint $\mathrm{trace}(A) = 1$. Nevertheless, the alternating optimization without this constraint was tested and empirically found to perform similarly to the case where the matrix was normalized to have its trace equal 1. The results presented include the additional step of $\overline{a}^{(u)}_{dd} \leftarrow \overline{a}^{(u)}_{dd} / \sum_d \overline{a}^{(u)}_{dd}$.

The SCSSAP with metric learning objective in eq. (28) is solved by means of an alternating maximization over the parameters. SCSSAP is employed in the optimization of the exemplars $\mathbf{c} = c_{11}, c_{12}, \ldots, c_{NN}$, while the optimization over $A_{l_1}, \ldots A_{l_N}$ is performed by solving eq. (31) (see Algorithm 2). Similar to SCSSAP, SCSSAP with metric learning terminates after the list of exemplars is unchanged for a given number of iterations or a maximum number of iterations is reached.

Algorithm 2. SCSSAP with Metric Learning

Initialize: $u = \{1,\ldots,N\}, A_u = \frac{1}{D} I_{D\times D}$, $\mathcal{U} = u$
while termination criteria not met **do**
  define $s(i,j) = -0.5\|x_i - x_j\|^2_{A_{l_i}} - 0.5\|x_i - x_j\|^2_{A_{l_j}}$
    for $i, j \in \{1,\ldots,N\}$
  update clusters: run SCSSAP
  update metric: solve $a^{(u)}_{dd}$ **for** $d \in \{1,,D\}$, $u \in \mathcal{U}$
end while

5.3 Results

Algorithm 2 is evaluated on the iris, wine, parkinsons, and soybean datasets from the UCI Machine Learning Repository [27] with regularizing functions $\overline{g}$ (eq. (29)) and $\hat{g}$ (eq. (30)) with $h = 0.1$. In learning the metric with $\hat{g}$, a very large value of h will select a uniform weight vector while setting $h = 0$ will assign all the weight to a single feature [37]. Both single metrics and cluster-specific metrics are learned for each dataset.

For some datasets, such as iris and soybean, metric learning can greatly improve the results of SCSSAP (see Fig. 7), while others do not benefit much from metric learning. Metric learning aids with clustering in the wine dataset as well, providing moderate improvements. The clustering performance with $\hat{g}$ as the regularizing function could potentially be improved by finding the optimal h for each dataset. In addition to $h = 0.1$, SCSSAP with metric learning with $h = 1$ was evaluated (results not shown), but the resulting accuracy was similar to that of SCSSAP without metric learning. This indicates uniform feature weights are not optimal for clustering in the iris, wine, and soybean datasets. Although the regularizing functions do not explicitly impose sparsity, when clusters are best characterized by a subset of features, SCSSAP with cluster-specific metric learning will provide an improved clustering over SCSSAP. Sparse solutions, however, may be undesirable in the case of single metric learning. In particular, when using $\overline{g}$ as the regularizing function, if there exists one or more features d for which $\sum_i (x_{id} - c_{id}) = 0$, then the metric will be uniform over these features and 0 elsewhere. This could result in a

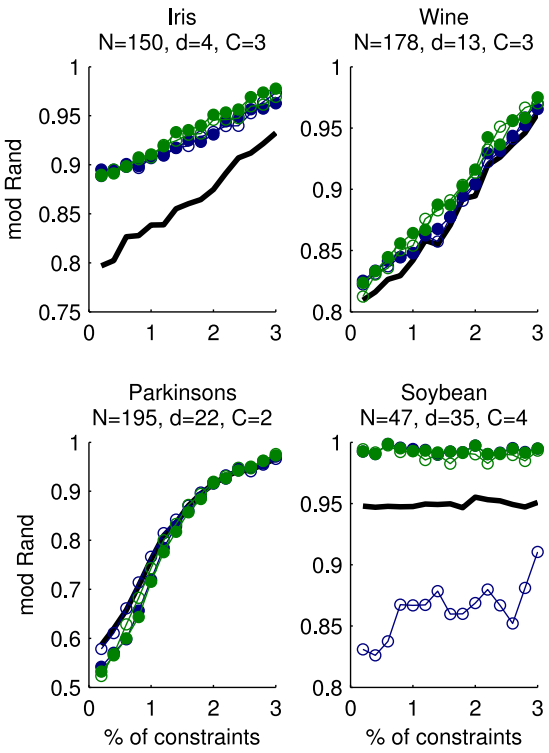

Fig. 7. SCSSAP without metric learning (black), with $\bar{g}$ in the objective (blue) and with $\hat{g}$ in the objective (green). Empty circles correspond to a single metric for the entire dataset, while filled circles correspond to cluster-specific metrics.

very sparse metric that does not capture the feature set necessary for optimal clustering and lead to deterioration of the performance of SCSSAP, as in single metric learning with $\bar{g}$ in the soybean dataset.

6 Conclusion

In this paper, a novel soft-constraint semi-supervised affinity propagation scheme is derived from a factor graph with additional factor nodes linking data instances in the constraint set. Instead of forcing must-link and cannot-link constraints to be met in the final clustering, these factor nodes allow constraints to be violated while imposing a penalty to the clustering objective. The penalty parameter can be tuned to represent a confidence level on the constraints. The algorithm follows the message updates from AP, but at each iteration the messages flowing from the instance-level constraint nodes affect the similarity between data points. In a noiseless setting, SCSSAP performs at least as well as constrained EM and semi-supervised AP, which strictly enforce instance-level constraints. In the presence of constraint noise or label noise, SCSSAP significantly outperforms both of the existing algorithms. The SCSSAP algorithm is also extended to alternately optimize the clustering and learn a global or cluster-specific metric by means of an unsupervised step with a closed form solution. Depending on the dataset, this extension can further improve the clustering performance. In conclusion, we derived a semi-supervised clustering algorithm, based on message passing, which does not strictly enforce constraints and is beneficial in a plethora of scenarios where the constraints are noisy. Moreover, we provided an extension that includes metric learning and often results in an increase in accuracy of the clustering.

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1110007 and Jack Kilby/Texas Instruments fellowship.

References

[1] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.
[2] I. E. Givoni and B. J. Frey, "A binary variable model for affinity propagation," *Neural Comput.*, vol. 21, no. 6, pp. 1589–1600, Jun. 2009.
[3] W. Li, "Clustering with uncertainties: An affinity propagation-based approach," in *Proc. 19th Int. Conf. Neural Inf. Process.—Volume Part V*, 2012, pp. 437–446.
[4] I. Givoni, C. Chung, and B. J. Frey. (2012, Feb.). "Hierarchical affinity propagation," arXiv e-print 1202.3722 [Online]. Available: http://arxiv.org/abs/1202.3722
[5] C.-D. Wang, J.-H. Lai, C. Y. Suen, and J.-Y. Zhu, "Multi-exemplar affinity propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2223–2237, Sep. 2013.
[6] I. E. Givoni and B. J. Frey, "Semi-supervised affinity propagation with instance-level constraints," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, 2009, pp. 161–168.
[7] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-means clustering with background knowledge," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 577–584.
[8] N. Shental, A. Bar-hillel, T. Hertz, and D. Weinshall, "Computing Gaussian mixture models with EM using equivalence constraints," in *Advances in Neural Information Processing Systems*, vol. 16. Cambridge, MA, USA: MIT Press, 2003.
[9] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Proc. Proc. Adv. Neural Inf. Process. Syst.*, 2002, vol. 15, pp. 505–512.
[10] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 11.
[11] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 11–18.
[12] S. Shalev-Shwartz, Y. Singer, and A. Y. Ng, "Online and batch learning of pseudo-metrics," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 94.
[13] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," *J. Artif. Intell. Res.*, vol. 11, pp. 131–167, 1999.
[14] L. S. Kennedy, S.-F. Chang, and I. V. Kozintsev, "To search or to label?: Predicting the performance of search-based automatic image classifiers," in *Proc. 8th ACM Int. Workshop Multimedia Inf. Retrieval*, 2006, pp. 249–258.
[15] W. Liu, J. Wang, and S.-F. Chang, "Robust and scalable graph-based semisupervised learning," *Proc. IEEE*, vol. 100, no. 9, pp. 2624–2638, Sep. 2012.
[16] N. D. Lawrence and B. Schölkopf, "Estimating a kernel fisher discriminant in the presence of label noise," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 306–313.
[17] M. Pechenizkiy, A. Tsymbal, S. Puuronen, and O. Pechenizkiy, "Class noise and supervised learning in medical domains: The effect of feature extraction," in *Proc. 19th IEEE Int. Symp. Comput.-Based Med. Syst.*, 2006, pp. 708–713.
[18] C. Bouveyron and S. Girard, "Robust supervised classification with mixture models: Learning from data with uncertain labels," *Pattern Recognit.*, vol. 42, no. 11, pp. 2649–2658, Nov. 2009.
[19] M. Leone, Sumedha, and M. Weigt, "Unsupervised and semi-supervised clustering by message passing: Soft-constraint affinity propagation," *Eur. Phys. J. B*, vol. 66, no. 1, pp. 125–135, 2008.
[20] M. Leone, Sumedha, and M. Weigt, "Clustering by soft-constraint affinity propagation: Applications to gene-expression data," *Bioinformatics*, vol. 23, no. 20, pp. 2708–2715, 2007.
[21] H. Wang, R. Nie, X. Liu, and T. Li, "Constraint projections for semi-supervised affinity propagation," *Knowl.-Based Syst.*, vol. 36, pp. 315–321, 2012.

[22] M. Zhu, F. Meng, and Y. Zhou, "Semisupervised clustering for networks based on fast affinity propagation," *Math. Problems Eng.*, vol. 2013, pp. 1–13, 2013.
[23] G. Jeh and J. Widom, "SimRank: A measure of structural-context similarity," in *Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, 2002, pp. 538–543.
[24] X. H. Shi, R. C. Guan, L. P. Wang, Z. L. Pei, and Y. C. Liang, "An incremental affinity propagation algorithm and its applications for text clustering," in *Proc. Int. Joint Conf. Neural Netw.*, 2009, pp. 2734–2739.
[25] C. Yang, L. Bruzzone, R. Guan, L. Lu, and Y. Liang, "Incremental and decremental affinity propagation for semisupervised clustering in multispectral images," *IEEE Trans. Geosci. Remote Sensing*, vol. 51, no. 3, pp. 1666–1679, Mar. 2013.
[26] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, Aug. 2006.
[27] K. Bache and M. Lichman. (2013). *UCI Machine Learning Repository*, School Inform. Comput. Sci., Univ. California, Irvine, CA, USA [Online]. Available: http://archive.ics.uci.edu/ml
[28] M. A. Little, P. E. McSharry, S. J. Roberts, D. A. Costello, and I. M. Moroz (2007, Jun.). Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMed. Eng. OnLine* [Online]. *6(23)*, Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1913514/
[29] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, vol. 17, pp. 513–520.
[30] S. Hoi, W. Liu, M. Lyu, and W.-Y. Ma, "Learning distance metrics with contextual constraints for image retrieval," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 2072–2078.
[31] S. Xiang, F. Nie, and C. Zhang, "Learning a Mahalanobis distance metric for data clustering and classification," *Pattern Recognit.*, vol. 41, no. 12, pp. 3600–3612, Dec. 2008.
[32] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 209–216.
[33] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon, "Metric and kernel learning using a linear transformation," *J. Mach. Learn. Res.*, vol. 13, pp. 519–547, 2012.
[34] L. Yang, R. Jin, and R. Sukthankar, "Bayesian active distance metric learning," in *Proc. 23rd Conf. Uncertainty Artif. Intell.*, 2007, pp. 442–449 [Online]. Available: http://arxiv.org/abs/1206.5283
[35] B. Conroy, Y. Xi, and P. Ramadge, "A supervisory approach to semi-supervised clustering," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Process.*, 2010, pp. 1858–1861.
[36] J. Ye, Z. Zhao, and H. Liu, "Adaptive distance metric learning for clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–7.
[37] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos. (2007, Feb.). Locally adaptive metrics for clustering high dimensional data *Data Mining Knowl. Discovery* [Online]. *14(1)*, pp. 63–97, Feb. 2007. [Online]. Available: http://link.springer.com/article/10.1007/s10618-006-0060-8
[38] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh, "Clustering complex data with group-dependent feature selection," in *Proc. 11th Eur. conf. Comput. Vis.: Part VI*, 2010, pp. 84–97.
[39] V. Roth and T. Lange, "Feature selection in clustering problems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, vol. 16.
[40] X. Shen, H.-C. Huang, and W. Pan, "Simultaneous supervised clustering and feature selection over a graph," *Biometrika*, vol. 99, pp. 899–914, Oct. 2012.
[41] G.-J. Qi, J. Tang, Z.-J. Zha, T.-S. Chua, and H.-J. Zhang, "An efficient sparse metric learning in high-dimensional space via l1-penalized log-determinant regularization," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 841–848.
[42] W. Liu, S. Ma, D. Tao, J. Liu, and P. Liu, "Semi-supervised sparse metric learning using alternating linearization optimization," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 1139–1148.
[43] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 607–616, Jun. 1996.
[44] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Jun. 2009.
[45] L. Yang, R. Jin, R. Sukthankar, and Y. Liu, "An efficient algorithm for local distance metric learning," in *Proc. 21st Nat. Conf. Artif. Intell.*, 2006, pp. 543–548.
[46] A. Globerson and S. Roweis, "Metric learning by collapsing classes," in *Advances in Neural Information Processing Systems*, vol. 18, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA, USA: MIT Press, 2006, pp. 451–458.
[47] S. Hoi, W. Liu, and S.-F. Chang, "Semi-supervised distance metric learning for collaborative image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–7.
[48] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons," in *Proc. Neural Inf. Process. Syst.*, 2003.
[49] N. Kumar and K. Kummamuru, "Semisupervised clustering with metric learning using relative comparisons," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 4, pp. 496–503, Apr. 2008.

Natalia M. Arzeno received the BS and MEng degrees in electrical engineering from the Massachusetts Institute of Technology in 2006 and 2007. She is currently working toward the PhD degree in the Electrical and Computer Engineering Department at the University of Texas at Austin. She received the National Science Foundation Graduate Research Fellowship. Her research interests include machine learning, data mining, and healthcare analytics.

Haris Vikalo received the BS degree from the University of Zagreb, Croatia, in 1995, the MS degree from Lehigh University in 1997, and the PhD degree from Stanford University in 2003, all in electrical engineering. He held a short-term appointment at Bell Laboratories, Murray Hill, NJ, in the summer of 1999. From January 2003 to July 2003, he was a postdoctoral researcher; and from July 2003 to August 2007, he was an associate scientist at the California Institute of Technology. Since September 2007, he has been with the Department of Electrical and Computer Engineering, the University of Texas at Austin, where he is currently an associate professor. He received the 2009 National Science Foundation Career Award. His research interests include signal processing, bioinformatics, machine learning, and communications. He is a member of the IEEE.