Constraints and Metric Learning in Semi-Supervised Clustering

Shashank Singh [1]

11-745 New Methods in Large-Scale Structured Learning

October 21, 2014

[1] Carnegie Mellon University, Pittsburgh, PA, USA

Two Papers

Xing et al., 2002, Distance Metric Learning, with application to Clustering with side-information

- first learn metric from labeled data ("preprocessing step"), and then cluster
- single metric for all data
- obey all constraints

Bilenko et al., 2004, Integrating Constraints and Metric Learning in Semi-Supervised Clustering

- leverage unlabeled data for metric learning by alternating clustering and metric learning
- separate metric for each cluster
- soft constraints tolerate noisy supervision

Constraints

Allow constraints of the form:

$$\ell_i = \ell_j \quad \text{and} \quad \ell_i \neq \ell_j,$$

where ℓ_i denotes the label of x_i.

Define "must-link" and "cannot-link" sets

$$\mathcal{M} := \{(i,j) : \ell_i = \ell_j \text{ is a constraint}\}$$

and

$$\mathcal{C} := \{(i,j) : \ell_i \neq \ell_j \text{ is a constraint}\}.$$

Metric Learning

Learn (pseudo)metrics of the form:

$$d_A(x, y) = \|x - y\|_A := \sqrt{(x - y)^T A (x - y)},$$

where $A \in \mathbb{R}^{d \times d}$ is positive semidefinite ($A \succeq 0$).

e.g.,

- $A = I$ gives Euclidean metric
- A diagonal reweights dimensions
- general A replaces features with linear combinations of features

Can learn nonlinear metrics via feature map

Approach

Minimize distance of pairs of points in $\mathcal{M}$, i.e.,

$$\min_A \sum_{(i,j)\in\mathcal{M}} \|x_i - x_j\|_A^2$$

subject to $A \succeq 0$.

Approach

Minimize distance of pairs of points in $\mathcal{M}$, i.e.,

$$\min_A \sum_{(i,j)\in\mathcal{M}} \|x_i - x_j\|_A^2$$

subject to $A \succeq 0$.

But $A = 0$ is a trivial solution.

Approach

Minimize distance of pairs of points in $\mathcal{M}$, i.e.,

$$\min_A \sum_{(i,j)\in\mathcal{M}} \|x_i - x_j\|_A^2$$

subject to $A \succeq 0$.

But $A = 0$ is a trivial solution.

So constrain points in $\mathcal{C}$ to be far apart.

Approach

This gives the following optimization problem:

$$\min_A \sum_{(i,j)\in\mathcal{M}} \|x_i - x_j\|_A^2$$

subject to $A \succeq 0$ and

$$\sum_{(i,j)\in\mathcal{C}} \|x_i - x_j\|_A \geq 1.$$

Approach

This gives the following optimization problem:

$$\min_A \sum_{(i,j)\in\mathcal{M}} \|x_i - x_j\|_A^2$$

subject to $A \succeq 0$ and

$$\sum_{(i,j)\in\mathcal{C}} \|x_i - x_j\|_A \geq 1.$$

Diagonal Case

Can show this is equivalent to minimizing

$$g(A) = \sum_{(i,j)\in\mathcal{M}} \|x_i - x_j\|_A^2 - \log\left(\sum_{(i,j)\in\mathcal{C}} \|x_i - x_j\|_A\right).$$

Easy to apply Newton's Method.

General Case

A has n^2 parameters, so $O(n^6)$ time to invert Hessian. So can't use Newton's Method.

General Case

A has n^2 parameters, so $O(n^6)$ time to invert Hessian. So can't use Newton's Method.

An equivalent problem is

$$\max_A \sum_{(i,j)\in\mathcal{C}} \|x_i - x_j\|_A$$

subject to $A \succeq 0$ and

$$\sum_{(i,j)\in\mathcal{M}} \|x_i - x_j\|_A^2 \leq 1.$$

use gradient descent with method of alternating projections to enforce (convex) constraints.

General Case

$$\max_A \sum_{(i,j)\in\mathcal{C}} \|x_i - x_j\|_A$$

subject to $A \succeq 0$ and

$$\sum_{(i,j)\in\mathcal{M}} \|x_i - x_j\|_A^2 \leq 1.$$

Now, constraints are convex and easy to project onto, so use gradient descent with method of alternating projections to enforce (convex) constraints.

General Case

Now, constraints are convex and easy to project onto, so use gradient descent with method of alternating projections to enforce (convex) constraints:

Iterate
  Iterate
    $A := \arg\min_{A'}\{||A' - A||_F : A' \in C_1\}$
    $A := \arg\min_{A'}\{||A' - A||_F : A' \in C_2\}$
  until A converges
  $A := A + \alpha(\nabla_A g(A))_{\perp \nabla_A f}$
until convergence

where C_1 is the constraint

$$\sum_{(i,j)\in\mathcal{M}} \|x_i - x_j\|_A^2 \leq 1.$$

and C_2 is the constraint $A \succeq 0$.

Examples of learned metrics

Learning A can be thought of as finding a linear transformation of the data $x \mapsto \sqrt{A}x$ that moves similar pairs together:

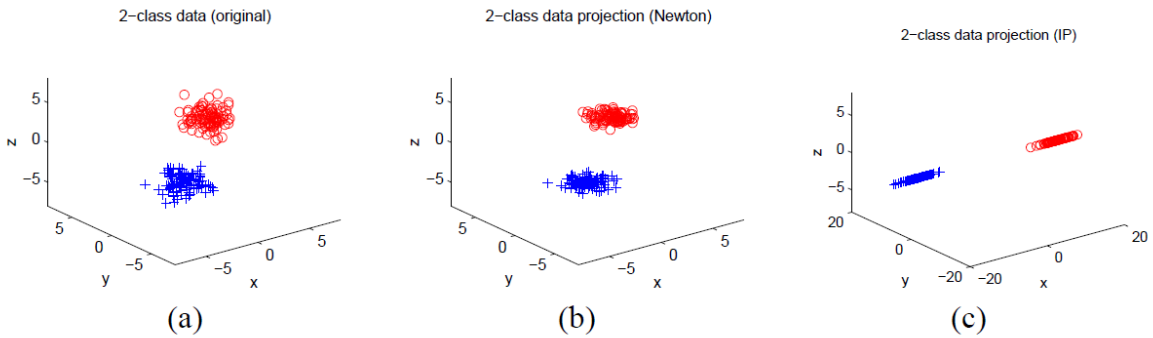


Figure 2: (a) Original data, with the different classes indicated by the different symbols (and colors, where available). (b) Rescaling of data corresponding to learned diagonal A. (c) Rescaling corresponding to full A.

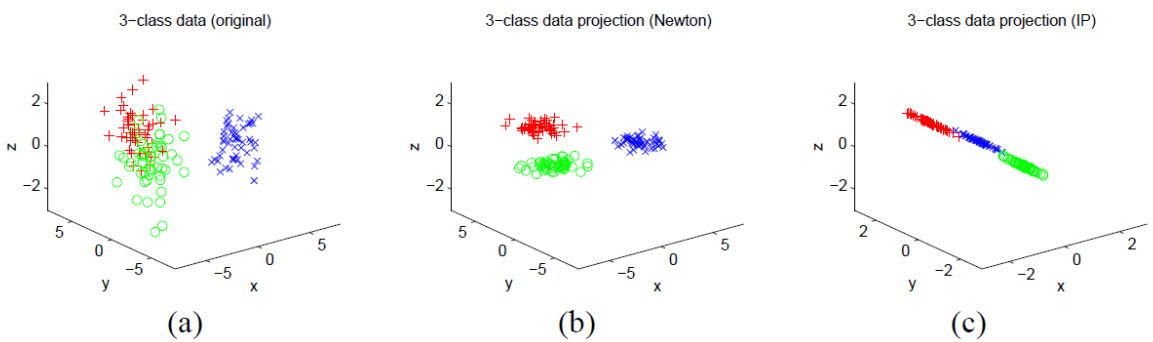
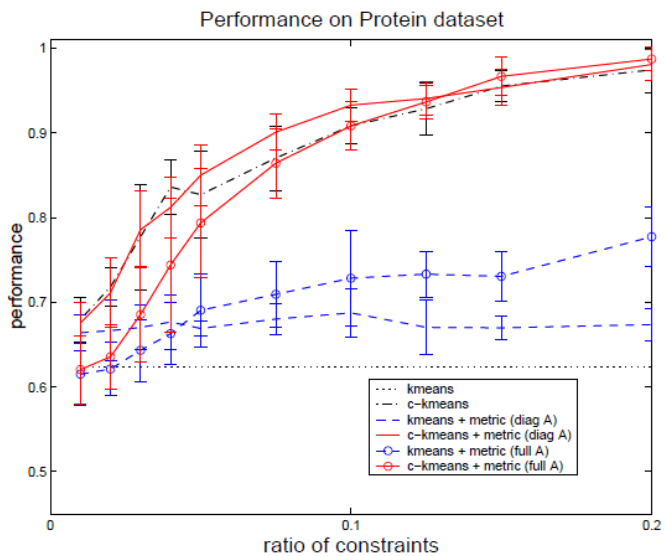


Figure 3: (a) Original data. (b) Rescaling corresponding to learned diagonal A. (c) Rescaling corresponding to full A.

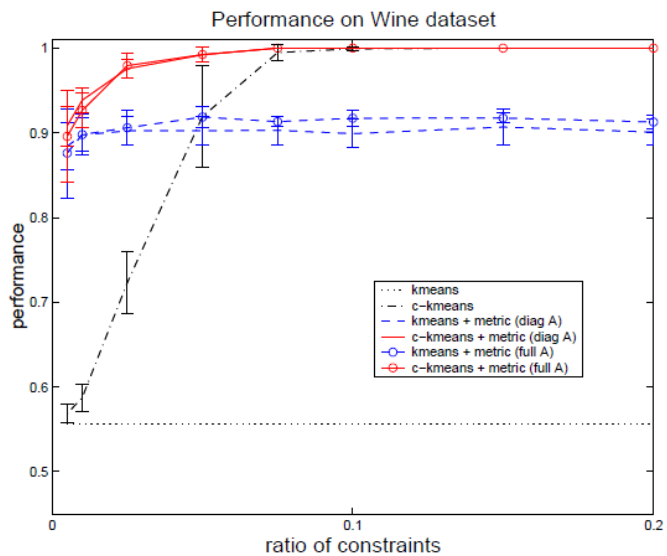
Algorithms Tested

1. K-means using the default Euclidean metric $||x_i - \mu_k||_2^2$ between points x_i and cluster centroids μ_k to define distortion (and ignoring $\mathcal{S}$).
2. Constrained K-means: K-means but subject to points $(x_i, x_j) \in \mathcal{S}$ always being assigned to the same cluster [12].[7]
3. K-means + metric: K-means but with distortion defined using the distance metric $||x_i - \mu_k||_A^2$ learned from $\mathcal{S}$.
4. Constrained K-means + metric: Constrained K-means using the distance metric learned from $\mathcal{S}$.

Test Results



(a)



(b)

Figure 7: Plots of accuracy vs. amount of side-information. Here, the x-axis gives the fraction of all pairs of points in the same class that are randomly sampled to be included in $\mathcal{S}$.

Approach

Previous metric-based semi-supervised clustering algorithms. . .

- use only labeled data for metric learning step and separate metric learning from clustering step
- learn a single metric for the entire data set
- force satisfaction of all constraints

Algorithms takes an iterative (EM) approach, alternating between

1. clustering the entire dataset according to constraints and current metric(s)
2. updating metric(s) according to current constraint violations

Pairwise Constrainted K-Means (PCK-means)

Minimize the K-means objective plus costs of constraint violations:

$$\sum_{i=1}^{n} \|x_i - \mu_{\ell_i}\|_2^2 + \sum_{(i,j)\in\mathcal{M}} w_{ij} 1_{\{\ell_i \neq \ell_j\}} + \sum_{(i,j)\in\mathcal{C}} w_{ij} 1_{\{\ell_i = \ell_j\}}$$

Multiple Metric K-Means (MK-means)

Learning separate metrics for each cluster better adapts to varied cluster shapes.

For each metric $\|x - y\|_{A_{\ell_i}} = \sqrt{(x-y)^T A_{\ell_i}(x-y)}$, minimize

$$\sum_{i=1}^{n} \|x_i - \mu_{\ell_i}\|_2^2 - \log(\det(A_{\ell_i}))$$

(subject to $A \succeq 0$)

Integrating Constraints and Metric Learning

Combining the previous objectives gives

$$\min_{A_1,\ldots,A_K \succeq 0, \ell} \quad \sum_{i=1}^{n} \|x_i - \mu_{\ell_i}\|^2_{A_{\ell_i}} - \log(\det(A_{\ell_i})) + \sum_{(i,j)\in\mathcal{M}} w_{ij} 1_{\{\ell_i \neq \ell_j\}} + \sum_{(i,j)\in\mathcal{C}} w_{ij} 1_{\{\ell_i = \ell_j\}}$$

Integrating Constraints and Metric Learning

Combining the previous objectives gives

$$\min_{A_1,\ldots,A_K \succeq 0, \ell} \quad \sum_{i=1}^{n} \|x_i - \mu_{\ell_i}\|^2_{A_{\ell_i}} - \log(\det(A_{\ell_i})) + \sum_{(i,j)\in\mathcal{M}} w_{ij} 1_{\{\ell_i \neq \ell_j\}} + \sum_{(i,j)\in\mathcal{C}} w_{ij} 1_{\{\ell_i = \ell_j\}}$$

This weights constraints independently of the distance between the corresponding points.

Distance weighting

To learn a good metric d

- violations of $(i, j) \in \mathcal{M}$ constraints should cost more when $d(x_i, x_j)$ is large
- violations of $(i, j) \in \mathcal{C}$ constraints should cost more when $d(x_i, x_j)$ is small

Define

$$f_M(x_i, x_j) = \frac{1}{2}\|x_i - x_j\|^2_{A_{\ell_i}} + \frac{1}{2}\|x_i - x_j\|^2_{A_{\ell_j}}$$

and

$$f_M(x_i, x_j) = \|x'_{\ell_i} - x''_{\ell_i}\|^2_{A_{\ell_i}} - \|x_i - x_j\|^2_{A_{\ell_i}}$$

where

$$(x'_{\ell_i}, x''_{\ell_i}) = \operatorname*{argmax}_{(x,y)\in\mathcal{X}} \|x - y\|_{A_{\ell_i}}.$$

Integrating Constraints and Metric Learning

Combining the previous objectives gives

$$\min_{A_1,\ldots,A_K \succeq 0, \ell} \quad \sum_{i=1}^{n} \|x_i - \mu_{\ell_i}\|^2_{A_{\ell_i}} - \log(\det(A_{\ell_i})) + \sum_{(i,j)\in\mathcal{M}} w_{ij} f_M(x_i, x_j) 1_{\{\ell_i \neq \ell_j\}} + \sum_{(i,j)\in\mathcal{C}} w_{ij} f_C(x_i, x_j) 1_{\{\ell_i = \ell_j\}}$$

E-step

- Sequentially add points greedily to cluster minimizing objective
- Theoretically depends on ordering
- Empirically, order does not matter much, so use random ordering

M-step

- Can set gradient of objective w.r.t. A_h to 0 and solve for A to get closed form update
- Project onto positive semidefinite cone (eliminates EM convergence guarantee, but works empirically)
- Requires inverting a $d \times d$ matrix, so can be too slow ($O(d^6)$) in high dimensions
- Restricting A_h to be diagonal is more efficient (will see performance later)

Initialization

- Good initialization is critical for greedy algorithms such as K-means.
- Use constraints to infer initial constraints as follows:
  1. Augment $\mathcal{M}$ with its transitive closure to create λ cliques
  2. Augment $\mathcal{C}$ with full bipartite graph between cliques with at least one edge
  3. If $\lambda < K$, add $K - \lambda$ random centroids
  4. If $\lambda > K$, run farthest-first algorithm weighted by clique size

Algorithms Tested

- MPCK-MEANS clustering, which involves both seeding and metric learning in the unified framework described in Section 2.4; a single metric parameterized by a diagonal matrix is used for all clusters;
- MK-MEANS, which is K-Means clustering with the metric learning component described in Section 3.3, without utilizing constraints for initialization; a single metric parameterized by a diagonal matrix is used for all clusters;
- PCK-MEANS clustering, which utilizes constraints for seeding the initial clusters and directs the cluster assignments to respect the constraints without doing any metric learning, as outlined in Section 2.2;
- K-MEANS unsupervised clustering;
- SUPERVISED-MEANS, which performs assignment of points to nearest cluster centroids inferred from constraints, as described in Section 3.1. This algorithm provides a baseline for performance of pure supervised learning based on constraints.

Empirical Observations

- Randomly sampled true labels to get constraints
- Performance measured via F-measure (combination of precision and recall)
- Generally, MPCK-means performed best

Empirical Observations

Tried variants with single metric or diagonal A's

- Fitting full matrix and multiple metrics generally performed best for many constraints (≈ 300)
- For smaller number of constraints, performance varied by data set

Questions

Both two papers learn a metric from data and cluster. Xing does the two tasks separately and Bilenko does them simultaneously. Xing does not use unlabeled data to learn the metric and Bilenko uses unlabeled data via a clustering framework which is based on k-means. Is there any other work that does metric learning solely (like Xing) but also utilizes unlabeled data (i.e., a true semi-supervised metric learning framework)?

Questions

Both two papers learn a metric from data and cluster. Xing does the two tasks separately and Bilenko does them simultaneously. Xing does not use unlabeled data to learn the metric and Bilenko uses unlabeled data via a clustering framework which is based on k-means. Is there any other work that does metric learning solely (like Xing) but also utilizes unlabeled data (i.e., a true semi-supervised metric learning framework)? Need to make an

assumption about how unlabeled data should affect the metric. Bilenko assumes the metric should conform to the clusters.