Probability-One Homotopy Maps for Tracking Constrained Clustering Solutions

David R. Easterling[*@], M. Shahriar Hossain[*],
Layne T. Watson[*+], N. Ramakrishnan[*]
Departments of Computer Science[*] and Mathematics[+]
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061
E-mail contact: dreast@vt.edu[@]

Keywords: probability-one homotopy, constrained clustering, machine learning, semisupervised learning

Abstract. Modern machine learning problems typically have multiple criteria, but there is currently no systematic mathematical theory to guide the design of formulations and exploration of alternatives. Homotopy methods are a promising approach to characterize solution spaces by smoothly tracking solutions from one formulation (typically an "easy" problem) to another (typically a "hard" problem). New results in constructing homotopy maps for constrained clustering problems are here presented, which combine quadratic loss functions with discrete evaluations of constraint violations are presented. These maps balance requirements of locality in clusters as well as those of discrete must-link and must-not-link constraints. Experimental results demonstrate advantages in tracking solutions compared to state-of-the-art constrained clustering algorithms.

1. Introduction

As machine learning permeates multiple fields of science and engineering, new objective functions are continually being proposed to suit the demands of new application domains. Multi-criteria objective functions especially are becoming more prevalent in areas such as mixing labeled and unlabeled data [1], [4], [16], incorporating constraints [8], [17], [19], and transfer learning [13], [14], [20], [21].

Although many of these multi-criteria problems have been approached with qualified success, there is currently a lack of a systematic mathematical theory to guide the design of formulations, understand tradeoffs, and explore alternatives. In particular, many formulations involve a parameter λ that balances or weights competing or alternative measures or formulations. For instance, λ might be used to balance between purely supervised and purely unsupervised learning to construct a hidden Markov model classifier [12], or to weight data compression relative to auxiliary information preservation in the information bottleneck [15], or to weight clustering constraints [10], [5], or to balance cluster coherence with cluster disparity in disparate clustering [11]. In general, existing theory does not deal elegantly with (1) how to efficiently compute solutions parametrically as λ varies, (2) how to find and deal with multiple solutions for a fixed λ and (3) how to canonically define the best choice of λ. Since most machine learning formulations involve multiple local optima, repeated optimization for discretely varying values of λ yields an incomplete picture of the solution space.

A systematic approach to characterize solution spaces by employing homotopy methods to smoothly track solutions from an unconstrained formulation to a constrained formulation is provided. This allows the effect of changing λ on the quality and nature of the solutions to be mathematically characterized. Smoothly tracking solutions as λ varies provides a holistic understanding of the interplay between the algorithm and a dataset. Beginning the homotopy zero curve tracking where the solution is (fairly) well-understood, the homotopy curve can then be tracked into regions where there is only a qualitative understanding of the solution space, finding multiple local minima (for the same λ) along the way. By connecting solutions across values of λ, homotopy methods can provide the raw material for obtaining multiple distinct solutions that can then be aggregated using ensemble techniques.

Initial efforts into the application of homotopy methods to machine learning have been made in [7], where classical continuation is used as a way to study how two diverse information sources should be combined in order to arrive at an integrated model. [12]shows that a general semisupervised formulation for hidden Markov models (HMMs) can be realized using a probability-one homotopy as well. However, the creation of homotopy maps remains a bit of a black art, especially for emerging machine learning formulations.

In this paper, new results in constructing homotopy maps for constrained clustering problems, which combine quadratic loss functions with discrete evaluations of constraint violations, are presented. In constrained clustering, the goal is not just to obtain clusters that are local in their respective spaces but also to obey a discrete set of a priori must-link (ML) and must-not-link (MNL) constraints between points.

A homotopy map between locality criteria and constraints, the formulation of a continuous selection function to overcome the problem of discrete assignments (of points to clusters), and the boundedness of the homotopy curve so that it does not diverge to infinity are demonstrated here. Experimental results demonstrate significant advantages in tracking solutions compared to state-of-the-art constrained clustering algorithms.

2. Background

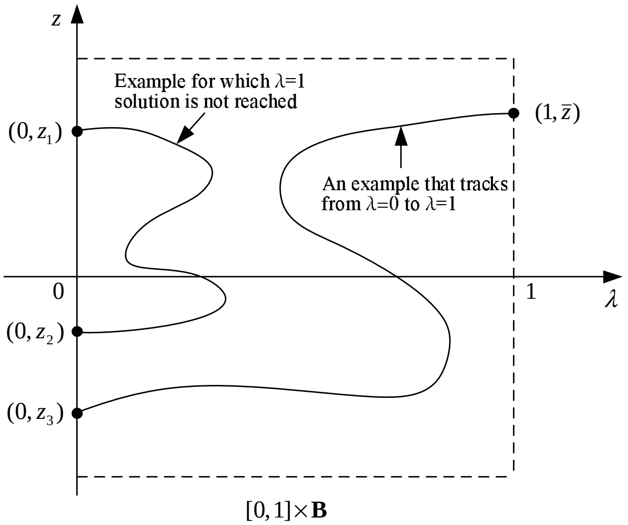
Some background information on probability-one homotopy theory and numerical homotopy algorithms is given first before new results for constrained clustering algorithms are described.

2.1 Globally convergent probability-one homotopies

The theory of globally convergent probability-one homotopy maps concerns finding zeros or fixed points of nonlinear systems of equations [6], [18]. The underlying idea is simple: Given a twice continuously differentiable function $F : \mathbb{R}^n \to \mathbb{R}^n$ of which a zero is sought, rather than solving the original difficult problem $F(z) = 0$ directly, start from an "easy" problem $G(z) = 0$ whose solution is readily identified, and gradually transform the "easy" problem into the original one, tracking the solutions along the transformation. Typically, one may choose a convex homotopy map, such as

$$H(\lambda, z) = (1 - \lambda)G(z) + \lambda F(z) \qquad (1)$$

and trace an implicitly defined zero curve $\gamma \in H^{-1}(0)$ from a starting point $(0, z^0)$ to a final point $(1, \bar{z})$. If this succeeds, then a zero point $\bar{z}$ of F is obtained. (1) also covers fixed point problems $z = f(z)$ by taking $F(z) = z - f(z)$. See Figure 1 for an example.



z
Example for which λ=1 solution is not reached
$(1, \bar{z})$
$(0, z_1)$
An example that tracks from λ=0 to λ=1
0
1
λ
$(0, z_2)$
$(0, z_3)$
$[0,1] \times \mathbf{B}$

Figure 1. Zero curve γ may return back to $\lambda = 0$ and there may not exist a path starting from some fixed points at $\lambda = 0$ and reaching a target solution at $\lambda = 1$.

Generally, there are two issues with respect to the homotopy method: (i) whether there indeed exists a smooth path of solutions starting from $\lambda = 0$ and reaching a target solution at $\lambda = 1$ in finite arc length, and (ii) development of numerical techniques for tracing this path. Three special homotopy maps [12] that assure the properties desired in (i), with probability one, follow; issue (ii) is discussed in [18].

Theorem 1. Suppose that $B \subset \mathbb{R}^n$ is a compact, convex subset and $f : B \to B$ is twice continuously differentiable. Then for almost all vectors $a \in \text{int}B$, there is a zero path γ of

$$H(\lambda, z) = (1 - \lambda)(z - a) + \lambda(z - f(z)), \qquad (2)$$

emanating from $(0,\ a)$, along which the $n \times (n + 1)$ Jacobian matrix $DH(\lambda, z)$ has full rank, that does not intersect itself and is disjoint from any other zeros of H, and reaches an accumulation point $(1,\ \bar{z})$ for which $f(\bar{z}) = \bar{z}$. Furthermore, if the Jacobian matrix $DH(1,\ \bar{z})$ is nonsingular, then the zero path γ between $(0,\ a)$ and $(1,\ \bar{z})$ has finite arc length [12].

A more general case is where $a : U \times B \to \text{int}B$ is a function of b and z, i.e., $a(b,\ z)$ and $U \subset \mathbb{R}^m$ is a nonempty open set, giving the homotopy map

$$H(b, \lambda, z) = (1 - \lambda)(z - a(b, z)) + \lambda(z - f(z)) \qquad (3)$$

where the parameter vector b is crucial for the probability-one homotopy theory, as shown in the theory in [12].

This works if for each $b \in U$, $a_b(z) = a(b, z)$ has a unique fixed point. If $a_b(z)$ has multiple fixed points, then consider the homotopy map

$$\begin{aligned} H(a_0, \lambda, z) = {} & (1 - \tanh(60\lambda))(z - a_0) \\ & + \tanh(60\lambda)[(1 - \lambda)(z - a_b(z)) \\ & + \lambda(z - f(z))], \end{aligned} \tag{4}$$

where $a_0 \in \mathbb{R}^n$ is a constant vector, and $\tanh(\cdot)$ is the hyperbolic tangent function. That (4) works is rigorously proven in [12], and used there for semisupervised HMM training.

3. Homotopy Maps for Constrained Clustering

3.1 Definitions

Let superscripts denote vector indices and subscripts denote components of vectors unless otherwise indicated. Let all norms be 2-norms and all distances be Euclidean distances. Given a set $\hat{X} = \{x^i \mid x^i \in \mathbb{R}^d,\ i = 1,\ 2,\ \ldots,\ k\}$ of k points (cluster prototypes) in d dimensions, let $X = \mathrm{vec}\big(x^1,\ x^2,\ \ldots,\ x^k\big) \in \mathbb{R}^{kd}$. Given a set $\hat{Y} = \{y^i \mid y^i \in \mathbb{R}^d,\ i = 1,\ 2,\ \ldots,\ n\}$ of n data points in d dimensions, let $Y = \mathrm{vec}\big(y^1,\ y^2,\ \ldots,\ y^n\big) \in \mathbb{R}^{nd}$. Represent a constraint by the vector $c = (a,\ b,\ z,\ w) \in \mathbb{R}^{2d+2}$ of two data points $a,\ b \in \hat{Y}$, an identifier $z = \pm 1$, and a degree-of-belief weight $w \in \mathbb{R}$, where an identifier of $z = 1$ means that a and b are bound by a must-link constraint (i.e., must be in the same cluster) and an identifier of $z = -1$ means that a and b are bound by a must-not-link or cannot-link constraint (must not be in the same cluster). Given a set $\hat{C} = \{c^i \mid c^i \in \mathbb{R}^{2d+2},\ i = 1,\ 2,\ \ldots,\ q\}$ of q constraints, let $C = \mathrm{vec}\big(c^1,\ c^2,\ \ldots,\ c^q\big) \in \mathbb{R}^{q(2d+2)}$.

If the distance between a data point and a cluster prototype is smaller than the distance between that data point and any other cluster prototype, define that data point as belonging to the cluster of all such data points. Note that if a data point is equidistant from multiple cluster prototypes, it is traditionally assigned to one of those clusters at random. Given sets $\hat{Y}$ and $\hat{C}$, the constrained clustering problem is to find a set $\hat{X}$ such that the largest number of constraints in $\hat{C}$ are satisfied. Since the number of constraints satisfied is a discrete value, some work is required to produce a smooth homotopy mapping connecting a local solution to the k-means clustering problem to a local solution of the constrained clustering problem.

3.2 Soft clustering – initial formulation

The traditional way to convert the discrete clustering problem to a continuous problem is through a distribution of probabilities, so that instead of assigning each data point to a single cluster prototype, each data point is assigned a probability corresponding with each cluster prototype based on the distance between the data point and each cluster prototype.

Let the probability-of-membership function $V : \mathbb{R}^{kd} \times \mathbb{R}^d \to \mathbb{R}^k$ be defined as $V_i(X, y^j) = e^{-\psi|y^j - x^i|^2} \,\big/\, \sum_{m=1}^{k} e^{-\psi|y^j - x^m|^2}$, $i = 1,\ 2,\ \ldots,\ k$, where $y^j \in \hat{Y}$, $x^m \in \hat{X}\ \forall m$, and $\psi > 0$ is a control constant that determines how soft the clustering is.

Let the soft must-link penalty function $P_1 : \mathbb{R}^{q(2d+2)} \times \mathbb{R}^{kd} \to \mathbb{R}$ be defined as

$$P_1(C, X) = \sum_{i=1}^{q} \frac{1 + z_i}{2} |V(X, a^i) - V(X, b^i)|^2,$$

where $c^i = \big(a^i, b^i, z_i, w_i\big) \in \hat{C}$.

Let the soft must-not-link penalty function $P_2 : \mathbb{R}^{q(2d+2)} \times \mathbb{R}^{kd} \to \mathbb{R}$ be defined as

$$P_2(C, X) = \sum_{i=1}^{q} \frac{1 - z_i}{2} \left(\frac{1}{\mu} - \frac{1}{3} \right),$$

where $\mu = 1 + |V(X, a^i) - V(X, b^i)|^2$.

Let the soft k-means approximation function $P_3 : \mathbb{R}^{kd} \times \mathbb{R}^{nd} \to \mathbb{R}$ be defined as

$$P_3(X, Y) = \sum_{i=1}^{k} \sum_{j=i}^{n} |y^j - x^i|^2 V_i(X, y^j),$$

and define the constrained clustering problem by letting Y and C be held constant. Then the proposed homotopy map $H : [0, 1) \times \mathbb{R}^{kd} \to \mathbb{R}^{kd}$ for soft clustering is

$$H(\lambda, X) = (1 - \lambda)\nabla P_3(X) + \lambda(\nabla P_1(X) + \nabla P_2(X)),$$

which is not a probability-one homotopy. Consequently, the Jacobian matrix DH is frequently rank deficient on $H^{-1}(0)$, making this a poor choice of homotopy map, as well as computationally expensive.

3.3 Hard clustering – geometric interpretation

A second way to generate a valid homotopy map from the nondifferentiable k-means function is to drop the repeated calculation of the k-means solution and simply select a starting point that corresponds to a solution provided by a single application of the k-means algorithm at $\lambda = 0$. This requires penalty functions based on the distances involved that satisfy a few key properties, outlined below.

For a data point $y \in \hat{Y}$ and two cluster prototypes $x^i, x^j \in \hat{X}$ define the function $D : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ by

$$D(x^i, x^j, y) = \left(\max\{0, |x^i - y|^2 - |x^j - y|^2\}\right)^4.$$

Note that D is three times continuously differentiable, $D \geq 0$, and $D(x^i, x^j, y) > 0$ if and only if the distance between y and x^i is larger than the distance between y and x^j.

Given $a, b \in \hat{Y}$, let the must-link function $F_m : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^{kd} \to \mathbb{R}$ be defined by

$$F_m(a, b, X) = \prod_{i=1}^{k} \left(\sum_{j=1, j \neq i}^{k} D(x^i, x^j, a) + D(x^i, x^j, b) \right)$$

and let the cannot-link function $F_c : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^{kd} \to \mathbb{R}$ be defined by

$$F_c(a, b, X) = \sum_{i=1}^{k} \left(\prod_{j=1, j \neq i}^{k} D(x^j, x^i, a) D(x^j, x^i, b) \right).$$

Then the following observations are easily verified.

Lemma 1. F_m and F_c are nonnegative and three times continuously differentiable.

Lemma 2. For any constraint $c = (a, b, 1, w) \in \hat{C}$, the must-link function $F_m(a, b, X) = 0$ if and only if constraint c is satisfied.

Lemma 3. For any constraint $c = (a, b, -1, w) \in \hat{C}$, the cannot-link function $F_c(a, b, X) = 0$ if and only if constraint c is satisfied.

Theorem 2. The penalty function

$$F(C, X) = \sum_{\{i : z_i = 1\}} F_m(a^i, b^i, X) + \sum_{\{i : z_i = -1\}} F_c(a^i, b^i, X)$$

is zero if and only if all the constraints in $\hat{C}$ are satisfied.

It is simple to add a degree-of-belief weight $w_i > 0$ to each component of the penalty function F without eliminating its properties:

$$F(C, X) = \sum_{\{i : z_i = 1\}} w_i F_m(a^i, b^i, X) + \sum_{\{i : z_i = -1\}} w_i F_c(a^i, b^i, X).$$

By Theorem 2, if it is possible to satisfy all of the constraints, then there exists a vector of cluster prototypes $\mathcal{X}$ such that $F(C, \mathcal{X}) = 0$. This vector of cluster prototypes represents a global minimum point of the function F at which $\nabla_{\mathcal{X}} F(C, \mathcal{X}) = 0$. This suggests the homotopy map

$$H(\lambda, X) = (1 - \lambda)(X - K_0) + \lambda \nabla_X F(C, X),$$

where $K_0 \in \mathbb{R}^{kd}$ is a vector of cluster prototypes that forms a solution to the unconstrained K-means clustering problem.

3.4 Bounding strategy

To avoid an unbounded zero curve of H, it is important to keep the cluster prototypes bounded. This can be accomplished with a simple modification to H. Write the cluster prototype $x^i \in \hat{X}$ as $x^i = \left(x_1^i, x_2^i, \ldots, x_d^i\right)$. Let $B \in \mathbb{R}^d$ be a bounding vector defined by $B_i = \max\left\{|y_i^1|, |y_i^2|, \ldots, |y_i^n|\right\}$, $i = 1$, 2, …, d. Map $x^i \in \mathbb{R}^d$ to $\xi^i \in \sqrt{d} S^d$ (the sphere of radius $\sqrt{d}$ in Re^{d+1}) by

$$\xi^i = \xi(x^i) = \left(\frac{x_1^i}{B_1}, \frac{x_2^i}{B_2}, \ldots, \frac{x_d^i}{B_d}, \sqrt{d - \left(\frac{x_1^i}{B_1}\right)^2 - \left(\frac{x_2^i}{B_2}\right)^2 - \cdots - \left(\frac{x_d^i}{B_d}\right)^2} \right),$$

let $\hat{\Xi} = \{\xi^i\}_{i=1}^{k}$, and let $\Xi = \mathrm{vec}\left(\xi^1, \xi^2, \ldots, \xi^k\right) \in \mathbb{R}^{k(d+1)}$. Then every cluster prototype $x^i \in \hat{X}$ can instead be represented by $\xi^i \in \hat{\Xi}$, and

$$x^i = x(\xi^i) = \left(\xi_1^i B_1, \xi_2^i B_2, \ldots, \xi_d^i B_d\right).$$

With $\Psi : \mathbb{R}^{d+1} \to \mathbb{R}$ defined as

$$\Psi(\xi) = (\xi_1)^2 + (\xi_2)^2 + \ldots + (\xi_{d+1})^2 - d,$$

$\Psi(\xi) = 0$ keeps ξ on the sphere $\sqrt{d} S^d$ and x in the box $\prod_{i=1}^{d} [-B_i \sqrt{d}, B_i \sqrt{d}]$. Thus the homotopy map $H_b : [0, 1) \times \mathbb{R}^{k(d+1)} \to \mathbb{R}^{k(d+1)}$ defined by

$$H_b(\lambda, \Xi) = \begin{pmatrix} (1 - \lambda)(x(\xi^1) - K_0^1) + \lambda(\nabla_{x(\xi^1)} F(C, X)) \\ \Psi(\xi^1) \\ (1 - \lambda)(x(\xi^2) - K_0^2) + \lambda(\nabla_{x(\xi^2)} F(C, X)) \\ \Psi(\xi^2) \\ \vdots \\ (1 - \lambda)(x(\xi^k) - K_0^k) + \lambda(\nabla_{x(\xi^k)} F(C, X)) \\ \Psi(\xi^k) \end{pmatrix}$$

will have bounded zero curves, but not a unique solution Ξ at $\lambda = 0$ because the coordinates ξ_{d+1}^i can have either sign.

3.5 Avoiding Degenerate Cases

In degenerate cases, the global minimum of F corresponds to a random assignment of points to clusters, which is geometrically represented when all cluster prototypes occupy the same point. In order to avoid this, it is necessary to include an additional penalty function $R : \mathbb{R}^{kd} \to \mathbb{R}$ that factors in the distance of each cluster prototype from each other cluster prototype.

Let $R(X) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \max(0, \ell - |x^i - x^j|^2)^3$, $x^i, x^j \in \hat{X}$. Then $R \geq 0$, R is twice continuously differentiable, and $R = 0$ unless two mean prototypes x^i and x^j are less than a distance ℓ from each other, where ℓ is the user-defined regularization parameter.

The bounded probability one homotopy map that includes regularization is

$$H_b(\lambda, \Xi) = \begin{pmatrix} (1-\lambda)(\nu_1) + \lambda(\nabla_{x(\xi^1)} F(C,X) + R(X)) \\ \Psi(\xi^1) \\ (1-\lambda)(\nu_2) + \lambda(\nabla_{x(\xi^2)} F(C,X) + R(X)) \\ \Psi(\xi^2) \\ \vdots \\ (1-\lambda)(\nu_k) + \lambda(\nabla_{x(\xi^k)} F(C,X) + R(X)) \\ \Psi(\xi^k) \end{pmatrix}.$$

where $\nu_a = x(\xi^a) - K_0^a$.

3.6 Final formulation

Note that since the terms in Ψ are squared and that ξ^i_{d+1} is not used to compute x^i, $H_b(0, \Xi) = 0$ will have multiple solutions. Therefore, incorporating the mapping in (4) yields a new probability-one homotopy map

$$H(\lambda, \Xi) = ((1-\tau)(\Xi - \Xi_0) + \tau H_b(\lambda, \Xi))$$

where $\tau = \tanh(60\lambda)$ and $\Xi_0 \in \mathbb{R}^{k(d+1)}$ is the vector of cluster prototypes that define the initial solution to the unconstrained K-means clustering problem.

4. Experimental Results

The capability of the probability-one homotopy maps listed above on real as well as synthetic datasets is demonstrated. The experiments are designed to answer the following questions:

1. Do homotopy maps help in trading off locality with constraint satisfaction?

2. Can homotopy maps help find better solutions than tailor made algorithms for constrained clustering?

3. Do homotopy maps reveal insight into the structure of solutions otherwise not obtainable using pointwise exploration of the parameter?

4. Is parallelization an effective method for increasing the speed of homotopy tracking?

The above questions are answered in the affirmative. Low dimensional datasets are deliberately used so that the results can be visualized and explained using visual means. Modern homotopy software such as HOMPACK90 [18] enables the tracking of solutions of systems involving thousands of variables.

A simple synthetic dataset involving 200 points gathered from four Gaussian distributions is constructed as a sample dataset. There are two natural two-cluster clusterings possible, depending on whether the clusters are organized horizontally or vertically. The first of these clusterings is used as the starting point, and as the homotopy curve is tracked, constraints are slowly introduced. A list of 50 constraints was generated in such a way that the must-link constraints are picked from two different initial clusters and the must-not-link constraints are picked from the same initial clusters. Thus, the clusters are forced to reorganize as λ is varied.

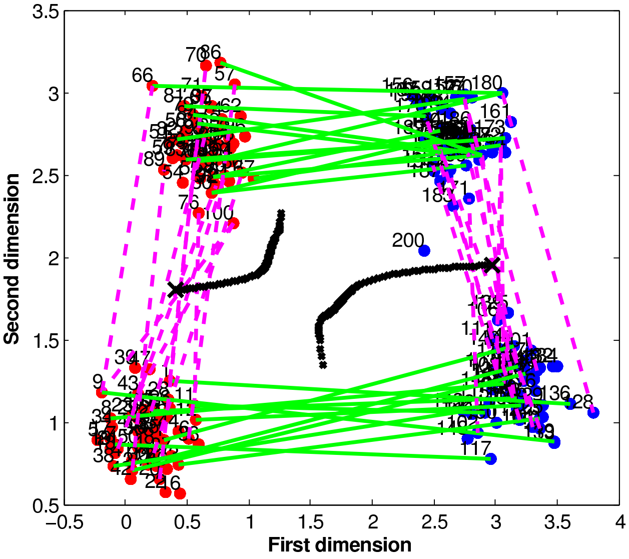


Figure 2. 200 points, two clusters, 50 constraints.

Figure 2 shows the data points and the constraints. The solid lines denote the must-link constrains and the dashed lines denote the must-not-link constraints. During the homotopy curve tracking, the cluster prototypes smoothly traverse the space and finally settle down to a position where a maximum of constraints are satisfied.

In addition to the data points and the constraints, Figure 2 shows the paths of the cluster prototypes as

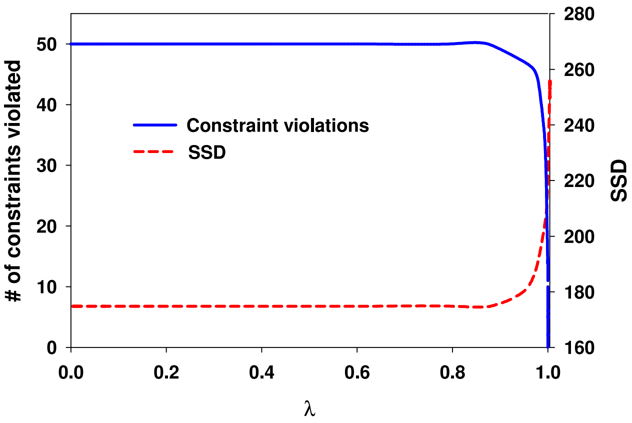


Figure 3. Tradeoff curve for sample clustering.

λ is varied. Figure 3 depicts that as λ increases during the tracking, the number of constraint violations reduces but the sum of squared distance (SSD) increases, as expected. Recall that the initial state of the homotopy curve, at $\lambda = 0.0$, represents a k-means local minimum, where the sum-of-squared distance is low but the number of constraint violations is high. At the other end of the curve, at $\lambda = 1.0$, the sum-of-squared distance reaches its peak but the all the constraints are satisfied. The actual crossing of these curves gives insights into how much one objective needs to be traded off in achieving another.

As shown in Figure 3, all the constraints are satisfied as the homotopy curve reaches $\lambda = 1$. To compare the homotopy results with a traditional constrained clustering algorithm, the MPCk-means algorithm [2] was applied on the same dataset and constraints. Only 52% of the constraints were satisfied by MPCk-means.

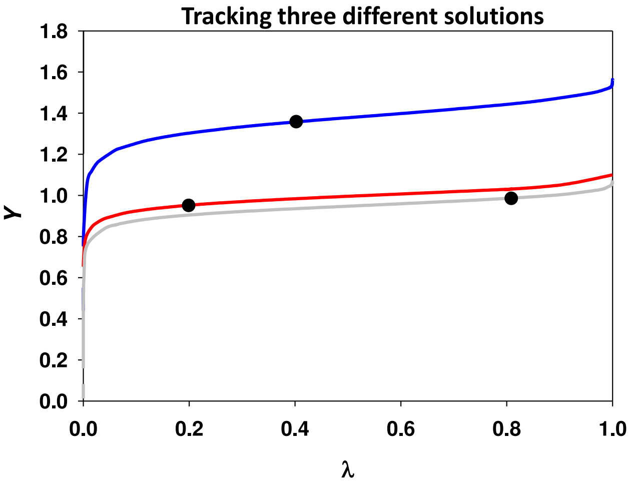


Figure 4. Tracking three different solutions.

Recall that the homotopy method can track multiple solutions for a given dataset and constraint set. Figure 4 shows an example where three different solutions are tracked for the same dataset and constraints as in Figure 2. One of these three tracking curves (the top curve) corresponds to Figure 3. Conventional optimization algorithms would require a fixed λ to balance the locality and constraint satisfaction trade-off of the objective function whereas homotopy smoothly tracks λ without requiring any user input for balancing the trade-offs. Discretely sampling λ and using a conventional local optimizer would likely detect solutions from different curves of the homotopy map. As an example, Figure 4 shows three points from three different curves which are obtained by a local optimizer at $\lambda = 0.2$, $\lambda = 0.4$, and $\lambda = 0.8$. This demonstrates that discrete sampling of λ does not give insight into the effect of λ as this would lead to the mistaken conclusion that the effect of λ has an inflexion point.

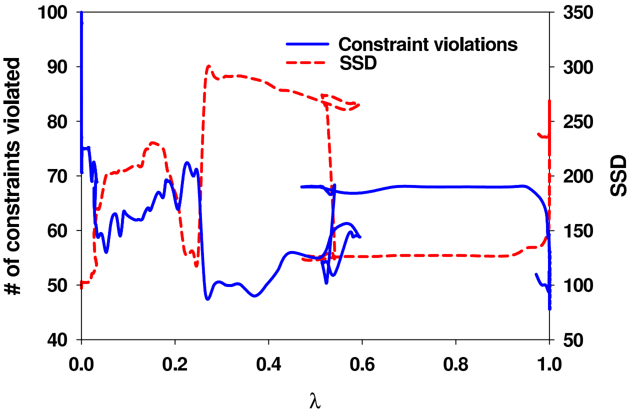


Figure 5. Iris dataset tradeoff curve.

The power of homotopy curve tracking in tracking a constrained clustering solution for the Iris dataset (from the UCI ML repository [9]) is demonstrated. Recall that the Iris dataset involves 150 points, four dimensions, and three class labels. One hundred random constraints are generated in a way that guarantees must-link constraints are from two different k-means clusters and must-not-link constraints are from the same cluster. The corresponding constraint violations and SSDs are shown in Figure 5. Figure 5 demonstrates that the SSD tends to be higher with smaller number of constraint satisfaction. It indicates that the zero curve automatically progresses with the trade-off parameter λ and smoothly tracks a solution. With the Iris dataset, with one specific starting solution, the homotopy based method was able to satisfy 54 constraints. The MPCk-means

Table 1. Parallel homotopy execution times (s).

no. processors	execution time	speedup
1	473.94	1.00
2	275.38	1.59
4	149.06	3.18
8	91.77	5.16
16	75.31	6.29

algorithm, on the other hand, was able to satisfy only 51 constraints for the same data. Note that λ changed direction during the solving of this particular dataset, a not uncommon occurrence, shown by the variations in the direction of the curves plotted in Figure 5.

Finally, a simple experiment was constructed to demonstrate the effectiveness of parallelization when applied to homotopy tracking. The homotopy map presented here is parallelizable in that it involves the sum of a large number of easily distributed calculations; in addition, HOMPACK90 itself has been parallelized independently [3]. This experiment focuses exclusively on parallelizing the homotopy map itself, although there is no reason that, with sufficient resources, both methods can't be employed to generate significant speedup. The homotopy map was parallelized by breaking up the calculations involved in the H_b function and its derivatives using OpenMP pragmas on a sixteen-node shared-memory cluster to parallelize the operations over the completely independent constraint penalty function calculations for a six dimensional problem with 324 data points with 200 constraints on three clusters. No parallelization was pursued for the homotopy tracking package HOMPACK90 for this experiment. The results are given in Table 1; note that this is perhaps the simplest method of parallelization available.

5. Discussion

New homotopy theory for constrained clustering problems has been developed and state-of-the-art mathematical software has been used to characterize multi-criteria problems in constrained clustering. Just as in other applications of homotopy methods to science and engineering, the application of homotopy methods to machine learning problems can usher in greater understanding of solution sets. Besides the strong mathematical foundations and rigorous formalisms brought to classical machine learning problems, this work has the potential to greatly reduce the ad hoc nature of methodological experimentation that is prevalent in practice. The approach given here not only helps extract better patterns from data, but to also helps formally understand the internal workings of machine learning techniques. While the homotopy maps presented here have worked in practice, improved maps from which more rigorous proofs can be derived are being developed. Continuing research involves the development and proper parallelization of these maps.

References

[1] M.F. Balcan; A. Blum. 2010. "A discriminative model for semi-supervised learning", J. ACM, 57, no. 3, issue 19, 1–46.

[2] M. Bilenko; S. Basu; R. J. Mooney. 2004. "Integrating constraints and metric learning in semi-supervised clustering", ICML '04, 11–18.

[3] A. Chakraborty; D.C.S. Allison; C. J. Ribbens; L.T.Watson. 1990. "Parallel homotopy curve tracking on a hypercube", 4th SIAM Conf. Parallel Processing for Scientific Computing, 149–153.

[4] O. Chapelle; B. Scholkopf; A. Zien. 2008. Semi-Supervised Learning, MIT Press, First Edition.

[5] H. Cheng; K. A. Hua; K. Vu. 2008. "Constrained locally weighted clustering", Proc. VLDB Endow., 1, no. 1, 90–101.

[6] S. N. Chow; J. Mallet-Paret; J. A. Yorke. 1978. "Finding zeros of maps: homotopy methods that are constructive with probability one", Math. Comput., 32, 887–899.

[7] A. Corduneanu; T. Jaakkola. 2002. "Continuation methods for mixing heterogeneous sources", UAI '02, 111–118.

[8] A. Demiriz, K. P. Bennett, P. S. Bradley. 2008. Constrained Clustering: Advances in Algorithms, Theory, and Applications. K. Wagstaff, I. Davidson, and S. Basu (eds.), Chapman & Hall/CRC, First edition.

[9] A. Frank; A. Asuncion. 2010, "UCI Machine Learning Repository", Technical Report.

[10] P. Hansen; B. Jaumard; K. Musitu. 1990. "Weight constrained maximum split clustering", Journal of Classification, 7, no. 2, 217–240.

[11] P. Jain; R. Meka; I.S. Dhillon. 2008. "Simultaneous unsupervised learning of disparate clusterings", SDM '08, 858–869.

[12] S. Ji; L.T. Watson; L. Carin. 2009. "Semisupervised learning of hidden Markov models via a homotopy method", IEEE Trans. Pattern Anal. Machine Intell., 31, 275–287.

[13] P. Luo; F. Zhuang; H. Xiong; Y. Xiong; Q. He. 2008. "Transfer learning from multiple source domains via consensus regularization", CIKM '08, 103–112.

[14] M.E. Taylor; G. Kuhlmann; P. Stone. 2008. "Autonomous transfer for reinforcement learning", AAMAS '08, 1, no. 283–290.

[15] N. Tishby; F.C. Pereira; W. Bialek. 1999. "The information bottleneck method", 37th Annual Allerton Conference on Communications, Control and Computing, 368–377.

[16] K. Sinha; M. Belkin. 2008. "The value of labeled and unlabeled examples when the model is imperfect", Advances in Neural Information Processing Systems 20, MIT Press.

[17] X. Wang; I. Davidson. 2010. "Flexible constrained spectral clustering", KDD '10, 563–572.

[18] L. T. Watson, M. Sosonkina, R. C. Melville, A. P. Morgan, H. F. Walker. 1997. "Algorithm 777: HOMPACK90:

a suite of Fortran 90 codes for globally convergent homotopy algorithms", ACM Trans. Math. Software, 23, 514–549.

[19] H. Yang; J. Callan. 2009. "A metric-based framework for automatic taxonomy induction", ACL '09, 1, 271–279.

[20] Q. Yang; Y. Chen; G. R. Xue; W. Dai; Y. Yu. "Heterogeneous transfer learning for image clustering via the social web", ACL '09, 1–9.

[21] D. Zhang; J. He; Y. Liu; L. Si; R. Lawrence. 2011. "Transfer learning from multiple source domains via consensus regularization", KDD '11, 1208–1216.