Probability-one Homotopy Maps for Constrained Clustering Problems

David R. Easterling[1], Layne T. Watson[1], Naren Ramakrishnan[1], M. Shahriar Hossain[2]
[1]Virginia Polytechnic Institute & State University, Blacksburg, VA 24061
[2]Virginia State University, Petersburg, VA, 23806
Email: dreast@vt.edu, mshossain@vsu.edu, {ltw, naren}@cs.vt.edu

Abstract. Many algorithms for constrained clustering have been developed in the literature that aim to balance vector quantization requirements of cluster prototypes against the discrete satisfaction requirements of constraint (must-link or cannot-link) sets. A significant amount of research has been devoted to designing new algorithms for constrained clustering and understanding when constraints help clustering. However, no method exists to systematically characterize solution sets as constraints are gently introduced and how to assist practitioners in choosing a sweet spot between vector quantization and constraint satisfaction. A homotopy method is presented that can smoothly track solutions from unconstrained to constrained formulations of clustering. Beginning the homotopy zero curve tracking where the solution is (fairly) well-understood, the curve can then be tracked into regions where there is only a qualitative understanding of the solution set, finding multiple local solutions along the way. Experiments demonstrate how the new homotopy method helps identify better tradeoffs and reveals insight into the structure of solution sets not obtainable using pointwise exploration of parameters.

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. Multicriteria objective functions especially are becoming more prevalent in areas such as mixing labeled and unlabeled data (Balcan & Blum, 2010), (Chapelle, 2008), (Sinha & Belkin, 2008), incorporating constraints (Demiriz et al., 2008), (Wang & Davidson, 2010), (Yang & Callan, 2009), and transfer learning (Luo et al., 2008), (Taylor et al., 2008), (Yang et al., 2009b), (Zhang et al., 2011).

One such multiobjective formulation is in the area of constrained clustering. In constrained clustering (Basu et al., 2008), the goal is not just to obtain clusters that are local in their respective spaces but that also obey a discrete set of a priori must-link (ML) and cannot-link (CL) or must-not-link constraints between points. Although there are many powerful constrained clustering algorithms published in the literature (Dai et al., 2007), (Sato & Iwayama, 2009), (Hossain et al., 2010), (Baghshah & Shouraki, 2011), (He et al., 2012), (Sese et al., 2004), there is currently a lack of a systematic mathematical theory to guide the design of formulations, understand tradeoffs, and explore alternatives.

The fundamental problem in algorithm design for constrained clustering problems is the tradeoff between conventional clustering objectives and the requirements of the linking constraints. Broadly speaking, there have been two types of algorithms designed to deal with this problem (Davidson & Ravi, 2007). The first uses the constraints to learn a distance function. The second strictly enforces the constraints as the algorithm iterates to a useful solution. The primary problem motivating the development of these two algorithmic approaches is that determining the feasibility of a set of constraints that contains CL and ML constraints both is an NP-complete problem, being equivalent to the graph coloring problem. When the feasibility of a solution can

not be determined in polynomial time, the usual approach is to fall back on heuristics, with the hope that the resulting solution will be good enough.

One traditional solution to such heuristically solved biobjective problems is to introduce a parameter λ that balances or weights competing considerations, in this case cluster locality versus constraint satisfaction. Although there is interesting theoretical insights into the complexity of constrained clustering problems (Davidson, 2012), there is no existing theory available that can deal 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 set.

Homotopy methods are systematic approaches to characterize solution sets by smoothly tracking solutions from one formulation to another (in this case, from an unconstrained formulation to a constrained formulation). This can allow 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 set, finding multiple local solutions (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 (Corduneanu & Jaakkola, 2002), 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. Ji et al. show that a general semisupervised formulation for hidden Markov models (HMMs) can be realized using a probability-one homotopy as well (2009). However, the creation of homotopy maps remains a bit of a black art, especially for emerging machine learning formulations.

The key contributions here are:

1. The *first* results in constructing homotopy maps, which combine quadratic loss functions with discrete evaluations of constraint violations, for constrained clustering problems are presented. This is a nontrivial task since there are several discrete aspects to the constrained clustering problem (e.g., discrete assignments of point to clusters, discrete satisfactions or violations of constraints) that need to accommodated in a traditional homotopy framework.
2. The construction of homotopy maps can often be problem specific and typically requires careful tweaking to ensure convergence. Here, a general construct that enables the map to be applied to any constrained clustering problem, similar to existing algorithms for this purpose, is demonstrated.
3. Numerous experimental results demonstrating the scalability, viability, usefulness, superiority, and interpretability of the homotopy map approach to constrained clustering are presented.

2. Homotopy Theory Background.

Homotopy methods generalize classical continuation methods. Continuation uses the known solutions for an 'easy' problem to find the solutions for a 'hard' problem. In a standard continuation method for root finding (Watson, 1979, Watson, 2002), given two differentiable functions $f(x)$ and $g(x)$ such that a zero x_0 for $g(x)$ is known and a zero $\bar{x}$ for $f(x)$ is sought, define the homotopy map

$$\rho(\lambda, x) = (1-\lambda)g(x) + \lambda\, f(x), \quad 0 \leq \lambda \leq 1,$$

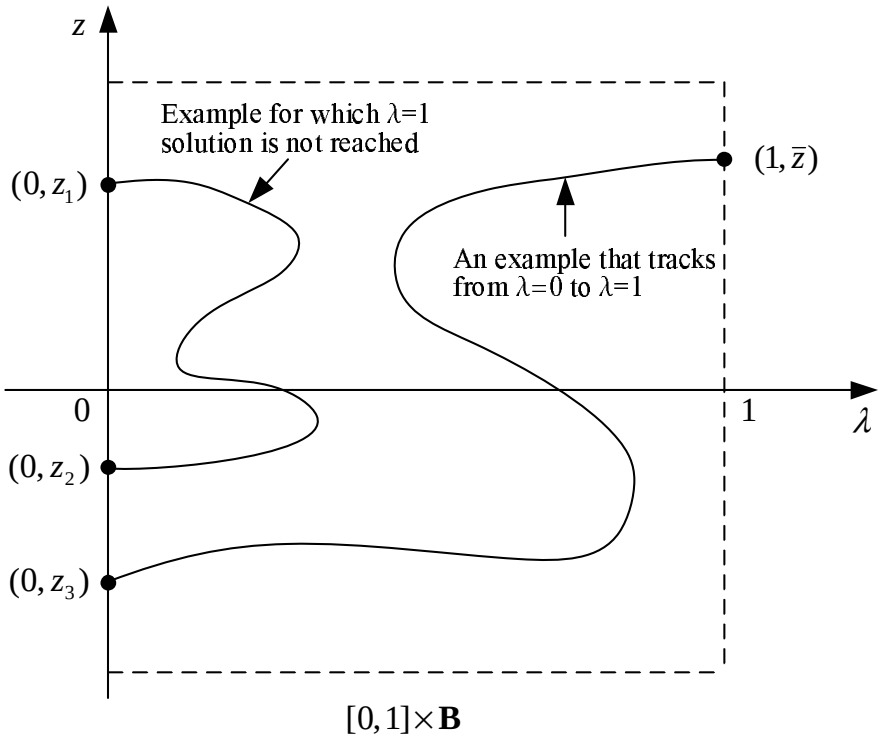

Figure 1. A classical continuation zero curve γ may return back to $\lambda = 0$ and there may not exist a curve starting from some zeros at $\lambda = 0$ and reaching a target solution at $\lambda = 1$. $B = \{z \in \mathbb{R}^n \mid \|z\|_2 \leq r\}$ is some ball in $\mathbb{R}^n$.

where λ is the homotopy variable, g is called the 'start' function, and f is called the 'target' function in the homotopy.

Since $\rho(0, x_0) = g(x_0) = 0$, standard local solution methods (such as Newton's method) may be employed iteratively as λ varies slowly from 0 to 1 to yield a zero $\bar{x}$ of f. Such a method, applied to the above problem, would yield a series of solutions along a solution curve γ, provided that a solution could be found for every λ. However, there is no guarantee that a given starting function g with zero x_0 will yield a zero of f, as the algorithm may fail as the continuation progresses. In particular, if some Jacobian matrix $D_x\rho(\tilde{\lambda}, x)$ is not invertible, the local solver will fail.

Continuation can fail if the zero curve γ emanating from $(0, x_0)$ of ρ fails to exist for some $\tilde{\lambda}$, or wanders off to infinity without ever reaching $\lambda = 1$, or if the Jacobian matrix $D_x\rho(\tilde{\lambda}, x)$ becomes singular due to a bifurcation or turning point in the zero curve γ. Unlike continuation methods, homotopy methods attempt to track the zero curve γ through turning points and bifurcation points, by locally parametrizing the curve $(\lambda, x) = (\lambda(t), x(t))$. All these issues with singular matrices, the existence of zero curves, and the mechanics of tracking them are addressed by modern *probability-one homotopy* methods (Watson, 1979), (Watson, 2002), (Chow et al., 1978), the essence of which is to guarantee almost surely (in the technical measure theoretic sense) the existence of a smooth, nonbifurcating, nonselfintersecting, bounded zero curve γ of a homotopy map $\rho_a(\lambda, x)$ that connects a start point $(0, x_0)$ to a point $(1, \bar{x})$ where $f(\bar{x}) = 0$.

These algorithms are implemented in FORTRAN 77 as HOMPACK (Watson et al., 1987), and extended in Fortran 90 as HOMPACK90 (Watson et al., 1997). The following are important theorems about probability-one homotopy maps and curves.

THEOREM 1: PARAMETRIZED SARD'S THEOREM. *Let $U \subset \mathbb{R}^m$, $V \subset \mathbb{R}^n$ be nonempty open sets, $\rho : U \times [0,1) \times V \to \mathbb{R}^n$ be a C^2 map, and define*

$$\rho_a(\lambda, x) = \rho(a, \lambda, x).$$

If ρ is transversal to zero (rank $D\rho = n$ *on* $\rho^{-1}(0)$), *then for almost all $a \in U$ the map ρ_a is also transversal to zero.*

THEOREM 2. *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ and $\rho : \mathbb{R}^m \times [0,1) \times \mathbb{R}^n \to \mathbb{R}^n$ be C^2, and define $\rho_a(\lambda, x) = \rho(a, \lambda, x)$. Assume that*
(1) *ρ is transversal to zero;*
(2) *for each fixed $a \in \mathbb{R}^m$, $\rho_a(0, x) = 0$ has a unique solution x_a at which* rank $D_x\rho_a(0, x_a) = n$*;*
(3) *$\rho_a(1, x) = F(x)$;*
(4) *for each $a \in \mathbb{R}^m$, the connected component of the zero set $\rho_a^{-1}(0)$ containing $(0, x_a)$ is bounded.*
Then for almost all $a \in \mathbb{R}^m$ there exists a zero curve γ of $\rho_a(\lambda, x)$, emanating from $(0, x_a)$, along which the $n \times (n+1)$ Jacobian matrix $D\rho_a(\lambda, x)$ has full rank, that does not intersect itself and is disjoint from any other zeros of ρ_a, and accumulates at a point $(1, \bar{x})$ for which $F(\bar{x}) = 0$. Furthermore, if rank $D\rho_a(1, \bar{x}) = n$*, then the curve γ connecting $(0, x_a)$ to $(1, \bar{x})$ has finite arc length.*

THEOREM 3. *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be C^2, and suppose there exist $r_0, r > 0$ such that for any $a \in \mathbb{R}^n$ with $\|a\|_2 < r_0$, $x - a$ and $F(x)$ do not point in opposite directions on $\left\{x \in \mathbb{R}^n \mid \|x\|_2 = r\right\}$. Define $\rho : \mathbb{R}^n \times [0,1) \times \mathbb{R}^n \to \mathbb{R}^n$ by*

$$\rho(a, \lambda, x) = (1 - \lambda)(x - a) + \lambda F(x),$$

and let $\rho_a(\lambda, x) = \rho(a, \lambda, x)$. Then for almost all vectors $a \in \mathbb{R}^n$ with $\|a\|_2 < r_0$ there exists a zero curve γ of $\rho_a(\lambda, x)$, emanating from $(0, a)$, along which the $n \times (n+1)$ Jacobian matrix $D\rho_a(\lambda, x)$ has full rank, that does not intersect itself and is disjoint from any other zeros of ρ_a, and accumulates at a point $(1, \bar{x})$ for which $F(\bar{x}) = 0$. Furthermore, if rank $D\rho_a(1, \bar{x}) = n$*, then the curve γ connecting $(0, a)$ to $(1, \bar{x})$ has finite arc length.*

Theorem 1 means that the set of points (λ, x) where $\rho_a(\lambda, x) = 0$ almost surely (with probability one, or for almost all points $a \in U$) looks like the curves in Figure 2. The hypotheses in Theorems 2 and 3 (Theorem 3 is a special case of Theorem 2) guarantee that the curve γ in Figure 2 is the only curve emanating from $\lambda = 0$, and that γ must reach (accumulate at) $\lambda = 1$. A probability-one homotopy algorithm is simply to track the zero curve γ of ρ_a, which is guaranteed to reach a solution $\bar{x}$ of $F(x) = 0$ at $\lambda = 1$, with probability one (almost surely).

In practice, the full rank of the Jacobian matrix at $(1, \bar{x})$, while convenient, is not necessary, as the zero curve usually approaches a solution as $\lambda \to 1$ with finite arc length. This is especially true when applied to the semisupervised clustering problem, as the desired clustering (satisfaction of all constraints) is presumed to be present at some point along γ before $\lambda = 1$, otherwise a direct optimization with $\lambda = 1$ would be preferable. Homotopy maps that fulfill the theorems' assumptions are called *globally convergent probability-one homotopy maps*. Given time to trace the finite (albeit potentially long) arc length of the solution (zero) curve with a robust enough curve tracker, such curves will inevitably yield a useful solution. More to the point in the current study, tracing the solution (zero) curve yields the entire parametrized solution trace (including multiple

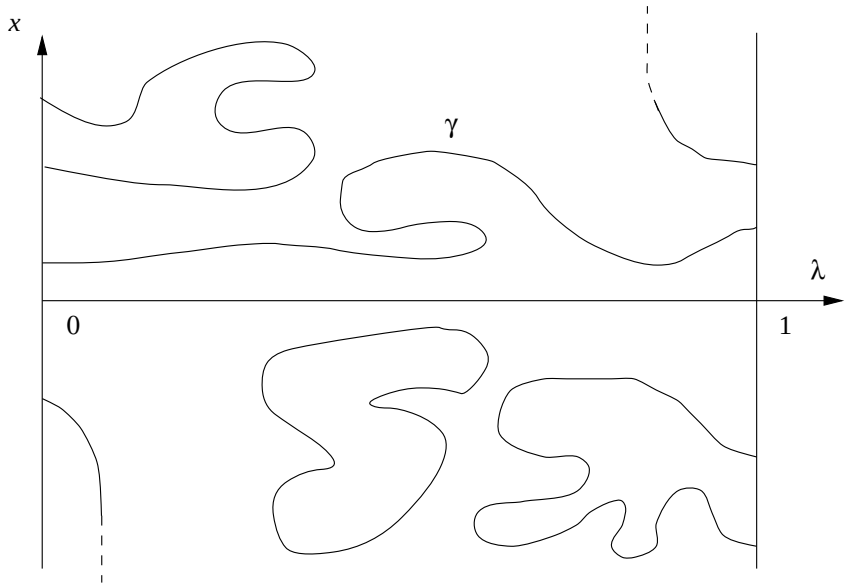

Figure 2. The inverse image $\rho_a^{-1}(0)$ for ρ_a transversal to zero.

solutions at the same λ values, obtained as the curve turns back and forth) for analysis, allowing for useful tradeoff information to be obtained for multiple objectives.

It is possible to convert a homotopy map $(1-\lambda)g(x)+\lambda f(x)$ where the "natural" start function $g(x)$ may have multiple zeros into one with a unique zero at $\lambda = 0$. Consider the map

$$\rho_a(\lambda, x) = (1 - \tanh(60\lambda))(x - a) + \tanh(60\lambda)\big[(1 - \lambda)g(x) + \lambda f(x)\big],$$

where tanh is the hyperbolic tangent function. The nature of the hyperbolic tangent function is such that $\tanh(60\lambda) \approx 1$ for $\lambda > 0.1$. Thus $\rho_a(\lambda, x) = 0$ has a unique solution $x = a$ at $\lambda = 0$, but for $\lambda > 0.1$ the map looks essentially like $(1 - \lambda)g(x) + \lambda f(x)$. Semisupervised learning problems often have such "natural" start functions g with multiple zeros, making this a useful trick for good homotopy map generation.

Sadly, there is no known "magical algorithm" that will generate a globally convergent probability-one homotopy map for a given biobjective problem. The key to the construction of any homotopy map in order to guarantee transversality and convergence is to ensure that the conditions in Theorem 2 are met while maintaining the desired properties that $\rho_a(0, x) = 0$ is easy to solve and that $\rho_a(1, \bar{x}) = f(\bar{x}) = 0$. Furthermore, if there are some situations where $f(x) = 0$ is not logically possible, then one of the above conditions must be impossible to meet (generally this means γ must be unbounded, and thus never reaches $\lambda = 1$). Thus, in practice, either a bounded zero curve γ must be assumed, such assumption being violated in the cases where $f(x) = 0$ has no solution, or the existence of a solution can be proven for a class of problems (as was done, e.g., with the semisupervised learning of HMM models (Ji et al., 2009)). Finally, if the homotopy map in question is intended to have value beyond the solution to $f = 0$, the map must reflect this in the construction of the start function g.

Note that homotopy algorithms typically do not follow the zero curve γ of ρ_a too closely, in order to avoid wasting computational effort. For a zero curve γ connecting a start point $(0, x_a)$ to

a final point $(1, \bar{x})$, there is a neighborhood W of $(0, x_a)$ such that for any start point $(0, x_z) \in W$ there is a zero curve γ_z of ρ_z close to γ that converges to the same final point $(1, \bar{x})$. Thus γ is surrounded by a "bundle" of zero curves (corresponding to different homotopy maps) that all lead to the same solution. Hence when numerically tracking γ it is only necessary to remain within this bundle (Watson, 1986). For the purposes of a homotopy map for solving the semisupervised clustering problem, close points on nearby zero curves correspond to close clusterings (close loss function values), so slight errors in curve tracking do not invalidate the utility of the method for measuring tradeoffs.

3. Homotopy Maps for Constrained Clustering.

Let $k^0 \in \mathbb{R}^{kd}$ be some (presumably poor) solution to the unsupervised clustering problem for k clusters in d dimensions, generated by a traditional clustering approach, such as the K-means algorithm. For the purposes of this map, consider each cluster assignment to be a "hard" assignment, that is, each data point is assigned to a single cluster determined by its distances from the cluster representatives, not assigned a probability of belonging to each cluster based on those distances.

Let superscripts denote vector indices and subscripts denote components of vectors unless otherwise indicated. Let all norms be 2-norms unless otherwise indicated and let all distances be Euclidean distances. Let $\mathbb{R}^n$ denote n-dimensional Euclidean space and let $\mathbb{R}^{n \times m}$ be the set of real $n \times m$ matrices. Let the ith row of a matrix $A \in \mathbb{R}^{n \times m}$ be denoted by $A_{i\cdot}$ and the jth column by $A_{\cdot j}$. Finally, for a vector $x \in \mathbb{R}^n$, $x > 0$ means all $x_i > 0$, $x \geqq 0$ means all $x_i \geqq 0$, and $x \geq 0$ means $x \geqq 0$ but $x \neq 0$.

Given a set $\hat{X} = \{x^i \mid x^i \in \mathbb{R}^d,\ i = 1,\ 2,\ \ldots,\ k\}$ of k points (cluster representatives) 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 cannot-link constraint (can 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 $C = \mathrm{vec}\big(c^1,\ c^2,\ \ldots,\ c^q\big) \in \mathbb{R}^{q(2d+2)}$.

3.1. Functions.

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

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

Note that D is three times continuously differentiable, $D \geqq 0$, 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.

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

Observation 2. For any must-link 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.

Observation 3. For any cannot-link 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.

Observation 4. 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:

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

It is worth noting at this juncture that disjunctive and conjunctive combinations of constraints can likewise be represented by the developed penalty functions, which are of particular value when ϵ- and δ-constraints are considered. ϵ- and δ-constraints are constraints that act upon groups of instances. ϵ-constraints are constraints that dictate that any data point in a cluster must have another data point in that cluster within ϵ distance, or be the only data point in the cluster. δ-constraints are constraints that dictate that any datapoint in a cluster must be at least δ distance from every datapoint that resides in a different cluster. Both of these types of constraints can be represented as disjunctions and conjunctions of must-link constraints (Davidson & Ravi, 2005).

Let C^1 and C^2 be constraints (must-link, cannot-link, or combinatorial) and let F^1 and F^2 be the corresponding penalty functions. Then $C^3 = C^1 \vee C^2$ has the corresponding penalty function $F^3 = F^1 F^2$. Similarly, $C^4 = C^1 \wedge C^2$ has the corresponding penalty function $F^4 = F^1 + F^2$. Observe that $F^3 = 0$ if and only if C^3 is satisfied and $F^3 > 0$ if and only if C^3 is not satisfied. Similarly, observe that $F^4 = 0$ if and only if C^4 is satisfied and $F^4 > 0$ if and only if C^4 is not satisfied. Finally, observe that any number of must-link and cannot-link constraints can thus be combined in conjunctive normal form by summing products of these penalty functions. As such, these penalty functions can easily be adapted to represent penalty functions for ϵ- and δ-constraints.

By Observation 4, if it is possible to satisfy all of the constraints, then there exists a vector of cluster representatives $\mathcal{X}$ such that $F(C, \mathcal{X}) = 0$. This vector of cluster representatives represents a global minimum point of the function F at which $\nabla_X F(C, \mathcal{X}) = 0$. This suggests the homotopy map (where $a = k^0$)

$$\check{\rho}_a(\lambda, X) = (1 - \lambda)(X - k^0) + \lambda \left(\nabla_X F(C, X)\right)^T.$$

The advantages of this homotopy map should be immediately evident. When $\lambda = 0$, the solution to the above map is simply the solution to the unsupervised clustering problem. When $\lambda = 1$, the solution, if one exists, represents a local minimum of the penalty function, which is based on the violation of constraints. This is not to say that the solution generated will satisfy all the constraints if such a solution is possible, as it is fairly easy to construct a degenerate set of constraints so that there is a local solution close to $x = k^0$. However, in practice this has not proven to be a problem.

This is a probability-one homotopy map, but while it satisfies conditions (1), (2), and (3) in Theorem 2, it fails to satisfy condition (4), bounded γ. Furthermore, there is a trivial solution to all constraints at $x^1 = x^2 = \ldots = x^k$, where all cluster representatives are equal. Thus, two modifications must be made to the above map. First, a bounding constraint $g(X) \leqq 0$ must be added to satisfy condition (4). Second, a constraint $h(X) = 0$ must be imposed to prevent the cluster representatives from degenerative stacking. Converting the minimum of this unconstrained penalty function to the minimum of a constrained function is simply a matter of converting the solution from a zero $\hat{X}$ of $\nabla_X F(C, X)$ to a Karush-Kuhn-Tucker (KKT) point (X^*, μ, ν) of the constrained minimization problem:

$$\min_X F(C, X) \quad \text{subject to} \quad g(X) \leqq 0, \quad h(X) = 0.$$

A KKT point for this constrained problem is a solution (X^*, μ, ν) of the KKT conditions:

$$\nabla_X F(C, X^*) + \sum_{i=1}^{m} \mu_i \nabla g_i(X^*) + \sum_{j=1}^{\ell} \nu_j \nabla h_j(X^*) = 0, \tag{1}$$
$$g(X^*) \leqq 0, \qquad h(X^*) = 0, \qquad \mu \geqq 0, \qquad \mu\, g(X^*) = 0.$$

3.2. Homotopy map.

First, consider the bounding constraint. A straightforward function $\Psi : \mathbb{R}^{kd} \to \mathbb{R}$ to achieve bounding is $\Psi(X) = B - \sum_{i=1}^{n} \|x^i\|^2 \geqq 0$, where $B > \|k^0\|^2$ is a given constant. The Lagrangian of the new bounded penalty function is $\hat{L}(X, \mu) = F(C, X) - \mu\Psi(X)$, and its derivative, replacing $\nabla_X F(C, X)$, is $\nabla_X \hat{L}(X, \mu) = \nabla_X F(C, X) - \mu \nabla_X \Psi(X)$. This yields a new variable, the Lagrangian multiplier μ, which in turn adds a new function to the map (since the map must be from $\mathbb{R}^{p+1} \to \mathbb{R}^p$ for some p), along with the requirement that $\mu \geqq 0$, $\Psi \geqq 0$, and $\mu\Psi = 0$. This naturally leads to the use of the Mangasarian NCP function presented in (Mangasarian, 1976) and modified in (Watson, 1979b), (Watson, 2001).

Define the function $\Phi : [0, 1] \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ by

$$\Phi\left(\lambda, \mu, \Psi(X), h^0\right) = -|\mu - \Psi(X)|^3 + \mu^3 + \Psi(X)^3 - (1 - \lambda)h^0$$

for some constant $h^0 > 0$. The constant term h^0 is designed to force the remaining terms to remain positive for $\lambda < 1$, which enforces the bounding of X for $\lambda < 1$, since $\Psi(X)$ must remain positive when $\Phi = 0$. When $\lambda = 1$, $\Phi(1, \mu, \Psi, h^0) = 0 \iff \mu \geqq 0$, $\Psi \geqq 0$, $\mu\Psi = 0$. The previous homotopy map $\check{\rho}_a$ is then modified to (where now $a = (k^0, h^0)$)

$$\hat{\rho}_a(\lambda, X, \mu) = \begin{pmatrix} (1 - \lambda)(X - k^0) + \lambda(\nabla_X \hat{L}(X, \mu))^T \\ \Phi\left(\lambda, \mu, \Psi(X), h^0\right) \end{pmatrix}. \tag{2}$$

Note that $\hat{\rho}_a(1, X, \mu) = 0$ is equivalent to the KKT conditions: $\nabla_X \hat{L} = 0$, $\Psi \geqq 0$, $\mu \geqq 0$, $\mu\Psi = 0$. Unfortunately, while the stated Φ enforces a lower bound on μ, it does not enforce an upper bound on μ; in fact, if $\Psi(X) \to 0$ as $\lambda \to \tilde{\lambda} < 1$, μ must potentially become arbitrarily large to compensate for this. While this map is better than the previous one in that it prevents cluster representatives from migrating arbitrarily far from the data set, it does not prevent μ from growing arbitrarily large, although in practice this is not a common occurrence.

Similarly, define the function $G : \mathbb{R}^{kd} \to \mathbb{R}$ by

$$G(X) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \max(0, \ell - \|x^i - x^j\|^2)^4, \quad x^i, x^j \in \hat{X}.$$

Then $G \geqq 0$, $G \in C^3$, and $G = 0$ unless two cluster representatives x^i and x^j are less than a distance $\sqrt{\ell}$ from each other, where $\ell > 0$ is a given regularization constant. This represents an equality constraint on the original problem. The updated Lagrangian becomes $\tilde{L}(X, \mu, \nu) = F(C, X) - \mu\Psi(X) + \nu G(X)$. In turn, $\nabla_X \tilde{L}(X, \mu, \nu) = \nabla_X F(C, X) - \mu\nabla_X \Psi(X) + \nu\nabla_X G(X)$. The additional function is much simpler here, as $G(X)$ is obviously bounded above by $k(k-1)\ell^4/2$ and below by 0. Let $G(X)$ serve as the final regularization function when $\lambda = 1$, thus fulfilling the equality constraint, and let ν be uniquely determined at $\lambda = 0$ by some initial $\mathbb{R} \ni \nu^0 > 0$. Since it can be assumed that $G(k^0) = 0$ for any reasonable ℓ, and $\Psi(k^0) > 0$, the final hard clustering map is (where now $a = (k^0, h^0, \nu^0)$)

$$\tilde{\rho}_a(\lambda, X, \mu, \nu) = \begin{pmatrix} (1-\lambda)(X - k^0) + \lambda(\nabla_X \tilde{L}(X, \mu, \nu))^T \\ \Phi\left(\lambda, \mu, \Psi(X), h^0\right) \\ (1-\lambda)(\nu - \nu^0) + G(X) \end{pmatrix}.$$

This map is also a probability-one homotopy map. Taking $a = \left(k^0, h^0, \nu^0\right)$ the map $\rho(a, \lambda, X, \mu, \nu) = \tilde{\rho}_a(\lambda, X, \mu, \nu)$ is transversal to zero — $D_a\rho = (\lambda - 1)I$, a multiple of the identity matrix, hence $D\rho$ has full rank. For $0 \leqq \lambda \leqq 1$ and $\Psi(k^0) > 0$, Φ is a strictly increasing function of μ, unbounded above, and therefore $\Phi(0, \mu, \Psi(k^0), h^0) = 0$ uniquely determines μ. Thus $\tilde{\rho}_a = 0$ has a unique solution at $\lambda = 0$, and a straightforward calculation shows that $D_{(X,\mu,\nu)}\tilde{\rho}_a(0, X, \mu, \nu)$ is invertible at this solution. It is also clear from the construction of $\tilde{\rho}_a$ that $\tilde{\rho}_a(1, X, \mu, \nu) = 0$ is equivalent to the KKT conditions for the problem of minimizing $F(X, C)$ subject to the bounding constraint $\Psi \geqq 0$ and the regularization constraint $G = 0$. Therefore, $\tilde{\rho}_a$ satisfies conditions (1), (2), and (3) of Theorem 2, but the bounded γ condition (4) is not satisfied without further assumptions. Conditions for the zero curve γ being bounded (and hence reaching a solution at $\lambda = 1$) are addressed in the next lemma.

LEMMA 1. *Let $\Psi(k^0) > 0$, $G(k^0) = 0$, γ be a zero curve of $\tilde{\rho}_a(\lambda, X, \mu, \nu)$ emanating from $\left(0, k^0, \mu^0, \nu^0\right)$ along which $D\tilde{\rho}_a$ has full rank, and assume that ν is bounded along γ. Then γ itself is bounded.*

Proof. $\Phi(\lambda, \mu, \Psi(X), h^0) = 0$ along γ implies that, for $\lambda < 1$, $\mu > 0$ and $\Psi(X) > 0$, which in turn implies that X is bounded along γ. Since $0 \leqq \lambda \leqq 1$, and ν is assumed to also be bounded along γ, it suffices to prove that μ is bounded along γ. Assume otherwise, so there exists a sequence of points $\left(\lambda_i, X^i, \mu_i, \nu_i\right)$ on γ with $\mu_i \geqq 0$, $\mu_i \to \infty$. Passing to a subsequence if necessary, it may be assumed (by compactness) that $\left(\lambda_i, X^i, \nu_i\right) \to (\bar{\lambda}, \bar{X}, \bar{\nu})$.

Suppose $\Psi(\bar{X}) > 0$. Then because Φ is strictly increasing and unbounded above ((Mangasarian, 1976), (Watson, 1979b), (Watson, 2001)) $\Phi\big(\bar{\lambda},\, \mu_i,\, \Psi(\bar{X}),\, h^0\big) \to 0$ implies $\{\mu_i\}$ is bounded, a contradiction. Hence $\Psi(\bar{X}) = 0$ must obtain.

Suppose then that $\Psi(\bar{X}) = B - \|\bar{X}\|^2 = 0$ and $\bar{\lambda} > 0$. In this case $\nabla\Psi(\bar{X}) = -2\bar{X} \neq 0$ and (from the first component of $\tilde{\rho}_a = 0$)

$$\mu_i \nabla\Psi(\bar{X}) \to \frac{1-\bar{\lambda}}{\bar{\lambda}}\big(\bar{X} - k^0\big)^T + \nabla_X F(C, \bar{X}) + \bar{\nu}\nabla G(\bar{X})$$

$\Longrightarrow \{\mu_i\}$ is bounded, a contradiction.

The remaining case is $\Psi(\bar{X}) = 0$ and $\bar{\lambda} = 0 \Longrightarrow \nabla\Psi(\bar{X}) \neq 0$ and $\lambda_i\mu_i\big(\nabla\Psi(\bar{X})\big)^T \to \bar{X} - k^0 \Longrightarrow w\big(-\nabla\Psi(\bar{X})\big)^T = k^0 - \bar{X}$ for some $w \geqq 0$. Since $-\Psi(X)$ is convex, $-\nabla\Psi(\bar{X})(k^0 - \bar{X}) \leqq -\Psi(k^0) - (-\Psi(\bar{X})) < 0$. Then $0 \geqq (k^0 - \bar{X})^T\big(-\nabla\Psi(\bar{X})\big)^T w = (k^0 - \bar{X})^T(k^0 - \bar{X}) > 0$, a contradiction. Therefore μ is bounded along γ. ■

Lemma 1 directly yields the next homotopy convergence theorem.

THEOREM 4. *Using the notation of this section, define* $\tilde{\rho} : \mathbb{R}^{kd} \times (0,\infty) \times (0,\infty) \times [0,1) \times \mathbb{R}^{kd} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^{kd+2}$ *by*

$$\tilde{\rho}\big(k^0, h^0, \nu^0, \lambda, X, \mu, \nu\big) = \begin{pmatrix} (1-\lambda)(X - k^0) + \lambda(\nabla_X \tilde{L}(X, \mu, \nu))^T \\ \Phi\left(\lambda, \mu, \Psi(X), h^0\right) \\ (1-\lambda)(\nu - \nu^0) + G(X) \end{pmatrix}.$$

Let $a = \big(k^0, h^0, \nu^0\big)$ *and* $\tilde{\rho}_a(\lambda, X, \mu, \nu) = \tilde{\rho}\big(k^0, h^0, \nu^0, \lambda, X, \mu, \nu\big)$. *Then* $\tilde{\rho}$ *is transversal to zero, and for almost all* $a \in \mathbb{R}^{kd} \times (0,\infty) \times (0,\infty)$ *there exists a zero curve* γ *of* $\tilde{\rho}_a$, *emanating from* $\big(0, k^0, h^0, \nu^0\big)$, *along which the* $(kd+2) \times (kd+3)$ *Jacobian matrix* $D\tilde{\rho}_a$ *has full rank, that does not intersect itself and is disjoint from any other zeros of* $\tilde{\rho}_a$. *If* k^0 *satisfies* $\Psi(k^0) > 0$, $G(k^0) = 0$ *and* ν *is bounded along* γ, *then* γ *accumulates at a point* $(1, \bar{X}, \bar{\mu}, \bar{\nu})$, *where* $(\bar{X}, \bar{\mu}, \bar{\nu})$ *is a KKT point for the constrained clustering problem*

$$\min_X F(C, X) \quad \textit{subject to} \quad -\Psi(X) \leqq 0, \quad G(X) = 0.$$

Furthermore, if rank $D\tilde{\rho}_a(1, \bar{X}, \bar{\mu}, \bar{\nu}) = kd + 2$, *then the curve* γ *connecting* $\big(0, k^0, h^0, \nu^0\big)$ *to* $(1, \bar{X}, \bar{\mu}, \bar{\nu})$ *has finite arc length.*

3.3. Alternative linear bounding constraints.

An alternative bounding function replaces the nonlinear bounding constraint $\Psi \geqq 0$ with a linear bounding constraint. Define $\tilde{\Psi} : [0,1) \times \mathbb{R}^{kd} \to \mathbb{R}^{kd+1}$ by $\tilde{\Psi}(\lambda, X) = AX - \hat{B} - (1-\lambda)b^0$, so that $\tilde{\Psi} \leqq 0$ implies X is bounded, where $A \in \mathbb{R}^{(kd+1)\times kd}$ is constructed to have full column rank, $\hat{B} \in \mathbb{R}^{kd+1}$, and $\mathbb{R}^{kd+1} \ni b^0 > 0$ is chosen such that $Ak^0 - \hat{B} - b^0 < 0$. In this formulation, the Lagrange multipliers are $\mu \in \mathbb{R}^{kd+1}$, $\mathbb{R}^{kd+1} \ni h^0 > 0$, and $D_X\tilde{\Psi}(\lambda, X) = A$. Define $\Omega : [0,1) \times \mathbb{R}^{kd+1} \times \mathbb{R}^{kd+1} \to \mathbb{R}^{kd+1}$ by

$$\Omega_i\left(\lambda, \mu, -\tilde{\Psi}(\lambda, X)\right) = \Phi\left(\lambda, \mu_i, -(\tilde{\Psi}(\lambda, X))_i, h_i^0\right) \quad \text{for } 1 \leqq i \leqq kd+1,$$

yielding the homotopy map

$$\rho_a(\lambda, X, \mu, \nu) = \begin{pmatrix} (1-\lambda)(X-k^0) + \lambda(\nabla_X L(X,\mu,\nu))^T \\ \Omega\left(\lambda, \mu, -\tilde{\Psi}(\lambda, X)\right) \\ (1-\lambda)(\nu-\nu^0) + G(X) \end{pmatrix},$$

where $L(X,\mu,\nu) = F(C,X) + \mu^T\tilde{\Psi}(\lambda,X) + \nu\, G(X)$, and $a = \left(k^0, b^0, h^0, \nu^0\right)$.

Letting $e = (1,\ldots,1)^T \in \mathbb{R}^{kd}$ and I be the $kd \times kd$ identity matrix, a suitable A and $\hat{B}$ would be

$$A = \begin{pmatrix} I \\ -e^T \end{pmatrix}, \qquad \hat{B} = \begin{pmatrix} B\,e \\ Bkd \end{pmatrix}.$$

The disadvantage to using this map is that it involves a doubling of the number of dimensions to be tracked, involving an increase in the number of computations to be performed at each step. The advantage is, compared to the previous map $\tilde{\rho}_a$, homotopy zero curves based on linear constraints tend to be shorter and easier to track than those based on nonlinear constraints, though this depends on how well the homotopy map is designed. As for $\tilde{\rho}_a$, a homotopy algorithm using the map ρ_a is provably globally convergent under mild assumptions (that the link constraint function $F(X,C)$ has a feasible local minimum point, the initial solution k^0 satisfies the bounding ($\tilde{\Psi} \leqq 0$) and regularity ($G = 0$) constraints, and ν is bounded along γ), satisfying all four requirements of Theorem 2 for global convergence.

4. Experimental Results.

Figure 3 shows the homotopy method for a simple synthetic dataset, involving 200 points gathered from four Gaussian distributions. The aim is to find two clusters from this dataset. There are multiple natural clusterings possible, depending on whether the clusters are organized horizontally or vertically, among other options. The constraints are carefully prepared in such a way that half of them are must-link, taken from two different initial clusters, and half of them are cannot-link, taken from the same initial clusters. Thus, the clusters are forced to reorganize as λ is varied. The solid (green) lines denote the must-link constraints and the dashed (pink) lines denote the cannot-link constraints. During the homotopy curve tracking, the cluster prototypes smoothly traverse the space and finally settle to a position where a maximum number of constraints are satisfied, in this case, the global maximum.

Figure 4 shows the potential of the homotopy method to track the zero curve γ of the homotopy map through turns in λ. One dimension of a cluster centroid was tracked against λ for this figure.

Experiments to discover the effectiveness of the homotopy tracking algorithm with the proposed homotopy map $\tilde{\rho_a}$, as compared to popular existing constrained clustering algorithms, are presented here. The homotopy map ρ_a produces results qualitatively similar to those with $\tilde{\rho}_a$, and thus ρ_a is not considered further. The constraints used involve combinations of ML and MNL constraints (problems involving solely ML constraints are fairly straightforward polynomial time graphing problems).

The existence of MNL constraints in the constraint sets is crucial to understanding the complexity of the test problems. Davidson et al. (2006) state that as a rough rule of thumb a set of constraints can be understood as fundamentally "difficult" for these iterative K-means approaches if any single datapoint appears in k or more MNL constraints. As such, for each dataset presented here, both an "easy" and a "difficult" set of constraints were generated. The "easy" constraint

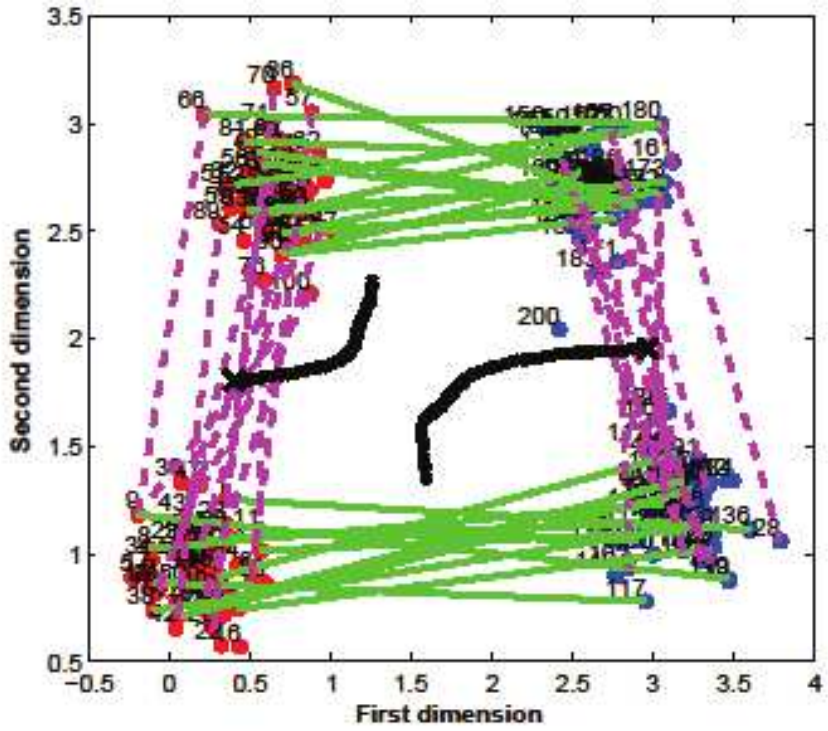

Figure 3. 200 points, two clusters, 50 constraints. The cluster prototypes change their positions during the tracking. All the constraints are satisfied at the final cluster prototypes.

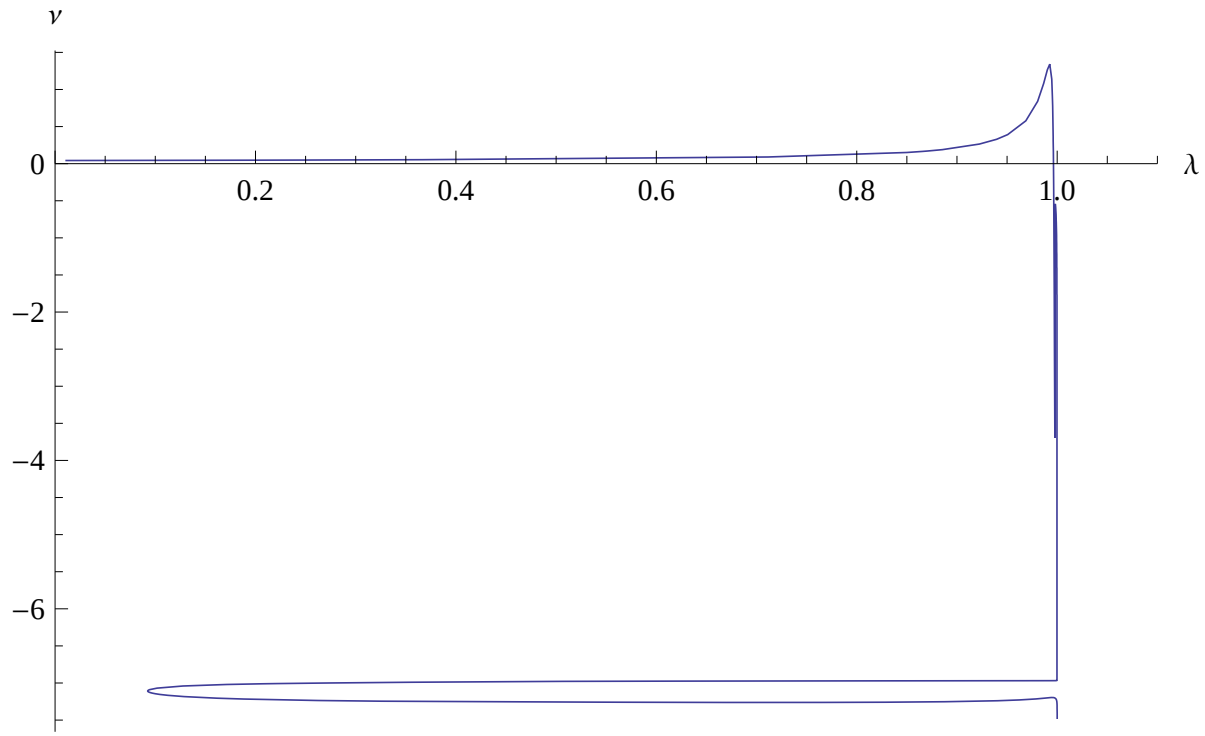

Figure 4. A turn in λ is tracked here to show the power of the homotopy method to deal with sharp turns in the zero curve γ. ν is tracked here to demonstrate this turning behavior.

set involves one hundred constraints such that no datapoint appears more than $k-1$ times in

a mix of ML and MNL constraints. The "difficult" constraint set, also one hundred constraints, involves predominately MNL constraints, and guarantees that at least one datapoint is involved in k MNL constraints. In both cases, the generated constraints were completely random, with no a priori knowledge about how well the generated constraints would guide the algorithms to a correct solution.

The datasets involved are all taken from the UCI machine learning dataset repository (Bache & Lichman, 2013), representing a balanced selection of moderately easy clustering problems without constraints, and should demonstrate some of the key differences between the homotopy algorithm developed here and the K-means algorithms used previously. The datasets are "Liver Disorders" (liver), "Pima Indians Diabetes" (pima), "Steel Plates Faults" (faults), "Wine" (wine), "Iris" (iris), "Ionosphere" (iono), "Glass Identification" (glass), and "PAMAP2 Physical Activity Monitoring" (pamap). The datasets "faults" and "pamap" were modified in the following manner: The first three classification categories of the dataset "faults" were treated as additional data, and the last classification category was used for classification. The dataset "pamap" was modified by eliminating all data points of the "0" classification (as recommended by the contributors) and any data point with a "NaN" data value. See Table 1 for the relevant details for each dataset.

Table 1
Dataset summary

	No. Instances	No. Features	No. Categories
iris	150	4	3
wine	178	12	3
glass	214	10	6
liver	345	6	2
iono	351	34	2
pima	768	7	2
faults	1941	31	2
pamap	175498	53	12

The K-means algorithms used for the comparison are those presented by Bilenko et al. (2004): metric pairwise constrained K-means (MPCK-means), metric learning K-means without pairwise constraints (MK-means), and pairwise constrained K-means without metric learning (PCK-means). The standard K-means result is also presented as a baseline. These algorithms were chosen for several reasons. First, K-means is by far the most popular clustering algorithm, if only because of its intuitive approach and ease of programming; thus, K-means algorithms modified for constrained clustering are the most likely to be employed by researchers who are interested in constrained clustering problems. Second, these constrained K-means algorithms minimize a summed penalty function based on the distance from the cluster centroids to the data points assigned to that cluster. While this penalty function may be discrete, it is still similar enough to the penalty function presented here to make comparisons between these algorithms and the homotopy approach reasonable.

Table 2 shows the adjusted Rand index of each dataset measured against the proper classification and the number of violated constraints in parentheses, for each algorithm discussed here, for 100 "easy" constraints. Table 3 shows the same data for 100 "hard" constraints. Note that

these tables only reveal the adjusted Rand index for the final clustering found by the homotopy tracking run, not the best across the entire zero trace. Each experiment was conducted 10 times with different random seeds, with the best result being shown here.

Table 4 shows the adjusted Rand index for each dataset measured against the proper classification for 100 "faulty" constraints. The faulty constraints are generated using the easy constraint set by replacing the final 30 constraints with incorrect constraints (simply converting must-link constraints to cannot-link constraints, and vice versa), to simulate unreliable constraints. The algorithms are then run on each dataset, with the best Dunn Index found along the trace generated by the homotopy method used to determine (according to the clustering assumption) the best clustering found using the faulty constraints. Again the best result from 10 experiments is reported.

For each of these experiments, the dataset "pamap" used 500 constraints, as 100 was not enough to provide sufficient differentiation among the results. Thus, 150 constraints were made incorrect for the experiments shown in Table 4.

Table 2
final clustering adjusted Rand index, "easy" constraints, constraint violations in parenthesis

	K-means	MK-means	PCK-means	MPCK-means	Homotopy
iris	0.7302 (43)	**0.8857** (**3**)	0.5195 (26)	0.5234 (26)	0.8841 (8)
wine	0.3711 (32)	**0.8309** (**3**)	0.3420 (19)	0.6211 (11)	0.4377 (25)
glass	0.2258 (56)	0.2482 (28)	0.1720 (**13**)	0.1824 (**13**)	**0.2812** (43)
liver	−0.0064 (53)	−0.0036 (42)	−0.0042 (41)	−0.0045 (43)	**0.0040** (**0**)
iono	0.1776 (50)	0.1776 (21)	0.1122 (41)	0.1122 (41)	**0.2450** (**41**)
pima	0.0744 (42)	0.1040 (17)	0.0510 (46)	0.0164 (56)	**0.1322** (**2**)
faults	**0.1358** (76)	0 (73)	0.1109 (**5**)	−0.0837 (**5**)	0.1084 (40)
pamap	**0.6457** (19)	0.3046 (18)	0.5660 (7)	0.2695 (**5**)	**0.6457** (19)

Table 3
final clustering adjusted Rand index, "hard" constraints, constraint violations in parenthesis

	K-means	MK-means	PCK-means	MPCK-means	Homotopy
iris	0.7302 (43)	0.8857 (3)	0.8015 (**0**)	**0.9222** (**0**)	0.9216(**0**)
wine	0.3711 (32)	0.8170 (4)	0.4451 (**0**)	**0.8636** (**0**)	0.4154 (17)
glass	0.2258 (56)	0.2482 (31)	0.2608 (**0**)	0.2102 (**0**)	**0.2812** (27)
liver	−0.0064 (53)	−0.0046 (63)	0.0102 (**0**)	0.0253 (**0**)	**0.0366** (**0**)
iono	0.1776 (50)	0.1776 (49)	0.1413 (49)	0.1413 (49)	**0.1943** (**45**)
pima	0.0744 (42)	**0.1043** (39)	0.0696 (63)	0.0422 (40)	0.0775 (**0**)
faults	**0.1358** (76)	0 (84)	0.1159 (**0**)	−0.0832 (**0**)	0.1104 (41)
pamap	**0.6457** (19)	0.2929 (**18**)	0.6454 (21)	0.2700 (19)	0.6315 (19)

Table 4
final clustering adjusted Rand index, "faulty" constraints

	K-means	MK-means	PCK-means	MPCK-means	Homotopy
iris	0.7302	**0.8857**	0.4677	0.4209	0.7865
wine	0.3711	**0.7994**	0.3081	0.6064	0.4720
glass	0.2258	0.2412	0.2142	0.1795	**0.2812**
liver	−0.0064	−0.0046	−0.0031	−0.0046	**0.0306**
iono	0.1776	0.1776	0.0705	0.0768	**0.1874**
pima	0.0744	**0.1073**	0.0681	0.0283	0.0744
faults	**0.1358**	−0.1028	0.1166	−0.0850	**0.1358**
pamap	**0.6457**	0.2084	0.6439	0.3198	**0.6457**

The chief purpose of the homotopy method, however, is not to compare the quality of the final solution found with other methods. Instead, the homotopy method is intended to show the entire trace of solutions presented as the constraints are gradually satisfied. To that end, Figures 5–8 demonstrate the Dunn index (Dunn, 1973) against the number of violated constraints with respect to arc length as λ varies (not necessarily monotonically) from 0 to 1 for two datasets (iris and liver) for the two constraint sets ("easy" and "hard"), comparing them against the Dunn index of the partitions found by the other algorithms (represented as "K" for K-means, "M" for MK-means, "P" for PCK-means, and "MP" for MPCK-means).

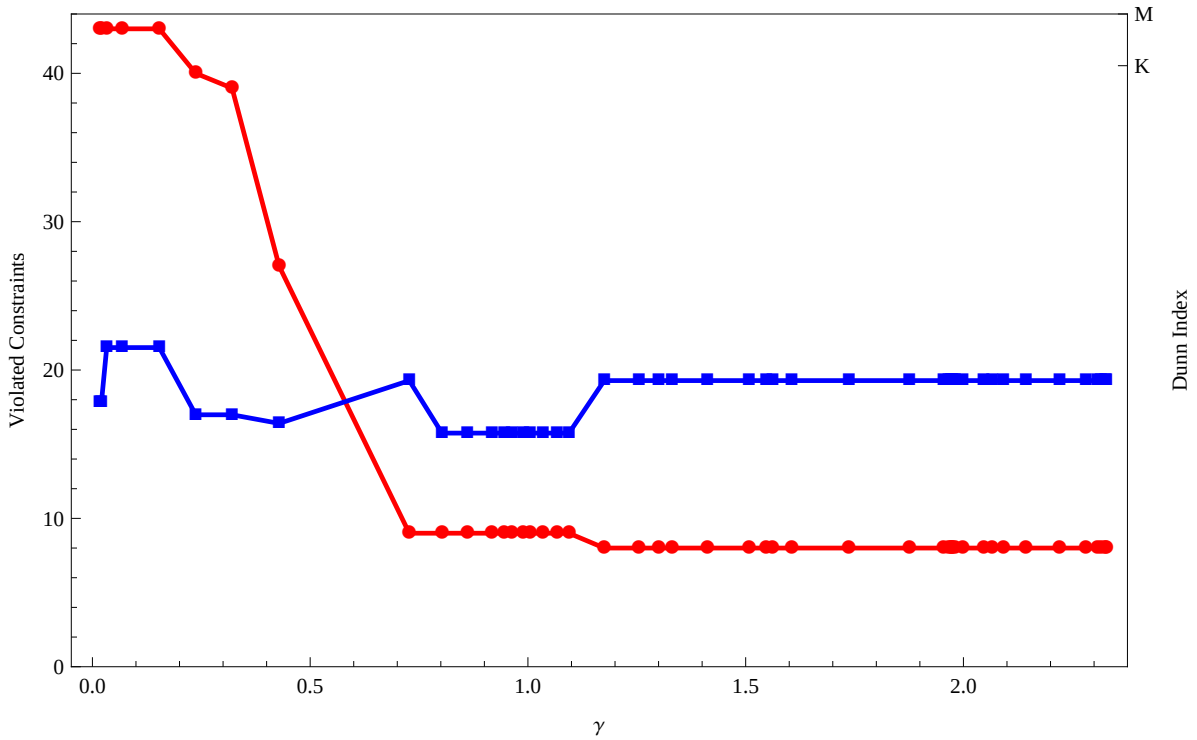

Figure 5. The dataset "iris" with "easy" constraints. The Dunn index is tracked against the arc length of γ (*), while the number of violated constraints is also tracked (•). The Dunn Index for other methods is shown for comparison. Note that the Dunn Index is not applicable to nonconvex clusterings, so methods with nonconvex results are not shown.

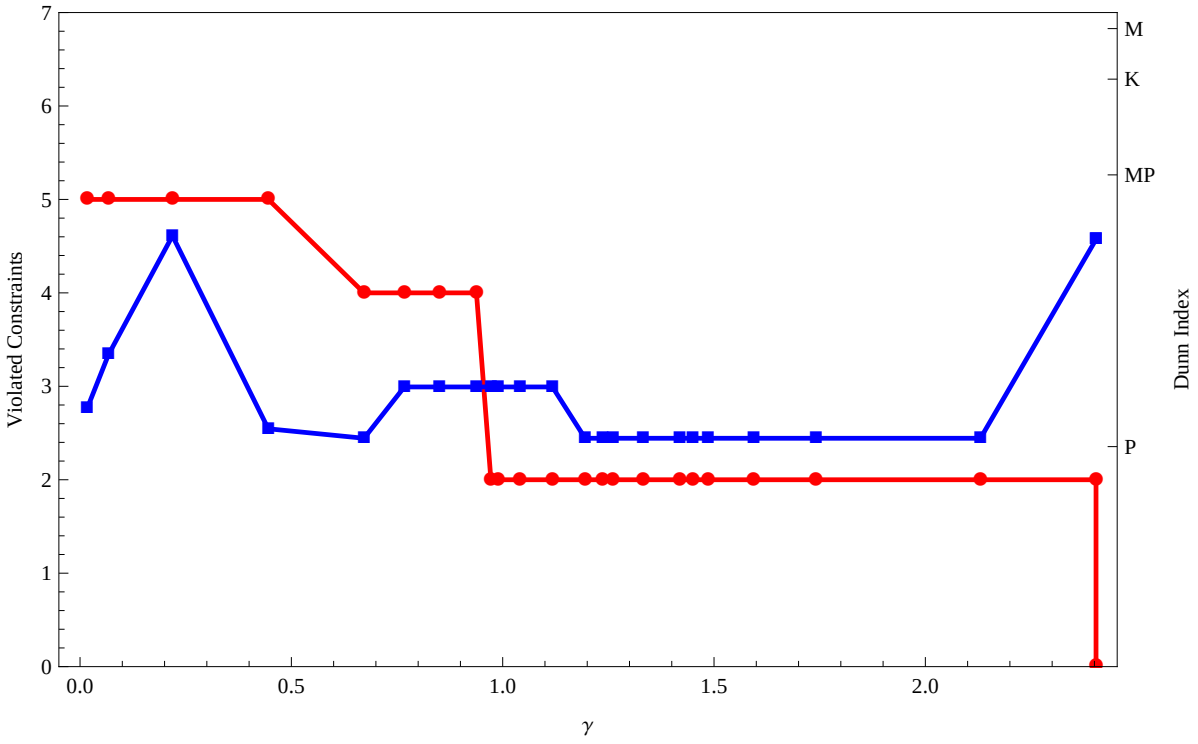

Figure 6. The dataset "iris" with "hard" constraints. The Dunn index is tracked against the arc length of γ (*), while the number of violated constraints is also tracked (•). The Dunn Index for other methods is shown for comparison.

Finally, in order to understand the effects that different datasets have on the execution time of a purely serial version of the code, experiments were run on a randomly generated Gaussian dataset with randomly generated constraints. The results, shown in Figures 9–12, reflect the bottleneck that the linear algebra of the homotopy method imposes on the process. Parallelizing this linear algebra with the ATLAS BLAS routines resulted in significant speedup. As expected, the growth of the execution time of the serial version of the code is linear with respect to the number of constraints and polynomial (d^2k^2) with the number of dimensions d and clusters k. Given that the number of clusters tends to remain fairly small even in large datasets, this makes for a fairly competitive constrained clustering algorithm, especially as the number of points in the dataset grows.

5. Discussion.

It is worth noting immediately that three things set the homotopy algorithm apart from the K-means algorithms presented here. *First*, for the K-means algorithms, the ordering of the constraints plays a nonnegligible role in the quality of the final result, meaning that finding the best result theoretically involves trying every permutation of a given constraint set (which is not computationally feasible). In fairness, there is also no guarantee that the homotopy algorithm will find the global optimum solution. For the homotopy algorithm, the ordering of the constraints is unimportant. *Second*, not only are problems involving concentrations of MNL constraints involving the same datapoint not qualitatively more "difficult" for the homotopy algorithm, but, since distances only need to be calculated once per iteration, problems involving concentrations of datapoints are computationally less intensive than problems where the constraints are more diverse, at least until

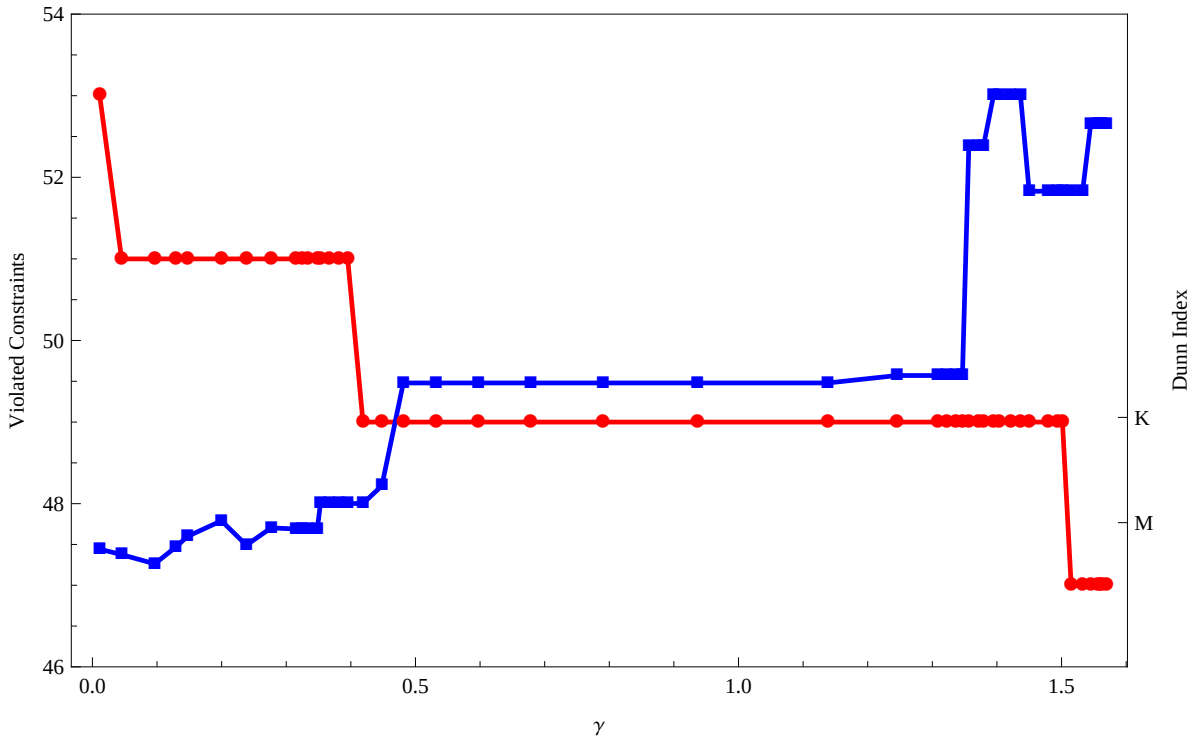

Figure 7. The dataset "liver" with "easy" constraints. The Dunn index is tracked against the arc length of γ (*), while the number of violated constraints is also tracked (•). The Dunn Index for the other methods are shown for comparison. Note that the Dunn Index is not applicable to nonconvex clusterings, so methods with nonconvex results are not shown.

each datapoint is involved in at least one constraint. *Finally*, the homotopy algorithm based on the homotopy map $\tilde{\rho_a}$, like the K-means algorithm, is limited to convex clusterings, which for some datasets can be potentially debilitating. In contrast, the adapted K-means algorithms presented here distinguish between cluster assignment and cluster centroids, which allows for nonconvex clusterings. There is no inherent reason why a homotopy map could not be developed for nonconvex clusterings, and doing so is a topic for further research.

For the "easy" constraint set (Table 2), the homotopy algorithm performed among the best for the datasets "glass", "liver", "iono", and "pima", reasonably well for both the datasets "iris" and "faults", and poorly for the dataset "wine", in terms of the adjusted Rand index. It tied with the K-means as the best algorithm for "pamap", in this case because it didn't vary appreciably from the original configuration. The "hard" constraints (Table 3) showed an improvement for the dataset "iris" and a degradation for the dataset "faults", along with a minor degradation for the dataset "pamap".

A few of the strengths and weaknesses of the homotopy method are visible from these results. First, number of constraints satisfied is not in itself a sufficient measure of comparison between algorithms. In several datasets, including the "hard" "glass", "faults", "pamap", and "pima" results (Table 3), the classifications closest to the true classifications were not those with the greatest number of satisfied constraints. In particular, the dataset "faults", with its large number of instances, is clearly less suited to a constraint set of 100 random elements than the other small datasets portrayed here. That said, it's interesting that as the size of the dataset grows, convex clustering methods regularly provide better results than nonconvex clustering methods given a constant number of constraints. This is probably due to the interaction of the clustering assumption

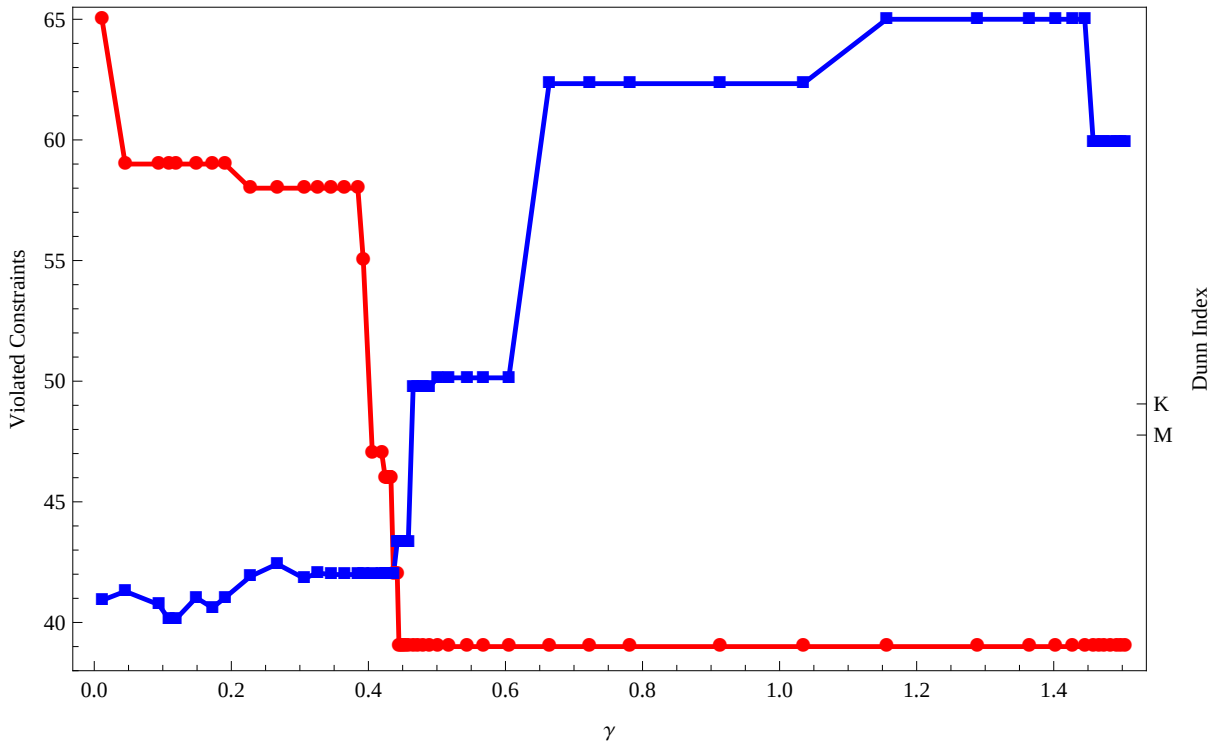

Figure 8. The dataset "liver" with "hard" constraints. The Dunn index is tracked against the arc length of γ (*), while the number of violated constraints is also tracked (•). The Dunn Index for the other methods are shown for comparison. Note that the Dunn Index is not applicable to nonconvex clusterings, so methods with nonconvex results are not shown.

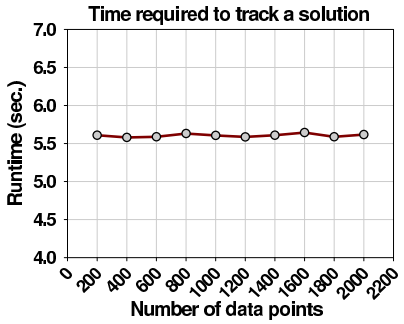

Figure 9. Runtime of the algorithm as a function of the number of data points.

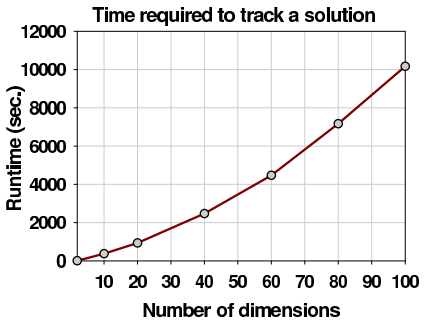

Figure 10. Runtime of the algorithm as a function of the number of dimensions.

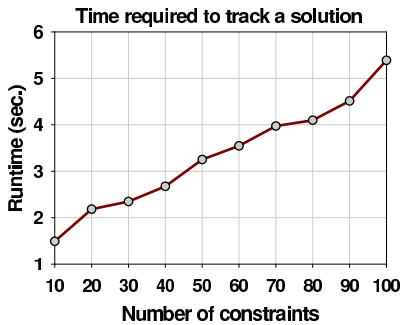

Figure 11. Runtime of the algorithm as a function of the number of constraints.

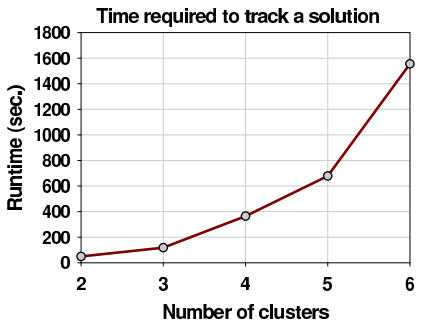

Figure 12. Runtime of the algorithm as a function of the number of clusters.

with the satisfaction of constraints, which increases the potential power of a given constraint by including nearby points in the changes of partition that result from it. This clearly demonstrates a strength of the homotopy method over the nonconvex alternatives presented here, and the numbers show the strength of the method against even the other convex clustering methods.

One of the major weaknesses, in terms of using the final result of the method as a partition without further analysis, similarly shows up in the "hard" "pima" results. Even though all constraints were satisfied, the final solution was not the best available along the trace, and the best solution failed to satisfy a fairly large number of constraints. "Pima" is a relatively large dataset, again showing the problem with applying small constraint sets to large datasets. The best the homotopy method (or any of the clustering methods presented) can do is a local optimum, and that's clearly not always going to be the global optimum.

All of the above is an aside to the true goal of the homotopy map. The adjusted Rand index is a good tool for a posteriori judgment of clusters, but in practice the classification is not already in hand. All that is known is intercluster and intracluster distances, with limited auxiliary information. For this, the Dunn index is a reasonable tool for measuring the validity of multiple clusterings of the same dataset. Figures 5–8 thus show the utility of the homotopy map $\tilde{\rho_a}$ without reference to the "correct" clustering, simply the constraints that a researcher may reasonably discover on their own, compared to the Dunn index yielded by the other methods. Note that in all cases, the Dunn index improves over that of the original K-means clustering as constraints are satisfied. Not only does satisfying the given set of constraints thus improve the quality of the discovered partitions, but the improvement to the Dunn index (or other cluster metric) can be viewed as a standard to judge imposed constraints. Here, the constraints are all known to be valid, but this is not necessarily the case in real world applications. Even when continuing to satisfy constraints ultimately reduces the viability of the partition, the result can still be judged based on the dual criteria: satisfaction of the clustering assumption and satisfaction of constraints.

Note the behavior of the Dunn index in Figure 4, and to a lesser extent all of the figures presented. As the number of constraints satisfied improves, the Dunn index initially increases and then sharply decreases. A region of a slightly larger Dunn index is then passed through before the end, where the satisfaction of all constraints results in a rapid increase in clustering quality. Assuming a proper classification has good clustering, this indicates that the given constraints are highly trustworthy, without recourse to the actual classification of the dataset: the constraints prevent convergence to a local minimum (twice) in favor of the minimum favored by the given constraint set, which clearly demonstrates stronger cluster properties. The Dunn index is not the only way to measure the internal strength of a given partition, of course, but it suffices for the purpose here.

The new homotopy theory for constrained clustering problems uses state-of-the-art numerical analysis to characterize solutions of multicriteria 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 also it helps formally understand the internal workings of machine learning techniques. Future work involves designing homotopy maps for other machine learning problems such as the information bottleneck, time series segmentation, and transfer learning.

References.

M. S. Baghshah and S. B. Shouraki. Learning low-rank kernel matrices for constrained clustering. *Neurocomput.*, vol. 74, no. 12–13, 2201–2211, 2011

K. Bache and M. Lichman. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. 2013.

M. F. Balcan and A. Blum. A discriminative model for semi-supervised learning. *J. ACM.*, vol. 57, no.3, 19:1–19:46, 2010

S. Basu, I. Davidson, and K. Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, Chapman and Hall, 2008

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

O. Chapelle, B. Scholkopf, and A. Zien. *Semi-Supervised Learning*, 1st ed., Cambridge, MA, MIT Press. 2008

S. N. Chow, J. Mallet-Paret, and J. A. Yorke. Finding zeros of maps: homotopy methods that are constructive with probability-one. *Math. Comput.*, vol. 32, 887–899, 1978

A. Corduneanu and T. Jaakkola. "Continuation methods for mixing heterogeneous sources." In *UAI '02*, 111–118, 2002

B. R. Dai, C. R. Lin, and M. S. Chen. Constrained data clustering by depth control and progressive constraint relaxation. *The VLDB Journal*, vol. 16, 201–217, 2007

I. Davidson. "Two approaches to understanding when constraints help clustering." In *KDD '12*, 1312–1320, 2012

I. Davidson and S. S. Ravi. "Clustering with constraints: feasibility issues and the K-means algorithm." In *SDM '05*, 201–211, 2005

I. Davidson and S. S. Ravi. "Identifying and generating easy sets of constraints for clustering." In *AAAI '06*, 336–341, 2006

I. Davidson and S. S. Ravi. The complexity of non-hierarchical clustering with instance and cluster level constraints. *Data Min. Knowl. Discov.*, vol. 14, 25–61, 2007

A. Demiriz, K. Bennett, and P. Bradley. K. Wagstaff, I. Davidson, and S. Basu, editors. "Chapter 9: Using assignment constraints to avoid empty clusters in K-means clustering." In *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, 1st ed. Chapman & Hall/CRC, 2008

J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, vol. 3, no. 3, 32–57, 1973

A. Frank and A. Asuncion. "UCI Machine Learning Repository." Available online at http://archive.ics.uci.edu/ml. 2010

D. Gondek and T. Hofmann. Non-redundant data clustering. *Knowl. Inf. Syst.*, vol. 12, no. 1, 1–24, 2007

P. He, X. Xu, and L. Chen. "Constrained clustering with local constraint propagation." In *ECCV '12*, 223–232, 2012

M. S. Hossain, N. Ramakrishnan, I. Davidson, and L. T. Watson. How to alternatize a clustering algorithm. *Data Mining and Knowledge Discovery*, 1–32, 2012

M. S. Hossain, S. Tadepalli, L. T. Watson, I. Davidson, R. Helm and N. Ramakrishnan. "Unifying dependent clustering and disparate clustering for non-homogeneous data." In *KDD '10*, 593–602, 2010

S. Ji, L. T. Watson, and L. Carin. Semisupervised learning of hidden Markov models via a homotopy method. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 31, 275–287, 2009

P. Luo, F. Zhang, H. Xiong, Y. Xiong, and Q. He. "Transfer learning from multiple source domains via consensus regularization." In *CIKM '08*, 103–112, 2008

O. Mangasarian. Equivalence of the complementarity problem to a system of nonlinear equations. *SIAM Journal on Applied Mathematics*, vol. 31, no. 1, 89–92, 1976

O. Mangasarian. *Nonlinear Programming*, McGraw-Hill, 1969

M. Meila. "Comparing clusterings: an axiomatic view." In *ICML '05*, 577–584, 2007

W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, vol. 66, no 336: 846–850, 1971

Y. Sato and M. Iwayama. "Interactive constrained clustering for patent document set." In *PaIR '09*, 17–20, 2009

J. Sese, Y. Kurokawa, M. Monden, K. Kato, and S. Moroshita. Constrained clusters of gene expression profiles with pathological features. *Bioinformatics*, vol. 20, no. 17, 3137–3145, 2004

K. Sinha and M. Belkin. "The value of labeled and unlabeled examples when the model is imperfect." In *Advances in Neural Information Processing Systems 20*, 2008

M. E. Taylor, G. Kuhlmann, and P. Stone. "Autonomous transfer for reinforcement learning." In *AAMAS '08*, vol.1, 283–290, 2008

X. Wang and I. Davidson. "Flexible constrained spectral clustering." In *KDD '10*, 563–572, 2010

L. T. Watson. A globally convergent algorithm for computing fixed points of C^2 maps. *Appl. Math. Comput.*, vol. 5, 297–311, 1979

L. T. Watson. Numerical linear algebra aspects of globally convergent homotopy methods. *SIAM Rev*, vol. 28, 529–545, 1987

L. T. Watson. Probability-one homotopies in computational science. *J. Comput. Appl. Math.*, vol. 140, 785–807, 2002

L. T. Watson. Solving the nonlinear complimentarity problem by a homotopy method. *SIAM J. Control Optimization*, vol. 17, 36–46, 1979

L. T. Watson. Theory of globally convergent probability-one homotopies for nonlinear programming. *SIAM Jounal on Optimization*, vol. 11, no. 3, 761–780, 2001

L. T. Watson, M. Sosonkina, and A.P. Morgan. HOMPACK: a suite of codes for globally convergent homotopy algorithms. *ACM Trans. Math. Software*, vol. 13, 281–310, 1987

L.T. Watson, M. Sosonkina, R. C. Melville, A. Morgan, and H. Walker. Algorithm 777: HOMPACK90: a suite of Fortran 90 codes for globally convergent homotopy algorithms. *ACM Trans. Math. Software.*, vol. 23, 514–549, 1997

H. Yang and J. Callan. "A metric-based framework for automatic taxonomy induction." In *ACL '09*, vol. 1, 271–279, 2009

Q. Yang, Y. Chen, G.R. Xue, W. Dai, and Y. Yu. "Heterogeneous transfer learning for image clustering via the social web." In *ACL '09*, 1–9, 2009

D. Zhang, J. He, Y. Liu, L. Si, and R. Lawrence. "Multi-view transfer learning with a large margin approach." In *KDD '11*, 1208–1216, 2011