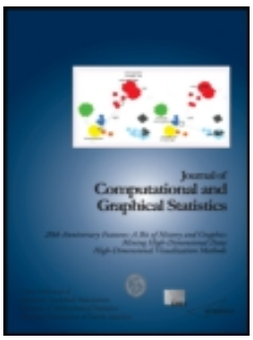


Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: http://www.tandfonline.com/loi/ucgs20

Splitting Methods for Convex Clustering

Eric C. Chi & Kenneth Lange

To cite this article: Eric C. Chi & Kenneth Lange (2015) Splitting Methods for Convex Clustering, Journal of Computational and Graphical Statistics, 24:4, 994-1013, DOI: 10.1080/10618600.2014.948181

To link to this article: http://dx.doi.org/10.1080/10618600.2014.948181

局部最小值问题，
局部最优利用凸松弛和
层次聚类可以得到全局最小值。
本文利用ADMM和AMA解决
凸聚类问题，提供了新范数？
并且利用真实数据模拟比较两种算法
，
AMA效率更高

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS

Splitting Methods for Convex Clustering

Eric C. CHI and Kenneth LANGE

Clustering is a fundamental problem in many scientific applications. Standard methods such as k-means, Gaussian mixture models, and hierarchical clustering, however, are beset by local minima, which are sometimes drastically suboptimal. Recently introduced convex relaxations of k-means and hierarchical clustering shrink cluster centroids toward one another and ensure a unique global minimizer. In this work, we present two splitting methods for solving the convex clustering problem. The first is an instance of the alternating direction method of multipliers (ADMM); the second is an instance of the alternating minimization algorithm (AMA). In contrast to previously considered algorithms, our ADMM and AMA formulations provide simple and unified frameworks for solving the convex clustering problem under the previously studied norms and open the door to potentially novel norms. We demonstrate the performance of our algorithm on both simulated and real data examples. While the differences between the two algorithms appear to be minor on the surface, complexity analysis and numerical experiments show AMA to be significantly more efficient. This article has supplementary materials available online.

Keywords: Alternating direction method of multipliers; Alternating minimization algorithm; Convex optimization; Hierarchical clustering; k-means; Regularization paths.

1. INTRODUCTION

In recent years, convex relaxations of many fundamental, yet combinatorially hard, optimization problems in engineering, applied mathematics, and statistics have been introduced (Tropp 2006). Good, and sometimes nearly optimal solutions, can be achieved at affordable computational prices for problems that appear at first blush to be computationally intractable. In this article, we introduce two new algorithmic frameworks based on variable splitting that generalize and extend recent efforts to convexify the classic unsupervised problem of clustering.

Lindsten, Ohlsson, and Ljung (2011) and Hocking et al. (2011) formulated the clustering task as a convex optimization problem. Given n points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in $\mathbb{R}^p$, they suggested

Eric C. Chi is Postdoctoral Research Associate, Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 (E-mail: *echi@rice.edu*). Kenneth Lange is Professor, Departments of Biomathematics, Human Genetics, and Statistics, University of California, Los Angeles, CA 90095 (E-mail: *klange@ucla.edu*).

Journal of Computational and Graphical Statistics, Volume 24, Number 4, Pages 994–1013
DOI: 10.1080/10618600.2014.948181
Color versions of one or more of the figures in the article can be found online at *www.tandfonline.com/r/jcgs*.

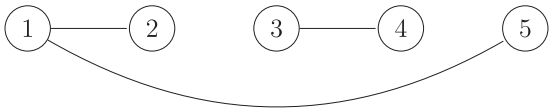


Figure 1. A graph with positive weights w_{12}, w_{15}, w_{34} and all other weights $w_{ij} = 0$.

minimizing the convex criterion 参考come on

由eg，根据他们的联系加权重

转置乘它

$$F_\gamma(\mathbf{U}) = \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{u}_i\|_2^2 + \gamma \sum_{i<j} w_{ij}\|\mathbf{u}_i - \mathbf{u}_j\|, \tag{1.1}$$

gamma可调节数

本文的范数用的是二范数

where γ is a positive tuning constant, $w_{ij} = w_{ji}$ is a nonnegative weight, and the ith column $\mathbf{u}_i$ of the matrix $\mathbf{U}$ is the cluster center (centroid) attached to point $\mathbf{x}_i$. Lindsten, Ohlsson, and Ljung (2011) considered an ℓ_q norm penalty on the differences $\mathbf{u}_i - \mathbf{u}_j$ while Hocking et al. (2011) considered ℓ_1, ℓ_2, and ℓ_∞ penalties. In the current article, an arbitrary norm defines the penalty.

ui是每一族的中心点
u1=u2，12归为一类

绝对值，分量p次方求和开p次方，向量分量最大值

The objective function bears some similarity to the fused lasso signal approximator (Tibshirani et al. 2005). When the ℓ_1 penalty is used in definition (1.1), we recover a special case of the general fused lasso (Hoefling 2010; Tibshirani and Taylor 2011). In the graphical interpretation of clustering, each point corresponds to a node in a graph, and an edge connects nodes i and j whenever $w_{ij} > 0$. Figure 1 depicts an example. In this case, the objective function $F_\gamma(\mathbf{U})$ separates over the connected components of the underlying graph. Thus, one can solve for the optimal $\mathbf{U}$ component by component. Without loss of generality, we assume the graph is connected.

When $\gamma = 0$, the minimum is attained when $\mathbf{u}_i = \mathbf{x}_i$, and each point occupies a unique cluster. As γ increases, the cluster centers begin to coalesce. Two points $\mathbf{x}_i$ and $\mathbf{x}_j$ with $\mathbf{u}_i = \mathbf{u}_j$ are said to belong to the same cluster. For sufficiently high γ, all points coalesce into a single cluster. Because the objective function $F_\gamma(\mathbf{U})$ in Equation (1.1) is strictly convex and coercive, it possesses a unique minimum point for each value of γ. If we plot the solution matrix $\mathbf{U}$ as a function of γ, then we can ordinarily identify those values of γ giving k clusters for any integer k between n and 1. In theory, k can decrement by more than 1 as certain critical values of γ are passed. Indeed, when points are not well separated, we observe that many centroids will coalesce abruptly unless care is taken in choosing the weights w_{ij}.

gamma足够大，第一项ui，ui都相等，第一项xi-u最小
ui为xi均值

严格凸，唯一最小解

The benefits of this formulation are manifold. As we will show, convex relaxation admits a simple and fast iterative algorithm that is guaranteed to converge to the unique global minimizer. In contrast, the classic k-means problem has been shown to be NP-hard (Aloise et al. 2009; Dasgupta and Freund 2009). In addition, the classical greedy algorithm for solving k-means clustering often gets trapped in suboptimal local minima (Forgy 1965; MacQueen 1967; Lloyd 1982).

Another vexing issue in clustering is determining the number of clusters. Agglomerative hierarchical clustering (Ward 1963; Johnson 1967; Lance and Williams 1967; Gower and Ross 1969; Murtagh 1983) finesses the problem by computing an entire clustering path. Agglomerative approaches, however, can be computationally demanding and tend to fall into suboptimal local minima since coalescence events are not reversed. The alternative

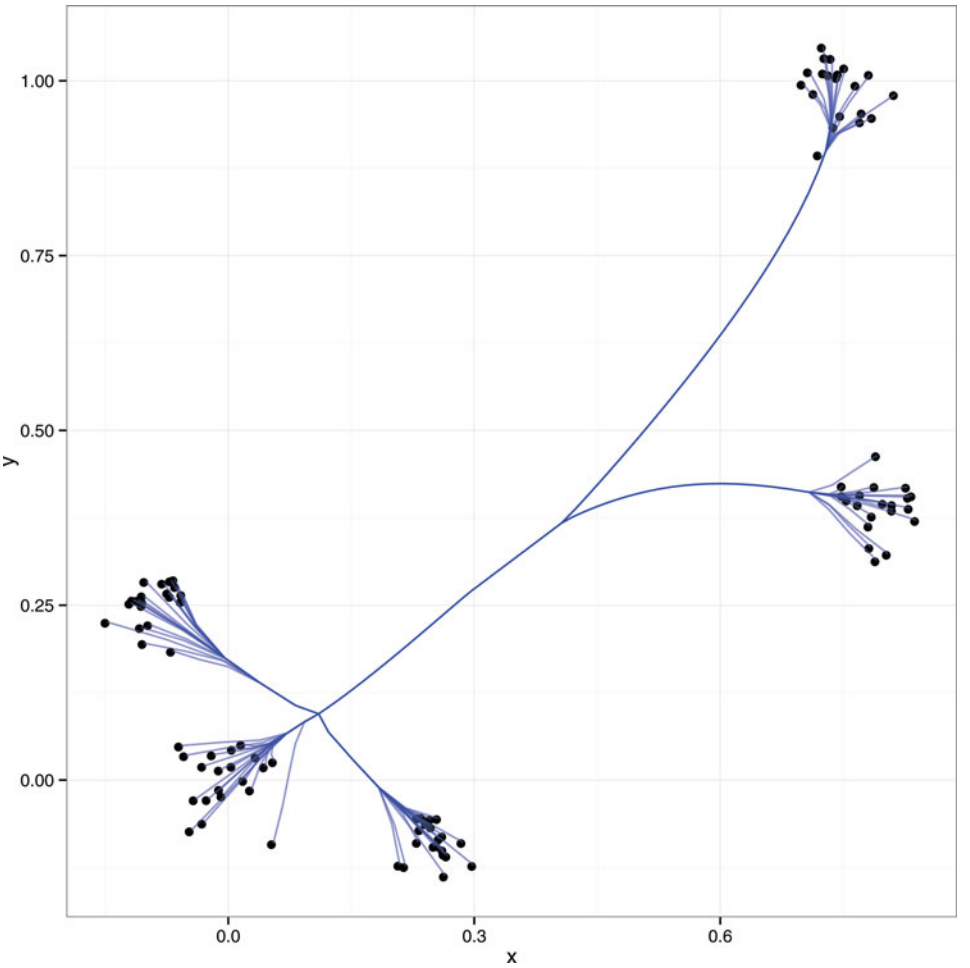


Figure 2. Cluster path assignment: The simulated example shows five well-separated clusters and the assigned clustering generated by the convex clustering algorithm under an ℓ_2-norm. The lines trace the path of the individual cluster centers as the regularization parameter γ increases.

convex relaxation considered here performs continuous clustering just as the lasso (Tibshirani 1996; Chen, Donoho, and Saunders 1998) performs continuous variable selection. Figure 2 shows how the solutions to the alternative convex problem traces out an intuitively appealing, globally optimal, and computationally tractable solution path.

1.1 CONTRIBUTIONS

Our main contributions are two new methods for solving the convex clustering problem. Relatively little work has been published on algorithms for solving this optimization problem. In fact, the only other article introducing dedicated algorithms for minimizing criterion (1.1) that we are aware of is Hocking et al. (2011). Lindsten, Ohlsson, and Ljung (2011) used the off-the-shelf convex solver CVX (CVX Research, Inc. 2012; Grant and Boyd 2008) to generate solution paths. Hocking et al. (2011) noted that CVX is useful

我们的主要贡献是解决凸聚类问题的两种新方法。
我们知道的另外一篇介绍最小化标准（1.1）专用算法的文章是Hocking等人（2011）。Lindsten、Ohlsson和Ljung（2011）使用CVX

他们针对三种最常见的规范引入了三种不同的算法

for solving small problems but a dedicated formulation is required for scalability. Thus, they introduced three distinct algorithms for the three most commonly encountered norms. Given the ℓ_1 norm and unit weights w_{ij}, the objective function separates, and they solved the convex clustering problem by the exact path following method designed for the fused lasso (Hoefling 2010). For the ℓ_1 and ℓ_2 norms with arbitrary weights w_{ij}, they employed subgradient descent in conjunction with active sets. Finally, they solve the convex clustering problem under the ℓ_∞ norm by viewing it as minimization of a Frobenius norm over a polytope. In this guise, the problem succumbs to the Frank–Wolfe algorithm (Frank and Wolfe 1956) of quadratic programming.

In contrast to this piecemeal approach, we introduce two similar generic frameworks for minimizing the convex clustering objective function with an arbitrary norm. One approach solves the problem by the alternating direction method of multipliers (ADMM), while the other solves it by the alternating minimization algorithm (AMA). The key step in both cases computes the proximal map of a given norm. Consequently, both of our algorithms apply provided the penalty norm admits efficient computation of its proximal map.

In addition to introducing new algorithms for solving the convex clustering problem, the current article contributes in other concrete ways: (a) We combine existing results on AMA and ADMM with the special structure of the convex clustering problem to characterize both of the new algorithms theoretically. In particular, the clustering problem formulation gives a minimal set of extra assumptions needed to prove the convergence of the ADMM iterates to the unique global minimum. We also explicitly show how the computational and storage complexity of our algorithms scales with the connectivity of the underlying graph. Examination of the dual problem enables us to identify a fixed step size for AMA that is associated with the Laplacian matrix of the underlying graph. Finally, our complexity analysis enables us to rigorously quantify the efficiency of the two algorithms so the two methods can be compared. (b) We provide new proofs of intuitive properties of the solution path. These results are tied solely to the minimization of the objective function objective function (1.1) and hold regardless of the algorithm used to find the minimum point. (c) We provide guidance on how to choose the weights w_{ij}. Our suggested choices diminish computational complexity and enhance solution quality. In particular, we show that employing k-nearest neighbor weights allows the storage and computation requirements for our AMA method to grow linearly in problem size.

1.2 RELATED WORK

The literature on clustering is immense; the reader can consult, for example, Hartigan (1975), Kaufman and Rousseeuw (1990), Mirkin (1996), Gordon (1999), Wu and Wunsch (2009) for a comprehensive review. The clustering function (1.1) can be viewed as a convex relaxation of either k-means clustering (Lindsten, Ohlsson, and Ljung 2011) or hierarchical agglomerative clustering (Hocking et al. 2011). Both of these classical clustering methods (Sørensen 1948; Sneath 1957; Ward 1963) come in several varieties. The literature on k-means clustering reports notable improvements in the computation (Elkan 2003) and quality of solutions (Kaufman and Rousseeuw 1990; Bradley, Mangasarian, and Street 1997; Arthur and Vassilvitskii 2007) delivered by the standard greedy algorithms. Faster methods for agglomerative hierarchical clustering have been developed as well

(Fraley 1998). Many statisticians view the hard cluster assignments of k-means as less desirable than the probabilistic assignments generated by mixture models (Titterington, Smith, and Makov 1985; McLachlan 2000). Mixture models have the advantage of gracefully assigning points to overlapping clusters. These models are amenable to an EM algorithm and can be extended to infinite mixtures (Ferguson 1973; Neal 2000; Rasmussen 2000).

Alternative approaches to clustering involve identifying components in the associated graph via its Laplacian matrix. Spectral clustering (Luxburg 2007) can be effective in cases when the clusters are nonconvex and linearly inseparable. Although spectral clustering is valuable, it does not conflict with convex relaxation. Indeed, Hocking et al. (2011) demonstrated that convex clustering can be effectively merged with spectral clustering. Although we agree with this point, the solution path revealed by convex clustering is meritorious in its own right because it partially obviates the persistent need for determining the number of clusters.

1.3 NOTATION AND ORGANIZATION

a常数，u向量，U矩阵，uj第j列，B大写字母表示集合

Throughout, scalars are denoted by lowercase letters (a), vectors by boldface lowercase letters ($\mathbf{u}$), and matrices by boldface capital letters ($\mathbf{U}$). The jth column of a matrix $\mathbf{U}$ is denoted by $\mathbf{u}_j$. At times, in our derivations, it will be easier to work with vectorized matrices. We adopt the convention of denoting the vectorization of a matrix ($\mathbf{U}$) by its lowercase letter in boldface ($\mathbf{u}$). Finally, we denote sets by uppercase letters (B).

矩阵向量化，把矩阵拼成一个列向量，方便计算

The rest of the article is organized as follows. We first characterize the solution path theoretically. Previous papers take intuitive properties of the path for granted. We then review the ADMM and AMA algorithms and adapt them to solve the convex clustering problem. Once the algorithms are specified, we discuss their convergence and computational and storage complexity. Practical issues such as weight choice and refinements such as accelerating the algorithms are discussed next. We then present some numerical examples of clustering. The article concludes with a general discussion.

复习ADMM和AMA算法，用来解决凸聚类问题，并讨论收敛性和计算复杂度

2. PROPERTIES OF THE SOLUTION PATH

The solution path $\mathbf{U}(\gamma)$ enjoys several nice properties as a function of the regularization parameter γ and the weight matrix $\mathbf{W} = (w_{ij})$. These properties expedite its numerical computation. Proofs of the following two propositions can be found in the supplementary materials.

凸问题解存在且唯一，依赖于gamma和权重，直观

Proposition 2.1. The solution path $\mathbf{U}(\gamma)$ exists, is unique, and depends continuously on γ. The path also depends continuously on the weight matrix $\mathbf{W}$.

Existence and uniqueness of $\mathbf{U}$ sets the stage for a well-posed optimization problem. Continuity of $\mathbf{U}$ suggests employing homotopy continuation. Indeed, empirically we find great time savings in solving a sequence of problems over a grid of γ values when we use the solution of a previous value of γ as a warm start or initial value for the next larger γ value.

图1，分成两类

It is plausible that the centroids eventually coalesce to a common point as γ becomes sufficiently large. For the example shown in Figure 1, we intuitively expect for sufficiently large γ that the columns of $\mathbf{U}$ satisfy $\mathbf{u}_3 = \mathbf{u}_4 = \bar{\mathbf{x}}_{34}$ and $\mathbf{u}_1 = \mathbf{u}_2 = \mathbf{u}_5 = \bar{\mathbf{x}}_{125}$, where $\bar{\mathbf{x}}_{34}$ is the mean of $\mathbf{x}_3$ and $\mathbf{x}_4$ and $\bar{\mathbf{x}}_{125}$ is the mean of $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_5$. The next proposition rigorously confirms our intuition.

gamma充分大F(U)就是均值

Proposition 2.2. Suppose each point corresponds to a node in a graph with an edge between nodes i and j whenever $w_{ij} > 0$. If this graph is connected, then $F_\gamma(\mathbf{U})$ is minimized by $\bar{\mathbf{X}}$ for γ sufficiently large, where each column of $\bar{\mathbf{X}}$ equals the average $\bar{\mathbf{x}}$ of the n vectors $\mathbf{x}_i$.

We close this section by noting that in general the clustering paths are not guaranteed to be agglomerative. In the special case of the ℓ_1 norm with uniform weights $w_{ij} = 1$, Hocking et al. (2011) proved that the path is agglomerative. In the same article, they give an ℓ_2 norm example where the centroids fuse and then unfuse as the regularization parameter increases. This behavior, however, does not seem to occur very frequently in practice. Nonetheless, in the algorithms we describe next, we allow for such fission events to ensure that the computed solution path is truly the global minimizer of the convex criterion (1.1).

3. ALGORITHMS TO COMPUTE THE CLUSTERING PATH

Having characterized the solution path $\mathbf{U}(\gamma)$, we now tackle the task of computing it. We present two closely related optimization approaches: the alternating direction method of multipliers (ADMM) (Glowinski and Marrocco 1975; Gabay and Mercier 1976; Boyd et al. 2011) and the alternating minimization algorithm (AMA) (Tseng 1991). Both approaches employ variable splitting to handle the shrinkage penalties in the convex clustering criterion (1.1).

Let us first recast the convex clustering problem as the equivalent constrained problem

无约束问题转化为有约束问题
因为这样就可以用lagrange乘数法
后者范数不确定

$$\begin{aligned} &\text{minimize } \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{u}_i\|_2^2 + \gamma \sum_{l\in\mathcal{E}} w_l \|\mathbf{v}_l\| \\ &\text{subject to } \mathbf{u}_{l_1} - \mathbf{u}_{l_2} - \mathbf{v}_l = \mathbf{0}. \end{aligned} \tag{3.1}$$

l为二元族

Here, we index a centroid pair by $l = (l_1, l_2)$ with $l_1 < l_2$, define the set of edges over the nonzero weights $\mathcal{E} = \{l = (l_1, l_2) : w_l > 0\}$, and introduce a new variable $\mathbf{v}_l = \mathbf{u}_{l_1} - \mathbf{u}_{l_2}$ to account for the difference between the two centroids. The purpose of variable splitting is to simplify optimization with respect to the penalty terms.

对惩罚项进行了简化

Splitting methods such as ADMM and AMA have been successfully used to attack similar problems in image restoration (Goldstein and Osher 2009). To clarify the similarities and differences between ADMM and AMA, we briefly review how these two methods iteratively solve the following constrained optimization problem:

回顾一般凸优化问题，
对偶问题的解的等价性
以及ADMM和AMA如何求解

$$\begin{aligned} &\text{minimize } f(\mathbf{u}) + g(\mathbf{v}) \\ &\text{subject to } \mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{v} = \mathbf{c}, \end{aligned} \tag{3.2}$$

C为列向量

which includes the constrained minimization problem Equation (3.1) as a special case. Recall that finding the minimizer to an equality constrained optimization problem is equivalent to identifying the saddle point of the associated Lagrangian function. Both ADMM and AMA invoke a related function called the augmented Lagrangian,

lamda列向量为lagrange乘子，与AB矩阵行数有关　加了惩罚项增广lagrange函数，v=0为正常lagrange

$$\mathcal{L}_\nu(\mathbf{u}, \mathbf{v}, \boldsymbol{\lambda}) = f(\mathbf{u}) + g(\mathbf{v}) + \langle \boldsymbol{\lambda}, \mathbf{c} - \mathbf{A}\mathbf{u} - \mathbf{B}\mathbf{v} \rangle + \frac{\nu}{2} \|\mathbf{c} - \mathbf{A}\mathbf{u} - \mathbf{B}\mathbf{v}\|_2^2,$$

v为惩罚参数

where the dual variable $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers and ν is a nonnegative tuning parameter. When $\nu = 0$, the augmented Lagrangian coincides with the ordinary Lagrangian.

ADMM minimizes the augmented Lagrangian one block of variables at a time before updating the dual variable $\boldsymbol{\lambda}$. This yields the algorithm

$$\begin{aligned} \mathbf{u}^{m+1} &= \arg\min_{\mathbf{u}} \mathcal{L}_\nu(\mathbf{u}, \mathbf{v}^m, \boldsymbol{\lambda}^m) \\ \mathbf{v}^{m+1} &= \arg\min_{\mathbf{v}} \mathcal{L}_\nu(\mathbf{u}^{m+1}, \mathbf{v}, \boldsymbol{\lambda}^m) \\ \boldsymbol{\lambda}^{m+1} &= \boldsymbol{\lambda}^m + \nu(\mathbf{c} - \mathbf{A}\mathbf{u}^{m+1} - \mathbf{B}\mathbf{v}^{m+1}). \end{aligned} \tag{3.3}$$

AMA takes a slightly different tack and updates the first block $\mathbf{u}$ without augmentation, assuming $f(\mathbf{u})$ is strongly convex. This change is accomplished by setting the positive tuning constant ν to be 0. Thus, we update the first block $\mathbf{u}$ as

更新u用一般lagrange乘数，初始v=0

$$\mathbf{u}^{m+1} = \arg\min_{\mathbf{u}} \mathcal{L}_0(\mathbf{u}, \mathbf{v}^m, \boldsymbol{\lambda}^m), \tag{3.4}$$

and update $\mathbf{v}$ and $\boldsymbol{\lambda}$ as indicated in Equation (3.3). Later, we will see that this seemingly innocuous change pays large dividends in the convex clustering problem. Although block descent appears to complicate matters, it often markedly simplifies optimization in the end. In the case of convex clustering, the updates are either simple linear transformations or evaluations of proximal maps, which we discuss next.

3.1 Proximal Map　近端映射，更新v用到

For $\sigma > 0$, the function

$$\text{prox}_{\sigma\Omega}(\mathbf{u}) = \arg\min_{\mathbf{v}} \left[\sigma\Omega(\mathbf{v}) + \frac{1}{2}\|\mathbf{u} - \mathbf{v}\|_2^2 \right]$$

is a well-studied operation called the proximal map of the function $\Omega(\mathbf{v})$. The proximal map exists and is unique whenever the function $\Omega(\mathbf{v})$ is convex and lower semicontinuous. Norms satisfy these conditions, and for many norms of interest the proximal map can be evaluated by either an explicit formula or an efficient algorithm. Table 1 lists some common examples. The proximal maps for the ℓ_1 and ℓ_2 norms have explicit solutions and can be computed in $\mathcal{O}(p)$ operations for a vector $\mathbf{v} \in \mathbb{R}^p$. The proximal map for the ℓ_∞ norm requires projection onto the unit simplex and lacks an explicit solution. However, there are good algorithms for projecting onto the unit simplex (Michelot 1986; Duchi et al. 2008). In particular, Duchi et al.'s projection algorithm makes it possible to evaluate $\text{prox}_{\sigma\|\cdot\|_\infty}(\mathbf{v})$ in $\mathcal{O}(p \log p)$ operations.

Table 1. Proximal maps for common norms; $\mathcal{P}_C$ is the projection onto the closed, convex set C

Norm	$\Omega(\mathbf{v})$	$\text{prox}_{\sigma\Omega}(\mathbf{v})$	Comment
ℓ_1	$\|\mathbf{v}\|_1$	$\left[1-\frac{\sigma}{\|v_l\|}\right]_+ v_l$	Elementwise soft-thresholding
ℓ_2	$\|\mathbf{v}\|_2$	$\left[1-\frac{\sigma}{\|\mathbf{v}\|_2}\right]_+ \mathbf{v}$	Blockwise soft-thresholding
ℓ_∞	$\|\mathbf{v}\|_\infty$	$\mathbf{v}-\mathcal{P}_{\sigma S}(\mathbf{v})$	S is the unit simplex

3.2 ADMM Updates

现解决（3.1）的解

The augmented Lagrangian for criterion Equation (3.1) is given by

$$\mathcal{L}_\nu(\mathbf{U},\mathbf{V},\mathbf{\Lambda}) = \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i-\mathbf{u}_i\|_2^2+\gamma\sum_{l\in\mathcal{E}}w_l\|\mathbf{v}_l\| + \sum_{l\in\mathcal{E}}\langle\boldsymbol{\lambda}_l,\mathbf{v}_l-\mathbf{u}_{l_1}+\mathbf{u}_{l_2}\rangle+\frac{\nu}{2}\sum_{l\in\mathcal{E}}\|\mathbf{v}_l-\mathbf{u}_{l_1}+\mathbf{u}_{l_2}\|_2^2. \tag{3.5}$$

凸优化+lanrange乘子+惩罚项

更新U，固定v和lamda

To update $\mathbf{U}$, we need to minimize the function

$$f(\mathbf{U}) = \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i-\mathbf{u}_i\|_2^2+\frac{\nu}{2}\sum_{l\in\mathcal{E}}\|\tilde{\mathbf{v}}_l-\mathbf{u}_{l_1}+\mathbf{u}_{l_2}\|_2^2,$$

where $\tilde{\mathbf{v}}_l=\mathbf{v}_l+\nu^{-1}\boldsymbol{\lambda}_l$. The gradient of this function vanishes at $\mathbf{U}$ satisfying the linear system

新算法

$$\mathbf{UM}=\mathbf{X}+\nu\sum_{l\in\mathcal{E}}\tilde{\mathbf{v}}_l(\mathbf{e}_{l_1}-\mathbf{e}_{l_2})^t, \tag{3.6}$$

where $\mathbf{M}=\mathbf{I}+\nu\sum_{l\in\mathcal{E}}(\mathbf{e}_{l_1}-\mathbf{e}_{l_2})(\mathbf{e}_{l_1}-\mathbf{e}_{l_2})^t$. If the edge set $\mathcal{E}$ contains all possible edges, then the update for $\mathbf{U}$ can be computed analytically as

$$\mathbf{u}_i=\frac{1}{1+n\nu}\mathbf{y}_i+\frac{n\nu}{1+n\nu}\bar{\mathbf{x}}, \tag{3.7}$$

where $\bar{\mathbf{x}}$ is the average column of $\mathbf{X}$ and

$$\mathbf{y}_i=\mathbf{x}_i+\sum_{l_1=i}[\boldsymbol{\lambda}_l+\nu\mathbf{v}_l]-\sum_{l_2=i}[\boldsymbol{\lambda}_l+\nu\mathbf{v}_l].$$

Detailed derivations of the equalities in Equations (3.6) and (3.7) can be found in the supplementary materials.

Before deriving the updates for $\mathbf{V}$, we remark that although a fully connected weights graph allows one to write explicit updates for $\mathbf{U}$, doing so comes at the cost of increasing the number of variables $\mathbf{v}_l$ and $\boldsymbol{\lambda}_l$. Such choices are not immaterial, and we will discuss the trade-offs later. To update $\mathbf{V}$, we first observe that the augmented Lagrangian $\mathcal{L}_\nu(\mathbf{U},\mathbf{V},\mathbf{\Lambda})$ is separable in the vectors $\mathbf{v}_l$. A particular difference vector $\mathbf{v}_l$ is determined by the proximal map

$$\begin{aligned}\mathbf{v}_l &= \arg\min_{\mathbf{v}_l}\frac{1}{2}\left[\|\mathbf{v}_l-(\mathbf{u}_{l_1}-\mathbf{u}_{l_2}-\nu^{-1}\boldsymbol{\lambda}_l)\|_2^2+\frac{\gamma w_l}{\nu}\|\mathbf{v}_l\|\right]\\ &= \text{prox}_{\sigma_l\|\cdot\|}(\mathbf{u}_{l_1}-\mathbf{u}_{l_2}-\nu^{-1}\boldsymbol{\lambda}_l),\end{aligned} \tag{3.8}$$

Algorithm 1 ADMM

Initialize $\mathbf{\Lambda}^0$ and $\mathbf{V}^0$.

for $m = 1, 2, 3, \ldots$ **do**
  for $i = 1, \ldots, n$ **do**
    $\mathbf{y}_i = \mathbf{x}_i + \sum_{l_1=i}[\boldsymbol{\lambda}_l^{m-1} + \nu \mathbf{v}_l^{m-1}] - \sum_{l_2=i}[\boldsymbol{\lambda}_l^{m-1} + \nu \mathbf{v}_l^{m-1}]$
  end for
  $\mathbf{U}^m = \frac{1}{1+n\nu}\mathbf{Y} + \frac{n\nu}{1+n\nu}\bar{\mathbf{X}}$
  for all l **do**
    $\mathbf{v}_l^m = \text{prox}_{\sigma_l\|\cdot\|}(\mathbf{u}_{l_1}^m - \mathbf{u}_{l_2}^m - \nu^{-1}\boldsymbol{\lambda}_l^{m-1})$
    $\boldsymbol{\lambda}_l^m = \boldsymbol{\lambda}_l^{m-1} + \nu(\mathbf{v}_l^m - \mathbf{u}_{l_1}^m + \mathbf{u}_{l_2}^m)$
  end for
end for

where $\sigma_l = \gamma w_l/\nu$. Finally, the Lagrange multipliers are updated by

$$\boldsymbol{\lambda}_l = \boldsymbol{\lambda}_l + \nu(\mathbf{v}_l - \mathbf{u}_{l_1} + \mathbf{u}_{l_2}).$$

Algorithm 1 summarizes the ADMM algorithm. To track the progress of ADMM, we use standard methods given by Boyd et al. (2011) based on primal and dual residuals. Details on the stopping rules that we employ are given in the supplementary materials.

更新不依赖于v

3.3 AMA Updates

Since AMA shares its update rules for $\mathbf{V}$ and $\mathbf{\Lambda}$ with ADMM, consider updating $\mathbf{U}$. Recall that AMA updates $\mathbf{U}$ by minimizing the ordinary Lagrangian. In the $\nu = 0$ case, we have

$$\mathbf{U}^{m+1} = \arg\min_{\mathbf{U}} \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{u}_i\|_2^2 + \sum_{l\in\mathcal{E}}\langle\boldsymbol{\lambda}_l^m, \mathbf{v}_l - \mathbf{u}_{l_1} + \mathbf{u}_{l_2}\rangle.$$

In contrast to ADMM, this minimization separates in each $\mathbf{u}_i$ and gives an update that does not depend on $\mathbf{v}_l$, namely

$$\mathbf{u}_i^{m+1} = \mathbf{x}_i + \sum_{l_1=i}\boldsymbol{\lambda}_l^m - \sum_{l_2=i}\boldsymbol{\lambda}_l^m. \tag{3.9}$$

揭示了AMA甚至不需要计算。
应用凸演算的标准结果，可以证明

Further scrutiny of the updates for $\mathbf{V}$ and $\mathbf{\Lambda}$ reveals that AMA does not even require computing $\mathbf{V}$. Applying standard results from convex calculus, it can be shown that $\mathbf{\Lambda}$ has the following update:

$$\boldsymbol{\lambda}_l^{m+1} = \mathcal{P}_{C_l}\big(\boldsymbol{\lambda}_l^m - \nu\mathbf{g}_l^{m+1}\big), \tag{3.10}$$

where $\mathbf{g}_l^m = \mathbf{u}_{l_1}^m - \mathbf{u}_{l_2}^m$, $C_l = \{\boldsymbol{\lambda}_l : \|\boldsymbol{\lambda}_l\|_\dagger \le \gamma\omega_l\}$, and $\|\cdot\|_\dagger$ is the dual norm of the norm defining the fusion penalty. Algorithm 2 summarizes the AMA algorithm. Detailed derivation of this simplification is in the supplementary materials. At the termination of Algorithm 2, the centroids can be recovered from the dual variables using Equation (3.9).

Algorithm 2 AMA

Initialize $\mathbf{\Lambda}^0$.

for $m = 1, 2, 3, \ldots$ **do**
  for $i = 1, \ldots, n$ **do**
    $\mathbf{\Delta}_i^m = \sum_{l_1=i} \boldsymbol{\lambda}_l^{m-1} - \sum_{l_2=i} \boldsymbol{\lambda}_l^{m-1}$
  end for
  for all l **do**
    $\mathbf{g}_l^m = \mathbf{x}_{l_1} - \mathbf{x}_{l_2} + \mathbf{\Delta}_{l_1}^m - \Delta_{l_2}^m$
    $\boldsymbol{\lambda}_l^m = \mathcal{P}_{C_l}(\boldsymbol{\lambda}_l^{m-1} - \nu \mathbf{g}_l^m)$
  end for
end for

凸聚类问题（3.1）的对偶本质上是一个约束最小二乘问题，因此可以用经典投影梯度算法（近梯度算法的一种特例）进行数值求解也就不足为奇了。除了提供AMA方法的简单解释外，对偶问题还允许我们基于对偶间隙推导出严格的AMA停止标准，即当前迭代中评估的原始目标超过其最佳值的范围。

Note that Algorithm 2 looks remarkably like a projected gradient algorithm. Indeed, Tseng (1991) showed that AMA is actually performing proximal gradient ascent to maximize a dual problem. Since the dual of the convex clustering problem (3.1) is essentially a constrained least squares problem, it is hardly surprising that it can be solved numerically by the classic projected gradient algorithm, a special case of the proximal gradient algorithm. In addition to providing a simple interpretation of the AMA method, the dual problem allows us to derive a rigorous stopping criterion for AMA based on the duality gap, a bound on how much the primal objective evaluated at the current iterate exceeds its optimal value. Due to space limitations, both the dual problem and duality gap computation are covered in the supplementary materials. Before proceeding, however, let us emphasize that AMA requires tracking of only as many dual variables $\boldsymbol{\lambda}_l$ as there are nonzero weights. We will find later that sparse weights often produce better quality clusterings. Thus, when relatively few weights are nonzero, the number of variables introduced by splitting does not become prohibitive under AMA.

ADMM和AMA都在合理的条件下收敛DMM的收敛性在任何ν>0时都是有保证的。当ν不太大时，保证AMA收敛。在本节中，我们将展示凸聚类问题的特定结构如何为ADMM提供更强的收敛结果，并为在AMA中选择ν提供指导，并在实践中提高AMA的收敛速度

4. CONVERGENCE

Both ADMM and AMA converge under reasonable conditions. Convergence of ADMM is guaranteed for any $\nu > 0$. Convergence for AMA is guaranteed when ν is not too large. In this section, we show how the specific structure of the convex clustering problem can give stronger convergence results for ADMM and provide guidance on choosing ν in AMA and improve AMA's rate of convergence in practice. All proofs can be found in the supplementary materials.

在最小假设下，可以证明ADMM迭代列的极限点与最小化目标函数的平稳点一致

4.1 ADMM

Under minimal assumptions, it can be shown that the limit points of an ADMM iterate sequence coincide with the stationary points of the objective function being minimized (Boyd et al. 2011). Note, however, this does not guarantee that the iterates $\mathbf{U}^m$ converge to $\mathbf{U}^*$. Since the convex clustering criterion $F_\gamma(\mathbf{U})$ defined by Equation (1.1) is strictly convex

这并不保证迭代UMU收敛到U .因为方程（1.1）定义的凸聚类准则F γ （U）是严格凸的

在凸条件下
我们得到了ADMM迭代序列收敛到唯一全局极小值U的更强的结果 $F_\gamma(\mathbf{U})$

and coercive, we have the stronger result that the ADMM iterate sequence converges to the unique global minimizer $\mathbf{U}^*$ of $F_\gamma(\mathbf{U})$.

Proposition 4.1. The iterates $\mathbf{U}^m$ in Algorithm 1 converge to the unique global minimizer $\mathbf{U}^*$ of the clustering criterion $F_\gamma(\mathbf{U})$.

4.2 AMA

The convergence of Algorithm 2 hinges on the choice of ν, which, in turn, depends on the connectivity of the associated graph.

Proposition 4.2. Let $\mathbf{u}_i^{m+1} = \mathbf{x}_i + \sum_{l_1=i} \boldsymbol{\lambda}_l^m - \sum_{l_2=i} \boldsymbol{\lambda}_l^m$, where $\boldsymbol{\lambda}_l^m$ are the iterates in Algorithm 2. Then, the sequence $\mathbf{U}^{m+1}$ converges to the unique global minimizer $\mathbf{U}^*$ of the clustering criterion $F_\gamma(\mathbf{U})$, provided that $\nu < 2/\rho(\mathbf{L})$, where $\rho(\mathbf{L})$ denotes the largest eigenvalue of $\mathbf{L}$, the Laplacian matrix of the associated graph.

In lieu of computing $\rho(\mathbf{L})$ numerically, one can bound it by theoretical arguments. In general, $\rho(\mathbf{L}) \leq n$ (Anderson and Morley 1985), with equality when the graph is fully connected and $w_{ij} > 0$ for all $i < j$. Choosing a fixed step size of $\nu < 2/n$ works in practice when there are fewer than 1000 data points and the graph is dense. For a sparse graph with bounded node degrees, the sharper bound

$$\rho(\mathbf{L}) \leq \max\{d(i) + d(j) : (i, j) \in \mathcal{E}\}$$

AMA在稀疏图上收敛性很好

applies, where $d(i)$ is the degree of the ith node (Anderson and Morley 1985). This bound can be computed quickly in $\mathcal{O}(n + \varepsilon)$ operations, where ε denotes the number of edges in $\mathcal{E}$. Section 7.2 demonstrates the overwhelming speed advantage that AMA has on sparse graphs.

优化算法的计算复杂度包括每次迭代的工作量、收敛前的迭代次数以及总体内存需求

5. COMPUTATIONAL COMPLEXITY

迭代次数不讨论

The computational complexity of an optimization algorithm includes the amount of work per iteration, the number of iterations until convergence, and the overall memory requirements. We discuss the first and third components. For the AMA and ADMM convex clustering algorithms, the number of iterations to achieve a desired level of accuracy can be derived from existing complexity results in the literature. Details can be found in the supplementary materials.

5.1 AMA

Inspection of Algorithm 2 shows that computing all $\boldsymbol{\Delta}_i$ requires $p(2\varepsilon - n)$ total additions and subtractions. Computing all vectors $\mathbf{g}_l$ in Algorithm 2 takes $\mathcal{O}(\varepsilon p)$ operations, and taking the subsequent gradient step costs $\mathcal{O}(\varepsilon p)$ operations. Computing the needed projections costs $\mathcal{O}(\varepsilon p)$ operations for the ℓ_1 and ℓ_2 norms and $\mathcal{O}(\varepsilon p \log p)$ operations for the ℓ_∞ norm. Finally, computing the duality gap costs $\mathcal{O}(np + \varepsilon p)$ operations. Details on the duality gap computation appear in the supplementary materials. The assumption

对偶间隙

我们限制一个节点到itsk最近邻的连通性，然后ε isO（kn）。因此，在最坏情况下，问题的计算复杂度在点的数量n上是二次的，在tok最近邻连通性的限制下是线性的。

that n is $\mathcal{O}(\varepsilon)$ entails smaller costs. A single iteration with gap checking then costs just $\mathcal{O}(\varepsilon p)$ operations for the ℓ_1 and ℓ_2 norms and $\mathcal{O}(\varepsilon p \log p)$ operations for the ℓ_∞ norm.

Total storage is $\mathcal{O}(p\varepsilon + np)$. In the worst case, ε is $\binom{n}{2}$. However, if we limit a node's connectivity to its k-nearest neighbors, then ε is $\mathcal{O}(kn)$. Thus, the computational complexity of the problem in the worst case is quadratic in the number of points n and linear under the restriction to k-nearest neighbors connectivity. Storage is quadratic in n in the worst case and linear in n under the k-nearest neighbors restriction. Thus, limiting a point's connectivity to its k-nearest neighbors renders both the storage requirements and operation counts linear in the problem size, namely $\mathcal{O}(knp)$.

存储在最坏情况下是二次的，在k-最近邻限制下是线性的。因此，将一个点的连接性限制为其k近邻会使存储需求和操作计数在问题大小中呈线性

5.2 ADMM

显式更新

We have two cases to consider. First, consider the explicit updates outlined in Algorithm 1 when the edge set $\mathcal{E}$ is full. By nearly identical arguments as earlier, the complexity of a single round of ADMM updates with primal and dual residual calculation requires $\mathcal{O}(n^2 p)$ operations for the ℓ_1 and ℓ_2 norms and $\mathcal{O}(n^2 p \log p)$ operations for the ℓ_∞ norm. Thus, the ADMM algorithm based on the explicit update Equation (3.7) requires the same computational effort as AMA in the worst case. In this setting, both ADMM and AMA also have $\mathcal{O}(pn^2)$ storage requirements.

The situation does not improve much when we consider the more frugal alternative in which $\mathcal{E}$ contains only node pairings corresponding to nonzero weights. In this case, the variables $\mathbf{\Lambda}$ and $\mathbf{V}$ have only as many columns as there are nonzero weights. Now, the storage requirements are $\mathcal{O}(p\varepsilon + np)$ like AMA, but the cost of updating $\mathbf{U}$, the most computationally demanding step, remains quadratic in n. Recall that we need to solve a linear system of Equations (3.6)

$$\mathbf{UM} = \mathbf{X} + \sum_{l \in \mathcal{E}} \tilde{\mathbf{v}}_l (\mathbf{e}_{l_1} - \mathbf{e}_{l_2})^t,$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$. Since $\mathbf{M}$ is positive definite and does not change throughout the ADMM iterations, the prudent course of action is to compute and cache its Cholesky factorization. The factorization requires $\mathcal{O}(n^3)$ operations to calculate but that cost can be amortized across the repeated ADMM updates. With the Cholesky factorization in hand, we can update each row of $\mathbf{U}$ by solving two sets of n-by-n triangular systems of equations, which together requires $\mathcal{O}(n^2)$ operations. Since $\mathbf{U}$ has p rows, the total amount of work to update $\mathbf{U}$ is $\mathcal{O}(n^2 p)$. Therefore, the overall amount of work per ADMM iteration is $\mathcal{O}(n^2 p + \varepsilon p)$ operations for the ℓ_1 and ℓ_2 norms and $\mathcal{O}(n^2 p + \varepsilon p \log p)$ operations for the ℓ_∞ norm. Thus, in stark contrast to AMA, both ADMM approaches grow quadratically, either in storage requirements or computational costs, regardless of how we might limit the size of the edge set $\mathcal{E}$.

因此，与AMA形成鲜明对比的是，无论是在存储需求还是计算成本方面，无论我们如何限制边缘集E的大小，这两种ADMM方法都是二次增长的

6. WEIGHTS, CLUSTER ASSIGNMENT, AND ACCELERATION

We now address some practical issues that arise in applying our algorithms.

Weights: The weight matrix $\mathbf{W}$ can dramatically affect the quality of the clustering path. We set the weight between the ith and jth points equal to $w_{ij} = \iota^k_{\{i,j\}} \exp(-\phi \|\mathbf{x}_i - \mathbf{x}_j\|_2^2)$,

权重矩阵W会显著影响聚类路径的质量

where the indicator function $\iota^k_{\{i,j\}}$ is 1 if j is among i's k-nearest-neighbors or vice versa and 0 otherwise. The second factor is a Gaussian kernel that slows the coalescence of distant points. The constant ϕ is nonnegative; the value $\phi = 0$ corresponds to uniform weights. As noted earlier, limiting positive weights to nearest neighbors improves both computational efficiency and clustering quality. Although the two factors defining the weights act similarly, their combination increases the sensitivity of the clustering path to the local density of the data.

Cluster Assignment: We determine clustering assignments as a function of γ by reading off which centroids fuse. For both ADMM and AMA, such assignments can be performed in $\mathcal{O}(n)$ operations using the differences variable $\mathbf{V}$. In the case of AMA, where we do not store a running estimate of $\mathbf{V}$, we compute $\mathbf{V}$ via the update (3.8) after the algorithm terminates for a given γ. Once we determine $\mathbf{V}$, we simply apply breadth-first search to identify the connected components of the graph induced by $\mathbf{V}$. This graph identifies a node with every data point and places an edge between the lth pair of points if and only if $\mathbf{v}_l = \mathbf{0}$. Each connected component corresponds to a cluster. Note that the graph described here varies with γ, through the matrix $\mathbf{V}$, and is unrelated to the graph discussed in the rest of this article, which depends solely on the weights $\mathbf{W}$ and is invariant to γ.

Acceleration: Both AMA and ADMM admit acceleration at little additional computational cost (Goldstein, O'Donoghue, and Setzer 2012). In our timing comparisons, accelerated variants of AMA and ADMM are used. Details on our acceleration techniques can be found in the supplementary materials.

权重的选择如何影响聚类解决方案的质量

Hocking等人只尝试了高斯核，所以在本节中，我们将跟进他们未经验证的建议，即将高斯核和k-近邻相结合。

7. NUMERICAL EXPERIMENTS

We now report numerical experiments on convex clustering for a synthetic and real dataset. Similar experiments on two additional real datasets (iris and Senate) can be found in the supplementary materials. In particular, we focus on how the choice of the weights w_{ij} affects the quality of the clustering solution. Prior research on this question is limited. Both Lindsten, Ohlsson, and Ljung (2011) and Hocking et al. (2011) suggested weights derived from Gaussian kernels and k-nearest neighbors. Because Hocking et al. tried only Gaussian kernels, in this section we follow up on their untested suggestion of combining Gaussian kernels and k-nearest neighbors.

We also compare the run times of our splitting methods to the run times of the subgradient algorithm employed by Hocking et al. for ℓ_2 paths. Our attention here focuses on solving the ℓ_2 path since the rotational invariance of the ℓ_2 norm makes it a robust choice in practice. Hocking et al. provided R and C++ code for their algorithms. Our algorithms are implemented in R and C. To make a fair comparison, we run our algorithm until it reaches a primal objective value that is less than or equal to the primal objective value obtained by the subgradient algorithm. Specifically, we first run the Hocking et al. code to generate a clusterpath and record the sequence of γ's generated by their code. We then run our algorithms over the same sequence of γ's and stop once our primal objective value falls below their value. We also retain the native stopping rule computations employed by our splitting methods, namely the dual loss calculations for AMA and residual calculations for ADMM. Since AMA already calculates the primal loss, this is not an additional burden.

限制最近邻的数量（k=10）和使用非平凡的高斯核权重（ϕ=0.5）可以产生最佳的聚类路径。使用过多的邻域并假设权重一致会导致很少的聚集聚类

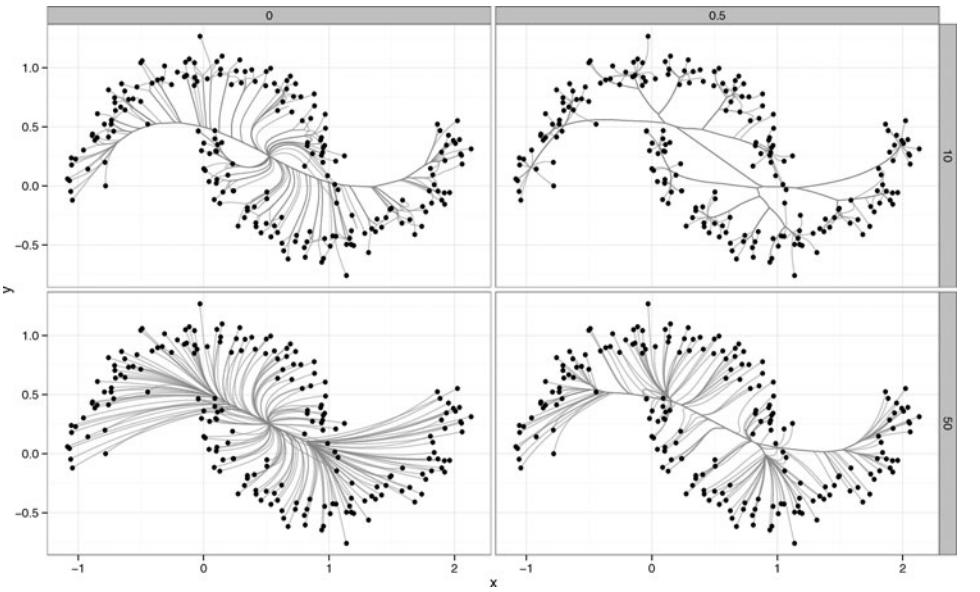


Figure 3. Halfmoons example: The first and second rows show results using $k = 10$ and 50 nearest neighbors, respectively. The first and second columns show results using $\phi = 0$ and 0.5, respectively.

高斯核

Although convergence monitoring creates additional work for ADMM, the added primal loss calculation at worst changes only the constant in the complexity bound. This follows since computing the primal loss requires $\mathcal{O}(np + \varepsilon p)$ total operations.

7.1 Qualitative Comparisons

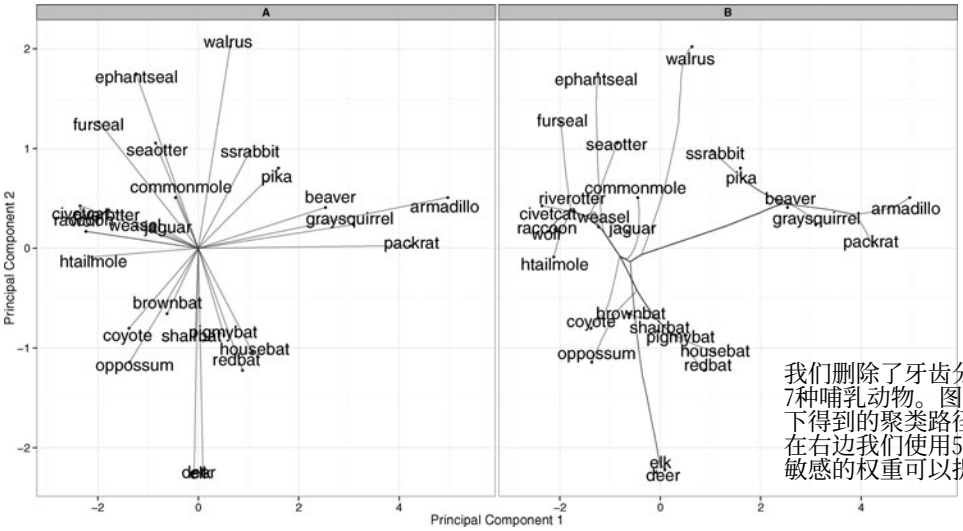
解决方案路径的特征如何随权重wij的选择而显著变化。

The following examples demonstrate how the character of the solution paths can vary drastically with the choice of weights w_{ij}.

Two Half Moons: Consider the standard simulated data of two interlocking half moons in $\mathbb{R}^2$ composed of 100 points each. Figure 3 shows four convex clustering paths computed assuming two different numbers of nearest neighbors (10 and 50) and two different kernel constants ϕ (0 and 0.5). The upper right panel makes it evident that limiting the number of nearest neighbors ($k = 10$) and using nontrivial Gaussian kernel weights ($\phi = 0.5$) produce the best clustering path. Using too many neighbors and assuming uniform weights results in little agglomerative clustering until late in the clustering path (lower left panel). The two intermediate cases diverge in interesting ways. The hardest set of points to cluster are the points in the upper half moon's right tip and the lower half moon's left tip. Limiting the number of nearest neighbors and omitting the Gaussian kernel (upper left panel) correctly agglomerates the easier points, but waffles on the harder points, agglomerating them only at the very end when all points coalesce at the grand mean. Conversely, using too many neighbors and the Gaussian kernel (lower right panel) leads to a clustering path that does not hedge but incorrectly assigns the harder points.

Dentition of mammals: Next, we consider the problem of clustering mammals based on their dentition (Hartigan 1975). Eight different kinds of teeth are tallied for each mammal:

坐标指的是两变量之间的距离。
距离越近（接近于0）则函数值越大，否则越小。
h越大，相同距离的情况下函数值会越小。因此我们可以选取适当的h值，得到满足上述要求的那种权重



我们删除了牙齿分布不独特的观察结
7种哺乳动物。图4显示了在两种不同
下得到的聚类路径。在左边wij=1f或
在右边我们使用5个最近邻，ϕ=0。
敏感的权重可以提供更好的结果

Figure 4. Clustering path under the ℓ_2 norm for the Mammal Data. Panel on the left (Set A) used $w_{ij} = 1$ for all $i \neq j$. Panel on the right (Set B) used $k = 5$ nearest neighbors and $\phi = 0.5$.

the number of top incisors, bottom incisors, top canines, bottom canines, top premolars, bottom premolars, top molars, and bottom molars. We removed observations with teeth distributions that were not unique, leaving us with 27 mammals. Figure 4 shows the resulting clustering paths under two different choices of weights. On the left $w_{ij} = 1$ for all $i \neq j$, and on the right we use 5-nearest neighbors and $\phi = 0.5$. Weights sensitive to the local density give superior results. Since there are eight variables, to visualize results we project the data and the fitted clustering paths onto the first two principal components of the data. The cluster path gives a different and perhaps more sensible solution than a two-dimensional principal component analysis (PCA). For example, the brown bat is considered more similar to the house bat and red bat, even though it is closer in the first two PCA coordinates to the coyote and oppossum.

将数据和拟合的聚类路径投影到数据的前两个主成分上
。与二维主成分分析（PCA）相比，聚类路径给出了一种
不同且可能更合理的解决方案。例如，棕色蝙蝠被认为
与家蝙蝠和红色蝙蝠更为相似，尽管它在前两个PCA坐标
中更接近郊狼和红蝙蝠

7.2 TIMING COMPARISONS

We now present results on two batches of experiments, with dense weights in the first batch and sparse ones in the second. For the first set of experiments, we compared the run times of the subgradient descent algorithm of Hocking et al. (2011), accelerated ADMM, and accelerated AMA on 10 replicates of simulated data consisting of 100, 200, 300, 400, and 500 points in $\mathbb{R}^2$ drawn from a multivariate standard normal. We limited our study to at most 500 points because the subgradient algorithm took several hours on a single realization of 500 points. Limiting the number of data points allowed us to use the simpler, but less storage efficient, ADMM formulation given in Equation (3.7). For AMA, we fixed the step size at $\nu = 1/n$. For all tests, we assigned full-connectivity weights based on the indicator $\iota_{\{i,j\}}^k \equiv 1$ and tuning constant $\phi = -2$. The parameter ϕ was chosen to ensure that the smallest weight was bounded safely away from zero. The full-connectivity assumption illustrates the superiority of AMA even under its least favorable circumstances. To trace

通过限制数据点的数量
我们可以使用方程式（3.7）
中给出的更简单
但存储效率更低的
ADMM公式。
对于AMA，
我们将步长固定为ν=1/n。
对于所有测试，
我们根据指标ιk{i，j}
分配完整的连通性权重
和调谐常数ϕ=−2.
选择参数ϕ是为了
确保最小权重安全地远离零。
全连通性假设说明了AMA的优越性，

我们比较了Hocking et al.（2011）的次梯度下降算法
、加速ADMM和加速AMA在10个模拟数据副本上的运行时

Table 2. Timing comparison under the ℓ_2 norm: Dense weights. Mean run times are in seconds. Different methods are listed on each row. Each column reports times for varying number of points

	100	200	300	400	500
Subgradient	44.40	287.86	2361.84	3231.21	13895.50
AMA	16.09	71.67	295.23	542.45	1109.67
ADMM	57.82	449.68	1430.05	3432.77	6745.82

一旦ADMM和AMA算法的质心迭代实现
于次梯度算法实现的原始损失，我们就
算法

out the entire clusterpath, we ran the Hocking subgradient algorithm to completion and invoked its default stopping criterion, namely a gradient with an ℓ_2 norm below 0.001. As noted earlier, we stopped our ADMM and AMA algorithms once their centroid iterates achieved a primal loss less than or equal to that achieved by the subgradient algorithm.

Table 2 shows the resulting mean times in seconds. Boxplots showing how the run time scales with the number of data points n can be found in the supplementary materials. All three algorithms scale quadratically in the number of points. This is expected for ADMM and AMA because all weights w_{ij} are positive. Nonetheless, the three algorithms possess different rate constants, with AMA possessing the slowest mean growth. The subgradient and ADMM algorithms performed commensurately, although the ADMM algorithm was almost twice as fast on average when $n = 500$. Again, to ensure fair comparisons with the subgradient algorithm, we required ADMM to make extra primal loss computations. This change tends to inflate its rate constant. In summary, we see that fast AMA leads to affordable computation times, on the order of minutes for hundreds of data points, in contrast to subgradient descent, which incurs run times on the order of hours for 400–500 data points.

AMA的平均增长速度最慢。
次梯度算法和ADMM算法的
性能相当，尽管当n=500时，

In the second batch of experiments, the same setup is retained except for assignments of weights and step length choice for AMA. We used $\phi = -2$ again, but this time we zeroed out all weights except those corresponding to the $k = \frac{n}{4}$ nearest neighbors of each point. For AMA, we used step sizes based on the bound (4.1). Table 3 shows the resulting mean run times in seconds. As before, more detailed boxplot comparisons can be found in the supplementary materials. As attested by the shorter run times for all three algorithms, incorporation of sparse weights appears to make the problems easier to solve. In this case, ADMM was uniformly better on average than the subgradient method for all n. Even more noteworthy is the pronounced speed advantage of AMA over the other two algorithms for large n. When clustering 500 points, AMA requires on average a mere 7 sec compared to 5–7 min for the subgradient and ADMM algorithms.

这三种算法的运行时间
都较短，这证明了稀疏
权重的加入似乎使问题
更容易解决。在这种情
况下，平均而言，ADMM
比次梯度法对所有n。值
得注意的是，与其他两
种算法相比，AMA在处理
大n时具有明显的速度优势

Table 3. Timing comparison under the ℓ_2 norm: Sparse weights. Mean run times are in seconds. Different methods are listed on each row. Each column reports times for varying number of points

	100	200	300	400	500
Subgradient	6.52	37.42	161.68	437.32	386.45
AMA	1.50	2.94	4.46	6.02	7.44
ADMM	3.78	21.35	61.23	139.37	297.99

引入两种分裂算法
解决凸聚类问题

范数任选，
但是近端映射要易于计算
这样才可以量化
算法复杂度和收敛特性

8. CONCLUSION AND FUTURE WORK

In this article, we introduce two splitting algorithms for solving the convex clustering problem. The splitting perspective encourages path following, one of the chief benefits of convex clustering. The splitting perspective also permits centroid penalties to invoke an arbitrary norm. The only requirement is that the proximal map for the norm be readily computable. Equivalently, projection onto the unit ball of the dual norm should be straightforward. Because proximal maps and projection operators are generally well understood, it is possible for us to quantify the computational complexity and convergence properties of our algorithms.

在第一次块更新中
最小化增广拉格朗日
未分段拉格朗日之间
一个重要的区别。
AMA可以在更短的
时间和空间内完成这

It is noteworthy that ADMM did not fare as well as AMA. ADMM has become quite popular in machine learning circles in recent years. Applying variable splitting and using ADMM to iteratively solve the convex clustering problem seemed like an obvious and natural initial strategy. Only later during our study did we implement the less favored AMA algorithm. Considering how trivial the differences are between the generic block updates for ADMM (3.3) and AMA (3.4), we were surprised by the performance gap between them. In the convex clustering problem, however, there is a nontrivial difference between minimizing the augmented and unaugmented Lagrangian in the first block update. This task can be accomplished in less time and space by AMA.

Two features of the convex clustering problem make it an especially good candidate for solution by AMA. First, the objective function is strongly convex and therefore has a Lipschitz differentiable dual. Lipschitz differentiability is a standard condition ensuring the convergence of proximal gradient algorithms. Second, a good step size can be readily computed from the Laplacian matrix generated by the edge set $\mathcal{E}$. Without this prior bound, we would have to employ a more complicated line-search.

Our complexity analysis and simulations show that the accelerated AMA method appears to be the algorithm of choice. Nonetheless, given that alternative variants of ADMM may close the performance gap (Deng and Yin 2012; Goldfarb, Ma, and Scheinberg 2012), we are reluctant to dismiss ADMM too quickly. Both algorithms deserve further investigation. For instance, in both ADMM and AMA, updates of $\mathbf{\Lambda}$ and $\mathbf{V}$ could be parallelized. Hocking et al. also employed an active set approach to reduce computations as the centroids coalesce. A similar strategy could be adopted in our framework, but it incurs additional overhead as checks for fission events have to be introduced. An interesting and practical question brought up by Hocking et al. remains open, namely under what conditions or weights are fusion events guaranteed to be permanent as γ increases? In all our experiments, we did not observe any fission events. Identifying those conditions would eliminate the need to check for fission in such cases and expedite computation.

For AMA, the storage demands and computational complexity of convex clustering depend quadratically on the number of points in the worst case. Limiting a point's connections to its k-nearest neighbors, for example, ensures that the number of edges in the graph is linear in the number of nodes in the graph. Eliminating long-range dependencies is often desirable anyway. Choosing sparse weights can improve both cluster quality and computational efficiency. Moreover, finding the exact k-nearest neighbors is likely not essential, and we conjecture that the quality of solutions would not suffer greatly if approximate nearest neighbors are used and algorithms for fast computation of approximately nearest neighbors

are leveraged (Slaney and Casey 2008). On very large problems, the best strategy might be to exploit the continuity of solution paths in the weights. This suggests starting with even sparser graphs than the desired one and generating a sequence of solutions to increasingly dense problems. A solution with fewer edges can serve as a warm start for the next problem with more edges.

The splitting perspective also invites extensions that impose structured sparsity on the centroids. Witten and Tibshirani (2010) discussed how sparse centroids can improve the quality of a solution, especially when only a relatively few features of data drive clustering. Structured sparsity can be accomplished by adding a sparsity-inducing norm penalty to the $\mathbf{U}$ updates. The update for the centroids for both AMA and ADMM then rely on another proximal map of a gradient step. Introducing a sparsifying norm, however, raises the additional complication of choosing the amount of penalization.

Except for a few hints about weights, our analysis leaves the topic of optimal clustering untouched. Recently, von Luxburg (2010) suggested some principled approaches to assessing the quality of a clustering assignment via data perturbation and resampling. These clues are worthy of further investigation.

SUPPLEMENTARY MATERIALS

The supplementary materials include proofs for all propositions, details on the more complicated derivations, the stopping criterions for both algorithms, a sketch of algorithm acceleration, derivation of the dual problem, additional discussion on computational complexity, additional figures showing timing comparisons, and application of convex clustering to two additional real datasets. (Supplement.pdf) [Code:] An R package, *cvxclustr*, which implements the AMA and ADMM algorithms in this article, is available on the CRAN website. The mammals dataset is included in it.

ACKNOWLEDGMENTS

The authors thank Jocelyn Chi, Daniel Duckworth, Tom Goldstein, Rob Tibshirani, and Genevera Allen for their helpful comments and suggestions. All plots were made using the open source R package ggplot2 (Wickham 2009). This research was supported by the NIH United States Public Health Service grants GM53275 and HG006139.

[Received March 2014. Revised July 2014.]

REFERENCES

Aloise, D., Deshpande, A., Hansen, P., and Popat, P. (2009), "NP-Hardness of Euclidean Sum-of-Squares Clustering," *Machine Learning*, 75, 245–248. [995]

Anderson, W. N., and Morley, T. D. (1985), "Eigenvalues of the Laplacian of a Graph," *Linear and Multilinear Algebra*, 18, 141–145. [1004]

Arthur, D., and Vassilvitskii, S. (2007), "*k*-means++: The Advantages of Careful Seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, SODA '07, Philadelphia, PA, pp. 1027–1035. [997]

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011), "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundation and Trends in Machine Learning*, 3, 1–122. [999,1002,1003]

Bradley, P. S., Mangasarian, O. L., and Street, W. N. (1997), "Clustering via Concave Minimization," in *Advances in Neural Information Processing Systems* (Vol. 9), Cambridge, MA: MIT Press, pp. 368–374. [997]

Chen, S. S., Donoho, D. L., and Saunders, M. A. (1998), "Atomic Decomposition by Basis Pursuit," *SIAM Journal on Scientific Computing*, 20, 33–61. [996]

CVX Research, Inc (2012), "CVX: Matlab Software for Disciplined Convex Programming, version 2.0 beta," available at *http://cvxr.com/cvx*. [996]

Dasgupta, S., and Freund, Y. (2009), "Random Projection Trees for Vector Quantization," *IEEE Transactions on Information Theory*, 55, 3229–3242. [995]

Deng, W., and Yin, W. (2012), "On the Global and Linear Convergence of the Generalized Alternating Direction Method of Multipliers," Technical Report, CAAM TR12-14, Rice University. [1010]

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. (2008), "Efficient Projections onto the ℓ_1-Ball for Learning in High Dimensions," in *Proceedings of the Twenty Fifth International Conference on Machine Learning*, New York: ACM, pp. 272–279. [1000]

Elkan, C. (2003), "Using the Triangle Inequality to Accelerate *k*-Means," in *Proceedings of the Twenty Fifth International Conference on Machine Learning*, New York: ACM, pp. 147–153. [997]

Ferguson, T. S. (1973), "A Bayesian Analysis of Some Nonparametric Problems," *The Annals of Statistics*, 1, 209–230. [998]

Forgy, E. (1965), "Cluster Analysis of Multivariate Data: Efficiency Versus Interpretability of Classifications," *Biometrics*, 21, 768–780. [995]

Fraley, C. (1998), "Algorithms for Model-Based Gaussian Hierarchical Clustering," *SIAM Journal on Scientific Computing*, 20, 270–281. [998]

Frank, M., and Wolfe, P. (1956), "An Algorithm for Quadratic Programming," *Naval Research Logistics Quarterly*, 3, 95–110. [997]

Gabay, D., and Mercier, B. (1976), "A Dual Algorithm for the Solution of Nonlinear Variational Problems via Finite-Element Approximations," *Computational and Applied Mathematics*, 2, 17–40. [999]

Glowinski, R., and Marrocco, A. (1975), "Sur Lapproximation Par Elements Finis Dordre un, et la Resolution Par Penalisation-Dualite Dune Classe de Problemes de Dirichlet Nonlineaires," *Revue Française d'Automatique, Informatique, Recherche Opérationnelle*, 2, 41–76. [999]

Goldfarb, D., Ma, S., and Scheinberg, K. (2012), "Fast Alternating Linearization Methods for Minimizing the Sum of Two Convex Functions," *Mathematical Programming*, 1–34. [1010]

Goldstein, T., O'Donoghue, B., and Setzer, S. (2012), "Fast Alternating Direction Optimization Methods," Technical Report cam12-35, University of California, Los Angeles. [1006]

Goldstein, T., and Osher, S. (2009), "The Split Bregman Method for L1-Regularized Problems," *SIAM Journal on Imaging Sciences*, 2, 323–343. [999]

Gordon, A. (1999), *Classification* (2nd ed.), London: Chapman and Hall/CRC Press. [997]

Gower, J. C., and Ross, G. J. S. (1969), "Minimum Spanning Trees and Single Linkage Cluster Analysis," *Journal of the Royal Statistical Society*, Series C, 18, 54–64. [995]

Grant, M., and Boyd, S. (2008), "Graph Implementations for Nonsmooth Convex Programs," in *Recent Advances in Learning and Control*, eds. V. Blondel, S. Boyd and H. Kimura, Berlin: Springer-Verlag Limited, Lecture Notes in Control and Information Sciences, pp. 95–110, available at *http://stanford.edu/boyd/graph_dcp.html*. [996]

Hartigan, J. (1975), *Clustering Algorithms*, New York: Wiley. [997,1007]

Hocking, T., Vert, J.-P., Bach, F., and Joulin, A. (2011), "Clusterpath: An Algorithm for Clustering Using Convex Fusion Penalties," in *Proceedings of the Twenty Eighth International Conference on Machine Learning*, New York: ACM, pp. 745–752. [994,995,996,997,998,999,1006,1008]

Hoefling, H. (2010), "A Path Algorithm for the Fused Lasso Signal Approximator," *Journal of Computational and Graphical Statistics*, 19, 984–1006. [995,997]

Johnson, S. (1967), "Hierarchical Clustering Schemes," *Psychometrika*, 32, 241–254. [995]

Kaufman, L., and Rousseeuw, P. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis*, New York: Wiley. [997]

Lance, G. N., and Williams, W. T. (1967), "A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems," *The Computer Journal*, 9, 373–380. [995]

Lindsten, F., Ohlsson, H., and Ljung, L. (2011), "Just Relax and Come Clustering! A Convexication of k-Means Clustering," Technical Report, Linköpings Universitet. [994,995,996,997,1006]

Lloyd, S. (1982), "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, 28, 129–137. [995]

Luxburg, U. (2007), "A Tutorial on Spectral Clustering," *Statistics and Computing*, 17, 395–416. [998]

MacQueen, J. (1967), "Some Methods for Classification and Analysis of Multivariate Observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, vol. 1, pp. 281–297. [995]

McLachlan, G. (2000), *Finite Mixture Models*, Hoboken, NJ: Wiley. [998]

Michelot, C. (1986), "A Finite Algorithm for Finding the Projection of a Point Onto the Canonical Simplex of $\mathbb{R}^n$," *Journal of Optimization Theory and Applications*, 50, 195–200. [1000]

Mirkin, B. (1996), *Mathematical Classification and Clustering*, Dordrecht, The Netherlands: Kluwer Academic Publishers. [997]

Murtagh, F. (1983), "A Survey of Recent Advances in Hierarchical Clustering Algorithms," *The Computer Journal*, 26, 354–359. [995]

Neal, R. M. (2000), "Markov Chain Sampling Methods for Dirichlet Process Mixture Models," *Journal of Computational and Graphical Statistics*, 9, 249–265. [998]

Rasmussen, C. E. (2000), "The Infinite Gaussian Mixture Model," in *Advances in Neural Information Processing Systems* (vol. 12), Cambridge, MA: MIT Press, pp. 554–560. [998]

Slaney, M., and Casey, M. (2008), "Locality-Sensitive Hashing for Finding Nearest Neighbors [Lecture Notes]," *IEEE Signal Processing Magazine*, 25, 128–131. [1011]

Sneath, P. H. A. (1957), "The Application of Computers to Taxonomy," *Journal of General Microbiology*, 17, 201–226. [997]

Sørensen, T. (1948), "A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species and Its Application to Analyses of the Vegetation on Danish Commons," *Biologiske Skrifter*, 5, 1–34. [997]

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, Series B, 58, 267–288. [996]

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005), "Sparsity and Smoothness via the Fused Lasso," *Journal of the Royal Statistical Society*, Series B, 91–108. [995]

Tibshirani, R. J., and Taylor, J. (2011), "The Solution Path of the Generalized Lasso," *The Annals of Statistics*, 39, 1335–1371. [995]

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985), *Statistical Analysis of Finite Mixture Distributions*, Hoboken, NJ: Wiley. [998]

Tropp, J. (2006), "Just Relax: Convex Programming Methods for Identifying Sparse Signals in Noise," *IEEE Transactions on Information Theory*, 52, 1030–1051. [994]

Tseng, P. (1991), "Applications of a Splitting Algorithm to Decomposition in Convex Programming and Variational Inequalities," *SIAM Journal on Control and Optimization*, 29, 119–138. [999,1003]

von Luxburg, U. (2010), "Clustering Stability: An Overview," *Foundation and Trends in Machine Learning*, 2, 235–274. [1011]

Ward, J. H. (1963), "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, 58, 236–244. [995,997]

Wickham, H. (2009), *ggplot2: Elegant Graphics for Data Analysis*, New York: Springer New York. [1011]

Witten, D. M., and Tibshirani, R. (2010), "A Framework for Feature Selection in Clustering," *Journal of the American Statistical Association*, 105, 713–726. [1011]

Wu, R., and Wunsch, D. (2009), *Clustering*, Hoboken, NJ: Wiley. [997]