Certifying clusters from sum-of-norms clustering

Tao Jiang[1] Stephen Vavasis[2]

[1]Cornell University, School of Operations Research and Information Engineering

[2]University of Waterloo, Combinatorics & Optimization

Clustering

Informally: Given n points $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n \in \mathbb{R}^d$, partition $\{1, \ldots, n\}$ into k subsets $C_1, \ldots, C_k$ such that for $i \in C_m, i' \in C_{m'}$, $\mathrm{dist}(\boldsymbol{a}_i, \boldsymbol{a}_{i'})$ is small iff $m = m'$.

Clustering

Informally: Given n points $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n \in \mathbb{R}^d$, partition $\{1, \ldots, n\}$ into k subsets $C_1, \ldots, C_k$ such that for $i \in C_m, i' \in C_{m'}$, $\mathrm{dist}(\boldsymbol{a}_i, \boldsymbol{a}_{i'})$ is small iff $m = m'$.

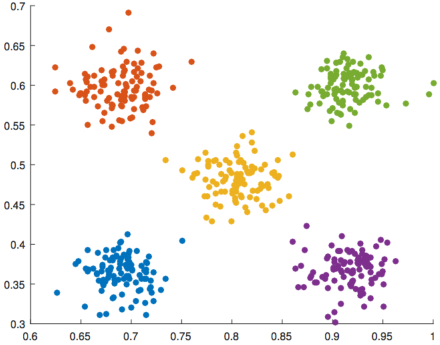
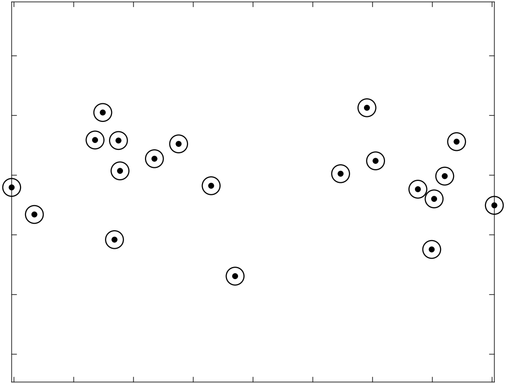
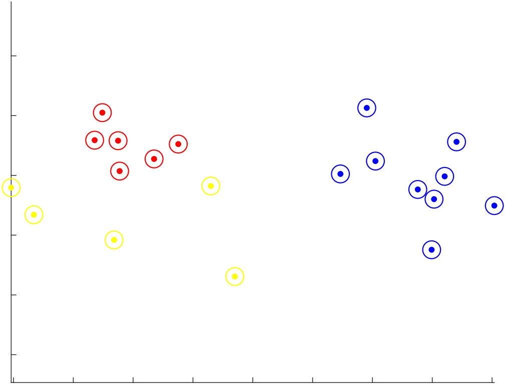


Figure: Visualization of a possible clustering

Example: Clustering ($d = 2$, $n = 20$)



Example: Clustering ($d = 2$, $n = 20$)



Yellow nodes are in singleton clusters.

Lloyd's algorithm

Best known method for clustering is Lloyd's algorithm ("k-means").
Assume k (number of clusters) is part of the input.

Lloyd's algorithm

Best known method for clustering is Lloyd's algorithm ("k-means"). Assume k (number of clusters) is part of the input.

1. Randomly partition n points into k random subsets $C_1, \ldots, C_k$, which serve as initial clusters.

Lloyd's algorithm

Best known method for clustering is Lloyd's algorithm ("k-means").
Assume k (number of clusters) is part of the input.

1. Randomly partition n points into k random subsets $C_1, \ldots, C_k$, which serve as initial clusters.
2. Alternate the following two operations until the cluster assignments stop changing:
   (a) For each cluster C_i, compute the cluster centroids.

Lloyd's algorithm

Best known method for clustering is Lloyd's algorithm ("k-means").
Assume k (number of clusters) is part of the input.

1. Randomly partition n points into k random subsets $C_1, \ldots, C_k$, which serve as initial clusters.
2. Alternate the following two operations until the cluster assignments stop changing:
   (a) For each cluster C_i, compute the cluster centroids.
   (b) Assign each observation to the cluster whose centroid is closest.

Lloyd's algorithm

Best known method for clustering is Lloyd's algorithm ("k-means").
Assume k (number of clusters) is part of the input.

1. Randomly partition n points into k random subsets $C_1, \ldots, C_k$, which serve as initial clusters.
2. Alternate the following two operations until the cluster assignments stop changing:
   (a) For each cluster C_i, compute the cluster centroids.
   (b) Assign each observation to the cluster whose centroid is closest.

Issues with Lloyd's algorithm

Corresponds to nonconvex optimization, so many local minimizers,

$\rightarrow$ sensitive to initialization;

$\rightarrow$ hard to prove properties of clustering output.

Sum-of-norms clustering

Find clusters by solving the convex optimization problem:

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_n} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a}_i\|^2 + \lambda \sum_{1\le i<j\le n} \|\mathbf{x}_i - \mathbf{x}_j\|,$$

which is known as the sum-of-norms clustering[1].

[1] Discovered independently by Pelckmans et al. (2005), Lindsten et al. (2011), Hocking et al. (2011).

[2] Hocking et al. (2011), Chiquet et al (2014)

[3] Panahi et al. (2017), Sun et al. (2018), Jiang, Vavasis, Zhai (2020)

Sum-of-norms clustering

Find clusters by solving the convex optimization problem:

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_n} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a}_i\|^2 + \lambda \sum_{1\le i<j\le n} \|\mathbf{x}_i - \mathbf{x}_j\|,$$

which is known as the sum-of-norms clustering[1].

	Lloyd's algorithm	**Sum-of-norms clustering**
convexity	non-convex	strongly convex
minimizers	many local minimizers	unique local (and global) minimizer
initialization	sensitive to initialization	independent of initialization
cluster output	hard to prove properties	agglomerative[2], recovery of MoG[3]

[1] Discovered independently by Pelckmans et al. (2005), Lindsten et al. (2011), Hocking et al. (2011).

[2] Hocking et al. (2011), Chiquet et al (2014)

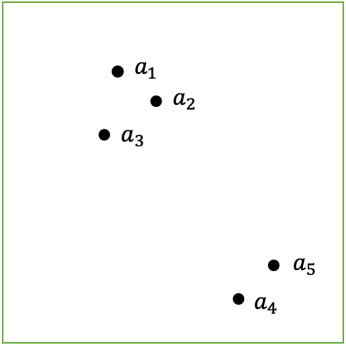
[3] Panahi et al. (2017), Sun et al. (2018), Jiang, Vavasis, Zhai (2020)

Sum-of-norms clustering formulation

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_n} \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i-\mathbf{a}_i\|^2+\lambda\sum_{1\le i<j\le n}\|\mathbf{x}_i-\mathbf{x}_j\|$$

Sum-of-norms clustering formulation

$$\min_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n} \frac{1}{2}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{a}_i\|^2 + \lambda \sum_{1\le i<j\le n} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$$



Data $\boldsymbol{a}_i$'s: Given n observations $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$

Sum-of-norms clustering formulation

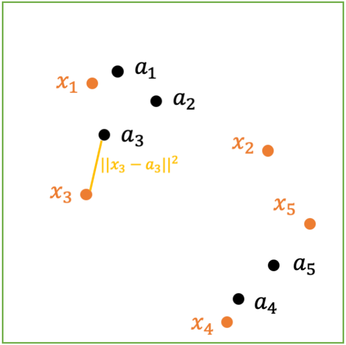
$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_n} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a}_i\|^2 + \lambda \sum_{1\le i<j\le n} \|\mathbf{x}_i - \mathbf{x}_j\|$$

Variable x_i's: Define unconstrained variable x_i for $i = 1, \ldots, n$.

We may interpret the optimal $\mathbf{x}_i^*$ as the cluster centroid that i is closest to.

Sum-of-norms clustering formulation

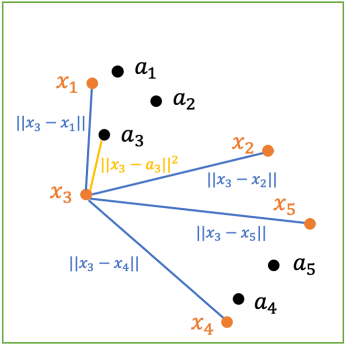
$$\min_{x_1,\ldots,x_n} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a}_i\|^2 + \lambda \sum_{1\le i<j\le n} \|\mathbf{x}_i - \mathbf{x}_j\|$$



Intuition: first term favors $\mathbf{x}_i^*$ close to $\mathbf{a}_i$, while second term tends to make $\mathbf{x}_i^*$ for many i's equal to each other.

Sum-of-norms clustering formulation

$$\min_{x_1,\ldots,x_n} \frac{1}{2}\sum_{i=1}^{n} \|x_i - a_i\|^2 + \lambda \sum_{1\le i<j\le n} \|x_i - x_j\|$$



Intuition: first term favors x_i^* close to a_i, while second term tends to make x_i^* for many i's equal to each other.

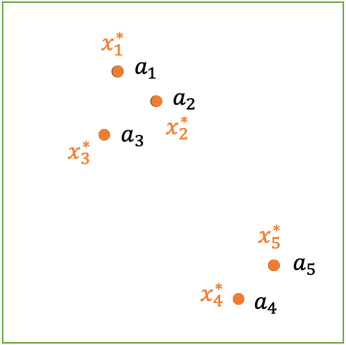
Sum-of-norms clustering formulation

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_n} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a}_i\|^2 + \lambda \sum_{1\le i<j\le n} \|\mathbf{x}_i - \mathbf{x}_j\|$$

Cluster recovery: points i, j get clustered together iff $\mathbf{x}_i^* = \mathbf{x}_j^*$

Sum-of-norms clustering formulation

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_n} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a}_i\|^2 + \lambda \sum_{1\le i<j\le n} \|\mathbf{x}_i - \mathbf{x}_j\|$$



Role of λ: when $\lambda = 0$, all noncoincident $\mathbf{a}_i$'s are in singleton clusters.

Sum-of-norms clustering formulation

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_n} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a}_i\|^2 + \lambda \sum_{1\le i<j\le n} \|\mathbf{x}_i - \mathbf{x}_j\|$$

Role of λ: there exists $\bar{\lambda}$ (depending on data) such that for all $\lambda \ge \bar{\lambda}$, all $\mathbf{a}_i$'s are in one large cluster.

Sum-of-norms clustering formulation

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_n} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a}_i\|^2 + \lambda \sum_{1\le i<j\le n} \|\mathbf{x}_i - \mathbf{x}_j\|$$

Role of λ: as λ increases, number of clusters goes down.

Thus, λ controls the number of clusters indirectly.

Identifying clusters

Issue: Identifying the clusters apparently requires exact knowledge of the optimizer, but all known algorithms are iterative.

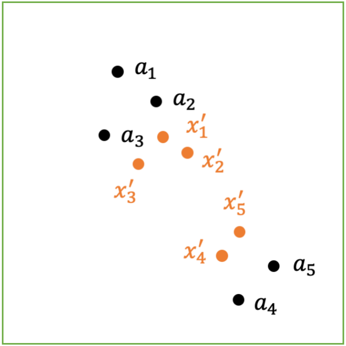
Identifying clusters

Issue: Identifying the clusters apparently requires exact knowledge of the optimizer, but all known algorithms are iterative.

How to identify clusters from *an approximate solution* with *mathematical guarantee* instead of using *an exact optimizer*?

Identifying clusters

Issue: Identifying the clusters apparently requires exact knowledge of the optimizer, but all known algorithms are iterative.



Let $\mathbf{x}'_1, \ldots, \mathbf{x}'_n$ be an approximate optimizer from an iterative method.

Identifying clusters

Issue: Identifying the clusters apparently requires exact knowledge of the optimizer, but all known algorithms are iterative.

Authors in practice use a tolerance:
say i, j are in the same cluster if $\left\| \mathbf{x}'_i - \mathbf{x}'_j \right\| \leq \epsilon$.

Identifying clusters

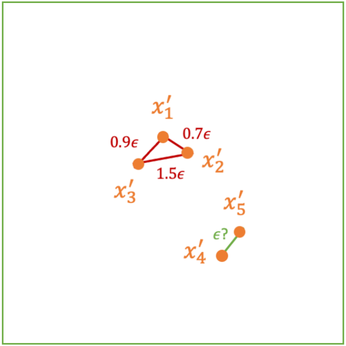
Issue: Identifying the clusters apparently requires exact knowledge of the optimizer, but all known algorithms are iterative.

For which ϵ is the recovery of the true clustering guaranteed?

Do the recovery of a MoG and the agglomerative property still hold?

Identifying clusters

Issue: Identifying the clusters apparently requires exact knowledge of the optimizer, but all known algorithms are iterative.



What if $\|\boldsymbol{x}'_1 - \boldsymbol{x}'_2\| < \epsilon$, $\|\boldsymbol{x}'_1 - \boldsymbol{x}'_3\| < \epsilon$, $\|\boldsymbol{x}'_2 - \boldsymbol{x}'_3\| > \epsilon$?

Our contributions

Test: We propose a test that takes an approximate solution and attempts to determine all clusters. The test may report 'success' or 'failure'.

Our contributions

Test: We propose a test that takes an approximate solution and attempts to determine all clusters. The test may report ‘success’ or ‘failure’.

Theorem 1. If the test reports ‘success’, then the clusters are correctly identified.

Our contributions

Test: We propose a test that takes an approximate solution and attempts to determine all clusters. The test may report 'success' or 'failure'.

Theorem 1. If the test reports 'success', then the clusters are correctly identified.

Theorem 2. If an interior-point algorithm is used, then the test is guaranteed to report 'success' after a finite number of iterations except . . .

When does the test fail?

. . . the test may never report 'success' for the particular values of λ at which clusters fuse to form a larger cluster.

When does the test fail?

. . . the test may never report ‘success’ for the particular values of λ at which clusters fuse to form a larger cluster.

Because of the agglomeration property, there are at most n such discrete values of λ for which the test may never succeed.

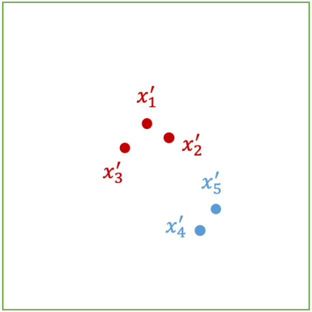
The test

Input: a primal and dual feasible solution $(\boldsymbol{x}', \omega')$, duality gap μ

The test

Input: a primal and dual feasible solution $(\boldsymbol{x}', \omega')$, duality gap μ

Step 0. Determine all clusters greedily according to $\boldsymbol{x}', \mu$:

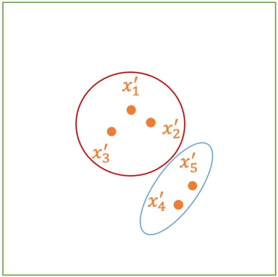


The test

Input: a primal and dual feasible solution $(\boldsymbol{x}', \omega')$, duality gap μ

Step 0. Determine all clusters greedily according to $\boldsymbol{x}', \mu$:

Step 1. Compute subgradients from $(\boldsymbol{x}', \omega')$ and check if they satisfy the CGR condition (will specify later).



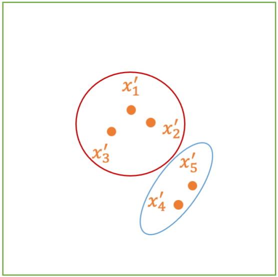
The test

Input: a primal and dual feasible solution $(\boldsymbol{x}', \omega')$, duality gap μ

Step 0. Determine all clusters greedily according to $\boldsymbol{x}', \mu$:

Step 1. Compute subgradients from $(\boldsymbol{x}', \omega')$ and check if they satisfy the CGR condition (will specify later).

True: points clustered in Step 0 indeed belong in the same cluster, proceed to Step 2.



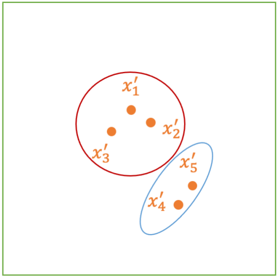
The test

Input: a primal and dual feasible solution $(\boldsymbol{x}', \omega')$, duality gap μ

Step 0. Determine all clusters greedily according to $\boldsymbol{x}', \mu$:

Step 1. Compute subgradients from $(\boldsymbol{x}', \omega')$ and check if they satisfy the CGR condition (will specify later).

- **True:** points clustered in Step 0 indeed belong in the same cluster, proceed to Step 2.
- **False:** the test reports "failure".



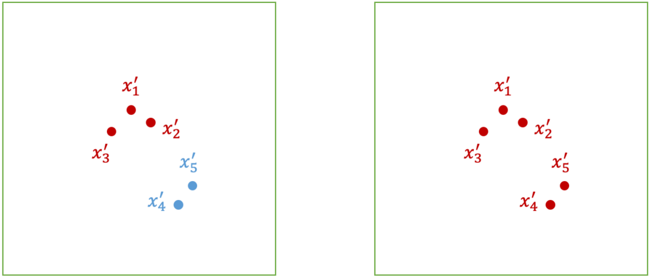
The test

Input: a primal and dual feasible solution $(\boldsymbol{x}', \omega')$, duality gap μ

Step 0. Determine all clusters greedily according to $\boldsymbol{x}', \mu$:

Step 1. Compute subgradients from $(\boldsymbol{x}', \omega')$ and check if they satisfy the CGR condition (will specify later).

- **True:** points clustered in Step 0 indeed belong in the same cluster, proceed to Step 2.
- **False:** the test reports "failure".



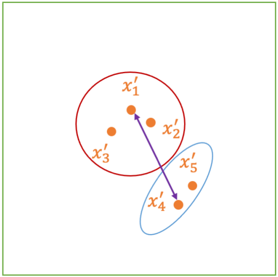
The test

Input: a primal and dual feasible solution $(\boldsymbol{x}', \omega')$, duality gap μ

Step 0. Determine all clusters greedily according to $\boldsymbol{x}', \mu$:

Step 1. Compute subgradients from $(\boldsymbol{x}', \omega')$ and check if they satisfy the CGR condition (will specify later).

Step 2. Check that no two clusters are distance $\leq c_n \mu^{1/2}$ of each other.



The test

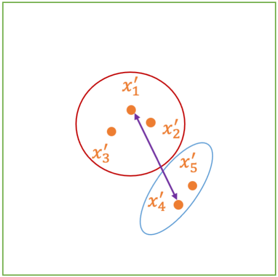
Input: a primal and dual feasible solution $(\boldsymbol{x}', \omega')$, duality gap μ

Step 0. Determine all clusters greedily according to $\boldsymbol{x}', \mu$:

Step 1. Compute subgradients from $(\boldsymbol{x}', \omega')$ and check if they satisfy the CGR condition (will specify later).

Step 2. Check that no two clusters are distance $\leq c_n \mu^{1/2}$ of each other.

True: all clusters identified in step 0 are distinct.



The test

Input: a primal and dual feasible solution $(\boldsymbol{x}', \omega')$, duality gap μ

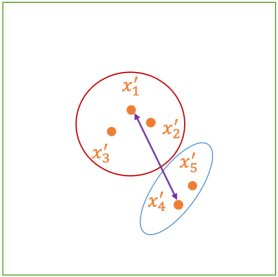
Step 0. Determine all clusters greedily according to $\boldsymbol{x}', \mu$:

Step 1. Compute subgradients from $(\boldsymbol{x}', \omega')$ and check if they satisfy the CGR condition (will specify later).

Step 2. Check that no two clusters are distance $\leq c_n \mu^{1/2}$ of each other.

True: all clusters identified in step 0 are distinct.

False: the test reports "failure".



The test

Input: a primal and dual feasible solution $(\boldsymbol{x}', \omega')$, duality gap μ

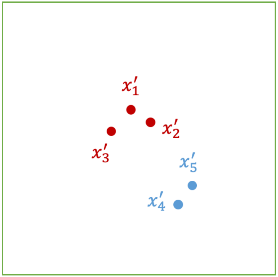
Step 0. Determine all clusters greedily according to $\boldsymbol{x}', \mu$:

Step 1. Compute subgradients from $(\boldsymbol{x}', \omega')$ and check if they satisfy the CGR condition (will specify later).

Step 2. Check that no two clusters are distance $\leq c_n \mu^{1/2}$ of each other.

True: all clusters identified in step 0 are distinct.

False: the test reports "failure".



CGR condition

Step 1. Compute subgradients from $(\boldsymbol{x}', \omega')$ and check if they satisfy the CGR condition.

True: points clustered in Step 0 indeed belong in the same cluster.

False: the test reports "failure".

CGR condition

Step 1. Compute subgradients from $(\boldsymbol{x}', \omega')$ and check if they satisfy the CGR condition.

> **True:** points clustered in Step 0 indeed belong in the same cluster.
>
> **False:** the test reports "failure".

CGR condition by Chiquet, Gutierrez and Rigaill (2017)

Suppose $\emptyset \neq C \subseteq \{1, \ldots, n\}$. Suppose there exists a solution $\boldsymbol{q}_{ij}^*$ for $j \in C - \{i\}$, $i \in C$ to the following system.

$$
\begin{aligned}
\boldsymbol{a}_i - \frac{1}{|C|} \sum_{l \in C} \boldsymbol{a}_l &= \lambda \sum_{j \in C - \{i\}} \boldsymbol{q}_{ij}^* && \forall i \in C, \\
\left\| \boldsymbol{q}_{ij}^* \right\| &\leq 1 && \forall i, j \in C, i \neq j, \\
\boldsymbol{q}_{ij}^* &= -\boldsymbol{q}_{ji}^* && \forall i, j \in C, i \neq j.
\end{aligned}
\tag{1}
$$

Then there exists an $\hat{\boldsymbol{x}} \in \mathbb{R}^d$ such that the minimizer $\boldsymbol{x}^*$ of our sum-of-norms clustering problem satisfies $\boldsymbol{x}_i^* = \hat{\boldsymbol{x}}$ for $i \in C$.

CGR condition

Step 1. Compute subgradients from $(\boldsymbol{x}', \omega')$ and check if they satisfy the CGR condition.

True: points clustered in Step 0 indeed belong in the same cluster.

False: the test reports "failure".

CGR condition by Chiquet, Gutierrez and Rigaill (2017)

Suppose $\emptyset \neq C \subseteq \{1, \ldots, n\}$. Suppose there exists a solution $\boldsymbol{q}^*_{ij}$ for $j \in C - \{i\}$, $i \in C$ to the following system.

$$\begin{aligned} \boldsymbol{a}_i - \frac{1}{|C|}\sum_{l \in C} \boldsymbol{a}_l &= \lambda \sum_{j \in C - \{i\}} \boldsymbol{q}^*_{ij} && \forall i \in C, \\ \left\| \boldsymbol{q}^*_{ij} \right\| &\leq 1 && \forall i, j \in C, i \neq j, \\ \boldsymbol{q}^*_{ij} &= -\boldsymbol{q}^*_{ji} && \forall i, j \in C, i \neq j. \end{aligned} \tag{1}$$

Then there exists an $\hat{\boldsymbol{x}} \in \mathbb{R}^d$ such that the minimizer $\boldsymbol{x}^*$ of our sum-of-norms clustering problem satisfies $\boldsymbol{x}^*_i = \hat{\boldsymbol{x}}$ for $i \in C$.

A sufficient condition for clustering: If there exists a solution to system (1), then points in C indeed belong to the same cluster.

Distinct clusters

Step 2. Check that no two clusters are distance $\leq c_n \mu^{1/2}$ of each other.

True: all clusters identified in step 0 are distinct.

False: the test reports "failure".

Distinct clusters

Step 2. Check that no two clusters are distance $\leq c_n \mu^{1/2}$ of each other.

True: all clusters identified in step 0 are distinct.

False: the test reports "failure".

Distinct clustering theorem

If there exist $i, j \in C$ such that $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| > 2\sqrt{2\mu}$, then i, j are not in the same cluster and C is not a cluster or part of a larger cluster. (The result holds due to strong convexity.)

Distinct clusters

Step 2. Check that no two clusters are distance $\leq c_n \mu^{1/2}$ of each other.

True: all clusters identified in step 0 are distinct.

False: the test reports "failure".

Distinct clustering theorem

If there exist $i, j \in C$ such that $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| > 2\sqrt{2\mu}$, then i, j are not in the same cluster and C is not a cluster or part of a larger cluster. (The result holds due to strong convexity.)

A sufficient condition for distinct clusters: If all clusters are pairwise far apart, then no cluster identified in Step 0 is actually a subcluster of a larger cluster.

Establishing Theorem 1 (correctness of the test)

If the test reports "success":

[4] Instead of using obvious subgradients, we constructed a version of subgradients that automatically satisfies the equalities in (1).

Establishing Theorem 1 (correctness of the test)

If the test reports "success":

Successful step 1: our subgradients[4]satisfy the CGR condition;

[4] Instead of using obvious subgradients, we constructed a version of subgradients that automatically satisfies the equalities in (1).

Establishing Theorem 1 (correctness of the test)

If the test reports "success":

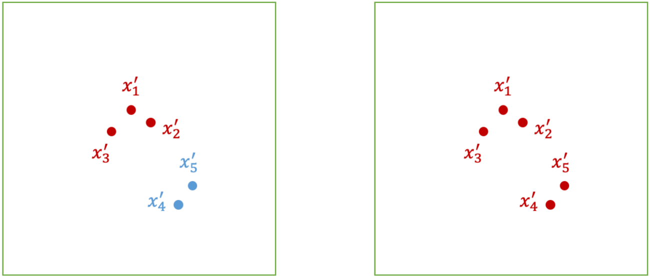
Successful step 1: our subgradients[4] satisfy the CGR condition;
$\rightarrow$ CGR condition certifies that points clustered in Step 0 indeed belong in the same cluster.

[4] Instead of using obvious subgradients, we constructed a version of subgradients that automatically satisfies the equalities in (1).

Establishing Theorem 1 (correctness of the test)

If the test reports "success":

Successful step 1: our subgradients[4] satisfy the CGR condition;
→ CGR condition certifies that points clustered in Step 0 indeed belong in the same cluster.



[4] Instead of using obvious subgradients, we constructed a version of subgradients that automatically satisfies the equalities in (1).

Establishing Theorem 1 (correctness of the test)

If the test reports "success":

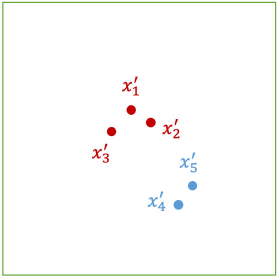
Successful step 2: Each pair of clusters are well separated;
$\rightarrow$ "distinct cluster theorem" certifies that no cluster identified in Step 0 is actually a subcluster of a larger cluster.

[4] Instead of using obvious subgradients, we constructed a version of subgradients that automatically satisfies the equalities in (1).

Establishing Theorem 1 (correctness of the test)

If the test reports “success”:

Successful step 2: Each pair of clusters are well separated;
→ “distinct cluster theorem” certifies that no cluster identified in Step 0 is actually a subcluster of a larger cluster.



[4] Instead of using obvious subgradients, we constructed a version of subgradients that automatically satisfies the equalities in (1).

Establishing Theorem 2 (eventual success)

Theorem 2. If an interior-point algorithm is used, then the test is guaranteed to report 'success' after a finite number of iterations except . . .

Establishing Theorem 2 (eventual success)

Theorem 2. If an interior-point algorithm is used, then the test is guaranteed to report 'success' after a finite number of iterations except ...

the test may never report 'success' for the particular values of λ at which clusters fuse to form a larger cluster.

Establishing Theorem 2 (eventual success)

Theorem 2. If an interior-point algorithm is used, then the test is guaranteed to report 'success' after a finite number of iterations except . . .

the test may never report 'success' for the particular values of λ at which clusters fuse to form a larger cluster.

Remark 1: Proof of Theorem 2 requires a deeper dive into duality.

Establishing Theorem 2 (eventual success)

Theorem 2. If an interior-point algorithm is used, then the test is guaranteed to report 'success' after a finite number of iterations except ...

the test may never report 'success' for the particular values of λ at which clusters fuse to form a larger cluster.

Remark 1: Proof of Theorem 2 requires a deeper dive into duality.

Remark 2: Ingredient of Theorem 2 proof is a result by Luo, Sturm and Zhang (1998) that, provided the optimizer satisfies strict complementarity, interior point iterates are $O(\mu)$ away from optimizer, where μ is the duality gap (scaled central path parameter).

Explanation of failure case

Not surprising that test fails when λ is exactly at a fusion value λ^*, since any arbitrarily small negative perturbation $\lambda^* - \epsilon$ yields a different clustering.

Explanation of failure case

Not surprising that test fails when λ is exactly at a fusion value λ^*, since any arbitrarily small negative perturbation $\lambda^* - \epsilon$ yields a different clustering.

Complete cluster identification for these values of λ^* is *ill-posed*; unreasonable to expect an algorithm to satisfy a guarantee for such a problem.

Summary

Test: We propose a test that takes an approximate solution and attempts to determine all clusters. The test may report 'success' or 'failure'.

Summary

Test: We propose a test that takes an approximate solution and attempts to determine all clusters. The test may report ‘success’ or ‘failure’.

Theorem 1. If the test reports ‘success’, then the clusters are correctly identified.

Summary

Test: We propose a test that takes an approximate solution and attempts to determine all clusters. The test may report 'success' or 'failure'.

Theorem 1. If the test reports 'success', then the clusters are correctly identified.

Theorem 2. If an interior-point algorithm is used, then the test is guaranteed to report 'success' after a finite number of iterations except when the clustering problem is ill-posed.

Appendix: Squared versus unsquared norm: simple example

Squared versus unsquared norm

If the norms in the second term were also squared, then it would almost never happen that $\boldsymbol{x}_i^* = \boldsymbol{x}_j^*$ when $i \neq j$.

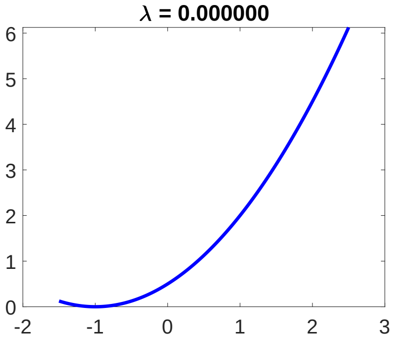
$$\min_x (x+1)^2/2 + \lambda|x-2|$$

Appendix: Squared versus unsquared norm: simple example

Squared versus unsquared norm

If the norms in the second term were also squared, then it would almost never happen that $\mathbf{x}_i^* = \mathbf{x}_j^*$ when $i \neq j$.

$$\min_x (x+1)^2/2 + \lambda|x-2|$$

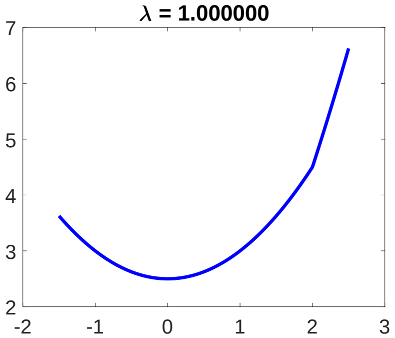


Appendix: Squared versus unsquared norm: simple example

Squared versus unsquared norm

If the norms in the second term were also squared, then it would almost never happen that $\mathbf{x}_i^* = \mathbf{x}_j^*$ when $i \neq j$.

$$\min_x (x+1)^2/2 + \lambda|x-2|$$

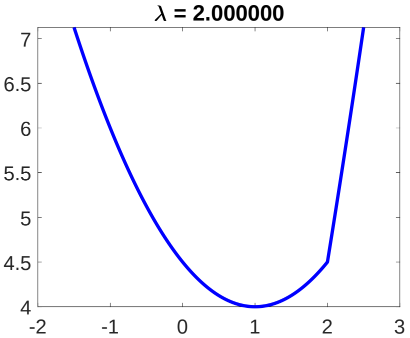


Appendix: Squared versus unsquared norm: simple example

Squared versus unsquared norm

If the norms in the second term were also squared, then it would almost never happen that $\mathbf{x}_i^* = \mathbf{x}_j^*$ when $i \neq j$.

$$\min_x (x+1)^2/2 + \lambda|x-2|$$

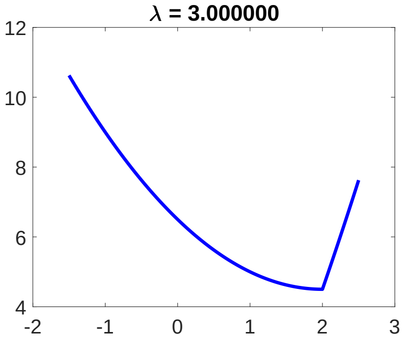


Appendix: Squared versus unsquared norm: simple example

Squared versus unsquared norm

If the norms in the second term were also squared, then it would almost never happen that $\mathbf{x}_i^* = \mathbf{x}_j^*$ when $i \neq j$.

$$\min_x (x+1)^2/2 + \lambda|x-2|$$

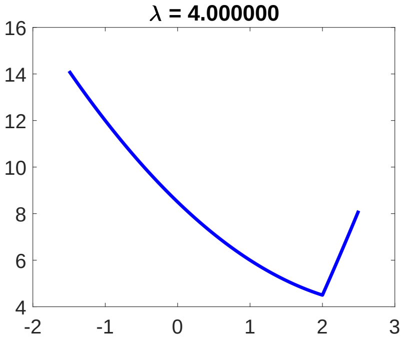


Appendix: Squared versus unsquared norm: simple example

Squared versus unsquared norm

If the norms in the second term were also squared, then it would almost never happen that $\mathbf{x}_i^* = \mathbf{x}_j^*$ when $i \neq j$.

$$\min_x (x+1)^2/2 + \lambda|x-2|$$



Appendix: Squared versus unsquared norm: simple example

Squared versus unsquared norm

If the norms in the second term were also squared, then it would almost never happen that $\mathbf{x}_i^* = \mathbf{x}_j^*$ when $i \neq j$.

$$\min_x (x+1)^2/2 + \lambda|x-2|$$