Clustering multi-way data: a novel algebraic approach

Anonymous Author(s)

Affiliation

Address

email

Abstract

In this paper, we develop a method for unsupervised clustering of two-way (matrix) data by combining two recent innovations from different fields: the Sparse Subspace Clustering (SSC) algorithm [6], which groups points coming from a union of subspaces into their respective subspaces, and the t-product [13], which was introduced to provide a matrix-like multiplication for third order tensors. Our algorithm is analogous to SSC in that an “affinity” between different data points is built using a sparse self-representation of the data. Unlike SSC, we employ the t-product in the self-representation. This allows us more flexibility in modeling; in fact, SSC is a special case of our method.

When using the t-product, three-way arrays are treated as matrices whose elements (scalars) are n-tuples or tubes. Convolutions take the place of scalar multiplication. This framework allows us to embed the 2-D data into a vector-space-like structure called a free module over a commutative ring. These free modules retain many properties of complex inner-product spaces, and we leverage that to provide theoretical guarantees on our algorithm. We believe the submodule clustering framework carries an advantage in some image clustering problems and we show the performance of the proposed method on Weizmann face database, the Yale Face database and the MNIST handwritten digits data base.

1 Introduction

In most clustering algorithms, the objects are assumed to be embedded in a normed linear space, and similarity is measured by a distance-like function. Among many different options, this can be followed by construction of a weighted graph in which similar points are joined by strong edges. Then, using tools from spectral graph theory, it is possible to find the clusters as connected graph components [1].

In contrast, subspace clustering techniques take a different perspective on the geometry of clustering. In these approaches, points are assumed to come from a union of subspaces rather than from disjoint, volume-filling clusters. For subspace clustering methods using spectral graph theory as a final step (which many do), any relevant notion of similarity should reflect whether points belong to the same subspace. Simple variations on distance are no longer effective. For this problem, a diverse array of subspace clustering methods [23], such as Sparse Subspace Clustering (SSC), can be employed to resolve the clusters. Some even reject outliers [6, 9,10,19].

For clustering data with two-way structure, such as images, typical subspace clustering methods must unfold the data or map it to a vector. This approach sacrifices the two-way structure, potentially failing to exploit useful information. Outside of subspace clustering, on the other hand, many methods take advantage of multi-way array structure, particularly for dimensionality reduction or finding the best (single) subspace in which to approximate the data. For examples of these, see [8], [7], [5], or [21] and references therein.
We know of no work exploiting multi-way structure in techniques similar to subspace clustering, and our goal is to fill that gap. In this paper we present a novel algebraic approach for clustering multi-way data. Whereas existing subspace clustering methods concatenate data as columns of a matrix, our method will group the data into a three-way array (a tensor), clustering slices of the tensor. Using the tensor products and factorizations outlined in [13], we will extend Sparse Subspace Clustering [6] for use on hypermatrix data. Although our strategy could process $N$-way ($N > 2$) data by incorporating technical tools in the style of [16], we chose to focus this work more narrowly and cluster only two-way data.

Before we begin with necessary background in Section 2, we would like to summarize our main contributions below.

**Our contributions:** We exploit the consequences of Theorem 2.1, which was proven in [4], to propose a new algebraic generative model, which we outline in Section 3. For inference with our model, we propose a novel clustering algorithm in Section 4.1. To characterize the algorithm’s performance, we bring in Definitions 2.2 through 2.5, which to our knowledge have not been used before, to derive performance bounds in Section 4.2. These results are of similar intuitive flavor as those in [6]. Generalizations of these geometrical notions for the t-product framework are not immediately obvious, and the key technical development which makes this possible is stated in Lemmas in the supplementary material. In Section 5, we conduct experiments with synthetic data, the Weizmann Face data base, the Yale Face Database, and the MNIST handwritten digits.

2 Background and Preliminaries

### 2.1 Sparse Subspace Clustering

Sparse Subspace Clustering (SSC) [6] is a recent method for solving the subspace clustering problem. Like many clustering methods, SSC constructs an affinity matrix whose $(i,j)$ entry is designed to be large when data points $i$ and $j$ are in the same subspace and relatively small or zero otherwise. To construct the affinity matrix, Sparse Subspace Clustering makes use of the fact that each datum can be expressed most efficiently as a linear combination of members of its own subspace. In particular, each data point is expressed as a linear combination of the others, with an $\ell_1$ regularization term to promote sparsity. [6], [20]. Spectral clustering [15] is then used to segment the data using the sparse coefficients as the affinity matrix. Variants of SSC have been considered in [19] and [24], which guarantee correctness for SSC with bounded noise. Similar algorithms include low-rank representation (LRR) [14], which uses a nuclear norm penalty in place of the $\ell_1$ regularization and also has theoretical guarantees on performance.

### 2.2 The t-product

We will now introduce the t-product [13, 4, 12] along with some notation. We will denote scalars with regular uppercase and lowercase font: $d_i, D \in \mathbb{R}$. Vectors will be in lowercase and bold, and matrices will be bold and uppercase, for example $v \in \mathbb{R}^n; X \in \mathbb{R}^{m \times n}$. In the following we will heavily rely upon the MATLAB notation for indexing the elements of a tensor; we treat tensors as multiway arrays stored in MATLAB. For readers unfamiliar with MATLAB “colon notation”, a colon in place of an index indicates that an entire cross-section of the array is being accessed; for example, column one of $X$ is $X(:, 1)$. We will also use the MATLAB function `fft` and its standard usage. On a three-way array, `fft(X, [], 3)` applies the transform to each “tube” $X(i, j, :)$. In Figure 1:

3-D tensors as operators on oriented matrices.

We view a 3-D tensor $X \in \mathbb{R}^{H \times L \times D}$ as an $H \times L$ matrix of tubes in $\mathbb{R}^{1 \times 1 \times D}$. These tubes will be denoted by adding an arrow over the vector notation: $\overrightarrow{v} \in \mathbb{R}^{1 \times 1 \times D}$. Similarly, we think of a $H \times 1 \times D$ tensor as a vector of tubes. We will call such tensors “oriented matrices” (see Figure 1) and denote them with arrows and calligraphic script: $\overrightarrow{M} \in \mathbb{R}^{H \times 1 \times D}$. In order to define matrix-like operations, we multiply “scalars” using a commutative operation between two tubes $\overrightarrow{v} \in \mathbb{R}^{1 \times 1 \times D}$ and $\overrightarrow{u} \in \mathbb{R}^{1 \times 1 \times D}$ resulting in another tube of same length. The commutative operation used is the

---

1See Section 2.2 for details of the theorem.
circular convolution; we will write * to denote the circular convolution. This algebra generalizes
matrix algebra because in the case of $D = 1$, convolution becomes scalar multiplication.

Under this construction, the product of a tensor $\mathbf{X} \in \mathbb{R}^{H \times L \times D}$ with an oriented matrix $\mathbf{M} \in \mathbb{R}^{H \times 1 \times D}$ is another oriented matrix of size $H \times 1 \times D$ whose $i$-th tubal element given by $\mathbf{X} * \mathbf{M}(i, j, :) = \sum_{j=1}^{L} \mathbf{X}(i, j, :) \odot \mathbf{M}(j, 1, :)$ as illustrated in Figure 1. Similarly one can extend this definition to define the multiplication of two tensors $\mathbf{X}$ and $\mathbf{Y}$ of sizes $H \times L \times D$ and $L \times K \times D$ respectively, resulting in a tensor $\mathbf{C} = \mathbf{X} * \mathbf{Y}$ of size $H \times K \times D$. This product between two tensors is referred to as the t-product, and its properties are developed in [13, 4, 12].

Computing the t-product using the FFT The t-product can be effectively computed using the Fast Fourier Transform. Using MATLAB notation and built-in functions resulting in a tensor $\mathbf{C}$ can be multiplied by computing $\mathbf{C} = \text{fft}(\mathbf{X}, 3), \mathbf{Y} = \text{fft}(\mathbf{Y}, 3)$ and $\mathbf{C} = \mathbf{X} * \mathbf{Y} = \text{ifft}(\mathbf{X} \odot \mathbf{Y}, 3)$ where $\mathbf{C} = \mathbf{X} \odot \mathbf{Y}$ is a tensor with the $d$-th frontal face is given by $\mathbf{C}(i, :, d) = \mathbf{X}(i, :, d) \mathbf{Y}(i, :, d)$. See [13] for proofs.

2.2.1 Linear algebra with the t-product: notation and facts

In linear algebra over a ring, the analog of a subspace is a free submodule. We begin this section by presenting a theorem from [4], because it and its corollaries imply that many useful properties of subspaces are still present with free submodules. In a variation on the notation in that work, we denote the set of length $D$ tubes by $\mathbb{K}_D$ and the set of oriented matrices of size $H \times D$ by $\mathbb{K}_D^H$. Likewise, $\mathbb{K}_{3D}^{L \times H}$ represents the set of tensors of size $H \times L \times D$. (We use $\mathbb{K}_D$ in place of $\mathbb{K}^{1 \times D}$ to signal that the algebraic structure associated with the set is the t-product rather than the more widespread framework of multilinear algebra.) We note that the set $\mathbb{K}_D$ forms a ring with identity using multiplication given by $*$ and the usual addition. Then we have the theorem from [4] as follows.

Theorem 2.1. The set of oriented matrices $\mathbb{K}_D^H$ forms a free module over $\mathbb{K}_D$.

Just as a subspace has a basis, a free submodule has a generating set so that any element of the free submodule can be written as a “t-linear combination” of generating elements $\mathcal{B}$. By “t-linear combination,” we mean a sum of oriented matrices multiplied, with the t-product, by coefficients from $\mathbb{K}_D$; we illustrate this in Figure 2. Furthermore, free submodules have the unique dimension property: if two generating sets are linearly independent, they will be the same size, and this size is what we refer to as the submodular dimension (we will define linear independence in Definition 2.2).

We now define a transpose for third order tensors. This definition is from [13], in which it is shown that the new transpose preserves the following property $(\mathcal{A} * \mathcal{B})^\top = (\mathcal{B}^\top * \mathcal{A}^\top)$.

Definition 2.1. Tensor Transpose. Let $\mathbf{X}$ be a tensor of size $n_1 \times n_2 \times n_3$. Then $\mathbf{X}^\top$ is the $n_2 \times n_1 \times n_3$ tensor obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through $n_3$.

Next, we will define ideas of sum, linear independence, and disjointness for submodules which are useful for subsequent analysis. In the case of scalars with depth one, these reduce to the linear algebraic definitions as seen in [23]. We also define linear independence for lists of oriented matrices and a transpose for elements of $\mathbb{K}_D^{H \times W}$. Submodules will be denoted by blackboard bold: $S \subset \mathbb{K}_D^H$. In this article, we will consider only free submodules.

Definition 2.2. The sum of two submodules $S_i$ and $S_j$ is defined as $S_i + S_j = \{ \mathbf{X} | \exists \mathbf{Y} \in S_i, \mathbf{W} \in S_j \text{ so that } \mathbf{X} = \mathbf{Y} + \mathbf{W} \}$.

Definition 2.3. A collection of submodules is linearly independent if each one intersects the sum of the others only at the zero element of $\mathbb{K}_D^H$.

\footnote{The adjective “free” applies when the submodule has a generating set.}
Definition 2.4. A collection of submodules is disjoint if each one intersects the union of the others only at the zero element of $\mathbb{R}^{H \times D}$.

Definition 2.5. For a t-linear combination with scalar tubes $\mathcal{V}_n$, suppose $\sum_{n=1}^{N} \mathcal{V}_n \ast \mathcal{V}_n = 0$ implies $\mathcal{V}_n = 0$. Then, the collection of oriented matrices $\mathcal{V}_1 \ldots \mathcal{V}_N$ is linearly independent. Here, the first zero is an oriented matrix, and the second is a tube.

3 Generative model: Union of Submodules

In this paper we will model the 2-D data as coming from union of free submodules and derive an algorithm in Section 4.1 for clustering data. Before that we would like to explain some nuances of the proposed model.

What is the relationship between linear combinations and t-linear combinations? The original definition of the t-product was written in terms of block circulant matrices, not convolutions, and we include it here to help interpret our model. Let $\mathcal{A}^{(d)}$ denote the d-th frontal slice of $\mathcal{A}$. Using functions $\text{birc}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} & \mathcal{A}^{(D)} & \ldots & \mathcal{A}^{(2)} \\ \vdots & \mathcal{A}^{(1)} & \ddots & \vdots \\ \mathcal{A}^{(D-1)} & \vdots & \ddots & \mathcal{A}^{(D)} \\ \mathcal{A}^{(D)} & \mathcal{A}^{(D-1)} & \ldots & \mathcal{A}^{(1)} \end{bmatrix}$, unfold $(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} \\ \vdots \\ \mathcal{A}^{(D)} \end{bmatrix}$, and fold () to invert unfold (), the t-product $\mathcal{A} \ast \mathcal{B}$ was given by fold (birc($\mathcal{A}$)unfold($\mathcal{B}$)).

Given a t-linear combination like the one in Figure 2, Figure 3 focuses in on a single oriented matrix and its “scalar” coefficient. For illustrative purposes, the coefficient is shown with only a single nonzero, in position 2. This nonzero element “selects” the second column of the block circulant matrix, which is then folded back into an oriented matrix. After folding, we observe that the new “scalar multiple” has been shifted relative to the original.

Why use submodules in place of subspaces? The takeaway from the demo above is that every oriented matrix spans a subspace containing multiples only of itself, but generates a submodule containing multiples of certain permutations of its column fibers as well. The presence of these shifted copies is what distinguishes t-linear combinations from linear combinations. To use terms from signal processing, the t-linear combination uses each coefficient tube to represent a filter. For our tactic to succeed, the structure imposed by the filter must capture the patterns in the data, and we will argue below that the block circulant matrices used here are suitable to represent data sets generated from imaging.

This shifting is not necessarily an advantage: we could construct a pathological case by creating two clusters, one consisting of shifted copies of the other’s elements. These would be distinguishable by subspace clustering, but not submodule clustering. Even though these pathological examples exist in theory, we believe the shifting of image columns that we allow is particularly suited to images, since many causes of within-cluster variation—for instance, moving subjects or camera panning—can be appropriately represented by shifts.

Modeling image collections with submodules is further warranted by its success in previous applications such as video restoration from missing pixels [26] and face recognition [7]. This prior work shows that the low-dimensional-submodule assumption can provide a useful framework to regulate model complexity while accurately capturing natural imaging data. What’s more, the variety of potential models phrased in terms of submodules is not limited to just shifting with circular boundary conditions. Ongoing work by our group adapts the t-product to use a large class of operations other than circular convolution, some of which are also well suited to imaging tasks. For example, one concrete benefit would be the potential to use convolutions with reflected boundary conditions, rather than cyclic; this has been shown to improve image de-blurring algorithms [17]. The methods presented in this paper will be easily extended to use those new adaptations.
4 Sparse Submodule Clustering

Our algorithm and the development below is based on the following algebraic assumption. We assume that the data, viewed as a list of oriented matrices, comes from a union of disjoint free submodules. The task is to find these submodules and group the data into their respective clusters.

4.1 The Algorithm

Let $\mathbf{Y}$ denote the data tensor of size $H \times N \times D$ obtained by arranging the $H \times D$ images (data points) laterally as oriented matrices $\tilde{Y}_n$, $n = 1, 2, \ldots, N$. Then the algorithm is based on the following principle. If we seek a sparse t-linear representation of an oriented matrix (image as a data point), then ideally only the matrices from the same submodule will contribute. In other words, samples from a submodule will provide an efficient generating set from the submodule. The idea is shown in Figure 4. This allows us build an “affinity tensor” $\mathbf{W}$ by solving for the following optimization problem.

$$
\mathbf{W} = \arg \min_{C} \| \mathbf{C} \|_{F1} + \lambda_h \| \mathbf{C} \|_{FF1} + \lambda_y \| \mathbf{Y} - \mathbf{Y} * \mathbf{C} \|_F^2
$$

subject to $C_{iik} = 0, i, k = 1, 2, \ldots, N$ (1)

By $\| \mathbf{A} \|_{F1}$, we mean $\sum_{i,j} \| A(i,j,:) \|_F$, a group LASSO type norm equal to the sum of the Frobenius norms (F-norms) of the tubes. $|A|_{FF1}$ denotes $\sum_i \| \mathbf{A}(i,:, :) \|_F$, a group LASSO norm equal to the sum of the F-norms of the horizontal slices (rows). Norms like these have been used to select for relevant groups of factors in regression problems [25], and they tend to force the terms inside the F-norms to survive as a group if the corresponding multiplicands contribute enough to the model or to be driven to zero as a group otherwise. Here, we hope to select relevant samples for the reconstruction of each cluster.

For affine submodules we add the constraint $\sum_i W(i,j,:) = [1, 0, \cdots, 0]$ (these are tubes, so the zeros go back into the page). Because the latter term is the identity element under convolution, this constraint gives the following property (proven below). Supposing the self-representation of a data point uses data from a submodule that has been shifted by addition of some $\tilde{M}$, we know $\mathbf{N}^2$ will be used exactly once in the representation. Submodules will then have a good chance of a successful self-representation for each datum, provided they remain unmixed. Specifically,

$$
\sum_i (\tilde{Y}_i + \tilde{M}) * W_{ij} = \sum_i \tilde{Y}_i * W_{ij} + \sum_i \tilde{M} * W_{ij} = \sum_i \tilde{Y}_i * W_{ij} + \tilde{M}
$$

where for compactness, $W_{ij} = W(i,j,:)$.

Because of convolution-multiplication properties, the computations can be carried out most efficiently in the Fourier domain. Viewed in Fourier space, the equivalent task is to solve for

$$
\widehat{W} = \arg \min_{\mathbf{C}} \| \mathbf{C} \|_{F1} + \lambda_h \| \mathbf{C} \|_{FF1} + \lambda_y \| \mathbf{Y} - \mathbf{Y} * \mathbf{C} \|_F^2
$$

subject to $\hat{c}_{ijk} = 0, i, k = 1, 2, \ldots, N$ (2)

Here, $\mathbf{A} \odot \mathbf{B}$ is the array resulting from face-wise matrix multiplication (see Section 2.2 notation). Then we build the affinity matrix $\mathbf{M}$ by taking tube-wise Frobenius norms of the affinity tensor and symmetricizing:

$$
\mathbf{M}(i,j) = \| \mathbf{C}(j,i,:) \|_F + \| \mathbf{C}(i,j,:) \|_F
$$

Finally, we run spectral clustering on the matrix, inserting it at step two of the procedure on page two of [1]. We chose to use the same clustering method used in [6]. It was proposed in [1] and the version we use was implemented by the authors of [22].

---

[1] For our purposes, an affine submodule is a submodule translated away from the origin.
4.2 Guarantees on performance

We now develop three results to give conditions under which the sparse representation of a point will use only others from its own submodule. These results are analogous to the results on Sparse Subspace Clustering in [6], which indicates the richness of proposed framework in its amenability to borrow intuition from the traditional vector space setting.

The first theorem requires a strict condition before it can apply: the submodules must be linearly independent. The second and third establish conditions under which submodules may be merely disjoint. Note that linear independence is a special case of disjointness, as the union is contained within the sum. The peril with disjointness is that a t-linear combination of points from a pair of submodules \( S_1 \) and \( S_2 \) may no longer lie within \( S_1 \cup S_2 \), so disjointness may allow elements of \( S_3 \) to be manufactured out of elements from \( S_1 \) and \( S_2 \).

For the case where \( S_i \cap \sum_{j \neq 1} S_j = 0 \), i.e. independent submodules, the following result holds. The symbol \( \sum \) indicates a sum of submodules as in Definition 2.2.

**Theorem 4.1.** Consider oriented data matrices \( \{ \mathbf{Y}_n \}_{n=1}^N \) in \( \mathbb{K}^H_D \). Suppose the data are within submodules \( \{ S_j \}_{j=1}^J \) with dimensions \( \{ d_j \}_{j=1}^J \). Given \( \mathbf{Y} \in S_i \subset \mathbb{K}^H_D \), denote by \( \mathbf{y}_i \) a tensor in \( \mathbb{K}^{H \times (N_i-1)}_D \) whose lateral slices contain all data from \( S_i \) except \( \mathbf{y} \) itself. Denote by \( \mathbf{y}_{-i} \) a tensor in \( \mathbb{K}^{H \times N-N_i}_D \) containing the data from other submodules. (Here, \( N_i \) is the number of data points in \( S_i \)). If the submodules are independent and

\[
\begin{bmatrix}
\mathbf{C}^+ \\
\mathbf{C}^-
\end{bmatrix} = \arg\min_{\mathbf{C}^+ \times \mathbf{C}^- \in \mathbb{K}^{H \times (N_i-1)}_D} \\| \begin{bmatrix}
\mathbf{C}^+ \\
\mathbf{C}^-
\end{bmatrix} \|_{F^1} \text{ s.t. } \sum_{j \neq 1} \mathbf{y}_j * \mathbf{C}^- = 0.
\]

Note that \( \mathbf{Y} = \mathbf{y}_i * \mathbf{C}^+ + \mathbf{y}_{-i} * \mathbf{C}^- \) by properties of the t-product. In other words, conformable block partitioning works the same way with the t-product that it does for matrices.

For cases where submodules are disjoint, but not linearly independent, we have the following result.

**Theorem 4.2.** Suppose the data are set up according to the hypotheses of the preceding theorem.

We define a pair of auxiliary quantities: given \( \mathbf{X} \),

\[
\mathbf{A} = \arg\min_{\mathbf{A}} \| \mathbf{A} \|_{F^1} \text{ s.t. } \mathbf{y}_i * \mathbf{A} = \mathbf{X} \text{ and } \mathbf{A}_{-i} = \arg\min_{\mathbf{A}} \| \mathbf{A} \|_{F^1} \text{ s.t. } \mathbf{y}_{-i} * \mathbf{A} = \mathbf{X}.
\]

If the submodules are disjoint and \( \forall \mathbf{X}, \| \mathbf{A} \|_{F^1} < \| \mathbf{A}_{-i} \|_{F^1} \), then the minimum argument

\[
\begin{bmatrix}
\mathbf{C}^+_i \\
\mathbf{C}^-_i
\end{bmatrix} = \arg\min_{\mathbf{C}^+_i \times \mathbf{C}^-_i \in \mathbb{K}^{H \times (N_i-1)}_D} \| \begin{bmatrix}
\mathbf{C}^+_i \\
\mathbf{C}^-_i
\end{bmatrix} \|_{F^1} \text{ s.t. } \mathbf{X} = [\mathbf{y}_i \mathbf{y}_{-i}] \begin{bmatrix}
\mathbf{C}^+_i \\
\mathbf{C}^-_i
\end{bmatrix} \text{ has a zero block: } \mathbf{C}^-_i = 0.
\]

Thus, only elements of \( S_i \) are used.

The performance guarantees are derived in terms of angles between 2-D data and coherency between submodules. We introduce two definitions to relay these notions in our algebraic setting.

Let \( \theta \in \mathbb{R}^{1 \times 1 \times D} \) represent a tubal angle as defined in [11], i.e. \( \cos(\theta(\mathbf{A}, \mathbf{B})) = \frac{\mathbf{A}^T \mathbf{B} + \mathbf{B}^T \mathbf{A}}{2 \| \mathbf{A} \|_F \| \mathbf{B} \|_F} \) for \( \mathbf{A} \) and \( \mathbf{B} \) oriented matrices in \( \mathbb{K}^H_D \). Cosine here acts on each entry of the tube individually.

**Definition 4.1.** For submodules \( \{ S_j \}_{j=1}^J \), define the tubal-coherence of \( S_i \) and \( S_j \) to be

\[
c_{ij} = \max_{\mathbf{V}_i \in S_i, \mathbf{V}_j \in S_j} \| \cos(\theta(\mathbf{V}_i, \mathbf{V}_j)) \|_F. \quad (4)
\]

We now have the following lemma which describes an important linear algebraic property (whose proof can be found in the supplementary material).

**Lemma 4.1.** Suppose \( \mathbf{y}_i \in \mathbb{K}^{H \times d_i}_D \) has lateral slices drawn from a submodule of (submodular) dimension \( d_i \) and that none of the Fourier-domain frontal slices of \( \mathbf{y}_i \) have any zero singular values. Then, the lateral slices of \( \mathbf{y}_i \) form a generating set for the submodule.
We can now state the main result on condition for exact submodule clustering.

**Theorem 4.3.** Suppose the data are set up as in the preceding theorems. Let \( W_i \) denote the set of all size \( H \times d_i \times D \) full-rank sub-tensors of \( Y_i \). In other words, every element of \( W_i \) is a tensor in \( \mathbb{R}^{H 	imes d_i \times D} \) whose lateral slices are drawn from \( Y_i \), and an element of \( W_i \), when viewed in the Fourier domain, has frontal faces with full rank.

For any \( \Lambda \in S_i \cap \left[ \sum_{j=1}^{k} S_j \right] \), suppose the condition

\[
\sqrt{d_i} \cdot \sqrt{H} \cdot \max_{i \not= j} |c_{ij}| \cdot \hat{\sigma}_{\max}(Y_j) < \max_{Y_i \in W_i} \{ \hat{\sigma}_{\min}(Y_i) \}
\]

holds, where \( \hat{\sigma}_{\min} \) returns the minimum singular value (a single real number) from the Fourier-domain frontal faces of a tensor and \( \hat{\sigma}_{\max} \) returns the maximum singular value (both will be nonzero due to the full-rank assumption). Then Theorem 4.2 will apply, or in other words \( \| \Lambda - \|_F < \| A - \|_F \),

5 Numerical Results

**Synthetic data -** In order to assess speed and reliability on modestly sized datasets, synthetic data were generated lying along multiple submodules. One set of test parameters was chosen to mimic, in size, a subset of the MNIST handwritten digit dataset. Each synthetic image is 28 by 28, with images distributed along 4 clusters. Total dataset size was varied from 40 to 200 “images.” The other set of test parameters was chosen to be larger, on the scale of a face database. Each synthetic image is 200 by 200, with images distributed along 3 clusters. Total dataset size was varied from 30 to 150 “images.” Runtime in seconds for both tests is displayed in Table 5. In our tests of SSmC, a visual cluster was made for each image.

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th>60.4360</th>
<th>290.1172</th>
<th>731.5302</th>
<th>539.5757</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face</td>
<td>74.1688</td>
<td>139.7167</td>
<td>231.1336</td>
<td>675.3414</td>
<td>2868.800</td>
</tr>
</tbody>
</table>

Figure 5: Run times for the SSmC algorithm implemented using Alternating Direction Method of Multipliers (ADMM) [3].

**Real data -** All of the tests in this section use an implementation of SSC contained as a special case of our implementation of SSmC. Our implementation lacks one portion of the SSC algorithm as presented in [6]; we do not make a provision for sparse outlying entries, which [6] does by solving this problem.

\[
W = \arg \min_C \| C \|_{F,1} + \lambda_h \| C \|_{F,1} + \lambda_z \| E \|_1 + \lambda_y \| Y - YC + E \|_F^2
\]

\[
s.t. \ C_{ii} = 0, \ i, k = 1, 2, ..., N
\]

The difference of note is the matrix \( E \), penalized with an \( \ell_1 \) norm to promote sparsity. To be clear, we could incorporate this extra term into SSmC and SSC, but this more detailed comparison is left for future work.

The t-product has performed well in the past as a tool for face recognition [7]. It is also known that images of the same object under various lighting conditions approximately form a low-dimensional subspace [2, 18]. So in the following we test the performance of the method on clustering images from various Data Bases (DB).

(1) **Weizmann Face Data Base** - http://www.wisdom.weizmann.ac.il/ vision/FaceBase/. In this experiment the aim is to group together faces regardless of lighting condition (same pose). Setting \( \lambda_h = 0 \), we selected \( \lambda_z \) with a training set of four faces at four lighting conditions, monitoring quality via a heuristic. Using those values, a test set of 28 faces at four illuminations was successfully grouped using SSmC and SSC into twenty-eight segments by SSmC, each containing only one person (no corresponding figure).

In another series of tests with no training stage, we find that SSmC displays robustness to the choice of \( \lambda_g \). This time, 36 images were used: four people, each at nine lighting conditions. SSC succeeded only in a narrow range with \( \lambda_g \) between \( 10^{-6} \) and \( 10^{-7} \), while SSmC succeeds with \( \lambda_g \) between \( 10^{-4} \) and \( 10^{-8} \). While this may not be a useful end in itself, we take it as a sign that a union of
submodules is a reasonable model for a database of this form and that the added complexity of the t-product may be warranted. Furthermore, SSmC can withstand a level of additive noise that foils SSC on the 36 images in the preceding paragraph. Pixel values ranged from zero to 255, and isotropic Gaussian noise of standard deviation 20 was added. SSmC succeeded for $\lambda_g$ between $10^{-7.3}$ and $10^{-8}$. All trials were run with SSC using identical parameters and noise instantiations, and SSC was unable to cluster the faces.

![Figure 6](image1.png)

Figure 6: Clustering error performance on original Yale DB.

8 subjects. **No other preprocessing in terms of centering, rotating etc. was performed.** Figure 6 illustrates the clustering error for various values of $\lambda_g$ for SSmC Vs SSC. Note that at optimal $\lambda_g$ the clustering error for SSmC is 0.09 whereas clustering error for SSC is 0.18.

We next performed experiments with Extended Yale Data Base. For this experiment we again reduced the size of the images from $480 \times 640$ to $120 \times 111$ by first downsampling by a factor of 4 along each axis and then cropping the columns (in order to approximately focus on the faces) by only keeping the indices from $30 : 140$. We normalize the pixel intensity to lie in $[0, 1]$. We then took subject numbers $11, 12, 15, 16, 17$ and randomly picked 25 images per person from the $65 \times 8 = 520$ images taken under various poses and different illuminations. Figure 7 shows the clustering performance of SSmC vs SSC averaged over 20 trials. The SSmC was able to achieve perfect clustering while SSC was not. We believe that this is due to the shift invariant nature of the SSmC method, which requires less preprocessing. This is further borne out by the following experiment on the MNIST handwritten character clustering problem.

![Figure 7](image2.png)

Figure 7: Clustering error performance on Extended Yale DB.

(2) **Yale Face Data Base** - We tested the approach on two Yale Data Bases, original (http://vision.ucsd.edu/datasets/yale_face_dataset_original/yalefaces.zip) and extended (http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html). The original data base contains contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. To reduce the processing time, we downsampled images from the original size of $240 \times 320$ to $81 \times 107$ and we picked the images corresponding to the first faces. We then took subject numbers $11, 15, 16, 17$ and randomly picked 25 images per person from the $65 \times 8 = 520$ images taken under various poses and different illuminations. Figure 7 shows the clustering performance of SSmC vs SSC averaged over 20 trials. The SSmC was able to achieve perfect clustering while SSC was not. We believe that this is due to the shift invariant nature of the SSmC method, which requires less preprocessing. This is further borne out by the following experiment on the MNIST handwritten character clustering problem.

![Figure 8](image3.png)

Figure 8: Clustering error performance on MNIST DB.

(3) **MNIST handwritten Data Base** (http://yann.lecun.com/exdb/mnist/) - In this experiment we seek to cluster digits 2, 4, 8 by randomly taking 20 instances for each digit from the labeled data. The clustering error performance curve, averaged over 25 such random trials, is shown in Figure 8. It turns out that for this data set SSC remains competitive to SSmC, with SSmC being slightly better at optimal $\lambda_g$.

In order to check the shift invariance property of SSmC we randomly shifted the digits horizontally, uniformly with respect to the center by 6 pixels either side. In this case for the best value of $\lambda_g$ SSmC exhibited a clustering error of 0.2 while clustering error for SSC remained at 0.5 for all values $\lambda_g$. This implies that **SSmC is more robust to processing data without needing to preprocess for centering and alignment**.

### 6 Conclusions and Future work

We presented a novel method to cluster 2-D data while preserving the multi-way aspect of the data. Initial results show robustness compared to existing approaches. In future we plan to carry out experiments with 3-D data sets, such as the American Sign Language Lexicon Video Dataset (ASLLVD). On the computational front, the current implementation of SSmC does not use multiple processors, and it is not suited to large datasets, as it takes minutes to run on the test examples. However, we believe the algorithm can be effectively parallelized. During this project, the paper [9] proposed a fast alternative to SSC, and we believe their work could be extended to quickly perform submodule clustering as well.
References


7 Appendix

7.1 Proof of Theorem 4.1

Proof. If $\overline{C}_{-i} \neq 0$, then $\overline{Y} = y_i * \overline{C}_{-i} + y_{-i} * \overline{C}_{-i} \Rightarrow y_{-i} * \overline{C}_{-i} = y_i * \overline{C}_{-i} + \overline{Y} \in S_i$, since the module generated by the lateral slices of $Y_{-i}$ is in $\oplus_{i \neq j} S_j$, it must be the case that $Y_{-i} * \overline{C}_{-i} \in \oplus_{i \neq j} S_j$, also, and so $Y_{-i} * \overline{C}_{-i}$ is in the intersection and must be zero. Thus $\overline{Y} = y_i * \overline{C}_{-i} + 0$, and $\overline{C}_{-i}$ may be replaced by zero. This lowers the F1 norm, contradicting the assumption that $\left[ \begin{array}{c} \overline{C}_{-i} \\ \overline{C}_{-i} \end{array} \right]$ is optimal.

7.2 Proof of Theorem 4.2

Proof. Suppose $\overline{C}_{-i} \neq 0$. Using the hypotheses of the theorem, we will derive a contradiction by constructing better coefficients. Let $\overline{X} = Y_{-i} * \overline{C}_{-i} = \overline{Y} - y_i * \overline{C}_{-i}$. Then $y_i * \overline{A}_i = \overline{X} = y_{-i} * \overline{A}_{-i} = \overline{Y} - y_i * \overline{C}_{-i}$, so

$$\begin{align*}
\overline{Y} &= \begin{bmatrix} y_i & y_{-i} \end{bmatrix} \begin{bmatrix} \overline{A}_i + \overline{C}_{-i} \end{bmatrix} \\
&= \begin{bmatrix} y_i & y_{-i} \end{bmatrix} \begin{bmatrix} \overline{C}_{-i} \overline{A}_{-i} \end{bmatrix}
\end{align*}$$

(6)

Also, $y_i * \overline{A}_{-i} = y_{-i} * \overline{C}_{-i}$, so $||\overline{A}_{-i}||_{F1} < ||\overline{C}_{-i}||_{F1}$ by optimality of $\overline{A}_{-i}$. Using the fact that

$$\begin{align*}
||\begin{bmatrix} \overline{V}_1 \\ \overline{V}_2 \end{bmatrix}||_{F1} &= ||\overline{V}_1||_{F1} + ||\overline{V}_2||_{F1}, \\
||\begin{bmatrix} \overline{A}_i + \overline{C}_{-i} \\ 0 \end{bmatrix}||_{F1} &\leq ||\overline{A}_i||_{F1} + ||\overline{C}_{-i}||_{F1}, \\
&< ||\overline{A}_{-i}||_{F1} + ||\overline{C}_{-i}||_{F1} \\
&\leq ||\overline{C}_{-i}||_{F1} + ||\overline{C}_{-i}||_{F1} \\
&= ||\begin{bmatrix} \overline{C}_{-i} \\ \overline{C}_{-i} \end{bmatrix}||_{F1}
\end{align*}$$

(7)

Since $\left[ \begin{array}{c} \overline{C}_{-i} \\ \overline{C}_{-i} \end{array} \right]$ was supposed to be optimal, we have reached a contradiction.

7.3 Proof of Theorem 4.3

Proof. Theorem 2 applies if $||\overline{A}_i||_{F1} < ||\overline{A}_{-i}||_{F1}$ (these are defined in relation to $\overline{X}$ as before). This proof will derive $\beta_i$ and $\beta_{-i}$ such that $||\overline{A}_i||_{F1} \leq \beta_i < \beta_{-i} \leq ||\overline{A}_i||_{F1}$ with the middle statement $\beta_i < \beta_{-i}$ holding exactly when the condition listed in the hypotheses holds.

To find $\beta_i$, note that since $\overline{Y}_i$ is an element of $W_i$, its columns are in $S_i$, so we can invoke Lemma 4.1 to claim that $\overline{X} \in S_i$ implies $\overline{X} = y_i * \overline{A}_i$ for some $\overline{A}_i$. This new variable $\overline{A}_i$ mimics $\overline{A}_i$ in terms of its role, but it may be suboptimal in terms of F1 norm. There may be many eligible candidates, and we choose among them as follows.

In the frequency domain, we can go to the dth face and write

$$\begin{align*}
\text{fft}(\overline{X}) &= \hat{x} \\
\text{fft}(\overline{Y}_i) &= \hat{Y} \\
\text{fft}(\overline{A}_i) &= \hat{a}
\end{align*}$$

(8) (9) (10)
Then, the constraint $\overline{x} = \overline{y}_i \ast \overline{A}_i$, can be written as $D$ independent constraints $\hat{x}(d) = \hat{Y}(d) \hat{a}(d)$, and we can make an independent choice of $\hat{a}(d)$ for each $d$. Specifically, we want $||\hat{a}(d)||_F$ to be minimal for each $d$. 

Then, by well-known properties of the pseudoinverse, $\hat{Y}(d) \ast \hat{x}(d) = \hat{a}(d)$, and if $\hat{U}(d) \hat{S}(d) \hat{V}(d)^H$ is a “compact” SVD of $\hat{Y}(d)$ where $\hat{S}$ is square and has no zero singular values, then $\hat{Y}(d) = \hat{V}(d) (\hat{S}(d)^{-1} \hat{U}(d)^H)$. The operator norm of a matrix pseudoinverse is the reciprocal of the smallest singular value, so $||\hat{a}(d)||_F \leq \frac{1}{\sigma_{min}} ||\hat{x}(d)||_F$ if $\sigma_{min}$ is the last singular value at the $d$th face. Taking $\sigma_{min}$ to be the smallest of these across all faces,

$$||\overline{A}_i||_F^2 = ||F \cdot (\overline{A}_i)||^2_F = \sum_d ||F \cdot (\overline{A}_i(d))||^2_F$$

$$\leq \sum_d \left[ \frac{1}{\sigma_{min}} ||F \cdot (\overline{x})||^2_F \right]^2$$

$$\leq \left[ \frac{1}{\sigma_{min}} ||F \cdot (\overline{x})||^2_F \right]^2 \quad (11)$$

Finally, we may pad the $d_i \times 1 \times D$ array $\overline{A}_i$ with zeros to form an $N_i \times 1 \times D$ array, $\overline{A}_i^{+0}$, placing the zeroes to hit elements of $\overline{Y}_i$ that aren’t present in the subtensor $\overline{Y}_i$. Then, $\overline{Y}_i \ast \overline{A}_i = \overline{Y}_i \ast \overline{A}_i^{+0}$, and we observe that $||\overline{A}_i^{+0}||_F = ||\overline{A}_i||_F$ and $\overline{Y}_i \ast \overline{A}_i^{+0} = \overline{x} = \overline{Y}_i \ast \overline{A}_i$. By the F1-optimality of $\overline{A}_i$, $||\overline{A}_i||_F \leq ||\overline{A}_i^{+0}||_F$, and in the following we also use relationship between F- and F1-norms: $||\overline{A}_i||_F = \sqrt{\sigma_{max}(\overline{Y}_i)}$. This can be shown by taking the F-norms of the tubes first and then applying the well known inequality $||v||_1 \leq \sqrt{\sigma_{max}(v)}||v||_F$. Using these, we can find a useful bound for one of the coefficient norms.

$$||\overline{A}_i||_F \leq ||\overline{A}_i^{+0}||_F = ||\overline{A}_i||_F \leq \sqrt{\sigma_{min}(\overline{Y}_i)} = \sigma_{min}(\overline{Y}_i) = \beta_i$$

(12)

To push the bound down, we may take $\beta_i = \min_{\overline{y}_i \in \overline{W}_i} \beta(\overline{y}_i)$, or equivalently $\sigma_{min} = \max_{\overline{y}_i \in \overline{W}_i} \sigma_{min}(\overline{y}_i)$.

To bound $||\overline{A}_i||_F$ below, we make use of lemmas 7.1 (in the first inequality) and 7.2 (in the fourth line. The largest singular value fourier coefficient of $\overline{y}_i - \overline{A}_i$ is labeled $\sigma_1^{\max}(\overline{y}_i - \overline{A}_i)$.

$$||\overline{x}||_F^2 = \sqrt{\sigma_{max}(\overline{Y}_i)} ||\overline{x}||_F$$

$$= \sqrt{\sigma_{max}(\overline{Y}_i)} ||\overline{Y}_i \ast \overline{A}_i||_F = \sqrt{\sigma_{max}(\overline{Y}_i)} \max_{i \neq j} \max_{\overline{y}_i \in \overline{W}_i} \sigma_1^{\max}(\overline{y}_i - \overline{A}_i) \cdot ||\overline{A}_i||_F \cdot ||\overline{x}||_F$$

The inequality

$$||\overline{x}||_F \leq \sqrt{\sigma_{max}(\overline{Y}_i)} \max_{i \neq j} \max_{\overline{y}_i \in \overline{W}_i} \sigma_1^{\max}(\overline{y}_i - \overline{A}_i) \cdot ||\overline{A}_i||_F \cdot ||\overline{x}||_F$$

can be rewritten as

$$||\overline{x}||_F/\sqrt{\sigma_{max}(\overline{Y}_i)} \max_{i \neq j} \max_{\overline{y}_i \in \overline{W}_i} \sigma_1^{\max}(\overline{y}_i - \overline{A}_i) \leq ||\overline{A}_i||_F$$

so if $\beta_i = ||\overline{x}||_F/\sqrt{\sigma_{max}(\overline{Y}_i)} \max_{i \neq j} \max_{\overline{y}_i \in \overline{W}_i} \sigma_1^{\max}(\overline{y}_i - \overline{A}_i)$, then $||\overline{A}_i||_F \leq \beta_i < \beta_i \leq \frac{1}{\sigma_{min}} ||\hat{x}(d)||_F$ exactly when the condition in the hypotheses holds. \qed

11
7.4 Lemmas Used in Proof of Theorem 4.3

We prove the three lemmas here, stating those not present earlier.

Lemma 7.1. For tensors of given size, say elements of $\mathbb{R}^{H \times W}$, we can bound $||A||^2_F$ by a scalar multiple of $||A^T \ast A||_F$: $||A||^2_F \leq \sqrt{HW}||A^T \ast A||_F$.

Proof.

$||A||^2_F = \sum_{i=1}^{D} \text{tr}(A(i)^T A(i))$

$= \text{tr}(\text{unfold}(A)^T \text{unfold}(A))$

$= \text{tr}(bcirc(A^T) \text{unfold}(A) [\text{top block}])$

$= \text{tr}((A^T \ast A)^{(1)})$

$= \text{tr}(X^{(1)})$

where $X = A^T \ast A$.

Now, $\text{tr}(X^{(1)}) \leq \sum_{i,j} |X_{i,j}^{(1)}| \leq \sqrt{HW}||X^{(1)}||_F \leq \sqrt{HW}||X||_F$. The second inequality is a relationship between vector 1-norms and 2-norms. It follows that

$||A||^2_F \leq \sqrt{HW}||A^T \ast A||_F$.

\[ \square \]

Lemma 7.2. The maximum singular value from taking facewise SVD’s in the Fourier domain acts like an induced matrix 2-norm. Formally, $||Y_{-i} \ast \overrightarrow{A}_{-i}||_F \leq [\hat{\sigma}_1^{\text{max}}] \cdot ||\overrightarrow{A}_{-i}||_F$.

Proof.

$||Y_{-i} \ast \overrightarrow{A}_{-i}||^2_F = \sum_d ||\text{fft}(Y_{-i})^{(d)} \text{fft}(\overrightarrow{A}_{-i})^{(d)}||^2_F$

$\leq \sum_d [\hat{\sigma}_1^{(d)}]^2 ||\text{fft}(\overrightarrow{A}_{-i})^{(d)}||^2_F$

$\leq [\hat{\sigma}_1^{\text{max}}]^2 \sum_d ||\text{fft}(\overrightarrow{A}_{-i})^{(d)}||^2_F$

$= [\hat{\sigma}_1^{\text{max}}]^2 \cdot ||\overrightarrow{A}_{-i}||^2_F$

\[ \square \]

Proof of Lemma 4.1

Proof. We first restate the conclusion: if the $H$ by $d_i$ by $D$ tensor $Y$ has lateral slices drawn from a free submodule $S_i$ of dimension $d_i$, then all the singular values of its Fourier-domain frontal slices are nonzero, then for any element $X$ of $S_i$, we must show the existence of a 1 by $d_i$ by $D$ coefficient tensor $C$ so that $X$ is the t-product of $Y$ and $C$. Here $S_i$ consists of $H$ by $W$ by $D$ tensors, with $H > d_i$.

We prove the lemma by contradiction. We first show that the module $S_i$ has dimension $D \times d_i$ when considered as a vector space over the complex numbers. Then, we suppose that some $X$ violates the claim, and this has the consequence of increasing the dimension.

To establish the dimension of $S_i$, we consider linearly independent generating elements $\{ \text{fft}(Y_j) \}$ of $S_i$, concatenated into a tensor $\mathcal{V}$. Such a generating set exists by the definition of a free module. The submodule may be decomposed into a direct sum of $D$ vector spaces each of whose Fourier-domain representations has elements with zeroes at every frontal slice but one. Here, $\oplus$ is a traditional sum of vector spaces, obtained by concatenating generating sets.
\[
S_i = \sum_{k=1}^{D} [S_i \cap \{ T \in \mathbb{C}^{H \times W \times D} | j \neq k \Rightarrow \text{fft}(T(\cdot, \cdot, j)) = 0 \}]
\]

Because the submodule \( S_i \) is generated via the T-product, linear combinations of \( V \)'s columns (lateral slices) can be represented simply in the Fourier domain as face-by-face matrix multiplication with the Fourier representation of \( V \) interacting with coefficient vectors at each face.

\[
\mathbf{\lambda} \in S_i \Rightarrow \exists \mathbf{c} s.t. \text{fft}(\mathbf{\lambda}^{(n)}) = \text{fft}(V^{(n)}) \cdot \text{fft}(\mathbf{c}^{(n)})
\]

At any given face, the \( N \) by \( d_i \) matrix containing Fourier coefficients of \( V \) has no zero singular values, because otherwise \( V \) would have a nontrivial null space and \( \{ \mathbf{\lambda} \} \) would not be linearly independent. Thus, the columns of that matrix are linearly independent and the subspace at that face contributes \( d_i \) degrees of freedom to the direct sum. With a total of \( D \) faces, the result is a complex vector space of dimension \( D \times d_i \).

Suppose, for the sake of contradiction, that some \( \mathbf{\lambda} \) violates the original claim. Then, for at least one face of the Fourier representation, there is no coefficient tensor \( \mathbf{c} \) for which \( \text{fft}(\mathbf{\lambda}^{(n)}) = \text{fft}(V^{(n)}) \cdot \text{fft}(\mathbf{c}^{(n)}) \). At that face, the augmented matrix \( \left[ \text{fft}(\mathbf{\lambda}^{(n)}) | \text{fft}(V^{(n)}) \right] \) must have linearly independent columns. If it did not, there would be coefficients so that \( \text{fft}(\mathbf{\lambda}^{(n)}) \cdot \text{fft}(c_x^{(n)}) = \text{fft}(V^{(n)}) \cdot \text{fft}(\mathbf{c}^{(n)}) \), and we could divide by \( -\text{fft}(c_x^{(n)}) \) to produce coefficients such that \( \text{fft}(\mathbf{\lambda}^{(n)}) = \text{fft}(V^{(n)}) \cdot \text{fft}(\mathbf{c}^{(n)}) \). This is impossible, so we deduce that either no \( \mathbf{\lambda} \) violates the original claim or \( -\text{fft}(c_x^{(n)}) \) is zero (preventing division).

In the latter case, the fact that \( \text{fft}(V^{(n)}) \) is full rank implies the rest of the coefficients are 0, so the columns of the augmented matrix are indeed linearly independent. Thus, at that face, the direct summand of complex dimension \( d_i \), \( \left[ S_i \cap \{ T \in \mathbb{C}^{H \times W \times D} | j \neq k \Rightarrow T(\cdot, \cdot, j) = 0 \} \right] \) contains the set

\[
\{ T \in \mathbb{C}^{H \times W \times D} | T(\cdot, \cdot, n) = \text{fft}(\mathbf{\lambda}^{(n)}) \cdot \text{fft}(\mathbf{c}_x^{(n)}) + \text{fft}(V^{(n)}) \cdot \text{fft}(\mathbf{c}^{(n)}) \text{ and } j \neq k \Rightarrow T(\cdot, \cdot, j) = 0 \}
\]

which has (complex) dimension \( d_i + 1 \). This is a contradiction. We conclude that no \( \mathbf{\lambda} \) may violate the original claim. \( \square \)