COMPUTING PERSISTENT FEATURES IN BIG DATA: A DISTRIBUTED DIMENSION REDUCTION APPROACH

Adam C. Wilkerson *Harish Chintakunta* *Hamid Krim*

North Carolina State University

ABSTRACT

Persistent homology has become one of the most popular tools used in topological data analysis for analyzing big data sets. In an effort to minimize the computational complexity of finding the persistent homology of a data set, we develop a simplicial collapse algorithm called the selective collapse. This algorithm works by representing the previously developed strong collapse as a forest and uses that forest data to improve the speed of both the strong collapse and of persistent homology. Finally, we demonstrate the savings in computational complexity using geometric random graphs.

Index Terms— Simplicial complex, persistent homology, simplicial collapse, topological data analysis, strong collapse

1. INTRODUCTION

Topological data analysis, particularly the tools of homology and persistent homology [3, 9] from algebraic topology, has found a wide variety of applications as a data analysis technique, including sensor network coverage [4, 6, 11, 21, 25], gene expression in cancer data [19], dynamics in biological systems [10], astronomical data [23], and social networks [17, 24]. The fundamental idea behind these tools is that a data set has a shape, which can inform a user on some of its global properties. Homology gives an algebraic description of topological features of the data, which are coarse geometry-free structures such as holes or voids in the data. These features may be viewed as generalizations of connected components (which are the result of many classification algorithms) to higher orders. It is also well known that there is no meaningful clustering algorithm [14] without determining the scale at which the data should be analyzed. When such an *a priori* scale is not available, it is advisable to use hierarchical clustering algorithms [18] which give a summary over all scales.

An analogue of this generalization to higher order features, which gives a summary of homology over all scales, is persistent homology. Owing to the importance of such analysis especially for large data sets, a fair amount of attention has been paid to efficient computation of homology [2,5,22,24,25] and to persistent homology [15,16] which has been shown to have the same complexity as that of computing homology at a single scale [15]. The worst case complexity of this is on the order of matrix multiplication time of a matrix with size equal to the number of simplices in a complex.

It is widely believed that the current state of the art complexity in the number of simplices is the best possible. As a result, much research has gone into developing techniques which might reduce the number and degree of the simplices, paralleling the efforts of dimension reduction in the field of machine learning. These techniques range from computing the persistent homology of some intelligently selected subsample of the data [7] to collapsing the data in ways that preserve its topology [1,16,22,24,25]. These collapsing methods come in several forms. One branch of study focuses on computing discrete Morse functions on the simplicial complexes in the persistence computation. [16] While the collapsed complexes yielded from this method are theoretically small, finding optimal Morse functions is known to be NP-hard in general. [13] Furthermore, there is no guarantee that the local structure about the topological features will be preserved.

Another method, introduced in [24], collapses a complex with the guarantee that cycle lengths are preserved: the tightest subset of the data around any topological feature is kept intact, thus allowing feature-localization algorithms to discover the original position of the features. Moreover, the algorithm can be distributed [25], and because persistence utilizes a particular family of simplicial complexes called Rips complexes, each node only needs knowledge of its immediate neighbors in the data in order to execute the collapse. Applying this collapsing algorithm to the persistent homology of a data set is our focus. In contrast to the parallel research on applying Morse theoretic collapses to the persistence algorithm [16], we introduce a new method here that uses the strong collapse to achieve 1) a distributed algorithm which can leverage the power of massively parallel computing architectures, and 2) automatically localizes the persistent features in the data set.

This paper is organized as follows: in section 2, we introduce the basics of the mathematics underlying persistent homology and the strong collapse. In section 3, we construct the algorithm and prove that the persistent homology of the collapsed filtration is the same as that of the original data. In section 4, we implement the algorithm and demonstrate its utility to persistence over the strong collapse. Finally, we provide a conclusion and future directions in section 5.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

2. PRELIMINARIES

This section introduces some background concepts for algorithm used in this paper. These concepts include the building blocks of simplicial complexes and homology, followed by an introduction to persistence computations, and finishing at the definition of the strong collapse used to reduce the data.

2.1. Simplicial Complexes and Homology

Simplicial complexes (often referred to as complexes) are the basic building blocks of algebraic topology. A complex can be formally defined as any set of sets closed under the subset operation. Geometrically, a simplicial complex is a generalization of a graph that includes not only vertices and edges, but also higher-dimensional structures such as triangles, tetrahedra, and higher order convex hulls of vertices in an ambient space. A simplicial complex may also be viewed abstractly as a special case of an hypergraph. Each of these structures is called a simplex, and the number of vertices in a simplex determines its dimension: A vertex is a 0-simplex, an edge is a 1-simplex, and the convex hull of n+1 points will be called an n-simplex, indicating it has dimension n.

Homology is a linear algebraic tool that takes as its input a simplicial complex X and outputs a sequence of real vector spaces $\{H_0(X), H_1(X), \cdots\}$. Of specific interest are the *Betti Numbers* $\{\beta_0(X), \beta_1(X), \cdots\}$ of X, where $\beta_i(X) = \text{rank}(H_i(X))$. While $\beta_0(X)$ counts the number of connected components in X, the i^{th} Betti number counts the number of $i+1$ dimensional holes: $\beta_1(X)$ yields the number of holes in X, $\beta_2(X)$ counts the 3-dimensional voids, and higher Betti numbers count higher dimensional *cycles*. The details of these computations are involved, and so are not included here, but any introductory algebraic topology text contains a good introduction. The authors suggest Hatcher [12], which is freely available at the authors' website.

2.2. Persistent Homology

Persistent homology is a higher order equivalent of hierarchical techniques used for clustering. It takes as its input a nested sequence of simplicial complexes, and outputs a summary of topological features at all scales. In fact, persistent $H_0(X)$ gives the same result as single-linkage clustering. A very common approach is to treat the data as a point cloud in some ambient metric space. Then, for any non-negative real number ϵ, a simplicial complex (known as the Vietoris-Rips complex) X^ϵ is constructed using the points in X as vertices, such that the pairwise distance between vertices in each simplex is no more than ϵ. As in the hierarchical clustering setting, ϵ is allowed to increase from 0 to ∞, but instead of observing only $H_0(X)$ over that interval, we also observed the other homology spaces and how they change with ϵ. It immediately follows from the definition that for $\epsilon < \delta$, $X^\epsilon \subset X^\delta$. Since our data set X is made up of a finite number of points, there are a finite number of pairwise distances between points in the data, and we can order them $0 < \epsilon_1 < \epsilon_2 < \cdots < \epsilon_m$. Hence, we observe a series of inclusions, called a *filtration*

$$\cdots X^{\epsilon_{j-1}} \xrightarrow{i^{j-1}} X^{\epsilon_j} \xrightarrow{i^j} X^{\epsilon_{j+1}} \xrightarrow{i^{j+1}} \cdots$$

This, in turn, induces a series of linear maps on the homology spaces of each complex in every dimension

$$\cdots H_*\big(X^{\epsilon_{j-1}}\big) \xrightarrow{i_*^{j-1}} H_*\big(X^{\epsilon_j}\big) \xrightarrow{i_*^j} H_*\big(X^{\epsilon_{j+1}}\big) \xrightarrow{i_*^{j+1}} \cdots$$

The persistent homology of a data set takes as input the data set and outputs a persistence diagram. This persistence diagram helps to keep track of the existence of persistent features in the complex over the range of the parameter ϵ. A homology class $v \in H_*(X^{\epsilon_b})$ is said to be born at ϵ_b if it is in the cokernel of i_*^{b-1} in the persistence diagram. It is said to persist from time ϵ_j to time ϵ_{j+1} if it is not in the kernel of i_*^p. Finally, it *dies* at the minimal d for which $i_*^d \circ \cdots \circ i_*^b(v) = 0$. The diagram can visually represent these births and deaths in many ways. [3] One popular visualization is the barcode, a chart whose x-axis is the range of ϵ, and whose y-axis is discrete, having one entry for each persistent homology class. Then, we display a bar for each class v which runs horizontally from ϵ_b to ϵ_d. By observing those features which persist for the largest parameter range $\epsilon_d - \epsilon_b$, we can infer a great deal about the topology of the data.

2.3. Strong Collapse

The final technique required to describe the new algorithm presented here is the *strong collapse* developed in [24, 25]. This collapse is based on a novel concept of "relevance" of the nodes, and has a simple implementation. In general the strong collapse requires the knowledge of all the simplices in the complex, and in some cases, these are determined by the problem at hand [17, 24]. But it is usually the case we are only presented with a point cloud, and a filtration using Rips complexes is generated as described in the previous section. Computing all the simplices (cliques) in the Rips complex is, however, very expensive. Fortunately, the relevance of the nodes can be computed directly (described below) without the need to compute all the cliques [25]. The strong collapse also has the added advantage of requiring only the local information, thus facilitating parallel and distributed computations.

In the distributed strong collapse, each node broadcasts its neighbor set to each of its neighbors. Every node then compares its own neighborhood to that of each of its neighbors. If its neighbor set contains the neighborhood of one of its neighbors, it tells that neighbor to turn off, thus removing it and all of its incident simplices from the simplicial complex. It can be shown that the collapse map, and the inverse inclusion, induce isomorphisms in homology [1, 24]. Each node adjacent to a deleted node then updates its neighbor set, and the collapse iterates again and again until it converges. This collapse has the benefit of not only preserving the topology of the simplicial complex, but also preserving the shortest generator of each homology class in the original complex. For example, when computing $H_1(X)$, the shortest path around every hole is preserved. This allows hole-localization algorithms to find the tightest bounds around topological features in the original data using only the collapsed data.

3. SELECTIVE COLLAPSE

Given a persistence diagram, we can exploit the techniques outlined in the previous section to build a new structure,

$$\begin{tikzcd}
\cdots X^{\epsilon_{j-1}} \arrow[r, "i^{j-1}"] \arrow[d, bend left, "C"] & X^{\epsilon_j} \arrow[r, "i^j"] \arrow[d, bend left, "C"] & X^{\epsilon_{j+1}} \arrow[r, "i^{j+1}"] \arrow[d, bend left, "C"] & \cdots \\
\cdots \tilde{X}^{\epsilon_{j-1}} \arrow[r, dotted, "f^{j-1}"] \arrow[u, bend left, "in"] & \tilde{X}^{\epsilon_j} \arrow[r, dotted, "f^j"] \arrow[u, bend left, "in"] & \tilde{X}^{\epsilon_{j+1}} \arrow[r, dotted, "f^{j+1}"] \arrow[u, bend left, "in"] & \cdots
\end{tikzcd}$$

The dotted maps are those induced by composing $f^* = C \circ i^* \circ in$, where $C : X \to \tilde{X}$ is the strong collapse map and $in : \tilde{X} \to X$ is the inclusion. Each square in the above diagram commutes by construction, and the induced maps on homology by the vertical maps are isomorphisms, as stated in Section 2.3. Thus, if a generator exists at $H_*(X^{\epsilon_{j-1}})$ and persists to $H_*(X^{\epsilon_j})$, then its image in $H_*(\tilde{X}^{\epsilon_{j-1}})$ will persist to $H_*(\tilde{X}^{\epsilon_j})$. Likewise, any class not persisting above will not persist below. Hence, the persistent homology of the original data above is isomorphic to the persistent homology of the reduced data below.

However, there is no guarantee that vertices collapsed by $C : X^{\epsilon_i} \to \tilde{X}^{\epsilon_i}$ will be collapsed by $C : X^{\epsilon_{i+1}} \to \tilde{X}^{\epsilon_{i+1}}$, and vice-versa. Hence, these dotted maps are not necessarily inclusions, which is a prerequisite for traditional persistence. However, they are simplicial maps. Thus, the bottom row is computable if we can compute persistence for a sequence of general simplicial maps. Such a technique has been recently presented in [8].

Unfortunately, reducing the complex at each stage is time-consuming, thus necessitating a careful use of the collapse at one stage to efficiently compute the collapse at the next. Given a complex X^ϵ and the strong collapse of that complex $C : X^\epsilon \to \tilde{X}^\epsilon$, we represent the collapse with a *forest*, or collection of directed trees, denoted F^ϵ. The nodes in this forest correspond to the vertices of the simplicial complex X^ϵ, and the collapsed complex $\tilde{X}^\epsilon$ is the subcomplex induced by the root nodes of the forest. Each node in the forest is a parent to all the nodes which collapse to it.

3.1. Selective Collapse Algorithm

In constructing the algorithm, we justify each step as it is presented. For ease of notation, we also define the *ϵ-neighborhood* $N_\epsilon(v) = \{w \in X | d(v, w) \leq \epsilon\}$ for every point $v \in X$. We can assume, without loss of generality, that each inclusion in the filtration corresponds to the addition of a single edge in the uderlying graph of the complex. The algorithm begins by collapsing the first complex in the filtration X^{ϵ_1}, and proceeds by distributively modifying this forest at each stage. Every node $v \in F^{\epsilon_i}$ is either a root ($v \in R(F^{\epsilon_i})$) in the forest or a non-root ($v \in NR(F^{\epsilon_i})$). In the case that $v \in NR(F^{\epsilon_i})$, v has a parent v_p. The algorithm continues to make the assumption that the starting forest represents a strong collapse, and that one edge is then added to the parent complex X^{ϵ_i} to obtain $X^{\epsilon_{i+1}}$. Then, to transform the forest $F^{\epsilon_i} \to F^{\epsilon_{i+1}}$, we add the new edge to X^{ϵ_i} to get $X^{\epsilon_{i+1}}$, and propogate the following algorithm at each node v, starting with the endpoints of the new edge, and working our way away from the new edge:

```
Receive $w : NR(F^{\epsilon_i}) \to R(F^{\epsilon_i})$ transition notification from neighbor
if $v \in NR(F^{\epsilon_i})$ then
    if $w \in N_{v_p}$ then
        $v$ remains unchanged
    else
        $v : NR(F^{\epsilon_i}) \to R(F^{\epsilon_i})$, notifies neighbors $w \in N_v$ of change
    end if
else
    Continue. Roots do not change state.
end if
```

After the algorithm has been completed, each node which is still a non-root maintained its status by either not having its neighborhood disturbed, or by testing any new roots entering its neighborhood to see if its parent also contained the root it its own neighborhood. In so doing, every non-root is dominated in the network, and as such, the resulting forest $F^{\epsilon_{i+1}}$ is a (partial) strong collapse of $X^{\epsilon_{i+1}}$. It may not be the complete strong collapse $C : X^{\epsilon_{i+1}} \to \tilde{X}^{\epsilon_{i+1}}$, in that there still may be collapsible nodes in the root set of the resulting forest. So, we take the final step of strong collapsing the remaining subcomplex to obtain $\tilde{X}^{\epsilon_{i+1}}$. The collapse, however, operates on a subcomplex of $X^{\epsilon_{i+1}}$, and therefore must complete far fewer computations to achieve the complete collapse.

4. SIMULATION RESULTS

In this section, we demonstrate the utility of the proposed selective collapse algorithm using an illustrative example of the Rips filtration of random points in a plane. A set of points V are chosen from a uniform distribution on a unit square, and a sequence of geometric graphs $G(V, \epsilon)$ is generated using an increasing sequence of values for ϵ. A geometric graph with parameter ϵ contains an edge (v_1, v_2) whenever the distance between v_1 and v_2 is less than or equal to ϵ, and therefore, the sequence $G(V, \epsilon)$ has the property that $G(V, \epsilon_1)$ is a subgraph of $G(V, \epsilon_2)$ whenever $\epsilon_1 \leq \epsilon_2$. An example sequence

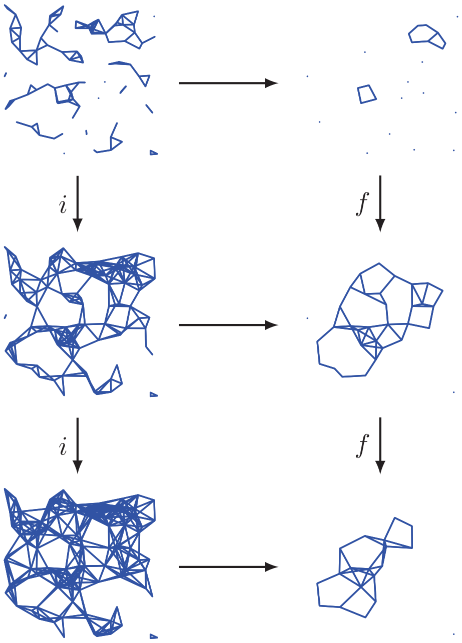

Fig. 1. The left column shows the sequence of geometric graphs obtained using increasing value for the parameter ϵ, along with the inclusion maps. The problem of interest is to compute the persistent homology of the filtration obtained using the corresponding flag complexes. The right column shows the collapsed complexes. Given the first collapsed complex, Section 3 presents an efficient algorithm to compute the next reduced complex along with the simplicial map f.

is shown in Figure 1. The filtration we consider here is the sequence of the Rips complexes of the geometric graphs.

Figure 1 shows a sequence of the graphs underlying the full and collapsed complexes in one run of the experiment. Note that the useful properties of the strong collapse that it preserves homology and shortest cycles are both clear in the examples found in figure 1. Even though the sequence of original complexes forms a filtration, note that the sequence of collapsed complexes does not. However, the simplicial map induced between the complexes is computed using the collapse forests described in Section 3.

The algorithm described in Section 3 also provides an efficient way to compute the sequence of collapsed complexes. Figure 2 demonstrates the utility of this algorithm by comparing the time required to compute strong collapse entirely for each complex without utilizing the information from the previous collapses. The figure shows the ratio of the time taken for selective collapse T_r by algorithm given in Section 3 to that taken for collapsing the entire complex T_f. The computations used to generate these ratios are, in fact, centralized. This figure shows that we see significant reductions in complexity, even in the centralized case. When the geometric random graph has 120 nodes, the algorithm runs about 15 times faster than if the entire complex is collapsed at each stage individually. The primary feature about the algorithm is that this ratio decreases as the number of data points increases, suggesting that for large data sets, the algorithm potentially provides a significant increase in performance. It should also be noted that the degree to which the collapse can be formed depends largely on the specific type of complexes. The specific purpose of these experiments is to show that a collapse obtained for a complex may be exploited to efficiently collapse a bigger complex into which it is included.

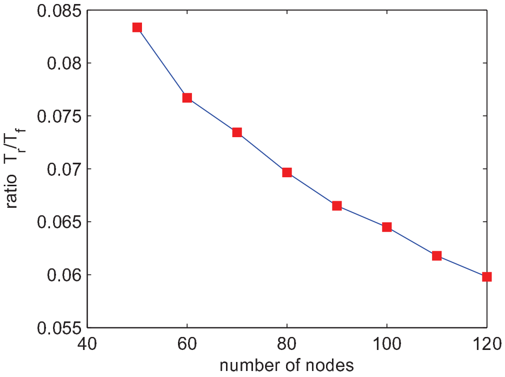

Fig. 2. Figures shows the ratio of the time taken by the algorithm presented here to that taken for collapsing the entire collapse for each step in the filtration. As seen, the ratio decreases with increasing number of nodes, a desirable feature for algorithms designed for large data sets.

5. SUMMARY AND CONCLUSION

We proposed an algorithm, called the selective collapse, which speeds up the computation of persistent homology, a tool with wide and various applications in topological data analysis. We presented the framework of collapsing the complex at each step and utilizing the persistence algorithm proposed in [8] to make the computations tractable. We represented the strong collapse with a forest, and used that information to ease the processing of persistence across collapsed complexes. We also outlined the distributed computations used to execute the selective collapse algorithm, and proved that the resulting forest corresponds to a strong collapse. Furthermore, we showed through simulations on geometric random graphs that the selective collapse algorithm is a faster method for collapsing the simplicial complexes in a persistence diagram, while maintaining all pertinent topological information within the complex. Finally, we showed that the collapse exhibits a trend towards greater efficiency with and increased number of data points, thus displaying the potential for this method in analyzing big data sets.

6. REFERENCES

[1] J. A. Barmak and E. G. Minian, “Strong homotopy types, nerves, and collapses,” *Discrete & Computational Geometry*, vol. 47, pp. 301–328, 2012.

[2] O. Busaryev, S. Cabello, C. Chen, T. K. Dey, and Y. Wang, “Annotating simplices with a homology basis and its applications,” *Algorithm Theory–SWAT 2012*, pp. 189–200, Springer, 2012.

[3] G. Carlsson, “Topology and data,” *Bulletin of the American Mathematical Society*, vol. 46, pp. 255–308, 2009.

[4] H. Chintakunta and H. Krim, “Divide and conquer: Localizing coverage holes in sensor networks,” in *Proceeding of the 2010 7th Annual IEEE Communications Society Conference Sensor Mesh and Ad Hoc Communications and Networks (SECON)*, Boston, MA, 2010, pp. 1–8.

[5] H. Chintakunta and H. Krim, ”Distributed computation of homology using harmonics,” arXiv:1306.1158 [math.AT].

[6] V. de Silva and R. Ghrist, “Coordinate-free coverage in sensor networks with controlled boundaries via homology,” *International Journal of Robotics Research*, vol. 25, pp. 1205–1222, 2006.

[7] V. de Silva and G. Carlsson, ”Topological estimation using witness complexes,” *Proceedings of the First Eurographics Conference on Point-Based Graphics*, pp. 157-66, 2004.

[8] T. K. Dey, F. Fan, and Y. Wang, ”Computing topological persistence for simplicial maps,” arXiv:1208.5018 [cs.CG].

[9] H. Edelsbrunner, and J. Harer, *Computational Topology, an Introduction*, American Mathematical Society Bookstore, 2010.

[10] S. Emrani, T. Gentimis, and H. Krim, ”Persistent homology of delay embeddings,” arXiv:1305.3879 [math.AT].

[11] J. Gamble, H. Chintakunta, and H. Krim, “Applied topology in static and dynamic sensor networks,” *Proceedings of the 2012 International Conference on Signal Processing and Communications (SPCOM)*, pp. 1–5, 2012.

[12] A. Hatcher, *Algebraic Topology*. Cambridge University Press: Cambridge, 2001.

[13] M. Joswig and M. Pfetsch, ”Computing optimal Morse matchings,” *SIAM Journal of Discrete Math*, vol. 20, pp. 11-25, 2006.

[14] J. Kleinberg, ”An impossibility theorem for clustering,” *Advances in Neural Information Processing Systems*, pp. 463-70, MIT Press, 2003.

[15] N. Milosavljević, D. Morozov, and P. Škraba, ”Zigzag persistent homology in matrix multiplication time,” *Proceedings of the 27th Annual Symposium of Computational Geometry*, pp. 216-25, 2011.

[16] K. Mischaikow, and V. Nanda, ”Morse theory for filtrations and efficient computation of persistent homology,” *Discrete & Computational Geometry*, vol. 50, pp. 330-53, Springer, 2013.

[17] T. J. Moore, R. J. Drost, P. Basu, R. Ramanathan, and A. Swami, “Analyzing collaboration networks using simplicial complexes: A case study,” in *Proceedings of the IEEE INFOCOM 2012 Workshop (NetSciCom)*, March 2012, pp. 238–243.

[18] F. Murtagh, ”A survey of recent advances in hierarchical clustering algorithms,” *The Computer Journal*, vol. 26, pp. 354-59, 1983.

[19] M. Nicolau, A. J. Levine, and G. Carlsson, ”Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival,” *Proceedings of the National Academy of Sciences*, vol. 108, pp 7265-70, 2011.

[20] M. Penrose, *Random Geometric Graphs*, Vol. 5, Oxford: Oxford University Press, 2003.

[21] A. Tahbaz-Salehi and A. Jadbabaie, “Distributed coverage verification in sensor networks without location information,” *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, 2008, pp. 4170–4176.

[22] A. Vergne, L. Decreusefond, and P. Martins, “Reduction algorithm for simplicial complexes,” preprint: http://hal.archives-ouvertes.fr/hal-00688919, pp. 1–8, 2012.

[23] R. van de Weygaert, G. Vegter, E. Platen, B. Eldering, and N. Kruithof, ”Alpha shape topology of the cosmic web,” arXiv:1006.2765 [astro-ph.CO].

[24] A. C. Wilkerson, T. J. Moore, A. Swami, A. H. Krim, ”Simplifying the homology of networks via strong collapses,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013.

[25] A. C. Wilkerson, H. Chintakunta, H. Krim, T. J. Moore, A. Swami, ”A distributed collapse of a network’s dimensionality,” in *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Austin, TX, USA, Dec. 2013.