SAND2017-1718C

Seismic Waveform Matching via Approximate Nearest Neighbor with Projection Deficiency Compensation

Antonio Gonzales
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-9999
aigonza@sandia.gov

Nicholas Blazier
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-9999
npblazi@sandia.gov

ABSTRACT

Waveform correlation is a key technology used to identify and locate the source of seismic signals. Due to the computational complexity of waveform correlation, there is a great interest in using approximate nearest neighbor (ANN) techniques to perform efficient searches in large historical seismic archives. In this paper, we examine using ANN methods combined with kernel projection to perform fast matching of seismic signals. We analyze the deficiencies of using correlation based kernel projection with seismic signals and introduce a new algorithm, ANN with projection deficiency compensation (ANN-PCD) , that overcomes these issues to dramatically increase performance.

KEYWORDS

approximate nearest neighbor, kernel projection, seismic data

1 INTRODUCTION

Seismic signal analysis is a core component of earthquake detection, energy exploration, and international nuclear treaty monitoring. An important problem in seismic signal analysis is determining the location on the Earth of the seismic event that generated an observed seismic signal. This is a very complex problem that usually requires signal detections from multiple stations and accurate models of the internal geology of the Earth.

Seismic signals have the interesting property that signals recorded at the same station from seismic events that are very nearly co-located look very similar (Figure 1). Signals that do not originate from the same location will look very different. This property allows researchers to take a new signal and compare it to an archive of historical signals. If a match is found then the new event can immediately be assigned to the same location that generated the historical signal [11][14]. These archives can be very large and continue to grow at a rapid rate. Due to the high computational costs associated with waveform correlation, literature historically focused on either building a small, optimized archives of historical waveforms or using high performance computing architectures to search very large archives. Recent work has started investigating leveraging approximate nearest neighbor (ANN) methods to allow for searching large historical waveform archives quickly [13].

In this paper, we discuss the difficulties of applying traditional ANN methods to seismic waveforms and introduce a new method that greatly improves the performance of ANN for seismic waveform matching in large historical archives.

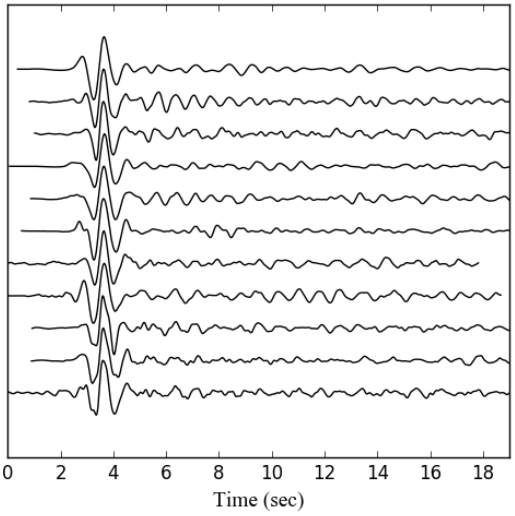

Figure 1: Example of seismic signals detected over 12 years at station MKAR in Kazakhstan from the same earthquake source in the Kuril Trench.

2 BACKGROUND

ANN methods build and populate a data structure known as an index that allows for fast queries while only failing to find a small percentage of the true nearest neighbors (high recall). To date several different approaches have been proposed for Approximate Nearest Neighbor (ANN) search.

Space partitioning methods aim to divide the search space for fast retrieval. These methods include KD-Trees, ball trees, and hierarchical K-Means trees[1][9]. They tend to be fast to construct and search but can be limited by poor space partitioning and in the distance metrics they support. Another popular approach is to work in the domain of *hash functions* such as Locality-Sensitive Hashing (LSH)[9]. These methods provide constant time queries but are limited by the fact that creating a hash function for a specific dataset can be a difficult problem. *Product quantization methods* represent the data distribution through a set of prototype vectors which act like real-valued hash functions [7][8]. These methods represent the state of the art in performance and are relatively fast to build and search. These examples represent just a fraction of the numerous methods that have been developed to build efficient ANN search indices.

KDD'17, Halifax, Nova Scotia, Canada
2017. 978-x-xxxx-xxxx-x/YY/MM…$15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

Most ANN indexing methods require that the distance between items be measured via a Minkowski metric which is of the form:

$$(\sum_{i=1}^{n} |x_i - y_i|^p)^{1/p} \quad (1)$$

i=1 Unfortunately, this type of metric is not appropriate for many types of data. To get around this limitation, datasets are often projected using kernel space methods from the machine learning literature. The concept is that we can perform a non-linear transformation on our data to take it from its original dimensional space to a much higher dimensional space where it becomes linearly separable. This projection creates a new representation of the dataset which can be compared via a Minkowski metric and, therefore, indexed by ANN methods.

A standard method of kernel projection is Kernel Principal Component Analysis (KPCA) [12]. KPCA combines the orthogonal transformation of standard principal component analysis (PCA) with a projection into a kernel space. As with PCA, KPCA can also provide dimensionality reduction. KPCA becomes extremely expensive as the size of the dataset increases due to the need to compute the full covariance of the dataset. To mitigate this problem Jiang *et.al* [6] introduced a method that approximates the covariance for the dataset using random sampling. This greatly reduces the cost of solving for and apply the data projections.

3 RELATED WORK

Yoon *et.al* recently introduced a new method of hashing seismic waveforms called FAST for use with LSH ANN indices [13]. Their method converts waveforms into spectrograms and uses wavelete decomposition to turn the spectrograms into binary hashes. They reported impressive results. The method described in this paper differs as it focuses on combining kernel projection with space partitioning ANN methods like KD-Trees as opposed to hashing methods.

4 KERNELS FOR SEISMIC WAVEFORMS

Seismic waveforms are compared via correlation because it is agnostic to amplitude changes. Correlation is not a Minkowski metric. Additionally, the starting time for each signal is chosen by a human analyst with an estimated uncertainty of ±0.5 seconds. Due to this uncertainty, the distance must be measured by checking a sliding set of alignments via 1D cross-correlation

$$xcorr(a, b) = \arg\max_i < a, b_i > \quad (2)$$

where a and b are waveforms and b_i represents a shift of waveform b by i in time. Since most ANN methods work in the space of distance, 0 indicating a perfect match, we work in the space of *correlation distance* defined as

$$xcorrDistance(a, b) = 1 - xcorr(a, b) \quad (3)$$

Correlation itself is not a valid kernel but it can be converted to one via the following transformation as described by Jiang[5].

$$\kappa(a, b) = exp^{xcorr(a,b)} \quad (4)$$

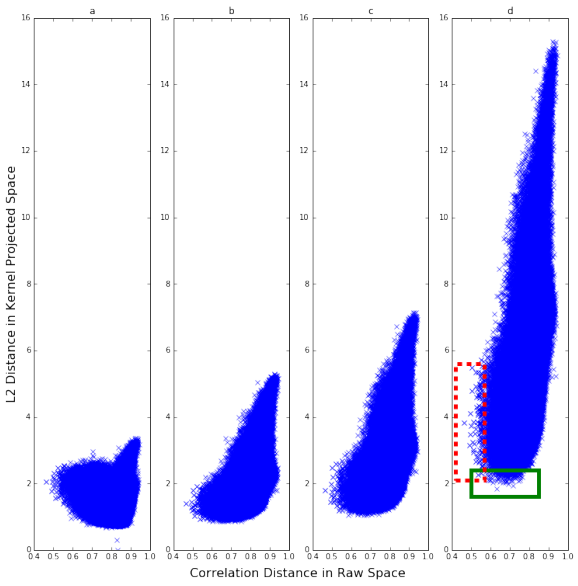

Figure 2: Relationship of the correlation distance in the raw data to the L2 distance of the kernel projected data between a representative query and all members of an archive when a) 200, b) 500, c) 1000, and d) 5000 random samples are used to build the covariance matrix during KPCA projection. In d) the dashed box represents the area of the true nearest neighbors in the raw correlation space. The solid box is the area of the nearest neighbors in kernel projected space.

Given our choice of kernel, we investigated projecting our seismic waveform archive via KPCA using Jiang *et.al*s random sampling method. Ideally, the Euclidean (L2) distance between kernel projected waveforms should have a linear relationship with the correlation distance between their raw waveforms. Figure 2 examines the relationship between these distances as we vary the number of members of the archive randomly selected to calculate the covariance in the KPCA projection. It is obvious that the size of this random set greatly effects the relationship between the distance metrics. The distribution approaches a linear relationship as the number of random samples increases. We also observe that the closest neighbors in the original raw correlation distance tend to sit on the left edge of the distribution and do not correlate very well to the minimum projected L2 distances which set at the bottom of the graph. We call this separation in the two distributions of the nearest neighbors distances *projection deficiency*.

We can use our knowledge of the source of the data to infer what may cause the projection deficiency. Since most seismic events are not common, even large archives will have very few strong correlations for any given query. Often the closest neighbor will barely be separable from the rest of the distribution. Without a good sample strongly correlating matches, it is very unlikely that the projection space will adequately model the local space around a given query. This means that we would likely have to use all the population to calculate covariance during KPCA to optimally represent the space. Doing this would equate to a brute force search

per query and negate any performance increase from ANN. Due to this issue, performance is likely to be poor regardless of the ANN method used.

These issues suggest two paths forward: investigate improved methods of kernel projection or find a way to compensate for the projection deficiency during the search. We note that even though the nearest neighbors in the correlation space do not map directly to the nearest neighbors in the kernel project space they do fall close to each other in the distribution. Inspired by the K-NN graph work of Dong *et.al* [2] and their principle that a neighbor of a neighbor is also likely to be a neighbor, we decided that improving the way an ANN index is searched to compensate for the projection deficiency had a higher return since improvements would not only improve this dataset, but possibly improve future datasets with less than perfect projections.

5 METHODOLOGY

5.1 Kernel Projection

In this section we detail the process used to project our waveform dataset D from its original dim dimensional space to the projected dataset D_ϕ. The covariance between two general vectors is defined as the kernel function:

$$\kappa(x, y) = \phi(x)^T \phi(y) \tag{5}$$

In our case, $\phi(x)$ is defined by equation 4. The kernel covariance matrix K over D is then defined as

$$K_{i,j} = \kappa(D_i, D_j) \tag{6}$$

where D_i is the i-th row of D. K is then decomposed using Principal Component Analysis (PCA) to produce a projection matrix W. As part of PCA, the dimensionality of the data can be reduced by choosing a new dimension $dimReduced < dim$ and preserving only the top $dimReduced$ eigenvectors when creating W. The projected dataset D_ϕ is then created as follows

$$D_\phi = KW \tag{7}$$

To mitigate this expense of computing the full covariance over D we randomly sample $numReps$ rows from D, where $numReps << N$ and use this set to construct K. We refer to this reduced set of random representatives as R. We then build a covariance matrix Kr over R

$$Kr_{i,j} = \kappa(R_i, R_j) \tag{8}$$

PCA is applied to Kr as above to create W and may include a further dimensionality reduction by choosing a $dimReduced < numReps$. The covariance for the dataset is now computed against R

$$K_{i,j} = \kappa(D_i, R_j) \tag{9}$$

and it is projected as in equation 7.

5.2 ANN with Projection Deficiency Compensation

ANN with projection deficiency compensation (ANN-PDC) uses an iterative process to improve the results of a query provided by a standard ANN index. The concept is that the candidate nearest neighbors which are returned during the process of a query provide improved information about the distribution or our difficult kernel space.

A key feature of ANN-PDC is that the distance between a query and its candidate neighbors is calculated independently from the internal queries of I_ϕ. This means that we can use the true distance in the raw correlation space to prioritize candidates for subsequent searches instead of the approximation in the projected data space. This makes it much more likely we are searching the optimal area of the distribution.

Algorithm 1: ANN-PDC

Input: Index built from projected data I_ϕ, projected query q_ϕ, distance function σ, max number of considered neighbors n, and internal query size n_s
Output: Set of candidate nearest neighbors C

begin
  $P \leftarrow$ empty$PriorityQueue$;
  $S \leftarrow \emptyset$;
  $C \leftarrow query(I_\phi, q_\phi, n_s)$;
  Calculate $\sigma(raw(q_\phi), raw(c_\phi)) \quad \forall c_\phi \in C$;
  $addToQueue(P, c_\phi) \quad \forall c_\phi \in C$;
  $cnt = |C|$;
  while $|P| > 0$ *and* $cnt < n$ **do**
    $s_\phi \leftarrow dequeue(P)$;
    $S \leftarrow S \cup s_\phi$;
    $C_s \leftarrow query(I_\phi, s_\phi, n_s)$;
    $C_{new} \leftarrow \{c | c \in C_s, c \notin S\}$;
    Calculate $\sigma(raw(q_\phi), raw(c_\phi)) \quad \forall c_\phi \in C_{new}$;
    $addToQueue(P, c_\phi) \quad \forall c_\phi \in C_{new}$;
    $C \leftarrow C \cup C_{new}$;
    $cnt \leftarrow cnt + |C_{new}|$;
  end
  return C
end

We use KPCA as described in Section 4.1 to project D to D_ϕ and we build our index I_ϕ from D_ϕ. Before we can use I_ϕ to search for a query q, we much project q into the same space as D_ϕ. We do this by preserving the projection matrix W and random representative set R from the projection of D use them to project query q to q_ϕ as shown in equations 10 and 11.

$$Kr_i = \kappa(q, R_i) \tag{10}$$

$$q_\phi = KrW \tag{11}$$

We define a function $query(I_\phi, q_\phi, n)$ that searches I_ϕ for neighbors of q_ϕ where n is the maximum number of considered neighbors. During a search, duplicates are counted as considered neighbors therefore the number of unique neighbors returned may be $\leq n$.

ANN-PDC starts by performing a query for q_ϕ in I_ϕ and then calculating the raw distance between q and each returned candidate neighbor. The candidates are added into a priority queue P that provides access to the candidates in ascending order by their distance

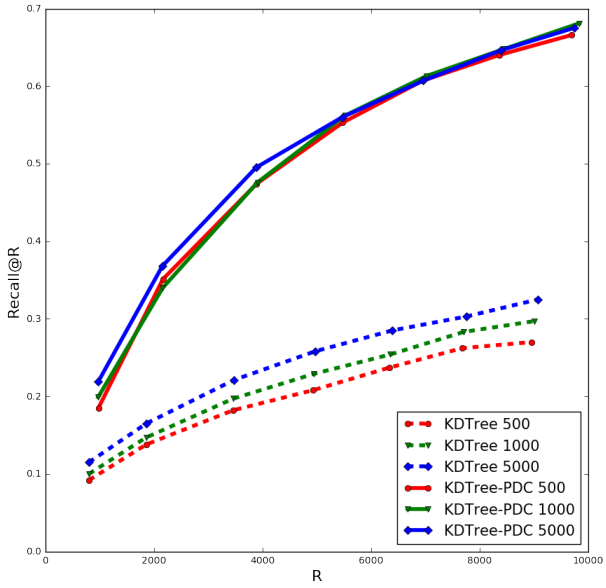

Figure 3: Comparison of performance for KD-Tree vs KD-Tree with PDC for varying number of random samples used during KPCA

to q. After this initialization, we repeat a process of dequeueing the first candidate in P, performing a query for that candidate in I_ϕ, calculating the distance between each resultant new candidate and q in the raw correlation space, and adding these to P. The process is repeated until either P is empty or we have considered a total of n candidates including multiple encounters of the same candidate.

ANN-PDC requires an internal query size parameter $n_s \leq n$ which dictates the number of candidates requested by each internal query. The value of n_s will vary per dataset and is chosen to produce as good of recall as possible while minimizing the amount of required raw distance calculations. ANN-PDC will perform approximately n/n_s internal queries.

To avoid repeating queries, ANN-PDC maintains a set S of candidates that have already been searched and does not let them be re-inserted into P. Additionally, any implementation should ensure that the raw distance between q and a candidate neighbor is only calculated the first time the pair is encountered and not if it is found by subsequent internal queries.

5.3 Complexity Discussion

Here we present a few observations on the cost of the ANN-PDC algorithm. The complexity of a single traversal through a KD-Tree is $O(log_2(N))$ [10]. Since we are limiting our randomized KD-Tree implementation to a constant number of visited leaves (nearest neighbor candidates) n, the complexity of a query remains the same order. After candidates are generated, their distance to the query must be calculated. While the complexity varies depending on the distance metric, the most basic ones, such as Minkowski metrics, will be $O(dim)$. This implies that the cost of calculating the distances will dominate algorithm time when

$$2^{dim} > N \tag{12}$$

which applies to most real-world data problems.

For each query, ANN-PDC performs n/n_s internal queries, calculates distances between the query and all candidates, and has additional bookkeeping costs such as managing the priority queue and set operations. Since n remains constant, none of these operations changes the overall complexity from $O(log_2(N))$ along with the $O(dim)$ for calculating the distances. Theoretically, this is the same as using KD-Trees. In practice, depending on implementation, it is likely that ANN-PDC will do slightly more work than a single KD-Tree query for the same number of distance calculations. With time being dominated by the distance calculations, the extra work will negligible.

It is important to note that there are several ANN use cases where the distance between the query and candidate nearest neighbors does not need to be calculated. In these cases, the metadata associated with the returned candidates is enough to support a decision. Since calculating distances is central to our method, ANN-PDC is not an appropriate choice for these use cases.

6 PERFORMANCE ANALYSIS

6.1 Seismic Waveform Dataset

In this section, we will examine the effect of using ANN-PDC over a real-world seismic waveform dataset. This dataset contains 308,219 analyst reviewed signals from known seismic events detected by the station MKAR, located in Kazakhstan, over the time period 2002-2013. We built a test set with 25,865 analyst reviewed signals from known events from the same station during 2014. These data are publicly available from the Incorporated Research Institutions for Seismology [3]. For each signal detection, we take a 30 second sample starting 2 seconds before the time that the signal was detected at the station. The data are sampled at 40Hz meaning that each signal is comprised of 1200 samples.

6.2 Analysis Setup

All analyses are performed using randomized KD-Tree forests as defined by Silpa-Anan and Hartley [10]. For the purpose of simplicity, we will use the term *KD-Tree Index* to refer to a randomized KD-Tree forest index for the remainder. While there are more state-of-the-art ANN methods available, KD-Tree indices are widely used, easy to implement, and fast to train. We feel they serve as a good baseline. We use our own implementation which is similar to the version presented in the FLANN library [9].

To compare performance, we use the measure Recall@R, as suggested by INRIA [4], which is considered a standard for comparing ANN performance. Recall@R is calculated as the percentage of queries where the true nearest neighbor is returned as part the N candidates returned by a search.

6.3 PDC Performance Analysis

Our initial analysis compares the performance of a basic KD-Tree index vs. a KD-Tree with PDC while varying the number of random samples used to calculate the covariance during KPCA projection. For this test there are 25 trees in each KD-Tree forest. For KD-

Tree with PDC ns = 250. Figure 3 shows that ANN-PDC averages an increase in recall@R over 200%. For the base KD-Tree index, performance increases with the number of random samples used for the covariance for both index types. This is expected as the larger this number becomes the better the space will be modeled. For ANN-PDC, this is true for smaller R but the performance equalizes as R increases. The number of random samples must be chosen carefully because the query time also increases linearly due to projection costs. In this case, we would choose 1000 as a good compromise between projection cost and performance.

6.4 Number of internal queries

The one new parameter that ANN-PDC introduces is the size of the internal queries n_s. Figure 4 shows the effect that the n_s parameter has on the performance of ANN-PDC for a KD-Tree index of 25 trees and 1000 random samples during KPCA. A value of $n_s = 250$ generally outperforms both $n_s = 50$ and $n_s = 1000$. At $n_s = 50$, we are likely searching too small a local area and not finding enough unique candidates. At $n_s = 1000$, we are likely searching too large an area and performing extra distance comparisons without any benefit.

7 DISCUSSION

After our initial success with the ANN-PDC method using KD-Tree indices, we assumed that moving to a more state-of-the-art ANN index would only improve the performance. We decided to use the LOPQ method from Kalantidis *et.al*[8]. As a baseline, we ran LOPQ against our projected dataset and found the performance is actually worse that KD-Trees as shown in Figure 5. This is not surprising as the projection deficiency issue effects LOPQ just the same as KD-Trees.

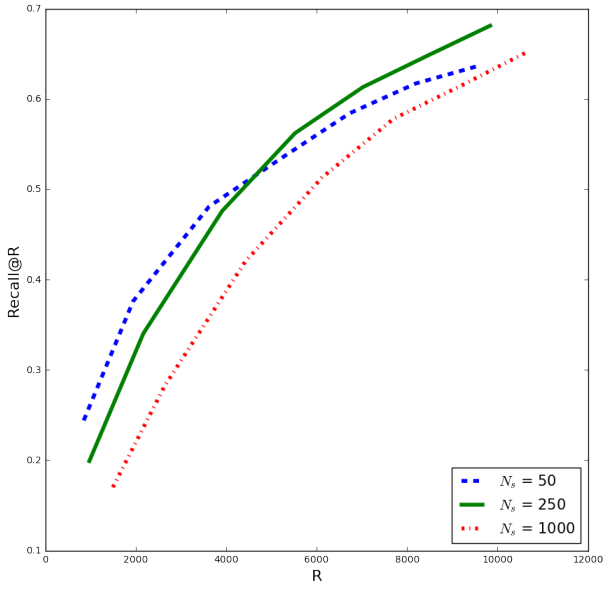

Figure 4: ANN-PDC performance for different n_s values for a KD-Tree index with PDC

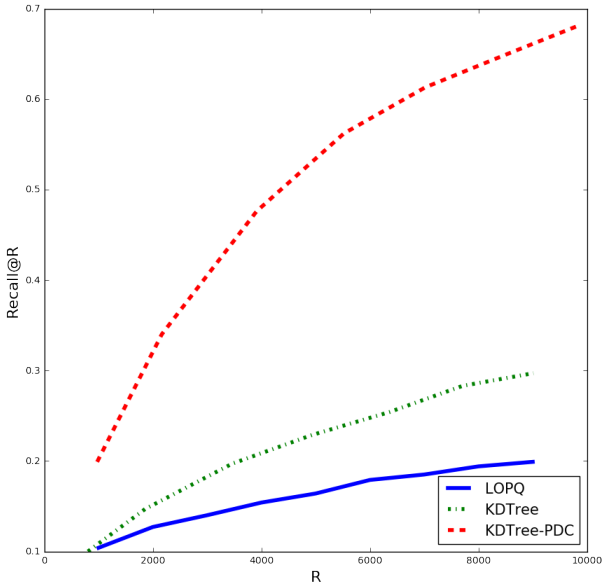

Figure 5: Performance comparison for LOPQ, KD-Tree, and KD-Tree with PDC

When we tried adding PDC to LOPQ, we were surprised to find that the performance degraded severely. LOPQ-PDC constantly terminated early due to an empty priority queue with an average of about 800 candidates returned. The value for recall@800 was nearly identical to the value produced for base LOPQ.

We investigated why this was happening by looking at the progress of a ANN-PDC search through its internal queries. Figure 6 shows that both KD-Tree and LOPQ initialize in roughly the same area with KD-Tree having a bit more variance in the returned candidates. LOPQ is so accurate at finding the neighbors in the immediate local area of the projected query that PDC is never able to move the search out of the local the area of projected neighbors and into the area of the true neighbors in the correlation space. It turns out that ANN-PDC needs the high variance in the returns of the KD-Tree index in order move the search space into the area around the true nearest neighbors.

8 CONCLUSION

Waveform correlation is emerging an important tool for locating and identifying the source of seismic signals. Due to the affordability of storage and increase in computer processing power, ANN methods are a prime candidate for fast seismic signal matching to historical events. Traditional kernel projection methods prove deficient to adequately model this tough real world dataset for use with ANN. Our algorithm, ANN-PDC, mitigates projection deficiency by compensating through iterative neighbor search providing drastically improved results. We leave for future work investigating modifying the KD-Tree index to better search the space.

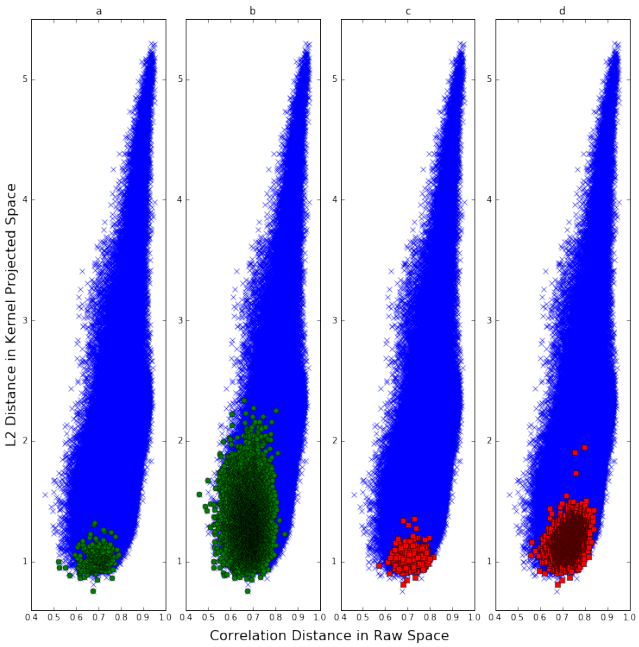

Figure 6: The initial 200 and final candidate nearest neighbors returned during a ANN-PDC query using a KD-Tree index (a, b) and using a LOPQ index (c, d).

9 ACKNOWLEDGEMENTS

We would like to thank Kurt Larson, Chris Young, Sandy Ballard, Rigo Tibi, and Jonathan Woodbridge for their fantastic ideas, expertise, and technical review.

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

REFERENCES

[1] Jon Louis Bentley. 1975. Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM* 18, 9 (Sept. 1975), 509–517. DOI: http://dx.doi.org/10.1145/361002.361007

[2] Wei Dong, Charikar Moses, and Kai Li. 2011. Efficient K-nearest Neighbor Graph Construction for Generic Similarity Measures. In *Proceedings of the 20th International Conference on World Wide Web (WWW '11)*. ACM, New York, NY, USA, 577–586. DOI: http://dx.doi.org/10.1145/1963405.1963487

[3] IRIS. 2016. Incorporated Research Institutions for Seismology. https://www.iris.edu. (2016).

[4] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2011. Product Quantization for Nearest Neighbor Search. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 1 (Jan. 2011), 117–128. DOI: http://dx.doi.org/10.1109/TPAMI.2010.57

[5] Hao Jiang and Wai-Ki Ching. 2012. Correlation Kernels for Support Vector Machines Classification with Applications in Cancer Data. *Computational and Mathematical Methods in Medicine* 10.1155/2012/205025 (2012). http://dx.doi.org/10.1155/2012/205025

[6] Ke Jiang, Qichao Que, and Brian Kulis. 2014. Revisiting Kernelized Locality-Sensitive Hashing for Improved Large-Scale Image Retrieval. *CoRR* abs/1411.4199 (2014). http://arxiv.org/abs/1411.4199

[7] Herv Jgou, Matthijs Douze, and Cordelia Schmid. 2011. Product Quantization for Nearest Neighbor Search. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 1 (2011), 117–128.

[8] Yannis Kalantidis and Yannis Avrithis. 2014. Locally optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2321–2328.

[9] Marius Muja and David G. Lowe. 2009. Fast approximate nearest neighbors with automatic algorithm configuration. In *In VISAPP International Conference on Computer Vision Theory and Applications*. 331–340.

[10] Chanop Silpa-Anan and Richard I. Hartley. 2008. Optimised KD-trees for fast image descriptor matching.. In *CVPR*. IEEE Computer Society.

[11] Megan Slinkard, Stephen Heck, David Schaff, Nedra Bonal, David Daily, Christopher Young, and Paul Richards. 2016. Detection of the Wenchuan Aftershock Sequence Using Waveform Correlation with a Composite Regional Network. *Bulletin of the Seismological Society of America* (2016). DOI: http://dx.doi.org/10.1785/0120150333 arXiv:http://bssa.geoscienceworld.org/content/early/2016/06/23/0120150333.full.pdf

[12] Quan Wang. 2012. Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models. *CoRR* abs/1207.3538 (2012). http://arxiv.org/abs/1207.3538

[13] Clara Yoon, Ossian O'Reilly, Karianne Bergen, and Gregory Beroza. 2015. Earthquake detection through computationally efficient similarity search. *Science Advances* 1, 11 (2015). DOI: http://dx.doi.org/10.1126/sciadv.1501057 arXiv:http://advances.sciencemag.org/content/1/11/e1501057.full.pdf

[14] Jie Zhang, Haijian Zhang, Ehong Chen, Yi Zheng, Wenhaun Kuang, and Xiong Zhang. 2014. Real-time earthquake monitoring using a search engine method. *Nature Communications* 5:5664 doi:10.1038/ncomms664 (2014).