# NNK-Means: Data summarization using dictionary learning with non-negative kernel regression

Sarath Shekkizhar, Antonio Ortega

*University of Southern California*

Los Angeles, CA, USA

{shekkizh, aortega}@usc.edu

*Abstract*—An increasing number of systems are being designed by gathering significant amounts of data and then optimizing the system parameters directly using the obtained data. Often this is done without analyzing the dataset structure. As task complexity, data size, and parameters all increase to millions or even billions, *data summarization* is becoming a major challenge. In this work, we investigate data summarization via dictionary learning (DL), leveraging the properties of recently introduced non-negative kernel regression (NNK) graphs. Our proposed NNK-Means, unlike previous DL techniques, such as kSVD, learns geometric dictionaries with atoms that are representative of the input data space. Experiments show that summarization using NNK-Means can provide better class separation compared to linear and kernel versions of kMeans and kSVD. Moreover, NNK-Means is scalable, with runtime complexity similar to that of kMeans.

*Index Terms*—Data summarization, dataset analysis, dictionary learning, neighborhood methods, kernel methods.

## I. INTRODUCTION

Massive high-dimensional datasets are becoming an increasingly common input for system design. While large datasets are easier to collect, the methods for exploratory (understanding or characterizing the data) and confirmatory (confirming the validity and stability of a system designed using the data) analysis are not as scalable and require new techniques that can cope with big data sizes [1], [2]. Data summarization methods aim to represent large datasets by a small set of elements, the insights from which can be used to organize the dataset into clusters, classify observations to its clusters, or detect outliers [3]. In datasets with label information, a summary can be obtained for each class, but summaries are in general decoupled from downstream data-driven system designs and thus different from coresets and sketches [4], [5].

Clustering methods such as kMeans [2], [6], vector quantization [7] and their variants [8], are among the most prevalent approaches to data summarization [9]. A desirable property for summarization, which can be obtained with clustering methods, is the geometric interpretability of elements in the summary. For example, in kMeans the elements in the summary are centroids, which are obtained by averaging points in the input data space, and thus are themselves in the same data space, so that one can associate properties (e.g., labels) to the summary points based on the data points these are derived from. In kMeans, each point in the dataset can be considered as a 1-sparse representation based on the nearest cluster center. This leads to hard partitioning of the input data space, which suggests that better summarization is possible if the optimization allows for points to be approximated by a sparse linear combination of summary points.

In this paper, we investigate data summarization using a dictionary learning (DL) framework where the summary, or dictionary, is optimized for $k$-sparsity, with $k > 1$, i.e., each data point is represented by a $k$-sparse combination of elements (atoms) from an adaptively learned set, the dictionary. It is important to note that using previous DL schemes such as the method of optimal directions (MOD) [10], the kSVD algorithm [11], and their kernel extensions [12]–[14], for DL-based data summarization is not possible for several reasons. Firstly, current DL methods learn dictionary atoms that are optimized to represent data and their *approximation residuals* [15]. This means that atoms in the dictionary are not guaranteed to be points that are on, or even near, the input data manifold and do not have geometric properties as those of cluster centers in kMeans. Secondly, although DL methods perform well in signal and image processing tasks, their application to machine learning problems is largely limited to learning class-specific dictionaries that can be later used for classification [16], [17]. This is because the individual atoms learned by DL cannot be directly associated to labels, or other properties of the data, and can only be assigned labels if separate class-wise dictionaries are learned. Finally, current DL schemes are impractical even for datasets of modest size [18], [19] and are thus not suitable for summarization involving large datasets.

To overcome these limitations and learn dictionaries with atoms that can be used for data summarization, we leverage our work on neighborhood definition with non-negative kernel regression (NNK) [20], [21]. Our proposed *NNK-Means* algorithm for data summarization is based on a dictionary obtained using a sparse coding technique based on NNK, where each selected neighbor corresponds to a direction in input space that is not represented by other selected neighbors. This representation can be interpreted geometrically as a polytope covering of the data by selected atoms [20].

In all DL methods, as in ours, learning is done by alternating two minimization steps, namely sparse coding and dictionary update, with differences between the methods arising given the choice of constraints or optimization in these two steps. The main novelty in NNK-Means comes from the use of a
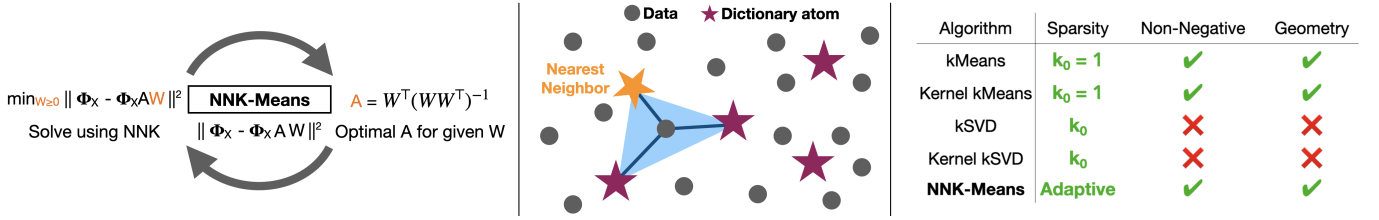
Fig. 1: **Left:** Proposed NNK-Means. The algorithm alternates between sparse coding ($\boldsymbol{W}$) using NNK and dictionary update ($\boldsymbol{A}$) until either the dictionary elements converge, or a given number of iterations or a reconstruction error is achieved. **Middle:** During sparse coding, kMeans assigns each data point to its nearest neighbor while NNK represents each data point in an *adaptively* formed convex polytope made of the dictionary atoms. **Right:** Comparative summary between dictionary learning methods and proposed NNK-Means approach. kMeans offers a 1-sparse dictionary learning approach while a kSVD offers a more general approach where the sparse coding stage accommodates for a chosen, fixed $k_0$-sparsity but lacks geometry. NNK-Means has adaptive sparsity that relies on the relative positions of atoms around each data to be represented.

non-negative sparse coding procedure in kernel space that can be described, similar to kMeans, in terms of the local data geometry. Non-negative DL and kernelized DL were separately studied in [22]–[24] and [13], [14], respectively. Closest to our work are [25], [26], where dictionaries are learned in kernel space with non-negative sparse coding performed using optimization schemes, such as an $l_1$ constrained quadratic solver with multiplicative updates [26]–[28] or make use of iterative thresholding algorithms with non-negativity constraint [22]–[24]. In contrast, our framework makes use of a geometric sparse coding approach based on local neighbors, a procedure previously unexplored in DL. The sparsity of the representation in our approach depends on the relative position of the data and atoms (i.e., the data geometry) and is thus interpretable and adaptive. Consequently, unlike earlier DL methods, individual atoms learned by NNK-Means have explicit geometric properties, with representations that are obtained as averages of input data examples similar to kMeans, and can be associated with data properties such as class labels. This makes the atoms learned by our approach suitable for data summarization. Note that previous DL methods and sparse coding approaches lacked such properties, so the proposed NNK-Means is the first DL framework to study these concepts with emphasis on geometry for use in data summarization.

Our experiments show that the NNK-Means i) selects atoms for summarization that belong to the data space, ii) outperforms DL methods in terms of downstream classification using class-specific summaries on several datasets (USPS, MNIST, and CIFAR10), and iii) achieves train and test runtimes similar to kernel kMeans, and $67\times$ and $7\times$ faster than kernel kSVD.

## II. PROBLEM SETUP AND BACKGROUND

*Sparse Dictionary Learning:* Given a dataset of $N$ data points represented by a matrix $\boldsymbol{X} \in \mathbb{R}^{d \times N}$, the goal of DL is to find a dictionary $\boldsymbol{D} \in \mathbb{R}^{d \times M}$ with $M << N$, and a sparse matrix $\boldsymbol{W} \in \mathbb{R}^{M \times N}$ that optimizes data reconstruction.

$$\hat{\boldsymbol{D}}, \hat{\boldsymbol{W}} = \underset{\boldsymbol{D}, \boldsymbol{W} \,:\, \forall i \;\; ||\boldsymbol{W}_i||_0 \leq \mathrm{k}}{\arg\min} ||\boldsymbol{X} - \boldsymbol{D}\boldsymbol{W}||_F^2 \qquad (1)$$

where the $\ell_0$ constraint on $\boldsymbol{W}$ corresponds to the sparsity requirements on the columns of the reconstruction coefficients

$\boldsymbol{W}_i \in \mathbb{R}^M$ and $|| \, . \, ||_F$ represents the Frobenius norm of the reconstruction error associated with the representation.

While kMeans can be written in terms of the DL objective (1) with a 1-sparse constraint on the sparse coding, i.e., each column of $\boldsymbol{W}$ can have only one nonzero value, there are several important differences between the two problems. In particular we can see that: (i) the coefficients involved in the sparse coding of kMeans are non-negative, (ii) in kMeans, the sparse coding is based on proximity of the data to the atoms (i.e., cluster centers), whereas in kSVD or MOD, coding is done by searching for atoms that maximally correlate with the residual, and (iii) the dictionary updates are different and lead to different dictionaries.

A straightforward way of kernelizing DL would involve replacing the input data by their respective Reproducing Kernel Hilbert Space (RKHS) representation. However, such a setup is unable to leverage the *kernel trick* [29], [30] and thus to overcome this problem, [13] suggest decomposing the dictionary and solving a modified objective (1), namely,

$$\hat{\boldsymbol{A}}, \hat{\boldsymbol{W}} = \underset{\boldsymbol{A}, \boldsymbol{W} \,:\, \forall i \;\; ||\boldsymbol{W}_i||_0 \leq \mathrm{k}}{\arg\min} ||\boldsymbol{\Phi} - \boldsymbol{\Phi}\boldsymbol{A}\boldsymbol{W}||_F^2 \qquad (2)$$

where $\boldsymbol{\Phi} = \phi(\boldsymbol{X})$ corresponds to the RKHS mapping of the data. In this setup, one learns a dictionary ($\boldsymbol{D} = \boldsymbol{\Phi}\boldsymbol{A}$) via the coefficient matrix $\boldsymbol{A} \in \mathbb{R}^{N \times M}$.

*Non-Negative Kernel Regression:* The starting point for our DL method is our graph construction framework using NNK [20], [21]. NNK formulates neighborhoods as a signal representation problem, where data points (represented as a RKHS function) is to be approximated by functions corresponding to its neighbors, i.e.,

$$\underset{\boldsymbol{\theta} \geq 0}{\min} \; ||\phi(\boldsymbol{x}_i) - \boldsymbol{\Phi}_S \boldsymbol{\theta}||^2, \qquad (3)$$

where $\boldsymbol{\Phi}_S$ contains the RKHS representation of a pre-selected set of data points that are good candidates for NNK neighborhood. Unlike k-nearest neighbor or $\epsilon$-neighborhood, where a neighbor is selected based on only $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$, and can be viewed as representation using thresholding, NNK leads to optimal neighborhoods, that avoids selecting two neighbors that are similar to each other. Geometrically, this

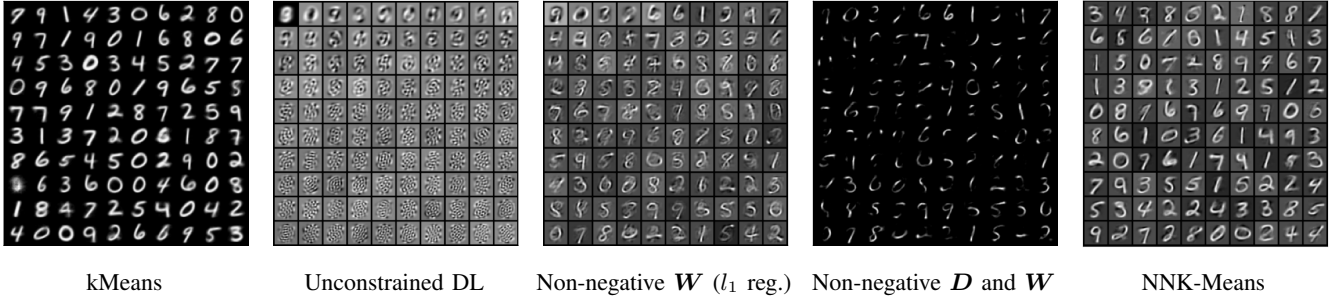| kMeans | Unconstrained DL | Non-negative $W$ ($l_1$ reg.) | Non-negative $D$ and $W$ | NNK-Means |

Fig. 2: 100 atoms for MNIST digits (contrast scaled for visualization) obtained using kMeans, DL methods and their constrained variants, NNK-Means (using cosine kernel [21]). Unlike earlier DL approaches, our method learns individual atoms that are linear combinations of the digits themselves and can be associated geometrically to the input data. Such explicit properties of atom are lost when working with $l_1$ regularized or thresholding based sparse coding methods used previously in DL.

can be explained using hyperplanes, one per selected NNK neighbor, which applied inductively leads to a convex polytope around the data such as the one in Figure 1.

## III. PROPOSED METHOD: NNK-MEANS

We propose a two-stage learning scheme where we solve sparse coding and dictionary update until convergence, or until a given number of iterations or reconstruction error is reached. We describe the two steps, the respective optimization involved, interpretation, and runtime complexity in this section[1].

*Sparse Coding:* Given a dictionary, $A$, in this step we seek to find a sparse matrix $W$ that optimizes data reconstruction in kernel space. We will additionally require the coefficients of representation to be non-negative with the number of nonzero coefficients at most $k$. Thus, the objective to minimize at this step is

$$\hat{W} = \underset{\forall i \ W_i \geq 0, \ ||W_i||_0 \leq k}{\arg\min} \sum_{i=1}^{N} ||\phi_i - \Phi A W_i||_2^2, \quad (4)$$

where $\phi_i$ corresponds to the RKHS representation of data $x_i$. Solving for each $W_i$ in equation (4) involves working with a $N \times N$ kernel matrix leading to run times that scale poorly with the size of the dataset. However, the geometric understanding of the NNK objective in [20], allows us to efficiently solve for the sparse coefficients ($W_i$) for each data point by selecting and optimizing starting from a small subset of data points, here the $k$-nearest neighbors. Objective (4) can be rewritten for each data point and solved with NNK to obtain $W_i$ as

$$\hat{W}_{i,S} = \underset{\theta_i \geq 0}{\arg\min} ||\phi_i - \Phi A_S \theta_i||_2^2 \text{ and } \hat{W}_{i,S^c} = 0 \quad (5)$$

where the set $S$ corresponds to the selected subset of indices corresponding to the set of the dictionary atoms $\Phi A$ that can have a nonzero influence in the sparse non-negative reconstruction. The above reduced objective can be solved efficiently as in NNK graphs [20]. Sparse coding using NNK allows us to explain the obtained sparse codes, leverage nearest-neighbor tools for scaling to large datasets, and analyze the obtained

[1]A longer version of the paper is posted on arxiv with proofs of theoretical statements and additional experiments [31]

atoms geometrically, very much similar to kMeans, where each data is represented by an adaptive set of non-redundant neighbors rather than just 1. This step includes a neighborhood search and a non-negative quadratic optimization with runtime complexities $\mathcal{O}(NMd)$ and $\mathcal{O}(Nk^3)$.

*Dictionary Update:* Assuming that the sparse codes for each training data, $W$, are calculated and fixed, the goal is to update $A$ such that the reconstruction error is minimized. Here, we propose an update similar to MOD, where the dictionary matrix $A$ is obtained based on $W$ as

$$\hat{A} = W^\top (WW^\top)^{-1} \quad (6)$$

The runtime associated with this step is $\mathcal{O}(M^3 + NMk)$, where we use the fact that $W$ has at most $Nk$ non zero elements. We note that using $k$-nearest neighbor directly for sparse coding, apart from lacking adaptivity, is sub-optimal and leads to instabilities at the dictionary update stage and thus is unsuitable for DL in a similar setup.

**Proposition 1.** *The dictionary update rule in (6) reduces to kMeans cluster update $A = W^\top \Sigma^{-1}$ when $W$ consists of $N$ columns from $(e_1 \ldots e_M)$, where $e_m$ is a basis vector, i.e., $e_{mi} = 0 \ \forall \ i \neq m$ and $e_{mm} = 1$ and $\Sigma \in \mathbb{R}^{M \times M}$ is a diagonal matrix containing the degree or number of times each basis vector $e_m$ appears in $W$.*

Proposition 1 shows that our proposed method reduces to the kMeans algorithm when the sparsity of each column in $W$ is constrained to 1 and can thus be considered a DL generalization that maintains the geometric and interpretable properties of kMeans. One can easily verify that our iterative procedure for DL, alternating between sparse coding and dictionary update, does converge (Theorem 1) and produces atoms that belong to the input data manifold.

**Theorem 1.** *The residual $||\Phi - \Phi A W||_F^2$ decreases monotonically under the NNK sparse coding step (5) for $W$ given matrix $A$. For a fixed $W$, the dictionary update (6) for $A$ is the optimal solution to $\min_A ||\Phi - \Phi A W||_F^2$. Thus, NNK-Means objective decreases monotonically and converges.*

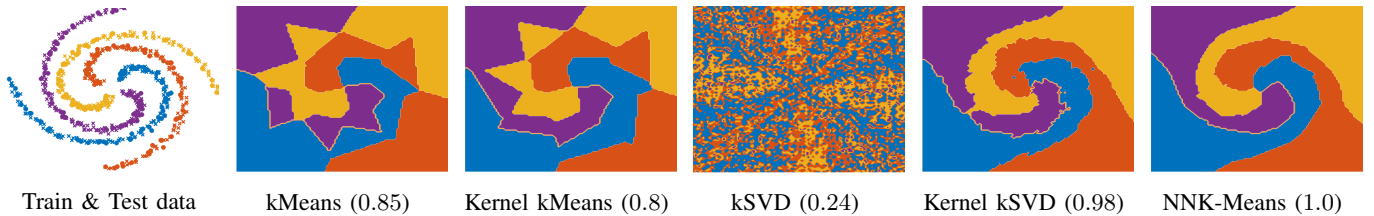| Train & Test data | kMeans (0.85) | Kernel kMeans (0.8) | kSVD (0.24) | Kernel kSVD (0.98) | NNK-Means (1.0) |

Fig. 3: Visualizing predictions (each color corresponds to a class) obtained using various dictionary based classification schemes on a 4-class synthetic dataset. Each method learns a 10 atom dictionary per class based on given training data ($N = 600$, $\times$) with sparsity constrain, where applicable, of 5. The learned per class dictionary is then used to classify a test data ($N = 200$, $\bullet$) with accuracy as indicated in parenthesis for each method. We see that the kSVD approach is unable to adapt to the nonlinear structure of data and adding a kernel is crucial in such scenarios. NNK-Means better adapts to the geometry of the data with a runtime comparable to that of kMeans while having $4\times$ and $2\times$ faster train and test times in comparison with kernel kSVD.

## IV. EXPERIMENTS

In this section, we validate properties of NNK-Means that make it suitable for data summarization. Figure 2 presents a visual comparison of the atoms obtained using our method with that of kMeans and previous DL approaches. Unlike standard DL approaches, we observe that atoms learned by NNK-Means have representations that are similar to the input data. We will now focus on a standard experiment setting in DL, namely DL-based classification [13], [19], to compare NNK-Means with previous DL approaches. Note that learning a good summary leads to better classification. Since existing DL methods cannot associate labels directly to the atoms obtained, experiments are constrained to learning a dictionary for each class ($\{\boldsymbol{A}_i\}_{i=1}^{C}$) in training data that are later used to classify queries based on the class-specific reconstruction error $\{e_i\}_{i=1}^{C}$, i.e., we sparse code a query $\boldsymbol{x}_q$ using each dictionary $\boldsymbol{A}_i$ and assign the query to the class ($c$) with lowest reconstruction error ($e_c$). NNK-Means outperforms all other methods consistently in classification while having desirable runtimes relative to kMeans, kSVD, and their kernelized versions[2] in both synthetic and real datasets. We use a Gaussian kernel $\kappa(\boldsymbol{x}, \boldsymbol{y}) = \exp(||\boldsymbol{x} - \boldsymbol{y}||_2^2/2)$ and report average performance over 10 runs for all experiments.

*Synthetic dataset:* We consider a 4-class dataset consisting of samples generated from a non-linear manifold and corrupted with gaussian noise (as in Figure 3). Since the data corresponding to each class have similar support, namely the entire space $\mathbb{R}^2$, dictionaries learned using kSVD are indistinguishable for each class and lead to at-chance performance in classification of test queries. On the contrary, a kernelized version of kSVD is able to handle the manifold, although it is not robust at some test locations, but at the cost of increased computational complexity. Interestingly, we observe that a non-negative neighborhood-based sparse coding is able to adapt to input space non-linearity even when constrained to 1-sparsity (kMeans) indicative of the importance of non-negativity and geometry in data summarization.

| Method | MNIST-S | MNIST | CIFAR-S | CIFAR |
|---|---|---|---|---|
| kMeans | 94.89 | 96.34 | 83.88 | 84.91 |
| K-kMeans | 91.56 | 93.19 | 84.22 | 85.06 |
| kSVD | 95.53 | 95.86 | 86.01 | 86.28 |
| K-kSVD | 96.45 | — | 86.71 | — |
| NNK-Means | **96**.**70** | **97**.**79** | **86**.**95** | **87**.**21** |

TABLE I: Classification accuracy (in %, higher is better) on MNIST, CIFAR10 and their subset (S, 20% of randomly sampled training set). Each method learns a 50-atom dictionary per class, initialized randomly, with sparsity constraint, where applicable, of 30 and run for at most 10 iterations. NNK-Means consistently produces better classification in terms of test accuracy while having a reduced runtime in comparison to kSVD approaches and comparable to that of kMeans. Kernel k-SVD produces comparable performance but at the cost of $67\times$ and $7\times$ slower train and test time relative to NNK-Means.

*USPS, MNIST, CIFAR10:* We use as features the pixel values of the images for USPS ($d = 256$) and MNIST ($d = 784$) dataset. For CIFAR10, we train a self-supervised model using SimCLR loss [33] on unlabelled training data to obtain features ($d = 512$) for our experiment. We use the standard train/test split for each dataset and standardize the feature vectors to zero mean and unit variance. We report here results of DL with a subset of the training data, namely MNIST-S and CIFAR10-S, for a fair comparison with kernel kSVD. We note that kernel kSVD scaled poorly with dataset size and timed out when working with the entire training set of MNIST and CIFAR10. NNK-Means is able to efficiently learn a compact set of atoms that are capable of representing each class which in turn provides better classification of test data in all settings as made evident in Figure 4 and the results in Table I.

## V. CONCLUSION

We investigate data summarization using DL and propose a framework, NNK-Means, that overcomes the limitations of previous DL methods for summarization. NNK-Means learns atoms that are geometric like kMeans centroids and leverages neighborhood tools to efficiently perform sparse coding and adaptively represent data using learned summary elements or

---

[2]We use the efficient implementations, as in [19], from omp-box and kSVD-box libraries [32] and Kernel kSVD code of [13].
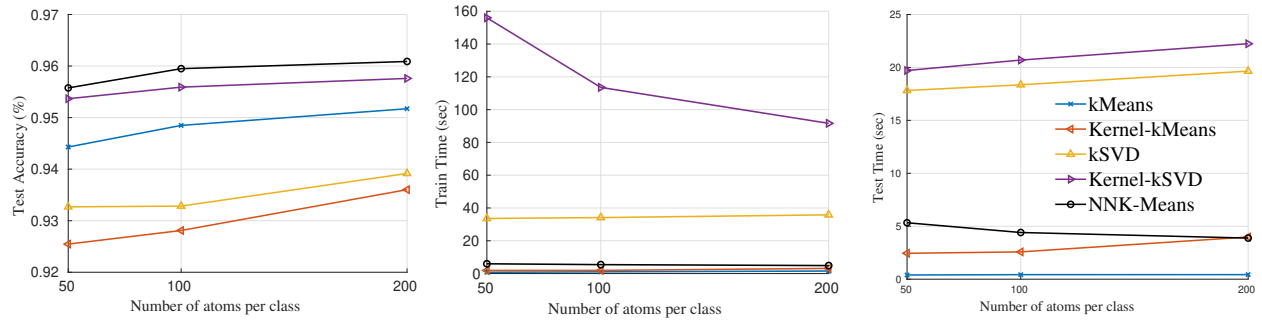
Fig. 4: **Plots:** Test classification accuracy, train time and test time as a function of number of dictionary atoms per class on USPS dataset for various DL methods. Each method is initialized similarly and is trained for a maximum of 10 iterations with sparsity constraint, where applicable, of 30. The plots demonstrate the benefits of NNK-Means both in accuracy of classification and runtime. The major gain in runtime for NNK-Means comes from the pre-selection of atoms in the form of nearest neighbors which leads to fast sparse coding (as can be seen via test time which performs only sparse coding) relative to kSVD approaches that sequentially perform a linear search for atoms that correlate with the residue at that stage. The training time in kSVD approaches decreases with more atoms since the sparse coding stage requires fewer atom selection steps compared to those where the dictionary is small. Source code for all experiments are made available at github.com/STAC-USC.

atoms of the dictionary. Experiments show that our method has runtimes similar to kMeans while learning dictionaries that can provide better discrimination than competing methods. In the future, we plan to study the trade-offs associated with summary size and the use of obtained summaries in improving analysis and design of data-driven machine learning systems.

## REFERENCES

[1] J. W. Tukey *et al.*, *Exploratory data analysis*, vol. 2. Addison Wesley Publishing Company, 1977.

[2] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, 2010.

[3] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive data sets*. Cambridge university press, 2020.

[4] J. M. Phillips, "Coresets and sketches," in *Handbook of discrete and computational geometry* (C. D. Toth, J. O'Rourke, and J. E. Goodman, eds.), CRC press, 2017.

[5] A. Munteanu and C. Schwiegelshohn, "Coresets-methods and history: A theoreticians design pattern for approximation and streaming algorithms," *KI-Künstliche Intelligenz*, vol. 32, no. 1, 2018.

[6] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. on Information theory*, vol. 28, no. 2, 1982.

[7] R. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, no. 2, 1984.

[8] M. Kleindessner, P. Awasthi, and J. Morgenstern, "Fair k-center clustering for data summarization," in *Intl. Conf. on Mach. Learning*, 2019.

[9] G. Gan, C. Ma, and J. Wu, *Data clustering: theory, algorithms, and applications*. SIAM, 2020.

[10] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *Intl. Conf. on Acoustics, Speech, and Signal Processing.*, vol. 5, IEEE, 1999.

[11] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal processing*, vol. 54, no. 11, 2006.

[12] L. Zhang, W.-D. Zhou, P.-C. Chang, J. Liu, Z. Yan, T. Wang, and F.-Z. Li, "Kernel sparse representation-based classifier," *IEEE Trans. on Signal Processing*, vol. 60, no. 4, 2011.

[13] H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Kernel dictionary learning," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2012.

[14] H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Design of non-linear kernel dictionaries for object recognition," *IEEE Trans. on Image Processing*, vol. 22, no. 12, 2013.

[15] B. Dumitrescu and P. Irofti, *Dictionary learning algorithms and applications*. Springer, 2018.

[16] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[17] T. H. Vu and V. Monga, "Fast low-rank shared dictionary learning for image classification," *IEEE Trans. on Image Processing*, vol. 26, 2017.

[18] D. Feldman, M. Feigin, and N. Sochen, "Learning big (image) data via coresets for dictionaries," *J. of Mathematical imaging and vision*, vol. 46, no. 3, 2013.

[19] A. Golts and M. Elad, "Linearized kernel dictionary learning," *J. of Selected Topics in Signal Processing*, vol. 10, no. 4, 2016.

[20] S. Shekkizhar and A. Ortega, "Graph construction from data by Non-Negative Kernel Regression," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020.

[21] S. Shekkizhar and A. Ortega, "Revisiting local neighborhood methods in machine learning," in *Data Science and Learning Workshop (DSLW)*, IEEE, 2021.

[22] M. Aharon, M. Elad, and A. M. Bruckstein, "K-SVD and its non-negative variant for dictionary design," in *Wavelets XI*, vol. 5914, Intl. Society for Optics and Photonics, 2005.

[23] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "K-web: Nonnegative dictionary learning for sparse image representations," in *2013 IEEE Intl. Conf. on Image Processing*, pp. 146–150, IEEE, 2013.

[24] Q. Pan, D. Kong, C. Ding, and B. Luo, "Robust non-negative dictionary learning," in *28th AAAI Conf. on Artificial Intelligence*, 2014.

[25] B. Hosseini, F. Hülsmann, M. Botsch, and B. Hammer, "Non-negative kernel sparse coding for the analysis of motion data," in *Intl. Conf. on Artificial Neural Networks*, pp. 506–514, Springer, 2016.

[26] Y. Zhang, T. Xu, and J. Ma, "Image categorization using non-negative kernel sparse representation," *Neurocomputing*, 2017.

[27] P. O. Hoyer, "Non-negative sparse coding," in *IEEE Workshop on Neural Networks for Signal Processing*, pp. 557–565, 2002.

[28] J. Zhou, S. Zeng, and B. Zhang, "Kernel nonnegative representation-based classifier," *Applied Intelligence*, 2021.

[29] J. Mercer, "Functions of positive and negative type, and their connection with the theory of integral equations," *Philosophical Trans. of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 1909.

[30] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, 2008.

[31] S. Shekkizhar and A. Ortega, "NNK-Means: Dictionary learning using non-negative kernel regression," *arXiv preprint arXiv:2110.08212*, 2021.

[32] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," tech. rep., Computer Science Department, Technion, 2008.

[33] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Intl. Conf. on Mach. Learning*, PMLR, 2020.