
Frank-Wolfe Algorithm for Exemplar Selection

Gary Cheng
UC Berkeley

Armin Askari
UC Berkeley

Laurent El Ghaoui
UC Berkeley

Kannan Ramchandran
UC Berkeley

Abstract

In this paper, we consider the problem of selecting representatives from a data set for arbitrary supervised/unsupervised learning tasks. We identify a subset S of a data set A such that 1) the size of S is much smaller than A and 2) S efficiently describes the entire data set, in a way formalized via auto-regression. The set S , also known as the exemplars of the data set A , is constructed by solving a convex auto-regressive version of dictionary learning where the dictionary and measurements are given by the data matrix. We show that in order to generate $|S| = k$ exemplars, our algorithm, Frank-Wolfe Sparse Representation (FWSR), only requires $\approx k$ iterations with a per-iteration cost that is quadratic in the size of A , an order of magnitude faster than state of the art methods. We test our algorithm against current methods on 4 different data sets and are able to outperform other exemplar finding methods in almost all scenarios. We also test our algorithm qualitatively by selecting exemplars from a corpus of Donald Trump and Hillary Clinton’s twitter posts.

non-exhaustive list of techniques include PCA (Wold et al., 1987), random projections (Candes and Tao, 2006), generalized discriminant analysis (Mika et al., 1999), local linear embeddings (Roweis and Saul, 2000) and non-negative matrix factorization (Lee and Seung, 1999).

A related problem is reducing the object-space, or reducing the number of data points in a data set. Exemplar selection is aimed at exactly solving this problem: finding a minimal set of representatives, or *exemplars*, of the data set that effectively represent the rest of the data points. This not only provides an efficient method of summarizing large data sets for a human observer, but also provides supervised/unsupervised learning algorithms with a smaller data set in place of the original. In a setting where multiple supervised tasks have to be done on the same large data sets, the extra cost of selecting exemplars first can be negligible compared to the speed up in the training of future models, with only a modest degradation in performance.

Exemplar selection methods can be separated into two groups: wrapper methods and filter methods. The former selects exemplars based on the accuracy obtained by a classifier, whereas the latter approach selects exemplars based on an objective function which is not based on a classifier (Olvera-López et al., 2010). In this paper, we work with filter methods.

1 INTRODUCTION

1.1 Overview

In the areas of computer vision, signal processing and machine learning, it has become important not only to improve the performance of models, but also to be able to train these models quickly and efficiently. This has motivated areas like dimensionality reduction that help save on computational resources and memory requirements by compressing the feature space; a

1.2 Paper contribution

Existing filter methods for exemplar selection are either fast but do not perform well on different learning tasks, or perform well on learning tasks but do not scale well with larger data sets. In this work, we strike a balance between the quality and the speed at which we select exemplars. We propose a Frank-Wolfe based algorithm, Frank-Wolfe Sparse Representation (FWSR), that solves the auto-regressive exemplar selection problem of finding a sparse subspace of the data that can efficiently span the entire data set. With our method, we are able to attain state-of-the-art results in supervised learning tasks while being at least ten times as fast and cutting down on the iteration cost by an order of magnitude. We employ our method on

the unsupervised learning task of selecting exemplars from the entire history of Donald Trump and Hillary Clinton’s tweets. Then, we consider the unsupervised problem of selecting exemplars as cluster representatives in a synthetic Gaussian data set. Finally, we compare our method against other filter methods in a supervised setting on four different data sets: Extended YaleB, 20 Newsgroup, credit card fraud, and EMNIST.

2 RELATED LITERATURE

The filter method of finding exemplars based on a sparse, auto-regressive model (SMRS) was introduced by Elhamifar et al. (2012). They justify their model using the self-expressiveness property which has been studied for subspace clustering and low-rank representations. Extensions of this work include Sparse Subspace Clustering (SSC) which uses the learned coefficient matrix as an affinity matrix in spectral clustering (Ng et al., 2001). SMRS and its variants such as D-SMRS (Dornaika and Aldine, 2015) and Kernelized SMRS (Dornaika et al., 2016) currently attain state of the results for exemplar selection on different supervised learning tasks. The aforementioned methods use the Alternating Direction Method of Multipliers (ADMM) to solve an optimization problem that requires a one-time inversion of a dense matrix, as well as dense matrix multiplications at every iteration; the complexity is cubic in the number of data points, making these methods unsuitable for even moderately-sized data sets. You et al. (2016) try to address this concern by introducing a greedy Orthogonal Matching Pursuit relaxation of SSC. However, in doing so, they remove the group lasso penalty from the objective and shift their focus to clustering, as opposed to exemplar selection.

The auto-regressive formulation of exemplar selection can be thought of as a specific instance of dictionary learning. Methods like K-SVD (Aharon et al., 2006) attempt to solve the regression problem

$$\min_{D, X} \|A - DX\|_F^2 : \forall i, \|X_{(i)}\|_0 \leq k,$$

where A is the data matrix, and $X_{(i)}$ represents the i th column of X . In the setting of exemplar selection, we restrict the dictionary D to be the data matrix A . SMRS and other similar works (Esser et al., 2012) can be seen as solving this particular instance of dictionary learning. Note that in K-SVD, simply replacing D by A generates the trivial solution $X = I$, motivating the introduction of the group lasso constraint.

An instance of exemplar selection that is not formulated as an auto-regressive optimization is k -medoids

(Kaufman and Rousseeuw, 1987). Unlike k -means, k -medoids requires that the centers of the clusters be data points, which can be treated as exemplars of the k classes. However, k -medoids in general does not converge to the global optimum and does not necessarily cluster points lying on the same subspace together.

There are other indirect methods whose solutions can be interpreted in the context of exemplar selection. For instance, Rank Revealing QR Decomposition (RRQR) (Hong and Pan, 1992) selects data points based on a permutation matrix of the data which gives a well conditioned submatrix. The Column Subset Selection Problem (CSSP) is also related to selecting exemplars. The problem is to identify k columns of a matrix A , called C , which minimize $\|A - P_C A\|_F$ where P_C is the projection operation onto C . Other ways of addressing this problem include randomized sketching methods like CUR decomposition (Drineas et al., 2008); Boutsidis et al. (2009) analyze a variant that combines ideas from CUR decomposition with RRQR.

There is also another body of work related to exemplar finding called cores set construction. Cores set construction is in the same spirit as exemplar selection and has had recent success in the context of PCA and k -means (Feldman et al., 2013, 2016). Despite this, these cores set construction methods are wrapper methods and it is unclear how to generalize their construction to arbitrary learning problems in a frequentist setting (Campbell and Broderick, 2017). We instead focus on filter methods, which are problem-agnostic.

3 PROBLEM FORMULATION

3.1 Notation

Let $\|\cdot\|_F$ be the Froebenius norm. Let $X^{(i)}$ and $X_{(i)}$ denote the i -th row and column, respectively, of a matrix X ; X_{ij} denotes the (i, j) th entry of a matrix X . For $q > 1$, we refer to $\sum_{i=1}^n \|X^{(i)}\|_q$ as the “group lasso” norm. We denote our feature matrix as $A \in \mathbb{R}^{d \times n}$ where each column represents a data point in d -dimensional space. We define the Gram matrix $K := A^T A$. Finally, $\mathbf{1}$ denotes a vector of ones of appropriate dimension.

3.2 Objective

We formulate exemplar selection as an auto-regressive version of the dictionary learning problem, where the dictionary is the data set itself. Given a data matrix $A \in \mathbb{R}^{d \times n}$ with n d -dimensional data points, the learn-

ing problem becomes

$$\begin{aligned} \min_X J(X) &:= \|AX - A\|_F^2 + \zeta^2 \|X^\top \mathbf{1} - \mathbf{1}\|_2^2 \quad (1) \\ \text{s.t. } \sum_{i=1}^n \|X^{(i)}\|_q &\leq \beta, \end{aligned}$$

where β, ζ are hyper-parameters. We assume that A is centered columnwise to remove the need for a bias term. The row-sparsity of the solution is controlled by β . Intuitively, (1) identifies a sparse subset of the data points that best span (*i.e.* represent) the entire data set. Note that the group lasso constraint with sufficiently small β not only ensures that the trivial solution of $X = I$ is not in the feasible set, but also encourages row-sparsity in the solution X . After solving (1), we select the data points corresponding to the non-zero rows of the X matrix as our exemplars. These exemplars then form our new training set which we use for different supervised/unsupervised learning tasks.

Whenever $q > 1$, the group lasso constraint encourages row sparsity of the X matrix. Because empirically $q = 2$ outperforms $q = \infty$, for the remainder of the paper we set $q = 2$.

The second term in (1) corresponds to a constraint introduced by Elhamifar et al. (2012) in order to enforce translation invariance on the data matrix A ; here we use a relaxed, penalized version of that constraint in order to make our algorithm simpler; ζ is the corresponding penalty parameter. In some instances, $\zeta = 0$ proves to be the most fitting; other times $\zeta = 100$ perform best.

4 CONTRIBUTIONS

We propose a Frank-Wolfe algorithm (Frank and Wolfe, 1956) for solving (1) that is faster than other exemplar selection methods and whose selected exemplars enjoy higher test set accuracies when trained on a variety of data sets. Frank Wolfe optimizes an objective over a closed, convex set by moving towards the minimizer of its linear approximation at each iteration while still remaining in the domain. In the case when the vertices of the feasible set are sparse, the Frank-Wolfe algorithm produces sparse iterates. After a fixed cost of $\mathcal{O}(n^2d)$, where typically $n \gg d$, is used to calculate the Gram Matrix, we can identify k exemplars after running our algorithm $\approx k$ iterations, with the cost of each iteration being $\mathcal{O}(n^2)$. This is achieved by noting that at each iteration we can make a rank 1 update to the gradient. The algorithm terminates when **NumExemplars**(X), the row sparsity of the iterate X , is equal to the number of desired exemplars k . Then **PickExemplars**(A, X, k) selects

Algorithm 1 Frank-Wolfe Sparse Representation

```

1: procedure FWSR( $A, K, k, \beta, \zeta$ )
2:    $\tilde{K} = K + \zeta^2 \mathbf{1}\mathbf{1}^\top$ 
3:    $X_0 = 0$ 
4:    $E = \text{NumExemplars}(X)$ 
5:    $\gamma_0, t = 0, 0$ 
6:   while  $E < k$  do
7:      $(\tilde{K}X)_t = (1 - \gamma_t)(\tilde{K}X)_{t-1} + \gamma_t \tilde{K}S_{t-1}$ 
8:      $\nabla J_t = 2(\tilde{K}X)_t - 2\tilde{K}$ 
9:      $j = \arg \max_i \|(\nabla J_t)^{(i)}\|_2$ 
10:     $S_t^{(j)} = -\beta \frac{(\nabla J_t)^{(j)}}{\|(\nabla J_t)^{(j)}\|_2}$ 
11:     $D_t = S_t - X_t$ 
12:     $g_t = -\langle \nabla J_t, D_t \rangle$ 
13:    if  $g_t < \delta$  then break
14:    end if
15:     $\gamma_t = \min\left(1, \frac{\text{Tr}(D_t^\top (\tilde{K} - \tilde{K}X_t))}{D_t^\top \tilde{K} D_t}\right)$ 
16:     $X_{t+1} = X_t + \gamma_t D_t$ 
17:     $E = \text{NumExemplars}(X_{t+1})$ 
18:     $t = t + 1$ 
19:  end while
20:  return PickExemplars( $A, X_t, k$ )
21: end procedure

```

the columns of A that correspond to the dense rows of X . Algorithm 1 outlines the method. Note that Algorithm 1 relies only on the Gram matrix K and not on A directly, which implies that we can trivially kernelize our algorithm.

SMRS and its variants that are able to attain state of the art results on different supervised learning tasks use ADMM. In addition to also requiring the $\mathcal{O}(n^2d)$ calculation of the Gram Matrix K , ADMM requires a dense matrix inversion before the first iteration and a dense matrix multiplication in every subsequent iteration, resulting in a $\mathcal{O}(n^3)$ cost per iteration which is inefficient for large n . Additionally, since the iterates generated by ADMM are not sparse, tuning the sparsity hyperparameter and the terminating tolerance of the algorithm is required in order to obtain a good set of exemplars. Since FWSR generates sparse iterates, the method is more interpretable. Moreover, we are able to terminate our algorithm whenever we have reached the desired number of exemplars, or have achieved a certain error bound. A convergence rate for our algorithm is given in Theorem 1.

Theorem 1. *Let X_t be the iterate generated by Algorithm 1 and let $\beta > n$. Then X_t satisfies*

$$\|\tilde{A}X_t - \tilde{A}\|_F^2 \leq \frac{4\beta^2 C^2 \lambda_{\max}(\tilde{K}) \nu^2}{C^2 \nu^{2-2t} + t}.$$

where $\tilde{A} := [\zeta \mathbf{1} \ A^\top]^\top$, $\tilde{K} := \tilde{A}^\top \tilde{A}$ and $C > 0, \nu \in (0, 1)$ are constants. For sufficiently large t , the convergence rate is linear.

The proof of Theorem 1 can be found in [Appendix A](#). One notable concern is the scenario where we are interested in selecting k exemplars but the algorithm converges to a solution with row-sparsity $r < k$. With Theorem 1, by setting $\beta > n$, either the algorithm converges to a dense solution such as the identity, in which case we can greedily terminate the algorithm once the iterate X_t is k row-sparse, or the algorithm converges to a $r < k$ row-sparse solution, in which case we select only r exemplars. In this case, Theorem 1 shows that we do not need to run the algorithm for a large number of steps.

5 EMPIRICAL RESULTS

We run three sets of experiments: two in an unsupervised learning setting and one in a supervised setting. The first is a qualitative experiment of obtaining a summary of the corpus of Donald Trump and Hillary Clinton tweets. The second experiment is on a synthetic Gaussian data set of k artificial clusters; the exemplar selection algorithms are tasked with picking an exemplar from each cluster. The last set of experiments are performed on labeled data sets where we compare our algorithm to other exemplar selection methods. We compare our algorithm against 4 different data reduction methods: random subset selection, SMRS, k -medoids, and RRQR. Each algorithm identifies k exemplars from each training class to be used as the training set for a classification model. The exemplar selection algorithms are then compared against one another based on the end to end data reduction time, training time, and validation accuracy. We do not consider D-SMRS and Kernelized SMRS since both introduce additional hyperparameters which, coupled with their runtimes on relatively larger data sets, make cross validation very computationally intensive.

In our algorithm, the effect of β is highly dependent on the number of data points, n . In an effort to disentangle this dependency, we parameterize β as n/α where α is a hyperparameter that we choose; typically $\alpha \in [0.5, 50]$. Additionally, to enforce that SMRS selects no more than k exemplars, we choose the data points corresponding to the k largest ℓ_2 norm rows of the returned coefficient matrix X as exemplars as done in [Elhamifar et al. \(2012\)](#).

5.1 Donald Trump & Hillary Clinton Tweets

We identify 10 exemplars from 10,000 randomly subsampled tweets, which corresponds to roughly one third of Donald Trump’s tweet corpus as of 9/26/18 (35,273 tweets in total) ([Trump, 2018](#)). We also identify 10 exemplars from the entirety of Hillary Clinton’s

tweets as of 10/02/18 (7,910 tweets in total). We preprocess each tweet by removing all twitter handles, numbers, urls, punctuation, and English stop words specified by sklearn ([Pedregosa et al., 2011](#)) and use a Count Vectorizer for our data matrix. Then using hyper-parameters of $(\alpha, \zeta) = (10, 0)$, FWSR selects 10 exemplar tweets from each corpus; this process takes about 10 seconds for both corpora. We show 3 notable tweets of the 10 exemplars that FWSR obtained from each twitter corpus in [Figure 1](#); the remaining seven exemplars can be found in [Appendix B](#). The content of each tweet provides a sampling of President Trump’s various positions, including his stances on: alleged Russia collusion, the Trump Dossier, fake news, border security, trade wars, Barack Obama, and Hillary Clinton. Hillary Clinton’s tweets are represented by her positions on equal rights, social security, and Donald Trump. Qualitatively, both summaries seem consistent with the positions and personas the two public figures have displayed.

5.2 Synthetic Data

To quantitatively demonstrate the performance of our algorithm, we first test it on synthetic data. We generate 1000 data points and disperse them evenly between k Gaussian clusters in 1500 dimensional space with covariance $\Sigma = 20^2 I$ using sklearn’s `make_blob` function. We then use FWSR, SMRS, RRQR, and k -medoids to find k exemplars from these k clusters and then calculate the fraction of the k clusters that were recovered. [Figure 2](#) plots the results.

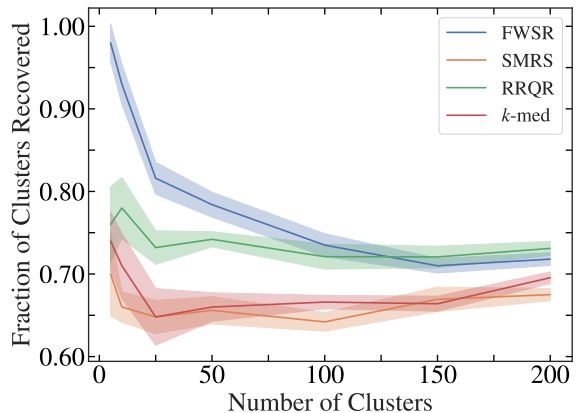


Figure 2: Average fraction of cluster centers recovered versus number of clusters on isotropic Gaussian data for FWSR, SMRS, RRQR, and k -medoids. The shaded regions represent one standard deviation over 10 experiments per cluster.



Figure 1: Six tweets that FWSR selected from Donald Trump's and Hillary Clinton's twitter accounts

Without any hyperparameter tuning, we set the sparsity hyperparameter $\alpha = 20$ for SMRS, which is in the range recommended by the authors, and $(\alpha, \zeta) = (10, 0)$ for FWSR. In Figure 2, it is clear that with only a few clusters, FWSR is able to recover exemplars from a large percentage of unique clusters compared to the other methods. In particular, for 5 clusters, FWSR is able to generate 5 exemplars that effectively recover all the unique clusters whereas the other methods on average can only recover exemplars from 3 or 4 of the unique clusters. As the number of clusters increases, all the methods tend to converge to a recovery rate between 0.70 and 0.75.

Although not shown, when the magnitude of the covariance is lowered, both FWSR and RRQR are able to recover the number of clusters with nearly 100% accuracy while SMRS and k -medoids had a recovery rate around 70%. This seems to indicate that for tightly clustered points, FWSR and RRQR are able to identify all the clusters while SMRS and k -medoids still

confuse the clusters with one another.

5.3 Labeled Data sets

Next, we compare FWSR against other exemplar selection methods to show how it compares as a pre-processing step for four different classification data sets. We use the exemplar selection methods to effectively reduce the size of the training set, and then feed the new reduced training set to a classifier and compare its accuracy against using the full training set. We consider 3 different classifiers: Balanced Linear Support Vector Machines (SVM), k -Nearest Neighbors (k -NN), and Multinomial Naive Bayes (MNB) all implemented using scikit-learn. We compare FWSR and Kernelized FWSR (K-FWSR) with a RBF kernel against random subset selection, SMRS, k -medoids, and RRQR. The same machine was used for all of the pre-processing methods and classification model training. The code used for each classification model is the same across different data reduction algorithms. The

Matlab code for SMRS is taken directly from Elhamifar et al. (2012).

The following data sets were used:

1. **Extended Yale Face Database B** (E-YaleB): 38 classes and 2,414 data points per class in 1,024 dimensional space (Georghiades et al., 2001). Each class corresponds to a human subject and the data points populating the class are different images of the subject. We take ≈ 13 data points from each class to form a validation data set.
2. **20 Newsgroups** (News20): 20 classes, with 11,314 term frequency-inverse document frequency (tf-idf) vectors in the training set and 7,532 tf-idf vectors in the validation set (Lang, 1995). Each class corresponds to text documents of a certain topic. We do not center and normalize to maintain sparsity of the data set, however we do reduce the dimension of the data set from 101,631 to 50,000 (5000 for our experiments with k -NN) using sklearn’s `feature_selection.chi2` function.
3. **Credit Fraud** (Credit): 2 imbalanced classes with 492 in the fraud class and randomly subsampled 5000 in the non-fraud class. We did stratified sampling of 80% to form a training set of size 4394 and a validation set of size 1098 (Dal Pozzolo et al., 2015). Because fraud data is rare, we only do exemplar selection on the non-fraud class. Because of the huge imbalance in class size, we use F1-scores instead of accuracy as our measure of quality of exemplars selected.
4. **EMNIST ByClass** (EMNIST): 62 classes in 784 dimensional space corresponding to numbers, upper case letters, and lower case letters (Cohen et al., 2017). To make cross validation viable, we sub-sample each class to make sure there are at most 5000 data points per class. The training set size is 253,523 and the validation set size is 116,323

Centering and normalizing the training and validation sets separately was a pre-processing step and was used for the Credit Fraud data set and left as a hyperparameter choice for the E-YaleB data set. It was not done for the News20 data set in order to preserve the sparsity in the data, and it was not done for E-MNIST because we empirically observed poor validation set performance for almost all the exemplar finding methods. A constraint often used in (1) with image or text data sets is $X \geq 0$ since it has real life interpretations (Esser et al., 2012). We noticed for our experiments, introducing this constraint did not significantly alter

the performance of any method (in fact it lowered the performance of most methods) and as a result we chose not to display results for this setting.

For each of the m classes in a labeled training data set, we select a fixed number k exemplars via an exemplar selection algorithm for each class. These mk exemplars are then used to train a classifier. We cross validate the hyper-parameters of the exemplar selection method and the classifier by comparing validation accuracies for the exemplar-trained classifiers. We repeat this process over (nearly) all combinations of data set, exemplar selection algorithm, and classification model. For non-deterministic methods such as random subset selection and k -medoids, we run the exemplar finding algorithm 20 times and average our results, optimizing hyper-parameters for each run. We display the best cross validation accuracies in Table 1. We also display the total time it takes for each algorithm to find the exemplars and train a Linear SVM Model in Table 2. Using the SVM classifier and the EMNIST data set, we compare FWSR to SMRS by plotting the end-to-end training time (i.e. sum of data reduction time and classifier training time) and test set accuracy as a function of the number of exemplars in Figure 3.

In Figure 3, FWSR is able to outperform SMRS while being significantly faster. Note as well that for essentially any number of exemplars, FWSR outperforms all the methods as seen in Figure 3 with the blue curve being consistently above the other curves. For the curve measuring the run time of the algorithms, the run time of FWSR grows linearly with the number of exemplars since the algorithm terminates once it has reached k exemplars. Note for a small fraction of exemplars (between 1% and 10%), FWSR is between 10 and 1000 times faster than SMRS.

For E-YaleB, FWSR is able to achieve only a 9% degradation in the performance of an SVM classifier when only using 13.7% of the original data set while SMRS has a final test set accuracy that is 3% lower than FWSR. Note that while RRQR is the top performing method with an SVM classifier, FWSR’s accuracy is only 0.5% lower. However, using k -NN all the methods seem to significantly under perform compared to the entire data set.

For News20, using a MNB training model, FWSR is able to outperform all other exemplar finding methods and outperforms RRQR by more than 20%. This is also seen in the SVM model where FWSR outperforms the other exemplar finding methods. As before, k -NN does not seem to be the best training algorithm for this data set. In fact, using the entire data set leads to a classification accuracy of 27% while using a subsampled version of the data, RRQR, K-FWSR and

	E-YALEB		News20			CREDIT		EMNIST
	SVM	k -NN	SVM	k -NN	MNB	SVM	k -NN	SVM
ALL	0.994	0.773	0.703	0.266	0.703	0.892	0.911	0.639
RANDOM	0.811	0.412	0.550	0.213	0.541	0.835	0.284	0.344
FWSR	0.903	0.473	0.601	0.343	0.625	0.885	0.182	0.515
K-FWSR	0.824	0.515	0.584	0.305	0.618	0.887	0.876	0.400
k -MED	0.851	0.480	0.567	0.162	0.566	0.848	0.434	0.461
RRQR	0.908	0.376	0.375	0.313	0.404	0.557	0.165	0.241
SMRS	0.876	0.456	0.568	0.274	0.576	0.812	0.167	–

Table 1: Accuracies for different exemplar selection algorithms using different training algorithms on 4 different data sets. We select 7 exemplars/class for E-YaleB, 50 exemplars/class for New20, 10 exemplars in the non-fraud class for Credit, and 10 exemplars/class for EMNIST, corresponding to 13.7%, 8.8%, 0.2%, and 0.2% of the data sets respectively. SMRS is not capable of running efficiently on the EMNIST data set due to its large size, so we use – as a placeholder. Bolded numbers in each column denote the best accuracy attained among all exemplar finding algorithms.

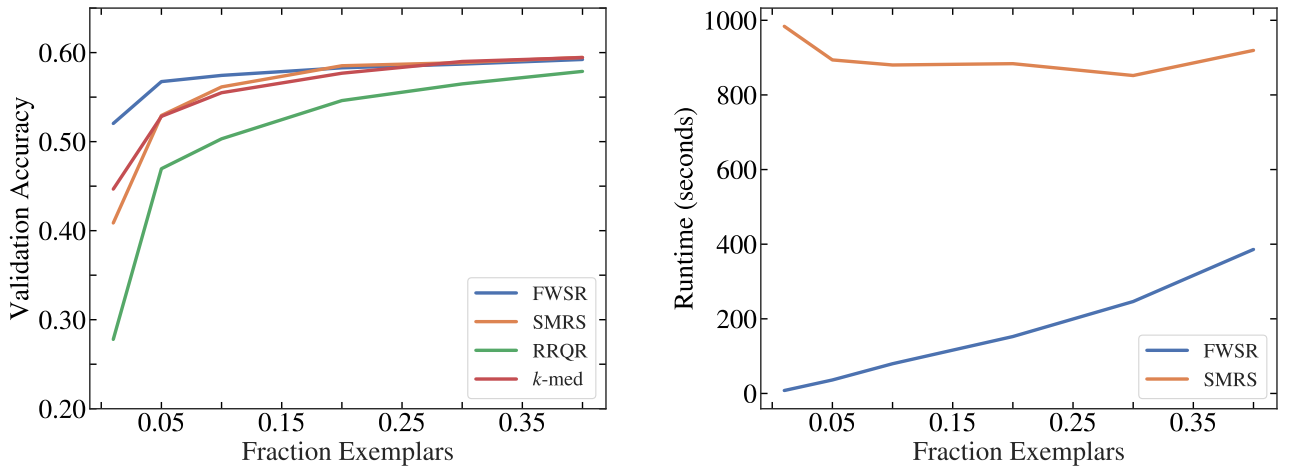


Figure 3: Validation accuracy and run time versus number of exemplars for EMNIST. The EMNIST data set was subsampled such that each class had at most 1000 data points so that SMRS could run in a reasonable time. Fraction exemplars denotes the number of exemplars as a percentage of the 1000 data points in each class. Not displayed: the run time for RRQR and k -medoids was ≤ 5 seconds along the abscissa.

FWSR are able to achieve a *higher* accuracy.

For the Credit Fraud data set, k -NN performs well compared to before. However, all the exemplar finding methods give extremely poor performance *except* for K-FWSR which is able to achieve an accuracy that is only 3.5% less than using the entire data set. For SVM, FWSR and K-FWSR achieve the highest accuracy, only under performing the entire data set by 0.7%. However, in this case, random subset selection does well and outperforms both RRQR and SMRS. It is interesting to note that while K-FWSR performs well on average, it performs especially well on the Credit Fraud data set which seems to imply there is some underlying structure in the data set that the other exemplar finding algorithms are unable to capture.

For EMNIST, using an SVM, we see that FWSR is able to outperform all the other exemplar finding methods. In particular, the performance of RRQR suffers and is even worse than random selection.

While Table 1 shows that FWSR is competitive and can outperform most exemplar finding algorithms in different settings, Table 2 shows that the algorithm also has a fast end-to-end training time.

Note that for all the data sets SMRS is the slowest algorithm while FWSR strikes a balance. On E-YaleB, it is 5 times faster than training on the entire data set and is in fact the fastest data reduction technique. However, on the Credit data set it is the second slowest but the best performing exemplar finding algorithm.

	E-YALEB	NEWS20	CREDIT	EMNIST
ALL	5.42	2.11	0.27	18133.05
FWSR	1.29	8.24	7.51	159.48
k -MED	8.21	5.79	0.22	17.39
RRQR	3.43	70.16	0.01	23.85
SMRS	14.80	3331.44	837.07	–

Table 2: Total reduction time and training time in seconds for an SVM across all the exemplar finding methods. Note that ALL has no reduction time and simply represents the training time of the SVM on the entire data set. Across 20 trials, the standard deviation of k -MED was 10.592, 0.097, 0.011, and 0.102 seconds in order from E-YALEB to EMNIST.

On EMNIST, we see that FWSR is more than 100 times faster than training on the entire data set. While RRQR is 7 times faster than FWSR, its accuracy as shown in Table 1 is 27% worse than FWSR. In this regime, it makes sense to employ our algorithm so that the training time for future machine learning models can be significantly reduced in exchange for a one time cost of finding the exemplars.

6 Conclusion

Finding exemplars within a training set not only helps summarize large or difficult-to-interpret data sets, but also helps reduce the training time for different types of supervised/unsupervised learning algorithms. In this paper, we proposed Franke-Wolfe Sparse Representation, an algorithm for solving the auto-regressive version of dictionary learning that helps identify a subset of the data that efficiently describes the entire data set. We show that our method is able to cut down the per iteration cost of state of the art methods by an order of magnitude and exhibits linear convergence. We employ our algorithm on a variety of data sets and show the computational gain as well as its performance against other exemplar finding algorithms.

Possible extensions of this work include looking at randomized variants of FWSR that consider a stochastic algorithm. Additionally, it would be interesting to study and compare the robustness properties of these different exemplar finding algorithms and methodically construct robust counterparts to the traditional algorithms. Another avenue of exploration would be in clustering. We could apply our algorithm with group lasso parameter $q = 1$ to the problems addressed in Elhamifar and Vidal (2013); in this setting, we could use Away Step and Pairwise variants of Frank Wolfe (Lacoste-Julien and Jaggi, 2013) that achieve a linear rate of convergence independent of β and strong convexity of the objective.

References

- Aharon, M., Elad, M., Bruckstein, A., et al. (2006). K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311.
- Boutsidis, C., Mahoney, M. W., and Drineas, P. (2009). An improved approximation algorithm for the column subset selection problem. pages 968–977.
- Campbell, T. and Broderick, T. (2017). Automated scalable bayesian inference via hilbert core-sets. *CoRR*, abs/1710.05053.
- Candes, E. J. and Tao, T. (2006). Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12):5406–5425.
- Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. (2017). Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*.
- Dal Pozzolo, A., Caelen, O., Johnson, R. A., and Bontemp, G. (2015). Calibrating probability with undersampling for unbalanced classification. In *Computational Intelligence, 2015 IEEE Symposium Series on*, pages 159–166. IEEE.
- Dornaika, F. and Aldine, I. K. (2015). Decremental sparse modeling representative selection for prototype selection. *Pattern Recognition*, 48(11):3714–3727.
- Dornaika, F., Aldine, I. K., and Hadid, A. (2016). Kernel sparse modeling for prototype selection. *Knowledge-Based Systems*, 107:61–69.
- Drineas, P., Mahoney, M., and Muthukrishnan, S. (2008). Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881.
- Elhamifar, E., Sapiro, G., and Vidal, R. (2012). See all by looking at a few: Sparse modeling for finding representative objects. pages 1600–1607.
- Elhamifar, E. and Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781.
- Esser, E., Moller, M., Osher, S., Sapiro, G., and Xin, J. (2012). A convex model for nonnegative matrix factorization and dimensionality reduction on physical space. *IEEE Transactions on Image Processing*, 21(7):3239–3252.
- Feldman, D., Schmidt, M., and Sohler, C. (2013). Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In *Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA

- '13, pages 1434–1453, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- Feldman, D., Volkov, M., and Rus, D. (2016). Dimensionality reduction of massive sparse datasets using coresets. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 2766–2774. Curran Associates, Inc.
- Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110.
- Georghiades, A., Belhumeur, P., and Kriegman, D. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660.
- Hong, Y. P. and Pan, C.-T. (1992). Rank-revealing qr factorizations and the singular value decomposition. *Mathematics of Computation*, 58(197):213–232.
- Kaufman, L. and Rousseeuw, P. (1987). *Clustering by means of medoids*. North-Holland.
- Lacoste-Julien, S. and Jaggi, M. (2013). An affine invariant linear convergence analysis for frank-wolfe algorithms. *arXiv preprint arXiv:1312.7864*.
- Lang, K. (1995). Newsweeder: Learning to filter net-news. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.
- Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788.
- Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Mullers, K.-R. (1999). Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop.*, pages 41–48. Ieee.
- Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS’01, pages 849–856, Cambridge, MA, USA. MIT Press.
- Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., and Kittler, J. (2010). A review of instance selection methods. *Artificial Intelligence Review*, 34(2):133–143.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.
- Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.
- Trump (2018). Trump twitter archive. <http://trumptwitterarchive.com/>.
- Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.
- You, C., Robinson, D., and Vidal, R. (2016). Scalable sparse subspace clustering by orthogonal matching pursuit.

Supplementary material

Appendix A Linear Convergence Proof

Appendix A.1 Prerequisites

We prove linear convergence for a not strongly convex $J(X)$ (i.e. when A is not full column rank) for $\beta > n$. For this appendix, we implicitly append a row of ζ 's to the top of A (i.e. we redefine $A := \hat{A}$) to accommodate the translational invariance constraint. We begin this proof by redefining some variables:

$$\begin{aligned} \hat{A} &:= \begin{bmatrix} A & 0 \\ & \ddots \\ 0 & A \end{bmatrix} & \hat{K} &:= \hat{A}^T \hat{A} \\ w_t &:= \begin{bmatrix} X_t^{(1)T} \\ \vdots \\ X_t^{(n)T} \end{bmatrix} & \hat{1}_i &:= \begin{cases} 1 & \text{if } i \bmod (d+1) = 0 \\ 0 & \text{o.w.} \end{cases} \\ s_t &:= \arg \min_s \langle \nabla J, s \rangle & d_t &:= s_t - w_t \\ &= \arg \max_s \langle \hat{A}s - \hat{A}w, \hat{A}\hat{1} - \hat{A}w \rangle \\ &= \arg \max_s (s - w)^T \hat{K} (\hat{1} - w) \end{aligned}$$

Observe that our objective and gradient using this notation can be rewritten as:

$$\begin{aligned} J(w_t) &:= \|AX_t - A\|_F^2 \\ &= \|\hat{A}w_t - \hat{A}\hat{1}\|_2^2 \\ \nabla J(w_t) &:= 2\hat{K}(w - \hat{1}) \end{aligned}$$

The next cost as a function of the previous cost is:

$$J(w_{t+1}) = J(w_t + \gamma d_t) \tag{2}$$

$$= (w_t + \gamma d_t - \hat{1})^T K (w_t + \gamma d_t - \hat{1}) \tag{3}$$

$$= J(w_t) + \gamma^2 d_t^T K d_t + 2\gamma(w_t - \hat{1})^T K d_t \tag{4}$$

$$\frac{\partial J}{\partial \gamma} = 2\gamma d_t^T K d_t + 2(w_t - \hat{1})^T K d_t = 0 \tag{5}$$

$$\gamma_t = \frac{d_t^T K (\hat{1} - w_t)}{d_t^T K d_t} \tag{6}$$

$$J(w_{t+1}) = J(w_t) - \frac{(d_t^T K (\hat{1} - w_t))^2}{d_t^T K d_t} \tag{7}$$

The last line is true due to lemma 1 which is introduced below.

We will also be using the following helpful lemmas:

Lemma 1. For $\beta > n$, the optimal step size satisfies: $0 \leq \gamma_t \leq 1$ for all t

Proof. Suppose $\gamma_t < 0$, this implies that $(s_t - w_t)^T \hat{K} (\hat{1} - w_t) < 0$, but because s maximizes the quantity (fact 1), it must be that:

$$0 > (s_t - w_t)^T \hat{K} (\hat{1} - w_t) \geq (\hat{1} - w_t)^T \hat{K} (\hat{1} - w_t) \geq 0$$

which is a contradiction.

Suppose that $\gamma_t > 1$, this implies that $(s_t - w_t)^T \hat{K}(\hat{1} - w_t) > d_t^T \hat{K} d_t^T$ (fact 2). However, in using $\gamma = 1$ in equation (4):

$$\begin{aligned} 0 &\leq J(w_{t+1}) = J(w_t) + 2d_t^T \hat{K}(w_t - \hat{1}) + d_t^T \hat{K} d_t \\ &< J(w_t) + d_t^T \hat{K}(w_t - \hat{1}) \\ &= (\hat{1} - w_t)^T \hat{K}(\hat{1} - w_t) + d_t^T \hat{K}(w_t - \hat{1}) \leq 0 \end{aligned}$$

where the first inequality comes from fact 2, and the second inequality comes from fact 1. This is also a contradiction. \square

Lemma 2. For $\beta > n$ and $n > d$, $\hat{A}\hat{1}$ is in the interior of the domain; furthermore, there exists $r > 0$ such that

$$Aw + (\|\hat{A}\hat{1} - \hat{A}w\|_2 + r) \frac{\hat{A}\hat{1} - \hat{A}w}{\|\hat{A}\hat{1} - \hat{A}w\|_2}$$

for all w is in the interior of the domain as well

Proof. The open mapping theorem proves the first point because \hat{A} is surjective. Given that the first point is true, then the second point arises from the fact that

$$\hat{A}w + (\|\hat{A}\hat{1} - \hat{A}w\|_2 + r) \frac{\hat{A}\hat{1} - \hat{A}w}{\|\hat{A}\hat{1} - \hat{A}w\|_2} = \hat{A}\hat{1} + r \frac{\hat{A}\hat{1} - \hat{A}w}{\|\hat{A}\hat{1} - \hat{A}w\|_2}$$

Since, there must exist an open ball around $\hat{A}\hat{1}$, there must exist an r such the above is true. \square

Lemma 3. The logistic equation

$$x_{n+1} = \alpha x_n (1 - x_n)$$

for $x_0, \alpha \in [0, 1]$ satisfies

$$\forall n \in \mathbb{N}, x_n \leq \frac{x_0}{\alpha^{-n} + x_0 n}$$

Proof. This is Lemma A.6 from (Campbell and Broderick, 2017). The proof of this lemma can be found in Appendix A of the aforementioned paper. \square

Appendix A.2 Main Proof

Starting with equation (7):

$$\begin{aligned} J(w_{t+1}) &= J(w_t) - \frac{(d_t^T \hat{K}(\hat{1} - w_t))^2}{d_t^T \hat{K} d_t} \\ &= J(w_t) \left(1 - \frac{1}{(w_t - \hat{1})^T \hat{K}(w_t - \hat{1})} \frac{(d_t^T \hat{K}(\hat{1} - w_t))^2}{d_t^T \hat{K} d_t} \right) \\ &= J(w_t) \left(1 - \left[\frac{(\hat{A}s_t - \hat{A}w_t)^T (\hat{A}\hat{1} - \hat{A}w_t)}{\|\hat{A}s_t - \hat{A}w_t\|_2 \|\hat{A}\hat{1} - \hat{A}w_t\|_2} \right]^2 \right) \end{aligned}$$

Furthermore, observe that because $s_t = \arg \max_s \langle \hat{A}s - \hat{A}w, \hat{A}\hat{1} - \hat{A}w \rangle$, replacing $\hat{A}s$ with any other point is a lower bound, so using equation A.57, for some $r > 0$ we can replace As with $\hat{A}w + (\|\hat{A}\hat{1} - \hat{A}w\|_2 + r) \frac{\hat{A}\hat{1} - \hat{A}w}{\|\hat{A}\hat{1} - \hat{A}w\|_2}$

using lemma 2. Notice that this vector is in the range of \hat{A} .

$$\begin{aligned}
 \left(\frac{\hat{A}s_t - \hat{A}w_t}{\|\hat{A}s_t - \hat{A}w_t\|_2} \right)^T \frac{(\hat{A}\hat{1} - \hat{A}w_t)}{\|\hat{A}\hat{1} - \hat{A}w_t\|_2} &\geq \left(\frac{(\|\hat{A}\hat{1} - \hat{A}w\|_2 + r) \frac{\hat{A}\hat{1} - \hat{A}w}{\|\hat{A}\hat{1} - \hat{A}w\|_2}}{\|\hat{A}s - \hat{A}w\|_2} \right)^T \frac{\hat{A}\hat{1} - \hat{A}w}{\|\hat{A}\hat{1} - \hat{A}w\|_2} \\
 &= \frac{\|\hat{A}\hat{1} - \hat{A}w\|_2 + r}{\|\hat{A}s - \hat{A}w\|_2} \\
 &= \frac{\sqrt{J(w_t)} + r}{\|\hat{A}s - \hat{A}w\|_2} \\
 &\geq \frac{\sqrt{J(w_t)} + r}{2\beta\sqrt{\lambda_{\max}(\hat{K})}} \\
 &\geq \frac{\sqrt{J(w_t)} + r}{C2\beta\sqrt{\lambda_{\max}(\hat{K})}}
 \end{aligned}$$

for some $C > 1$. This implies that:

$$\begin{aligned}
 J(w_{t+1}) &\leq J(w_t) \left(1 - \left(\frac{\sqrt{J(w_t)} + r}{2\beta C \sqrt{\lambda_{\max}(\hat{K})}} \right)^2 \right) \\
 &\leq J(w_t) \left(1 - \frac{J(w_t)}{4\beta^2 C^2 \lambda_{\max}(\hat{K})} - \frac{r^2}{4\beta^2 C^2 \lambda_{\max}(\hat{K})} \right) \\
 &= J(w_t) \left(\nu^2 - \frac{J(w_t)}{4\beta^2 C^2 \lambda_{\max}(\hat{K})} \right)
 \end{aligned}$$

where $\nu^2 := 1 - \frac{r^2}{4\beta^2 C^2 \lambda_{\max}(\hat{K})}$

With this relationship we can derive the critical recursive relationship:

$$J(w_{t+1}) \leq J(w_t) \nu^2 \left(1 - \frac{J(w_t)}{4\beta^2 C^2 \lambda_{\max}(\hat{K}) \nu^2} \right) \quad (8)$$

$$x_{t+1} \leq x_t \nu^2 (1 - x_t) \quad (9)$$

$$x_t := \frac{J(w_t)}{4\beta^2 C^2 \lambda_{\max}(\hat{K}) \nu^2} \quad (10)$$

Claim 1.

$$0 \leq \nu^2 = 1 - \frac{r^2}{4\beta^2 C^2 \lambda_{\max}(\hat{K})} < 1$$

Proof. Recall that r is the magnitude of change in the direction of $\frac{\hat{A}\hat{1} - \hat{A}w}{\|\hat{A}\hat{1} - \hat{A}w\|_2}$. So this means we can always pick r small enough such that the numerator of $\frac{r^2}{4\beta^2 C^2 \lambda_{\max}(\hat{K})}$ is smaller than the denominator for all t (assuming the total number of iterations is finite) and that the new point that we chose is still in the interior of the domain; thereby making the claim true. \square

Claim 2. For a sufficiently large C ,

$$0 \leq x_t \leq 1$$

Proof. x_t is non-negative because all components of the fraction are nonnegative.

$$\begin{aligned} \frac{J(w_{t+1})}{4\beta^2 C^2 \lambda_{\max}(\hat{K}) \nu^2} &= \frac{\|\hat{A}(w - \hat{1})\|_2^2}{4\beta^2 C^2 \lambda_{\max}(\hat{K}) \nu^2} \\ &\leq \frac{1}{C^2 \nu^2} \end{aligned}$$

We can always pick a C large enough such that the quantity for all t (assuming the total number of iterations is finite) is less than 1. \square

Then finally in using lemma 3 with x_t from (10), the proof is complete. The convergence rate is:

$$\begin{aligned} x_t &\leq \frac{x_0}{\nu^{-2t} + x_0 t} \\ \frac{J(w_t)}{4\beta^2 C^2 \lambda_{\max}(\hat{K}) \nu^2} &\leq \frac{\frac{J(w_0)}{4\beta^2 C^2 \lambda_{\max}(\hat{K}) \nu^2}}{\nu^{-2t} + \frac{J(w_0)}{4\beta^2 C^2 \lambda_{\max}(\hat{K}) \nu^2} t} \\ &\leq \frac{C^{-2} \nu^{-2}}{\nu^{-2t} + C^{-2} \nu^{-2} t} \\ &= \frac{1}{C^2 \nu^{2-2t} + t} \end{aligned}$$

We can upper bound $J(w_0) \leq 4\beta^2 \lambda_{\max}(\hat{K})$ because $\frac{a}{at+b} = \frac{1}{t} \frac{a}{a+b/t}$ and $\frac{a}{a+c}$ is monotonically increasing in a for all $a, c \geq 0$.

This implies for sufficiently large t the convergence rate is linear:

$$\begin{aligned} J(w_t) &= \|AX_t - A\|_F^2 \\ &\leq \frac{4\beta^2 C^2 \lambda_{\max}(\hat{K}) \nu^2}{C^2 \nu^{2-2t} + t} \\ &\leq 2\beta^2 \lambda_{\max}(\hat{K}) \nu^{2t} \end{aligned}$$

Appendix B Donald Trump & Hillary Clinton Tweets



Here we display the remaining 7 exemplars of Donald Trump's and Hillary Clinton's tweets found using FWSR. "The Briefing"'s tweet below was retweeted by Hillary Clinton.

 **Donald J. Trump** 
@realDonaldTrump

.@RepClayHiggins has been a great help to me on Cutting Taxes, creating great new healthcare programs at low cost, fighting for Border Security, our Military and are Vets. He is tough on Crime and has my full Endorsement. The Great State of Louisiana, we want Clay!

6:08 PM - Jun 24, 2018



♡ 61.6K 💬 23.3K people are talking about this

 **Donald J. Trump** 
@realDonaldTrump

Statement by me last night in Florida: "Honestly, I don't think the Democrats want to make a deal. They talk about DACA, but they don't want to help..We are ready, willing and able to make a deal but they don't want to. They don't want security at the border, they don't want.....

5:57 AM - Jan 15, 2018



♡ 90.4K 💬 40.2K people are talking about this

 **Donald J. Trump** 
@realDonaldTrump

Google search results for "Trump News" shows only the viewing/reporting of Fake News Media. In other words, they have it RIGGED, for me & others, so that almost all stories & news is BAD. Fake CNN is prominent. Republican/Conservative & Fair Media is shut out. Illegal? 96% of....

8:02 AM - Aug 28, 2018



♡ 94.4K 💬 60.4K people are talking about this

 **Donald J. Trump** 
@realDonaldTrump

WOW, @foxandfrlends "Dossier is bogus. Clinton Campaign, DNC funded Dossier. FBI CANNOT (after all of this time) VERIFY CLAIMS IN DOSSIER OF RUSSIA/TRUMP COLLUSION. FBI TAINTED." And they used this Crooked Hillary pile of garbage as the basis for going after the Trump Campaign!

6:24 AM - Dec 26, 2017


♡ 114K 💬 86.2K people are talking about this

 **Hillary Clinton** 
@HillaryClinton

"When Donald Trump says he'll make America great, he means make it even greater for rich guys just like Donald Trump." — @ElizabethForMA

7:43 AM - Jun 27, 2016

♡ 3,520 💬 1,734 people are talking about this

 **Hillary Clinton** 
@HillaryClinton

"Get out and vote. Get out and vote for Hillary. Vote early. Vote right now!" — @FLOTUSIWillVote.com



12:13 PM - Oct 27, 2016

 **Hillary Clinton** 
@HillaryClinton

Let's expand Social Security—not cut or privatize it. Let's offer paid family leave. Let's guarantee equal pay for women once and for all.

9:14 AM - Jul 12, 2016



♡ 4,068 💬 1,786 people are talking about this

 **The Briefing** 
@TheBriefing2016

While Donald Trump was...being Donald Trump, Hillary Clinton was fighting to get kids health care. #debate

7:02 PM - Oct 9, 2016

♡ 4,378 💬 2,643 people are talking about this

 **Hillary Clinton** 
@HillaryClinton

"I support Hillary because Hillary supports me." @OITNB's @UzoAduba & Dascha Polanco are on Team Hillary.

4:33 PM - Nov 1, 2015

♡ 1,579 💬 1,013 people are talking about this



Donald J. Trump 
@realDonaldTrump



BIG CPAC STRAW POLL RESULTS: 93% APPROVE OF THE JOB PRESIDENT TRUMP IS DOING (Thank you!). 50% say President Trump should Tweet MORE or SAME (funny!). 79% say Republicans in Congress should do a better job of working with President Trump (starting to happen).

3:26 PM - Feb 24, 2018

 96K  51.8K people are talking about this



Donald J. Trump 
@realDonaldTrump



"Barack Obama talked a lot about hope, but Donald Trump delivered the American Dream. All the economic indicators, what's happening overseas, Donald Trump has proven to be far more successful than Barack Obama. President Trump is delivering the American Dream." Jason Chaffetz

6:32 AM - Sep 9, 2018

 95.3K  47.5K people are talking about this



Donald J. Trump 
@realDonaldTrump



Great night for Republicans! Congratulations to John Cox on a really big number in California. He can win. Even Fake News CNN said the Trump impact was really big, much bigger than they ever thought possible. So much for the big Blue Wave, it may be a big Red Wave. Working hard!

6:16 AM - Jun 6, 2018

 83.8K  30.9K people are talking about this



Hillary Clinton 
@HillaryClinton



"Madam President! Madam President! Madam President!"

10:01 AM - Jun 13, 2015

 1,671  888 people are talking about this



Hillary Clinton 
@HillaryClinton



"We don't need to make America great again. America never stopped being great. But we do need to make America whole again." —Hillary in SC

5:53 PM - Feb 27, 2016

 4,473  2,811 people are talking about this

