Clustering with feature selection using alternating minimization and a projection-gradient method

Cyprien Gilet GILET@I3S.UNICE.FR
I3S, Univ. Côte d'Azur & CNRS, F-06900 Sophia Antipolis.

Marie Deprez DEPREZ@IPMC.CNRS.FR
IPMC, Univ. Côte d'Azur & CNRS, F-06560 Sophia Antipolis.

Jean-Baptiste Caillau JEAN-BAPTISTE.CAILLAU@UNIV-COTEDAZUR.FR
LJAD, Univ. Côte d'Azur & CNRS/Inria, F-06108 Nice.

Michel Barlaud BARLAUD@I3S.UNICE.FR
I3S, Univ. Côte d'Azur & CNRS, F-06900 Sophia Antipolis.

Editor:

Abstract

This paper deals with unsupervised clustering with feature selection in high dimensional space. The problem is to estimate both labels and a sparse projection matrix of weights. To address this combinatorial non-convex problem maintaining a strict control on the sparsity of the matrix of weights, we propose an alternating minimization of the Frobenius norm criterion. We provide a new efficient algorithm named k-sparse which alternates k-means with projection-gradient minimization. The projection-gradient step is a method of splitting type, with exact projection on the ℓ^1 ball to promote sparsity. The convergence of the gradient-projection step is addressed, and a preliminary analysis of the alternating minimization is made. Experiments on Single Cell RNA sequencing datasets show that our method significantly improves the results of PCA k-means, spectral clustering, SIMLR, and Sparcl methods. The complexity of our method is linear in the number of samples (cells), so that the method scales up to large datasets.

1. Introduction

This paper deals with unsupervised clustering and feature selection in high dimensional space. Early work on feature selection were based on support vector machine (see Guyon et al. (2002)) or logistic regression (Shevade and Keerthi (2003)). We advocate the use of sparsity promoting methods as they allow not only to perform feature selection (a crucial task in biological applications, *e.g.* where features are genes), but also to use efficient state-of-the-art algorithms from convex optimization. Clustering in high dimension using classical algorithms such as k-means (McQueen (1967); Arthur and Vassilvitski (2007)) suffers from the curse of dimensionality. As dimensions increase, vectors become indiscernible and the predictive power of the aforementioned methods is drastically reduced (Aggarwal (2005); Radovanovic et al. (2010)). In order to overcome this issue, a popular approach for high-dimensional data is to perform *Principal Component Analysis* (PCA) prior to clustering. This approach is however difficult to justify in general (Wei-Chien (1983)). An alternative approach proposed in (de la Torre and Kanade (2006); Ding and Li (2007)) is to combine clustering and dimension

reduction by means of *Linear Discriminant Analysis* (LDA). The heuristic used in (Ding and Li (2007)) is based on alternating minimization, which consists in iteratively computing a projection subspace by LDA, using the labels y at the current iteration and then running k-means on the projection of the data onto the subspace. Departing from this work, Bach and Harchaoui (2008) propose a convex relaxation in terms of a suitable semi-definite program (SDP). Another efficient approach is spectral clustering where the main tools are graph Laplacian matrices (Ng et al. (2002); Von Luxburg (2007)). However, methods such as PCA, LDA or, more recently SIMLR, do not provide sparsity. A popular approach for selecting sparse features in supervised classification or regression is the *Least Absolute Shrinkage and Selection Operator* (LASSO) formulation (Tibshirani (1996)). The LASSO formulation uses the ℓ^1 norm instead of ℓ^0 (Candès (2008); Candès et al. (2008); Donoho and Elad (2003); Donoho and Logan (1992)) as an added penalty term. A hyperparameter, which unfortunaltely does not have any simple interpretation, is then used to tune sparsity. Witten and Tibshirani (2010) use a lasso-type penalty to select the features and propose a sparse k-means method. A main issue is that optimizing the values of the Lagrangian parameter λ (Hastie et al. (2004); Witten and Tibshirani (2010)) is computationally expensive (Mairal and Yu (2012)). All these methods (Bach and Harchaoui (2008); de la Torre and Kanade (2006); Ding and Li (2007); Witten and Tibshirani (2010)) require a k-means heuristic to retrieve the labels. The alternating scheme we propose combines such a k-means step with dimension reduction, as well as feature selection using an ℓ^1 sparsity constraint.

2. Constrained unsupervised classification

2.1 General Framework

Let X be the (nonzero) $m \times d$ matrix made of m line samples $x_1, \dots, x_m$ belonging to the d-dimensional space of features. Let $Y \in \{0,1\}^{m \times k}$ be the matrix of labels where $k \geqslant 2$ is the number of clusters. Note that we assume that this number is known; It is indeed the case for the applications we present in Section 3, while estimating k is in general a delicate matter out of the scope of this paper. Each line of Y has exactly one nonzero element equal to one, $y_{ij} = 1$ indicating that the sample x_i belongs to the j-th cluster. Let $W \in \mathbb{R}^{d \times \bar{d}}$ be the projection matrix, where the dimension in the projected space, $\bar{d}$, is understood to be much smaller than d. Let then μ be the $k \times \bar{d}$ matrix of centroids of the projected data, XW:

$$\mu(j,:) := \frac{1}{\sum_{i=1}^{m} y_{ij}} \sum_{i \text{ s.t. } y_{ij}=1} (XW)(i,:).$$

The j-th centroid is the model for all samples x_i belonging to the j-th cluster ($y_{ij} = 1$). The clustering criterion can be cast as the *Within-Cluster Sum of Squares* (WCSS, Selim and Ismail (1984); Witten and Tibshirani (2010)) in the projected space

$$\frac{1}{2}\|Y\mu - XW\|_F^2 \to \min \tag{1}$$

where $\|.\|_F$ is the Frobenius norm induced by the Euclidean structure on $m \times \bar{d}$ matrices,

$$(A|B)_F := \mathrm{tr}(A^T B) = \mathrm{tr}(AB^T), \quad \|A\|_F := \sqrt{(A|A)_F}\,.$$

The matrix of labels is constrained according to

$$y_{ij} \in \{0,1\}, \quad i = 1,\dots,m, \quad j = 1,\dots,k, \tag{2}$$

$$\sum_{j=1}^{k} y_{ij} = 1, \quad i = 1,\dots,m, \tag{3}$$

$$\sum_{i=1}^{m} y_{ij} \geqslant 1, \quad j = 1,\dots,k. \tag{4}$$

Note that (3) implies that each sample belongs to exactly one cluster while (4) ensures that each cluster is not empty (no fusion of clusters). This prevents trivial solutions consisting in $k-1$ empty clusters and $W = 0$. In contrast with the Lagrangian LASSO formulation, we want to have a direct control on the value of the ℓ^1 bound, so we constrain W according to

$$\|W\|_1 \leqslant \eta \quad (\eta > 0), \tag{5}$$

where $\|.\|_1$ is the ℓ^1 norm of the vectorized $d \times \bar{d}$ matrix of weights:

$$\|W\|_1 := \|W(:)\|_1 = \sum_{i=1}^{d}\sum_{j=1}^{\bar{d}} |w_{ij}|.$$

The problem is to estimate labels Y together with the sparse projection matrix W. As Y and W are bounded, the set of constraints is compact and existence of minimizers holds.

Proposition 1 *The minimization of the norm (1), jointly in Y and W under the constraints (2)-(5), has a solution.*

To attack this difficult nonconvex problem, we propose an alternating (or Gauss-Seidel) scheme as in de la Torre and Kanade (2006); Ding and Li (2007); Witten and Tibshirani (2010). Another option would be to design a global convex relaxation to address the joint minimization in Y and W (see, *e.g.*, Bach and Harchaoui (2008); Flammarion et al.) The first convex subproblem is to find the best projection from dimension d to dimension $\bar{d}$ for a given clustering.

Problem 1 *For a fixed clustering Y (and a given $\eta > 0$),*

$$\frac{1}{2}\|Y\mu - XW\|_F^2 \to \min$$

under the constraint (5) on W.

Given the matrix of weights W, the second subproblem is the standard k-means on the projected data.

Problem 2 *For a fixed projection matrix W,*

$$\frac{1}{2}\|Y\mu - XW\|_F^2 \to \min$$

under the constraints (2)-(4) on Y.

2.2 Exact gradient-projection splitting method

To solve Problem 1, we use a gradient-projection method. It belongs to the class of splitting methods (Boyd and Vandenberghe (2009); Combettes and Wajs (2005); Combettes and Pesquet (2011); Lions and Mercier (1979); Mosci et al. (2010); Sra et al. (2012); Parikh and Boyd (2014)). It is designed to solve minimization problems of the form

$$\varphi(W) \to \min, \quad W \in C, \tag{6}$$

using separately the convexity properties of the function φ on one hand, and of the convex set C on the other. We use the following forward-backward scheme to generate a sequence of iterates:

$$V_n := W_n - \gamma_n \nabla\varphi(W_n), \tag{7}$$
$$W_{n+1} := P_C(V_n) + \varepsilon_n, \tag{8}$$

where P_C denotes the projection on the convex set C (a subset of some Euclidean space). Under standard assumptions on the sequence of gradient steps $(\gamma_n)_n$, and on the sequence of projection errors $(\varepsilon_n)_n$, convergence holds (see, *e.g.*, Bauschke and Combettes (2011)).

Theorem 1 *Assume that (6) has a solution. Assume that φ is convex, differentiable, and that $\nabla\varphi$ is β-Lipschitz, $\beta > 0$. Assume finally that C is convex and that*

$$\sum_n |\varepsilon_n| < \infty, \quad \inf_n \gamma_n > 0, \quad \sup_n \gamma_n < 2/\beta.$$

Then the sequence of iterates of the forward-backward scheme (7-8) converges, whatever the initialization. If moreover $(\varepsilon_n)_n = 0$ (exact projections), there exists a rank N and a positive constant K such that, for $n \geqslant N$,

$$\varphi(W_n) - \inf_C \varphi \leqslant K/n. \tag{9}$$

In our case, $\nabla\varphi$ is Lipschitz since it is affine,

$$\nabla\varphi(W) = X^T(XW - Y\mu), \tag{10}$$

and we recall the estimation of its best Lipschitz constant.

Lemma 1 *Let A be a $d \times d$ real matrix, acting linearly on the set of $d \times k$ real matrices by left multiplication, $W \mapsto AW$. Then, its norm as a linear operator on this set endowed with the Frobenius norm is equal to its largest singular value, $\sigma_{\max}(A)$.*

Proof. The Frobenius norm is equal to the ℓ^2 norm of the vectorized matrix,

$$\|W\|_F = \| \begin{bmatrix} W^1 \\ \vdots \\ W^h \end{bmatrix} \|_2, \quad \|AW\|_F = \| \begin{bmatrix} AW^1 \\ \vdots \\ AW^h \end{bmatrix} \|_2, \tag{11}$$

where $W^1, \ldots, W^h$ denote the h column vectors of the $d \times h$ matrix W. Accordingly, the operator norm is equal to the largest singular value of the $kd \times kd$ block-diagonal matrix

whose diagonal is made of k matrix A blocks. Such a matrix readily has the same largest singular value as A. □

As a byproduct of Theorem 1, we get

Corollary 1 *For any fixed step $\gamma \in (0, 2/\sigma^2_{\max}(X))$, the forward-backward scheme applied to the Problem 1 with an exact projection on ℓ^1 balls converges with a linear rate towards a solution, and the estimate (9) holds.*

Proof. The ℓ^1 ball being compact, existence holds. So does convergence, provided the condition of the step lengths is fulfilled. Now, according to the previous lemma, the best Lipschitz constant of the gradient of φ is $\sigma_{\max}(X^TX) = \sigma^2_{\max}(X)$, hence the result. □

Algorithm 1 Exact gradient-projection algorithm

Input: $X, Y, \mu, W_0, N, \gamma, \eta$
$W \leftarrow W_0$
for $n = 0, \dots, N$ **do**
  $V \leftarrow W - \gamma X^T(XW - Y\mu)$
  $W \leftarrow P^1_\eta(V)$
end for
Output: W

Exact projection. In Algorithm 1, we denote by $P^1_\eta(W)$ the (reshaped as a $d \times \bar{d}$ matrix) projection of the vectorized matrix $W(:)$. An important asset of the method is that it takes advantage of the availability of efficient methods (Condat (2016); Duchi et al. (2008)) to compute the ℓ^1 projection. For $\eta > 0$, denote $B^1(0, \eta)$ the closed ℓ_1 ball of radius η in the space $\mathbb{R}^{d\times\bar{d}}$ centered at the origin, and Δ_η the simplex $\{w \in \mathbb{R}^{d\times\bar{d}} \mid w_1 + \cdots + w_{d\bar{d}} = 1,\ w_1 \geqslant 0, \dots, w_{d\bar{d}} \geqslant 0\}$. Let $w \in \mathbb{R}^{d\times\bar{d}}$, and let v denote the projection on Δ_η of $(|w_1|, \dots, |w_{d\bar{d}}|)$. It is well known that the projection of w on $B^1(0, \eta)$ is

$$(\varepsilon_1(v_1), \dots, \varepsilon_{kd}(v_{d\bar{d}})), \quad \varepsilon_j := \operatorname{sign}(w_j), \quad j = 1, \dots, d\bar{d}, \tag{12}$$

and the fast method described in (Condat (2016)) is used to compute v with complexity $O(d \times \bar{d})$.

Fista implementation. A constant step of suitable size γ is used in accordance with Corollary 1. In our setting, a useful normalization of the design matrix X is obtained replacing X by $X/\sigma_{\max}(X)$. This sets the Lipschitz constant in Theorem 1 to one. The $O(1/n)$ convergence rate of the algorithm can be speeded up to $O(1/n^2)$ using a FISTA step (Beck and Teboulle (2009)). In practice we use a modified version (Chambolle and Dossal (2015)) which ensures convergence of the iterates, see Algorithm 2. Note that for any fixed step $\gamma \in (0, 1/\sigma^2_{\max}(X))$, the FISTA algorithm applied to Problem 1 with an exact projection on ℓ^1 balls converges with a quadratic rate towards a solution, and the estimate (9) holds.

Algorithm 2 Exact gradient-projection algorithm with FISTA

Input: $X, Y, \mu, W_0, N, \gamma, \eta$
$W \leftarrow W_0$
$t \leftarrow 1$
for $n = 0, \dots, N$ **do**
  $V \leftarrow W - \gamma X^T(XW - Y\mu)$
  $W_{\text{new}} \leftarrow P^1_\eta(V)$
  $t_{\text{new}} \leftarrow (n+5)/4$
  $\lambda \leftarrow 1 + (t-1)/t_{\text{new}}$
  $W \leftarrow (1-\lambda)W + \lambda W_{\text{new}}$
  $t \leftarrow t_{\text{new}}$
end for
Output: W

2.3 Clustering algorithm

The resulting alternating minimization is described by Algorithm 3. (One can readily replace the gradient-projection step by the FISTA version described in Algorithm 2.) Labels Y are for instance initialized by spectral clustering on X, while the k-means computation relies on standard methods such as k-means++ (Arthur and Vassilvitski (2007)).

Algorithm 3 Alternating minimization clustering.

Input: $X, Y_0, \mu_0, W_0, L, N, k, \gamma, \eta$
$Y \leftarrow Y_0$
$\mu \leftarrow \mu_0$
$W \leftarrow W_0$
for $l = 0, \dots, L$ **do**
  for $n = 0, \dots, N$ **do**
    $V \leftarrow W - \gamma X^T(XW - Y\mu)$
    $W \leftarrow P^1_\eta(V)$
  end for
  $Y \leftarrow \texttt{kmeans}(XW, k)$
  $\mu \leftarrow \texttt{centroids}(Y, XW)$
end for
Output: Y, W

Convergence of the algorithm. Similarly to the approaches advocated in (Bach and Harchaoui (2008); de la Torre and Kanade (2006); Ding and Li (2007); Witten and Tibshirani (2010)), our method involves non-convex k-means optimization for which convergence towards local minimizers only can be proved (Bottou and Bengio (1995); Selim and Ismail (1984)). In practice, we use k-means++ with several replicates to improve each clustering step. We assume that the initial guess for labels Y and matrix of weights W is such that the associated k centroids are all different. We note for further research that there have been recent attempts to convexify k-means (see, *e.g.*, Bunea et al. (2016); Condat (2017); Mixon et al. (2017);

Peng and Wei (2017)). As each step of the alternating minimization scheme decreases the norm in (1), which is nonnegative, the following readily holds.

Proposition 2 *The Frobenius norm $\|Y\mu - XW\|_F$ converges as the number of iterates L in Algorithm 3 goes to infinity.*

This property is illustrated in the next section on biological data. Further analysis of the convergence may build on recent results on proximal regularizations of the Gauss-Seidel alternating scheme for non convex problems (Attouch et al. (2010); Bolte et al. (2014)).

Gene selection. Feature selection is based on the sparsity inducing ℓ^1 constraint (5). The projection $P^1_\eta(W)$ aims at sparsifying the W matrix so that the gene j will be selected if $\|W(j,:)\| > 0$. For a given constraint η, the practical stopping criterion of the alternating minimization algorithm involves the evolution of the number of the selected genes. At the higher level loop on the bound η itself, the evolution of accuracy versus η is analyzed. We also note that the extension to multi-label classification is straightforward as it suffices to allow several unit values on each line of the matrix Y by relaxing constraint (3).

3. Experimental evaluation on single cell RNA-seq clustering

3.1 Experimental settings

We normalize the features and use the FISTA implementation with constant step $\gamma = 1$ in accordance with Corollary 1, and we set $\bar{d} = k + 4$. Methods based on k-means provide different labels depending on the initial conditions, thus we select the best result over 40 replicates of k-means++ (Arthur and Vassilvitski (2007)). The problem of estimating the number of clusters is out of the range of this study, and we refer to the popular GAP method (Tibshirani et al. (2001)). We compare the labels obtained from our clustering with the true labels to compute the clustering accuracy. We also report the popular *Adjusted Rank Index* (ARI) (Lawrence and Phipps (1985)) and *Normalized Mutual Information* (NMI) criteria. Processing times are obtained on a computer using an i7 processor (2.5 Ghz).
We compare our method with PCA k-means, spectral clustering (Von Luxburg (2007)), SIMLR (Single-cell Interpretation via Multikernel Learning) (Wang et al. (2017); Bach et al. (2004)) and Sparcl (Sparse k-means clustering) (Witten and Tibshirani (2010)). The first two methods (PCA k-means and spectral clustering) are standard and easily tested, while we have used the R software package `Sparcl` provided by (Witten and Tibshirani (2010)) for Sparcl method. And we refer for SIMLR to the codes available online: See `https://github.com/BatzoglouLabSU/SIMLR/tree/SIMLR/MATLAB`.

3.2 Application to computational biology: Synthetic datasets

The simulation software was downloaded from `https://github.com/DeprezM/SCsim`. We use default parameters. The decay of the Frobenius norm (1) is portrayed Figure 1, while the evolution of the number of selected genes vs. the sparsity constraint (ℓ^1 constraint) or the accuracy is shown Figure 2. Both graphs illustrate the good properties of our method in terms of convergence, feature selection (a plateau in accuracy is reached as soon as the number of selected genes is large enough). As is clear from Tables 1 and 2, k-sparse behaves

better than any of the four other methods on synthetic data. We also provide `tsne` (Van der Maaten and Hinton (2008)) for a 2D visual evaluation of each method (see Figure 6). The results, quite comparable for SIMLR and k-sparse, provide a clear confirmation of those in Tables 1 and 2 .

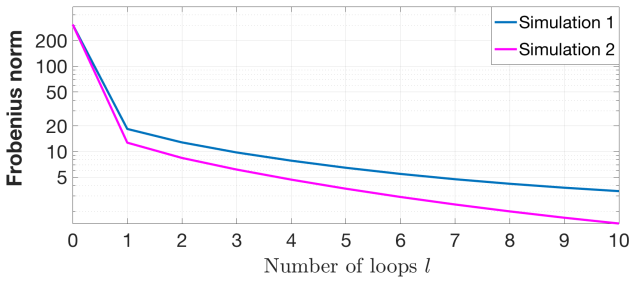

Figure 1: Decay of the Frobenius norm for the two synthetic datasets versus the number of loops of the alternating minimization scheme emphasizes the fast and smooth convergence of our algorithm.

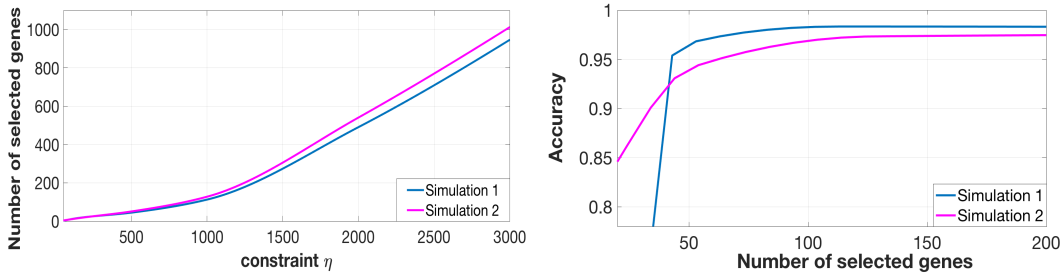

Figure 2: Left: The evolution of the number of selected genes versus the constraint is a smooth monotonous function. The bound η for the ℓ^1 constraint is thus easily tuned. Right: Accuracy versus number of genes. These results show that a minimum number of genes is required to get the best possible clustering accuracy.

Table 1: Comparison between methods for the synthetic dataset 1 (4 clusters, 600 cells, 5,000 genes). For $\eta = 1000$ k-sparse selected 113 genes (see Figure 2. Left.) and outperforms others methods in terms of accuracy, ARI and NMI.

Simulation 1	PCA	Spectral	SIMLR	k-sparse
Accuracy (%)	62.33	74.00	97.33	**98.33**
ARI (%)	37.21	56.43	93.77	**95.27**
NMI	0.50	0.63	0.89	**0.92**
Time (s)	**0.36**	0.48	10.04	13.73

Table 2: Comparison between methods for the synthetic dataset 2 (4 clusters, 600 cells, 10,000 genes). For $\eta = 3000$ k-sparse selected $1,089$ genes (see Figure 2. Left.) and outperforms others methods in terms of accuracy, ARI and NMI.

Simulation 2	PCA	Spectral	SIMLR	k-sparse
Accuracy (%)	61.33	74.50	97.50	**97.83**
ARI (%)	34.75	57.36	93.26	**94.03**
NMI	0.49	0.60	0.90	**0.91**
Time (s)	**0.63**	0.75	13.82	62.92

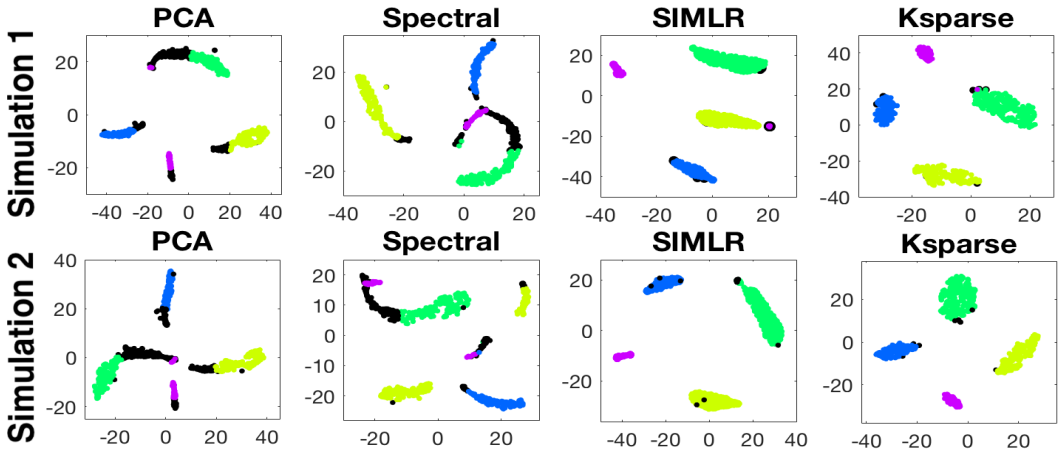

Figure 3: Comparison of 2D visualization using `tsne` (Van der Maaten and Hinton (2008)). Each point represents a cell. Misclassified cells in black are reported for each method. This figure shows the nice small ball-shaped clusters computed by k-sparse and SIMLR methods.

3.3 Application to computational biology: Single cell datasets

Our algorithm can be readily extended to multiclass clustering of high dimensional databases in computational biology (single cell clustering, mass-spectrometric data...), pattern recognition, combinatorial chemistry, social networks clustering, decision making, *etc.* We provide an experimental evaluation on Single-cell sequencing dataset. The new Single-cell technology has been elected "method of the year" in 2013 by *Nature Methods* (Evanko (2014)). The widespread use of such methods has enabled the publication of many datasets with ground truth cell type annotations (Kiselev (2017)). Thus we compare algorithms on three of those public single-cell RNA-seq datasets: Klein dataset (Klein (2015)), Zeisel dataset (Zeisel et al. (2015)) and Usoskin (Usoskin et al. (2015)) dataset.

Klein scRNA-seq dataset. Klein (2015) characterized the transcriptome of 2,717 cells (*Mouse Embryonic Stem Cells*, mESCs), across four culture conditions (control and with 2, 4 or 7 days after leukemia inhibitory factor, LIF, withdrawal) using InDrop sequencing. Gene expression was quantified with *Unique Molecular Identifier* (UMI) counts (essentially tags that identify individual molecules allowing removal of amplification bias). The raw UMI counts and cells label were downloaded from `hemberg-lab.github.io/scRNA.seq.datasets`. After filtering out lowly expressed genes (10,322 genes remaining after removing genes that have less than 2 counts in 130 cells) and Count Per Million normalization (CPM) to reduce cell-to-cell variation in sequencing, we report clustering into four cell sub-populations, corresponding to the four culture conditions.

Zeisel scRNA-seq dataset. Zeisel et al. (Kiselev (2017); Zeisel et al. (2015)) collected 3,005 mouse cells from the primary somatosensory cortex (S1) and the hippocampal CA1 region, using the Fluidigm C1 microfluidics cell capture platform followed. Gene expression was quantified with UMI counts. The raw UMI counts and metadata (batch, sex, labels) were downloaded from `linnarssonlab.org/cortex`. We applied low expressed gene filtering (7,364 remaining genes after removing genes that have less than 2 counts in 30 cells) and CPM normalization. We report clustering into the nine major classes identified in the study.

Usoskin scRNA-seq dataset. Uzoskin et al. (Usoskin et al. (2015)) collected 622 cells from the mouse dorsal root ganglion, using a robotic cell-picking setup and sequenced with a 5' single-cell tagged reverse transcription (STRT) method. Filtered (9,195 genes) and normalized data (expressed as Reads Per Million) were downloaded with full sample annotations from `linnarssonlab.org/drg`. We report clustering into four neuronal cell types.

3.4 Comparison between methods

We provide accuracy, ARI, NMI and time processing for five different methods: PCA k-means, spectral clustering (Von Luxburg (2007)), SIMLR (Single-cell Interpretation via Multikernel Learning) (Wang et al. (2017)), Sparcl (Sparse k-means clustering) (Witten and Tibshirani (2010)), and our method k-sparse. As in the previous section on synthetic data, we provide an evaluation of k-sparse on each of three bases (Klein, Usoskin and Zeisel) in terms of convergence (see Figure 4), feature selection and accuracy (Figure 5). Our method

significantly improves the results of Sparcl and SIMLR in terms of accuracy, ARI and NMI. For each of the three databases, k-sparse obtains the best results when compared to the four other methods, not only for accuracy but also for ARI and NMI (see Tables 3, 4 and 5. We note however that k-sparse, though faster than SIMLR in two cases out of three, has larger execution times than much less precise methods such as PCA k-means (for which very efficient codes exist). We provide again `tsne` (Van der Maaten and Hinton (2008)) for visual evaluation and comparison of the five methods (Figure 6).

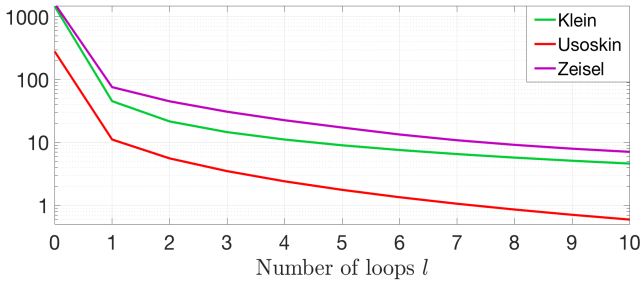

Figure 4: Left: Decay of the Frobenius norm for the three datasets versus the number of loops of the alternating minimization scheme emphasizes the fast and smooth convergence of our algorithm.

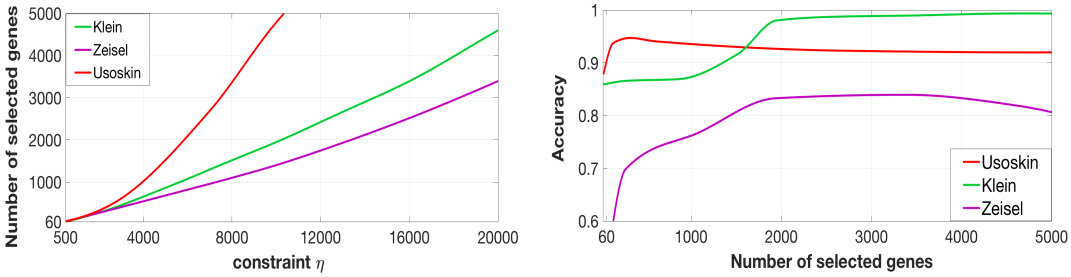

Figure 5: Left: The evolution of the number of selected genes versus the constraint is a smooth monotonous function. In our constrained approach, parameter η is directly connected to the number of genes. An estimate of this number is known by biologists. Thus we tune η in order to obtain the desired number of genes. Right: We use Accuracy, ARI and NMI (which indeed uses ground-true labels) for evaluation only of our algorithm, and comparison with contenders. These results show that a minimum number of genes is required to get the best possible clustering accuracy and that accuracy is constant over a large plateau, making the tuning of the parameter very easy.

Table 3: Comparison between methods for Usoskin dataset (4 clusters, 622 cells, 9,195 genes). For $\eta = 8000$, k-sparse selected $3,325$ genes (see Figure 5. Left.) and outperforms others methods in terms of accuracy by 15%.

Usoskin dataset	PCA	Spectral	SIMLR	Sparcl	k-sparse
Accuracy (%)	54.82	60.13	76.37	57.24	**91.96**
ARI (%)	22.33	26.46	67.19	31.30	**85.85**
NMI	0.29	0.33	0.75	0.39	**0.83**
Time (s)	1.06	**0.91**	15.67	1,830	57.07

Table 4: Comparison between methods for Klein dataset (4 clusters, 2,717 cells, 10,322 genes). For $\eta = 20000$, k-sparse selected $4,599$ genes (see Figure 5. Left.) and has an accuracy close to 100%. SIMLR has similar performances (accuracy, ARI and NMI) than k-sparse (which is 5 times faster than SIMLR).

Klein dataset	PCA	Spectral	SIMLR	Sparcl	k-sparse
Accuracy (%)	68.50	63.31	99.12	65.11	**99.33**
ARI (%)	44.82	38.91	98.34	45.11	**98.77**
NMI	0.55	0.54	0.96	0.56	**0.97**
Time (s)	**10.91**	20.81	511	30,384	97.10

Table 5: Comparison between methods for Zeisel dataset (9 clusters, 3,005 cells, 7,364 genes). For $\eta = 16000$, k-sparse selected $2,497$ genes (see Figure 5. Left.) and outperforms others methods in terms of accuracy by 11%. K-sparse is 6 times faster than SIMLR.

Zeisel dataset	PCA	Spectral	SIMLR	Sparcl	k-sparse
Accuracy (%)	39.60	59.30	71.85	65.23	**83.26**
ARI (%)	34.67	50.55	64.8	59.06	**75.06**
NMI	0.54	0.68	0.75	0.69	**0.77**
Time (s)	**11**	23	464	28,980	71.60

3.5 Scalability

K-sparse converges within around $L = 10$ loops. The complexity of the inner iteration of k-sparse is $O(d \times \bar{d} \times d_n(\eta))$ for the gradient part (sparse matrix multiplication $X^T XW$), plus $O(d \times \bar{d})$ for the projection part, where $d_n(\eta)$ is the average number of nonzero entries of the sparse matrix W. This number depends on the sharpness of the ℓ^1 constraint (5) defined by η, and on the iteration n. (As n ranges from 0 to N, sparsity is increased as illustrated by the numerical simulations.) The number of genes decreases rapidly with the iterates which allows to use sparse computing. One must then add the cost of k-means, that is expected to be $O(m \times \bar{d})$ in average. This allows k-sparse to scale up to large or very large databases. In contrast, optimizing the values of the Lagrangian parameter using permutations

Sparcl is computationally expensive, with complexity $O(m^2 \times d)$. Naive implementation of Kernel methods SIMLR results in $O(m^2)$ complexity. The computational cost can be reduced to $O(p^2 \times m)$ (p is the low rank) using low rank kernel matrix approximation (Bach (2013)). The computational cost is improved (see Table 7) while the performance (ARI) drop significantly (see Table 6) when using low rank kernel matrix approximation in Large SIMLR (`https://github.com/BatzoglouLabSU/SIMLR/tree/SIMLR/MATLAB`).

Table 6: Comparison between SIMLR, Large SIMLR and k-sparse in terms of ARI (%) on large datasets. K-sparse outperforms Large SIMLR by 36% on Klein dataset and 20% on Zeisel dataset in terms of ARI.

Methods	SIMLR	Large SIMLR	k-sparse
Klein (2,717 cells, $10,322$ genes, $k = 4$)	98.34	61.49	**98.77**
Zeisel (3,005 cells, $7,364$ genes, $k = 9$)	64.8	56.39	**75.06**

Table 7: Comparison between SIMLR, Large SIMLR and k-sparse in terms of time (s) on large datasets. K-sparse is 8 times faster on Klein dataset and 10 times faster on Zeisel dataset than SIMLR. Large SIMLR is faster than k-sparse but Table 6 shows that the clusters performed by Large SIMLR are not similar to real clusters.

Methods	SIMLR	Large SIMLR	k-sparse
Klein (2,717 cells, $10,322$ genes, $k = 4$)	511	**8.64**	97.10
Zeisel (3,005 cells, $7,364$ genes, $k = 9$)	464	**8.19**	71.60

4. Conclusion

In this paper, we focus on unsupervised classification. We provide a new efficient algorithm based on alternating minimization that achieves feature selection by introducing an ℓ^1 constraint in the gradient-projection step. This step, of splitting type, uses an exact projection on the ℓ^1 ball to promote sparsity, and is alternated with k-means. Convergence of the projection-gradient method is established, and each iterative step of our algorithm necessarily lowers the cost. Experiments on single-cell RNA-seq dataset in Section 3 demonstrate that our method is very promising compared to other algorithms in the field. Ongoing developments deal with the application of k-sparse to very large datasets.

References

C. Aggarwal. On k-anonymity and the curse of dimensionality. *Proceedings of the 31st VLDB Conference, Trondheim, Norway*, 2005.

D. Arthur and S. Vassilvitski. k-means++: The advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007.

H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the kurdyka-lojasiewicz inequality. *Mathematics of Operations Research*, 2010.

F. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

F. R. Bach. Sharp analysis of low-rank kernel matrix approximations. *International Conference on Learning Theory (COLT).*, 2013.

F. R. Bach and Z. Harchaoui. Diffrac: a discriminative and flexible framework for clustering. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 49–56. Curran Associates, Inc., 2008.

H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, Aug 2014.

L. Bottou and Y. Bengio. Convergence properties of the k-means algorithms. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 585–592. MIT Press, 1995.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2009.

F. Bunea, C. Giraud, M. Royer, and N. Verzelen. PECOK: A convex optimization approach to variable clustering. (1606.05100), 2016.

E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Acad Sciences Paris*, 346(1):589–592, 2008.

E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted $\ell 1$ minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.

A. Chambolle and C. Dossal. On the convergence of the iterates of "fista". *Journal of Optimization Theory and Applications, Springer Verlag*, (166), 2015.

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

L. Condat. Fast projection onto the simplex and the l1 ball. *Mathematical Programming Series A*, 158(1):575–585, 2016.

L. Condat. A convex approach to k-means clustering and image segmentation. *HAL*, (01504799), 2017.

F. de la Torre and T. Kanade. Discriminative cluster analysis. *ICML 06 Proceedings of the 23rd international conference on Machine learning, Pittsburgh, Pennsylvania, USA*, 2006.

C. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 521–528. ACM, 2007.

D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell 1$ minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

D. L. Donoho and B. F. Logan. Signal recovery and the large sieve. *SIAM Journal on Applied Mathematics*, 52(2):577–591, 1992.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008.

D. Evanko. Method of the year 2013: Methods to sequence the dna and rna of single cells are poised to transform many areas of biology and medicine. *Nature Methods, Vol 11*, 2014.

N. Flammarion, B. Palaniappan, and F. R. Bach. Robust discriminative clustering with sparse regularizers. *Journal of Machine Learning Research*.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

V. Y. et al Kiselev. Sc3: consensus clustering of single-cell rna-seq data. *Nature Methods*, 2017.

A. M. et al Klein. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 2015.

H. Lawrence and A. Phipps. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

P.-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

J. Mairal and B. Yu. Complexity analysis of the lasso regularization path. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 353–360, 2012.

J.-B. McQueen. Some methods for classification and analysis of multivariate observations. *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

D. G. Mixon, S. Villar, and R. Ward. Clustering subgaussian mixtures with k-means. *Information and inference*, 00:1–27, 2017.

S. Mosci, L. Rosasco, M. Santoro, A. Verri, and S. Villa. Solving structured sparsity regularization with proximal methods. In *Machine Learning and Knowledge Discovery in Databases*, pages 418–433. Springer, 2010.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2002.

N. Parikh and S. Boyd. *Proximal Algorithms*. Foundations and Trends in Optimization, 2014.

J. Peng and Y. Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM J. Optim.*, 18:186–205, 2017.

M. Radovanovic, A. Nanopoulos, and M. Ivanovic. Hubs in space : Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research. 11: 2487?2531.*, 2010.

S. Z. Selim and M. A. Ismail. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Trans. Patt. An. Machine Intel.*, PAMI-6:81–87, 1984.

S. K. Shevade and S. S. Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19(17):2246–2253, 2003.

S. Sra, S. Nowozin, and S. J. Wright. *Optimization for Machine Learning*. MIT Press, 2012.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 411–423, 2001.

D. Usoskin, A. Furlan, S. Islam, H. Abdo, P. Lönnerberg, D. Lou, H. Hjerling-Leffler, J. Haeggström, O. Kharchenko, P. V. Kharchenko, S. Linnarsson, and Ernfors P. Unbiased classification of sensory neuron types by large-scale single-cell rna sequencing. *Nature Neuroscience*, 18:145–153, 2015.

L. J. P. Van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 2007.

B. Wang, J. Zhu, E. Pierson, D Ramazzotti, and S. Batzoglou. Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning. *Nature methods*, (14), 2017.

C. Wei-Chien. On using principal components before separating a mixture of two multivariate normal distributions. *Journal of the Royal Statistical Society*, 32(3), 1983.

D. M Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010.

A. Zeisel, A. B. Muñoz Manchado, S. Codeluppi1, P. Lönnerberg, G. La Manno, A. Juréus, S. Marques, H. Munguba, L. He, C. Betsholtz, C. Rolny, G. Castelo-Branco, J. Hjerling-Leffler, and S. Linnarsson. Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347(6226):1138–1142, 2015.

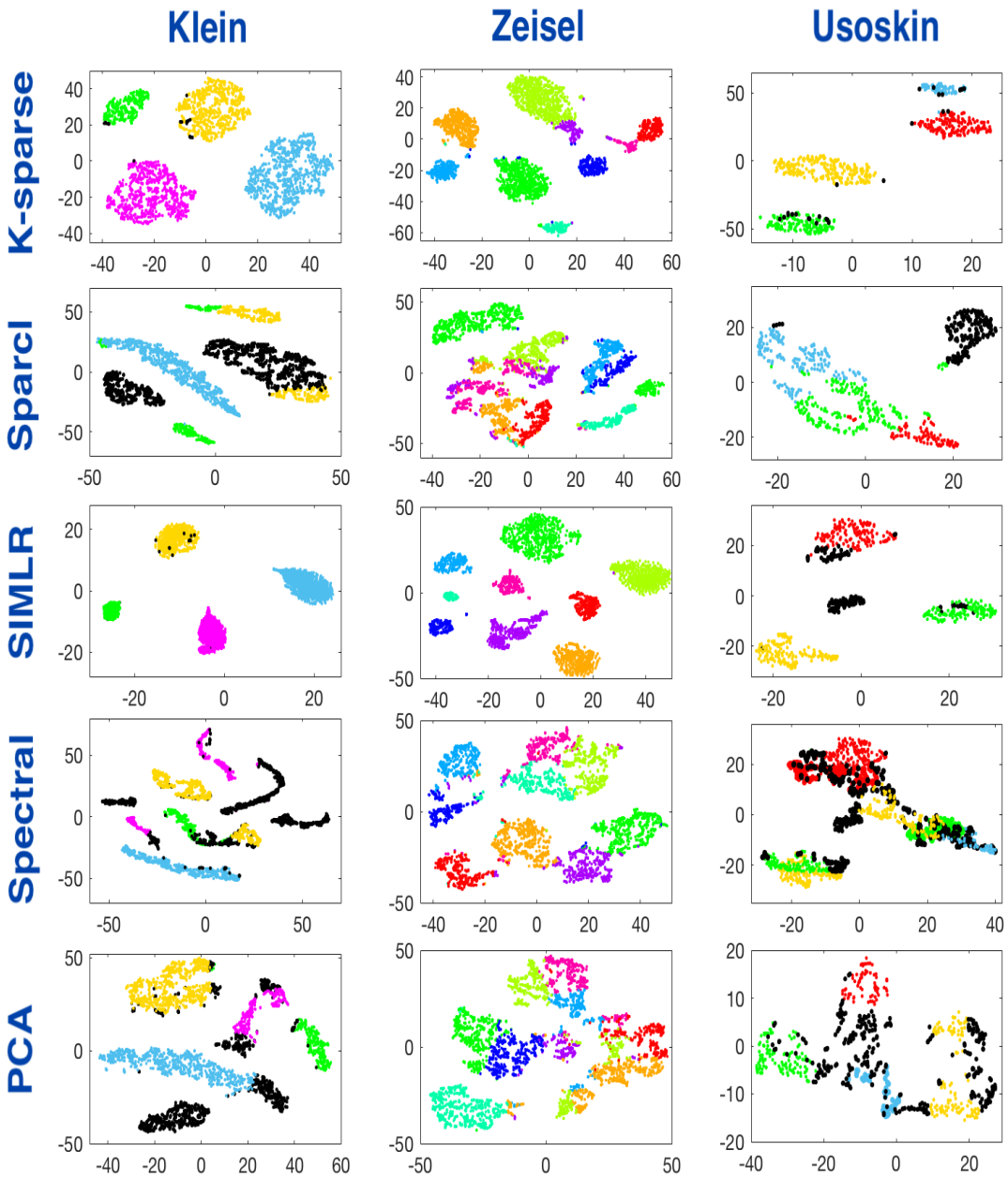

Figure 6: Comparison of 2D visualization using `tsne` (Van der Maaten and Hinton (2008)). Each point represents a cell. Misclassified cells in black are reported for two datasets: Klein and Usoskin. k-sparse significantly improves visually the results of Sparcl and SIMLR (note that SIMLR fails to discover one class on Usoskin). This figure shows the nice small ball-shaped clusters computed by k-sparse and SIMLR methods.