SCALING UP DISCRETE DISTRIBUTION CLUSTERING USING ADMM

Jianbo Ye and Jia Li

The Pennsylvania State University, University Park, PA, USA

ABSTRACT

The discrete distribution as a sparse representation, equipped with the Kantorovich-Wasserstein metric, has been proven effective in learning tasks on imagery data. However, clustering based on the Kantorovich metric under a principled optimization criterion is computationally challenging, and has not been adequately explored. In this paper, we focus on the scalability issue and develop a new algorithm for clustering distributions. An optimal centroid or representative distribution in the sense of the Kantorovich metric is solved for each cluster. The key idea is to adapt the state-of-the-art distributed optimization approach called alternating direction method of multipliers (ADMM). The new algorithm achieves linear complexity in the update of each centroid and can be easily parallelizable, improving significantly over the existing method. It is also observed that in practice, satisfactory results can be obtained after a few tens of iterations. We conduct experiments on both synthetic and real data to demonstrate the computational efficiency and accuracy of the new algorithm.

***Index Terms*—** Discrete Distribution, Clustering, Large-Scale Learning, ADMM

1. INTRODUCTION

In data representation, the discrete distribution, including histogram as a special case, is a typical way to quantize data and to summarize a mass of data. The Kantorovich-Wasserstein distance between two discrete distributions can be understood intuitively as the minimum amount of transportation work from one distribution to the other. It is known as the Mallows distance in statistics and histogram-based earth mover's distance (EMD) in computer science. The Kantorovich distance is a true metric [1]. Because of its consideration of cross-term relationships, the Kantorovich metric has demonstrated strength in abundant applications in computer vision [2, 3, 4], data mining [5], information retrieval [6], computer graphics [7], and multimedia [8]. It provides better distinguishability between objects characterized by discrete distributions than classical metrics such as Euclidean distance [6]. Despite its usefulness, Kantorovich distance is computationally costly because the distance has no closed form and is super-linear subject to the number of bins or support points in the distribution. A number of approximation algorithms and effective filtering have been proposed for fast evaluation and similarity search [9, 3, 10].

The central problem for this paper is to develop a clustering algorithm under the Kantorovich metric subject to the generally accepted optimization criterion of minimizing the total squared distances between all the objects and their corresponding centroids. This criterion is also used by K-means for vector data under the Euclidean distance, mathematically much simpler to treat. When the data structure is complicated or not even available, clustering can be performed based only on the pairwise distances between data objects. For such methods, a centroid for each cluster is either not provided or picked from the data objects themselves according to some heuristics, as is done in some generalized versions of K-means. In our case, however, it is important to ensure certain optimality of the centroid and solve the optimization problem rigorously. Serving as effective summarization of the clusters, the centroids play a critical role in the construction of generative models for different categories of data, as demonstrated in a real-time image annotation system, namely, ALIPR [11]. An iterative optimization algorithm called *D2-clustering* [11] has been developed to solve the optimization problem.

The main technical challenge tackled in D2-clustering is to compute an optimal centroid given a cluster of data objects. D2-clustering achieves optimality by large-scale linear programming (LP). However, the algorithm is of poor scalability because the number of variables in LP grows with the cluster size and classical LP methods, e.g., simplex method, interior point method, are not scalable in this regard, preventing the algorithm from computing a centroid for a large number of objects. There have been approaches to get around this difficulty, for example, computing a centroid heuristically that might be far from the exact centroid [12, 13], or losing some important properties of data samples via wavelet approximation, e.g., sparsity and high dimensionality [5].

In this paper, instead of avoiding the optimization of the centroid, we tackle the problem directly and exploit state-of-the-art optimization methods to improve scalability. We found that one particular distributed optimization

This work is supported by the National Science Foundation under the grant CCF-0936948. Jianbo Ye is with the College of Information Science and Technology, Email: jxy198@ist.psu.edu Jia Li is with the Department of Statistics, Email: jiali@stat.psu.edu

framework, called alternating direction method of multipliers (ADMM) [14, 15, 16], fits well for this challenge and makes D2-clustering algorithm tractable when scaling up (linear complexity even on a single processor). ADMM, developed in 1970s [14, 15], is a distributed optimization method that could scale up many statistical learning problems. Its global convergence has been established in the literature [17, 18]. It has received renewed attention recently due to the tremendous demands from large-scale and data-distributed machine learning algorithms [16]. Our major contribution is to design a new clustering algorithm based on ADMM to achieve scalability (see section 2 and 3). We analyze the computational complexity of the new algorithm and compare it experimentally with the existing D2-clustering (see section 4).

2. PRELIMINARIES

D2-clustering attempts to minimize the total squared Kantorovich distances between the data and the centroids. At the outer level, the D2-clustering algorithm iterates the update of data partition and the update of the centroids. Given the centroids, a data object is assigned to its closest centroid. The computational core is the update of the optimal centroids given the partition of clusters, elaborated below.

Let $\{p^{(1)}, \ldots, p^{(N)}\}$ be the dataset of N discrete distributions. For clarity of discussion, we now focus on computing the centroid of a cluster. Hence, here, N is the number of objects in one cluster under consideration, instead of the whole data set. The discrete distribution $p^{(i)}$ is represented by $\{\binom{x_1^{(i)}}{w_1^{(i)}}, \ldots, \binom{x_{m_i}^{(i)}}{w_{m_i}^{(i)}}\}$, where m_i is the number of support values, $x_j^{(i)}$ is the support value with a probability of $w_j^{(i)} > 0$ (therefore $\sum_{j=1}^{m_i} w_j^{(i)} = 1$). The L_2 Kantorovich distance $D(p^{(a)}, p^{(b)}) > 0$ between two discrete distributions, $p^{(a)}$ and $p^{(b)}$, is computed by a linear programming problem, that is,

$$D^2(p^{(a)}, p^{(b)}) = \min_{\pi} \quad \sum_{i,j} \pi_{i,j} \left\| x_i^{(a)} - x_j^{(b)} \right\|^2 \tag{1}$$

subject to $\sum_{j=1}^{m_b} \pi_{i,j} = w_i^{(a)}$, $i = 1, \ldots, m_a$, $\sum_{i=1}^{m_a} \pi_{i,j} = w_j^{(b)}$, $j = 1, \ldots, m_b$, $\pi_{i,j} \geq 0$, $1 \leq i \leq m_a$, $1 \leq j \leq m_b$.

We want to to find $\beta : \{\binom{z_1}{u_1}, \ldots, \binom{z_M}{u_M}\}$, such that $\sum_{i=1}^{N} D^2(\beta, p^{(i)})$ is minimized with respect to z_j's and u_j's. With detailed derivation omitted, we state that the centroid is solved by the following optimization

$$\min_{\Pi, U, Z} \quad \sum_k \sum_{i,j} \pi_{i,j}^{(k)} \left\| x_i^{(k)} - z_j \right\|^2 \tag{2}$$

subject to $\sum_{j=1}^{M} \pi_{i,j}^{(k)} = w_i^{(k)}$, $i = 1, \ldots, m_k,\ k = 1, \ldots, N$, $\sum_{i=1}^{m_k} \pi_{i,j}^{(k)} = u_j$, $j = 1, \ldots, M,\ k = 1, \ldots, N$, $\sum_{j=1}^{M} u_j = 1$, $\pi_{i,j}^{(k)} \geq 0, u_j \geq 0$, $\forall i, j, k$. Here we use Π to denote the set of $\{\pi_{i,j}^k\}$, U for the set of $\{u_j, j = 1, ..., M\}$, and Z for the set of $\{z_j, j = 1, ..., M\}$. The optimization problem described by (2) is not convex in general, so we might not expect to have a unique minimizer. We solve the minimization by iterating two steps: in the first step, we fix U, Π and update Z, which reduces to a small-size least square problem with closed solution, in particular, $z_j = \frac{1}{N} \sum_{i,k} \pi_{i,j}^{(k)} x_i^{(k)}$; in the second step, we fix Z and update U, Π, which is solved by a large-scale linear programming with super-linearly growth with respect to N in D2-clustering. Then we iterate the procedure until convergence (subject to some empirical stopping criteria). The first step of computing Z is parallelizable, while the second based on standard LP package is not. The key difference between the existing D2-clustering and our new algorithm is the approach to solve U, Π given Z.

3. ALGORITHM

We now refer to the new algorithm the new D2-clustering algorithm. The outer level iteration of the new D2-clustering remains the same as the existing D2-clustering, which is an iteration of cluster label assignment versus centroid update in the same style as K-means. We use Algorithm 1 to update the centroid of each updated partition with a warm start from the previous centroid, which is the technical core.

Suppose we have initialized a solution. As explained in the previous section, we iterate the update of Z and (U, Π). With fixed Z, the update of U and Π is a linear programming problem:

$$\begin{aligned} \min_{\Pi, U} \quad & \sum_k \sum_{i,j} c_{i,k,j} \pi_{i,j}^{(k)} \\ s.t. \quad & \textstyle\sum_{j=1}^{M} \pi_{i,j}^{(k)} = w_i^{(k)}, \quad i = 1, \ldots, m_k,\ k = 1, \ldots, N \\ & \textstyle\sum_{i=1}^{m_k} \pi_{i,j}^{(k)} = u_j, \quad j = 1, \ldots, M,\ k = 1, \ldots, N \\ & \textstyle\sum_{j=1}^{M} u_j = 1 \\ & \pi_{i,j}^{(k)} \geq 0, u_j \geq 0, \quad \forall i, j, k \end{aligned} \tag{3}$$

where $c_{i,k,j} = \left\| x_i^{(k)} - z_j \right\|^2$. Directly solving this problem via LP [11], usually an interior-point based optimization, is not scalable. One major reason is the time complexity of linear programming (3) is super-linear with respect to the number of samples, N. Besides, since our interest of the solution is U, rather than Π, a generic optimization routine treating all variables equally may not in-design come up with an acceptable approximation to U after a few iterations. Therefore, a hierarchical optimization approach is much more desired. Using ADMM, our algorithm decomposes the original centroid optimization problem described by (2) into iterative and distributed small-size subproblems, which can be efficiently solved by standard optimization packages.

Because problem (3) has multiple sets of constraints including equalities and inequalities, applying ADMM to solve the optimization is not straightforward. We propose to relax all equality constraints related to U by substituting with

Fig. 1. Visualization of the clustering result. For each cluster, five images nearest to the centroid are displayed. The color stripe images show the computed color-descriptors of the centroids.

corresponding augmented Lagrangians and use the rest to determine a convex set on which we are to optimize. As in the method of multipliers, we form the scaled augmented Lagrangian $L_\rho(\Pi, U, \Lambda, \mu)$ for the second and third set of constraints (with different penalty parameters τ) in (3) as

$$\begin{aligned} L_\rho(\Pi, U, \Lambda, \mu) &= \sum_k \sum_{i,j} c_{i,k,j} \pi_{i,j}^{(k)} \\ &+ \rho \sum_{j,k} \lambda_{j,k} \left(\sum\nolimits_{i=1}^{m_k} \pi_{i,j}^{(k)} - u_j \right) + \tau\rho\mu \left(\sum\nolimits_{j=1}^{M} u_j - 1 \right) \\ &+ (\rho/2) \sum_{j,k} \left(\sum\nolimits_{i=1}^{m_k} \pi_{i,j}^{(k)} - u_j \right)^2 + \tau(\rho/2) \left(\sum\nolimits_{j=1}^{M} u_j - 1 \right)^2. \end{aligned} \tag{4}$$

Here ρ is a parameter to balance the objective function and augmented Lagrangians. ADMM is an algorithm built to blend the decomposability of dual ascent with superior convergence properties of the method of multipliers. The algorithm typically solves problem with two sets of variables (for our problem, they are $\Pi = (\pi_{i,j}^{(k)})$ and $U = (u_1, \ldots, u_M)$), which are only coupled in constraints, *i.e.*, with the objective function separable across this splitting (for our problem, U is not present in the objective function). Problem (3) can be solved using ADMM iteratively as follows,

$$\Pi^{n+1} := \operatorname*{argmin}_{\Pi \in \Omega_\Pi} L_\rho(\Pi, U^n, \Lambda^n, \mu^n) \tag{5a}$$

$$U^{n+1} := \operatorname*{argmin}_{U \in R_+^M} L_\rho(\Pi^{n+1}, U, \Lambda^n, \mu^n) \tag{5b}$$

$$\begin{aligned} \lambda_{j,k}^{n+1} :=& \lambda_{j,k}^n + \sum\nolimits_{i=1}^{m_k} \pi_{i,j}^{(k),n+1} - u_j^{n+1} \\ \mu^{n+1} :=& \mu^n + \sum\nolimits_{j=1}^{M} u_j^{n+1} - 1, \end{aligned} \tag{5c}$$

where $\Omega_\Pi = \{(\pi_{i,j}^{(k)}) | \sum_{j=1}^M \pi_{i,j}^{(k)} = w_i^{(k)}, \pi_{i,j}^{(k)} \geq 0\}$. The key motivation to use ADMM is that step (5a) can be separable, such that we instead are to solve N disjoint constrained quadratic programming, for $k = 1, \ldots, N$ respectively:

$$\begin{aligned} \min_{\pi_{i,j}^{(k)}} \quad & \sum_{i,j} c_{i,k,j} \pi_{i,j}^{(k)} + (\rho/2) \sum_{j=1}^{M} \left(\sum_{i=1}^{m_k} \pi_{i,j}^{(k)} - u_j^n + \lambda_{j,k}^n \right)^2 \\ \text{s.t.} \quad & \sum\nolimits_{j=1}^{M} \pi_{i,j}^{(k)} = w_i^{(k)} \quad i = 1, \ldots, m_k \\ & \pi_{i,j}^{(k)} \geq 0 \quad \forall i, j. \end{aligned} \tag{6}$$

Since we need to solve small-size problem (6) in multiple rounds of loop, we prefer active set method with warm start.

We can rewrite step (5b) as

$$\min_{u_j > 0} \quad \sum_{j,k} \left(d_{j,k}^{n+1} - u_j \right)^2 + \tau \left(\sum\nolimits_{j=1}^{M} u_j - 1 + \mu^n \right)^2, \tag{7}$$

where $d_{j,k}^{n+1} = \sum_{i=1}^{m_k} \pi_{i,j}^{(k),n+1} + \lambda_{j,k}^n$.

A reasonable termination criterion as suggested by [16] is that the primal and dual residuals must be small, *i.e.*,

$$\begin{aligned} \sum_{j,k} \left\| \sum_{i=1}^{m_k} \pi_{i,j}^{(k),n+1} - u_j^{n+1} \right\|^2 + \tau \left\| \sum_{i=1}^{M} u_j^{n+1} - 1 \right\|^2 \leq \epsilon^{\text{pri}} \\ \text{and } \sum\nolimits_{j=1}^{M} \left\| u_j^{n+1} - u_j^n \right\|^2 \leq \epsilon^{\text{dual}}. \end{aligned} \tag{8}$$

Empirically, we found a fixed number of iterations is sufficient for obtaining satisfactory solutions. We summarize the computation of centroid in Algorithm 1.

The time complexity of Algorithm 1 is $\hat{t}\hat{n}N\sigma(M^2)$, where $\hat{t}$ is the average number of iterations regarding iterator t, $\hat{n}$ is the average number of iterations regarding iterator n, and $\sigma(M^2)$ is the average time to solve sample-wise subproblem (6), which is in general super-linear with respect to M^2. Because the algorithm takes linear complexity in N, the computational cost on large dataset can therefore be effectively distributed via parallelization.

The implementation of Algorithm 1 involves the specification of parameter ρ and τ. Empirically, we found setting $\rho \sim |c|$, where c is the mean of $\{|c_{i,k,j}|\}$, and $\tau \sim N$ is adequate to obtain convergence in a few iterations. We prototype our proposed new approach using MATLAB Parallel Computing Toolbox.

4. EXPERIMENT

We conduct two experiments to evaluate our proposed algorithm. In the first experiment, we generate synthetic discrete distributions, and compare the running time and solution of our Algorithm 1 to the original interior point LP in solving problem (3). In generating synthetic data, we fix the number of supports m_k to 5, set the dimension of support $x_i^{(k)}$ to 3, and independently sample each scalar component of $x_i^{(k)}$ and $w_i^{(k)}$ from the uniform distribution on $[0, 1]$. All sampled

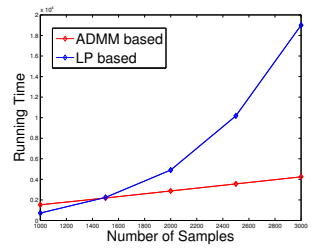
Input: given set of observations $L = \{p^{(1)}, \ldots, p^{(N)}\}$ and an initial guess $\beta^{(0)} = (Z^{(0)}, U^{(0)})$.
Output: centroid $\beta^{(t)} = (Z^{(t)}, U^{(t)})$.

Initialize $\pi_{i,j}^{(k),(0)}$ via minimizing $D(p^{(k)}, \beta^{(0)})$ in (1) and set $t = 1$;
while *termination criterion not satisfied* **do**
  update Z: update $Z^{(t)}$ from $\Pi^{(t-1)}$ based on $z_j^{(t)} = \frac{1}{N} \sum_{j,k} \pi_{i,j}^{(k),(t-1)} x_i^{(k)}$;
  update U: Initialize $\Lambda^0 = 0$ and $\mu^0 = 0$, and reset $U^0 = U^{(t-1)}$ and $n = 1$;
  while *termination criterion* (8) *not satisfied* **do**
    for $k = 1, \ldots, N$ **do**
      update $\pi_{i,j}^{(k),n}$ from U^{n-1}, $\lambda_{j,k}^{n-1}$ and μ^{n-1} based on Eq. (6);
      $d_{j,k}^n := \sum_{i=1}^{m_k} \pi_{i,j}^{(k),n} + \lambda_{j,k}^n \quad j = 1, \ldots, M;$
    end
    update U^n from $\{d_{j,k}^n\}$, Λ^{n-1}, μ^{n-1} based on (7);
    update Λ^n and μ^n from Π^n and U^n based on (5c);
    increase n;
  end
  renormalize U^n; $U^{(t)} = U^n$;
  increase t;
end
Output centroid $\beta^{(t)} = (Z^{(t)}, U^{(t)})$;

Algorithm 1: Compute centroids using ADMM.

$w_i^{(k)}$'s are then normalized to form valid probabilities. Given the mean and covariance of all supports $x_i^{(k)}$, we use the same initial guess for two methods, which are based on multivariate normal sampling. In this performance experiment, we manually fix the number of iterations as $t = 1, \ldots, 20$ and $n = 1, \ldots, 10$, which are used in Algorithm 1. Detailed statistics about their timings and solution differences are shown in Table 1, respectively. It is shown that the running time of our method grows linearly with the number of samples, while LP based method grows much faster and is super-linear. By comparing solution β of our method to the LP based method, the relative difference $D(\beta_a, \beta_b) / \|Z_b\|$ between them are indeed small ($\leq 3\%$).

In the second experiment, we test our clustering algorithm on a real-world image dataset. We crawl images from flickr.com using 5 distinctive concepts (1000 images each): mountains, sky, water, flowers and house, extract discrete distribution representations based on the color descriptor [11], and perform a 10-centroid clustering using our approach. Because we have tested the computational efficiency using synthetic data, the image dataset used here is not of impressively large size. Moreover, as we can see from the nature of the algorithm, the bottleneck for computation is not the overall size of the data, but the size of individual clusters. We thus feel that as proof-of-concept, the data in the experiments are adequate. After several rounds of assignment-update (i.e., outer level) iterations, the number of data objects that change their cluster labels is less than 2% (100). We visualize computed centroids of the largest clusters by displaying the five images with data objects nearest to the centroids (see Fig 1). In addition, we show cluster-based statistical profiles for each concept (see Fig 2) from both our ADMM based clustering and LP based clustering. We understand that the clusters are not guaranteed to be the exact same because of different initializations and non-unique minimums of centroid based clustering. But from the results shown, both approaches have similar distinguishibility among different concepts based on the distributions of clusters



#samples	Rd
1000	2.9%
1500	1.2%
2000	3.0%
2500	0.8%
3000	1.6%

Table 1. The running time and relative difference (Rd) of two methods computing the centroid of discrete distributions. Red line is our approach, and blue one is interior point based linear programming used in [11]. The statistics are based on our prototypes in a Mac PC with 2.4 GHz dual-core Intel Core i5, 4 GB 1600 MHz DDR3.

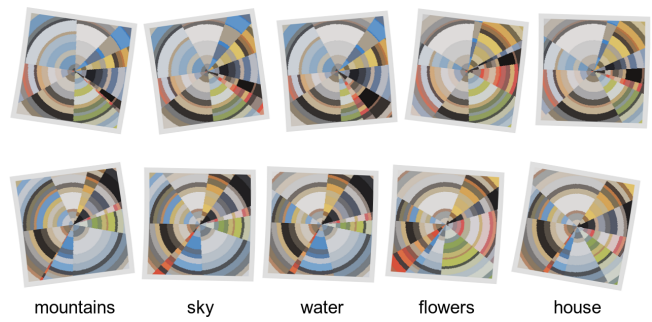


Fig. 2. Cluster-based color profiles of different concepts: mountain, sky and water are similar concepts because of their frequent co-existence in images, but are substantially different from flowers and house. First row: our new approach; Second row: LP based D2-clustering.

5. CONCLUSION

We have developed a new discrete distribution clustering algorithm under the Kantorovich metric using ADMM. The new method is of linear complexity and can be easily parallelized, making it suitable for large scale objects clustering problems. Experiments have been conducted to compare the new method with the existing method in terms of computational efficiency and accuracy.

6. REFERENCES

[1] Svetlozar T Rachev, "The monge-kantorovich mass transference problem and its stochastic applications," *Theory of Probability & Its Applications*, vol. 29, no. 4, pp. 647–676, 1985.

[2] Zhou Ren, Junsong Yuan, and Zhengyou Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 1093–1096.

[3] Ofir Pele and Michael Werman, "Fast and robust earth mover's distances," in *Computer vision, 2009 IEEE 12th international conference on*. IEEE, 2009, pp. 460–467.

[4] Haibin Ling and Kazunori Okada, "An efficient earth mover's distance algorithm for robust histogram comparison," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 5, pp. 840–853, 2007.

[5] David Applegate, Tamraparni Dasu, Shankar Krishnan, and Simon Urbanek, "Unsupervised clustering of multidimensional distributions using earth mover distance," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 636–644.

[6] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[7] Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich, "Displacement interpolation using lagrangian mass transport," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6, pp. 158, 2011.

[8] Marc Wichterich, Ira Assent, Philipp Kranen, and Thomas Seidl, "Efficient emd-based similarity search in multimedia databases via flexible dimensionality reduction," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008, pp. 199–212.

[9] Sameer Shirdhonkar and David W Jacobs, "Approximate earth movers distance in linear time," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[10] Yu Tang, U Leong Hou, Yilun Cai, Nikos Mamoulis, and Reynold Cheng, "Earth movers distance based similarity search at scale," *Proceedings of the VLDB Endowment*, vol. 7, no. 4, 2013.

[11] Jia Li and James Ze Wang, "Real-time computerized annotation of pictures," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 6, pp. 985–1002, 2008.

[12] Yu Zhang, James Z Wang, and Jia Li, "Parallel d2-clustering: Large-scale clustering of discrete distributions," *arXiv preprint arXiv:1302.0435*, 2013.

[13] Antonio Irpino, Rosanna Verde, and Francisco de AT De Carvalho, "Dynamic clustering of histogram data based on adaptive squared wasserstein distances," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3351–3366, 2014.

[14] R. Glowinski and A Marroco, "Sur lapproximation, par elements nis dordre un, et la resolution, par penalisation- dualite, dune classe de problems de dirichlet non lineares," *Revue Francaise dAutomatique, Informatique, et Recherche Operationelle*, vol. 9, no. 2, pp. 41–76, 1975.

[15] Daniel Gabay and Bertrand Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[16] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[17] Daniel Gabay, "Chapter ix applications of the method of multipliers to variational inequalities," *Studies in mathematics and its applications*, vol. 15, pp. 299–331, 1983.

[18] Jonathan Eckstein and Dimitri P Bertsekas, "On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1-3, pp. 293–318, 1992.