Dirac Mixture Reduction Using Wasserstein Distances on Projected Cumulative Distributions

Dominik Prossel and **Uwe D. Hanebeck**
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
dominik.prossel@kit.edu, uwe.hanebeck@ieee.org

***Abstract*—The reapproximation of discrete probability densities is a common task in sample-based filters such as the particle filter. It can be viewed as the approximation of a given Dirac mixture density with another one, typically with fewer samples. In this paper, the Wasserstein distance is established as a suitable measure to compare two Dirac mixtures. The resulting minimization problem is also known as location-allocation or facility location problem and cannot be solved in polynomial time. Therefore, the well-known sliced Wasserstein distance is introduced as a replacement and its ties to the projected cumulative distribution (PCD) are shown. An iterative algorithm is proposed to minimize the sliced Wasserstein distance between the given distribution and approximation.**

***Index Terms*—Deterministic sampling, density reapproximation, Dirac mixtures, nonlinear filtering, least-squares, Wasserstein distance.**

I. Introduction

Sample-based filters play an important role when dealing with nonlinear filtering problems. In these filters, the involved probability densities are represented by a discrete set of samples or particles. This allows them to handle the complex densities that can arise when working with nonlinear models.

A common step in such filters is the reapproximation or resampling of the sample density to keep a good representation of the actual density without the use of exponentially many samples. For example, in a particle filter this is often done with sequential importance resampling. In the filter step, each particle is weighted according to the measurement likelihood. This resulting density is then reapproximated with unweighted samples by randomly selecting samples with replacement according to their weight. Without this step, the filter can either suffer from particle degeneracy or has to increase the number of samples to be able to keep more samples in relevant regions. This makes fast and accurate resampling and sample reduction basic building blocks for nonlinear filtering. In this paper, the problem of reapproximating a set of samples with a new set containing fewer samples is investigated. The new samples should be placed in a way to optimally represent the given samples as illustrated for two dimensions in Fig. 1.

To this end, the samples are first represented as Dirac mixture (DM) distributions

$$f(\underline{x}; \hat{X}, W) = \sum_{i=1}^{N} w_i \delta(\underline{x} - \hat{\underline{x}}_i) \ . \tag{1}$$

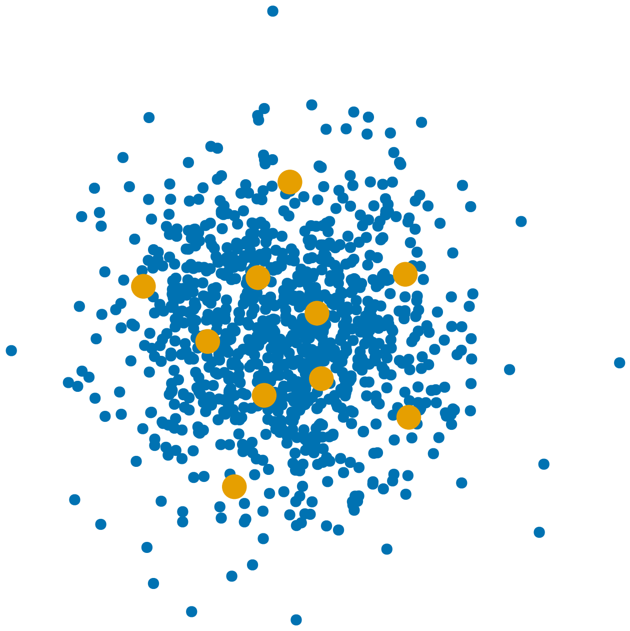


Fig. 1: Result of reducing 1000 random samples of a standard normal distribution to 10 samples using projected cumulative distributions.

They consist of N Dirac impulses at the sample positions $\hat{X} = \{\hat{x}_0, \dots, \hat{x}_N\}$ and the according weights $W = \{w_0, \dots, w_N\}$ for each sample, which sum to one.

When these weights are chosen to be equal for all Dirac impulses as $1/N$, the distribution is defined only by the sample positions. This essentially makes the mixture unweighted, as no additional information is encoded in the weights. When they are chosen differently for each sample, it is possible to encode different probability distributions without altering the sample positions, which is used in the filter step of a particle filter.

Throughout this paper it is assumed, that both the given and the reduced DM are equally weighted. This restriction is made because it is in general NP-hard to find the weighted reduced DM that optimally represents the original DM, even in one dimension [1]. In contrast, finding the best equally weighted approximation can be done in polynomial time. Furthermore, when reducing a weighted DM to an equally weighted DM some additional challenges would arise, which will be discussed towards the end of the paper.

A. *Optimal Reduction*

One approach to optimal DM reduction is the optimization of some measure of quality of the approximation.

There are different measures to define such measures of quality. A straightforward approach would be to compare the probability density functions (PDFs) of the involved distributions. This can for example be done by calculating some metric based on the difference of the PDFs or also some more specialized methods like the Kullback-Leibler divergence [2].

However, all of these methods only give meaningful results when the PDFs of both of the involved distributions are nonzero on the same sample space. This is not the case when DM densities are involved as they are zero almost everywhere, rendering these distances unusable for DM reduction.

Instead, it is much easier to employ the cumulative density functions (CDFs) of the DMs in the measure of quality [3]. The CDF of a scalar Dirac mixture is a staircase function, which is well defined on a continuous support and can be easily compared to other functions. Unfortunately, when moving to more than one dimension, the CDF of a probability distribution is not uniquely defined anymore [3].

One method to get around this limitation is the use of localized cumulative distributions (LCDs) [3]. This technique has also been used for Dirac mixture reduction both in Euclidean space [4] and on spheres [5], [6]. It creates a smooth representation of the involved densities by integrating over kernels of all different sizes. While this works for all kinds of densities, it is quite computationally expensive.

In this paper, the idea of PCD introduced in [7] will be extended with the Wasserstein distances as a measure of quality for optimal Dirac mixture reduction. We will first give some background on the Wasserstein distance and optimal transport theory. The task of Dirac mixture reduction is then modeled as allocation-location problem stemming from the field of logistics. It will be shown how this problem can be solved approximately using PCDs and how these relate to the sliced Wasserstein distance [8].

Furthermore, the resulting optimization problem will be reformulated as a least-squares problem and an efficient iterative solution algorithm will be proposed.

B. *Wasserstein Distance*

The p-Wasserstein distance is a popular measure to compare two continuous distributions $\mu(x)$ and $\nu(y)$ according to a distance function $d(\underline{x},\underline{y})$

$$W_p(\mu,\nu) := \left(\inf_{\gamma\in\Gamma(\mu,\nu)} \int d\left(\underline{x}-\underline{y}\right)^p \mathrm{d}\gamma(\underline{x},\underline{y})\right)^{\frac{1}{p}} \quad . \tag{2}$$

It has its origin in transportation theory and is the optimal value of the optimal transport problem to find the transport plan γ that minimizes the overall cost to transform one of the distributions into the other one. It is also known as *Earth Mover's* distance [9] as the transport plan can be interpreted as describing how much earth or mass is moved from every point $\underline{x}$ in the support of μ to every point $\underline{y}$ in the support of ν.

The distance function describes the cost to move mass between the two points $\underline{x}$ and $\underline{y}$ and is often chosen to be the Euclidean or Manhattan distance. In the remainder of this paper the 2-Wasserstein distance in combination with the Euclidean distance as metric will be used.

Another way to think of the Wasserstein distance is that at each point of one of the distributions a specific demand is present. Each point in the other distribution can satisfy a specific amount of this demand, such that the overall supply and demand are the same. The optimal transport plan now assigns every point of supply to every point of demand, so that the demand is satisfied with minimal transportation cost.

The Wasserstein distance is also widely used to compare discrete densities like image data [8] and has also been used as loss function in neural networks [10].

In the case that both distributions are Dirac mixtures, the integral in the Wasserstein distance reduces to a sum and the transport plan is a sparse matrix describing how much mass is moved between two samples.

Given the two Dirac mixtures $f(\underline{x};\hat{X}) = \frac{1}{N}\sum_i^N \delta(\underline{x}-\hat{\underline{x}}_i)$ and $g(\underline{y};\hat{Y}) = \frac{1}{M}\sum_j^M \delta(y-\hat{\underline{y}}_j)$, the optimal transport matrix $\mathbf{T}$ with entries t_{ij} and the 2-Wasserstein distance can be found by solving the linear program

$$W(f,g) = \min_{t_{ij}} \sum_i^N \sum_j^M t_{ij} \|\hat{\underline{x}}_i - \hat{\underline{y}}_j\|^2 \tag{3a}$$

$$\text{s.t.} \sum_i^N t_{ij} = \frac{1}{M} \quad \forall j \tag{3b}$$

$$\sum_j^M t_{ij} = \frac{1}{N} \quad \forall i \tag{3c}$$

$$t_{ij} \geq 0\,. \tag{3d}$$

The constraints (3b) to (3d) define the set of all valid transport plans. This set is often called the transport polytope. These constraints make sure that the supply and demand at each point are satisfied and all mass is moved between the two distributions.

C. *Optimizing the Sample Positions*

The Wasserstein distance can be used as a measure of quality between distributions and as such for DM reduction. Given a distribution with sample positions $\hat{Y} = \{\hat{\underline{y}}_0,\ldots,\hat{\underline{y}}_M\}$ the optimal reduced DM can be found by minimizing the Wasserstein distance with respect to the new sample positions $\hat{X} = \{\hat{\underline{x}}_0,\ldots,\hat{\underline{x}}_N\}$.

For a given transport plan $\mathbf{T}$ the goal is to find the positions of unweighted samples, so that the transport cost between these samples and the given distribution is minimized. This transport cost for a fixed matrix $\mathbf{T}$ is simply given as

$$D(\hat{X},\hat{Y}) = \sum_{i=1}^{N}\sum_{j=1}^{M} t_{ij} \|\hat{\underline{x}}_i - \underline{\hat{y}}_j\|^2 \ . \tag{4}$$

Minimizing this with respect to the new sample positions gives the unconstrained convex optimization problem

$$\min_{\hat{\underline{x}}_i} D(\hat{X},\hat{Y}) \ . \tag{5}$$

The gradient of $D(\hat{X},\hat{Y})$ is given as a vector G with entries

$$G_i = \frac{\mathrm{d}D}{\mathrm{d}\hat{\underline{x}}_i} = 2\sum_{i=1}^{N}\sum_{j=1}^{M} t_{ij}(\hat{\underline{x}}_i - \underline{\hat{y}}_j) \ . \tag{6}$$

Setting this gradient equal to zero and using that the sum over one row of the transport plan, $\sum_{j=1}^{M} w_{ij}$, is $1/N$ the optimal position of the i-th sample $\hat{\underline{x}}_i$ can be determined in closed form

$$\hat{\underline{x}}_i = N\sum_{j=1}^{M} t_{ij}\underline{\hat{y}}_j \tag{7}$$

This is the weighted average of all given samples, that the new sample is assigned to, weighted with the amount that is assigned.

However, the transport plan is typically not given for DM reduction and needs to be found according to (3). But in addition, the sample positions $\hat{\underline{x}}_i$ now become decision variables as well. This type of problem is also known as location-allocation problem or (Capacitated) Multi-Source Weber problem in different variants [11]. Unfortunately, this problem is now a non-convex nonlinear optimization problem, making it much harder, in fact NP-hard, to solve.

Calculating the exact solution to this problem is only feasible for relatively small instances with hundreds of samples. A general and exact solution method would be for example to enumerate all vertices of the transport polytope, that is described by the constraints of the problem and find the one with the smallest Wasserstein distance. This number of vertices depends on the number of possible assignments between the original and reduced DM, which grows extremely fast with increasing number of samples.

There also exist some special cases, where further assumptions about the solution can be made [11]. One such special case, which will later be used to derive a new approximate solution algorithm, occurs, when all samples are located on a line, reducing the problem to a single dimension.

An approximate algorithm was first proposed by Cooper [12] and is also known as Alternating Transportation-Location method or Cooper-ATL. It is an iterative scheme starting with some initial guess for the sample positions and alternates between solving the transport problem and the positioning problem with each step. As such, the solution found is highly dependent on the starting values used. Over time many different heuristics have been proposed to guide this algorithm closer to the global optimum [13], [14].

II. Projections onto one dimension

The concept of PCDs [7] will now be introduced and used as a basis to propose a new algorithm to approximately solve the location-allocation problem. PCDs were originally introduced for deterministic sampling of continuous probability densities in Euclidean space. However, they have also been extended for use on different manifolds, for example, circles [15]. The main idea behind PCDs is to represent the given distribution and the sample distribution by the set of all of their one-dimensional projections, known as the Radon transform. The goal now is to find sample positions, so that the difference between these projected distributions is minimal for each projection. By doing so, the optimization is broken down from one multi-dimensional problem into infinitely many one-dimensional problems, that are easier to solve. To make the Radon transform computationally tractable, a finite subset of projections needs to be chosen. This is commonly done by uniform random sampling [16] or deterministic sampling [7] from the according hypersphere.

The sample positions are then found by minimizing the average Cramér-von Mises distance $\tilde{C}$ over all projections. The Cramér-von Mises distance C between two one-dimensional probability distributions with CDFs $F(x)$ and $G(x)$ is defined as

$$C(F(x), G(x)) = \int_{\mathbb{R}} \left(F\left(x\right) - G\left(x\right)\right)^2 \mathrm{d}x \ . \tag{8}$$

This means that a given density $f(\underline{x})$ with projected CDFs $F_0(r), \ldots, F_V(r)$ for V projection vectors $\underline{v}_0, \ldots, \underline{v}_V$, is approximated by a DM $g(\underline{x}; \hat{X})$ with projected CDFs $G(r; \hat{R}_k), \ldots, G(r; \hat{R}_V)$ by minimizing

$$\tilde{C}\left(f\left(\underline{x}\right), g\left(\underline{x}; \hat{X}\right)\right) = \frac{1}{V}\sum_{k=1}^{V} C\left(F_k\left(r\right), G\left(r; \hat{R}_k\right)\right) \ . \tag{9}$$

Note, that the one-dimensional projection of a DM with N components is another DM with impulses at the projected positions $\hat{R}_k = \{\underline{v}_k^\top \hat{\underline{x}}_0, \ldots, \underline{v}_k^\top \hat{\underline{x}}_N\}$. The CDF is then a staircase function with steps at these positions

$$G(r; \hat{R}_k) = \sum_{i=1}^{N} H(r, \hat{r}_i) \ . \tag{10}$$

Here, the Heaviside function $H(r, \hat{r})$ is defined as

$$H(r,\hat{r}) = \begin{cases} 0 & r < \hat{r} \\ 0.5 & r = \hat{r} \\ 1 & r > \hat{r} \end{cases} \ . \tag{11}$$

To minimize (9), gradient-based optimization algorithms can be used. So far, PCDs have not been used for Dirac mixture reduction or solving the location-allocation problem. One reason for this is that the Cramér-von Mises distance is not well suited as an objective function in this case. When both of the distributions are Dirac mixtures, there is in general no unique solution that minimizes this distance.

This is evident, when looking at an example of reducing two samples to one: Two given one-dimensional samples are located at $\hat{y}_1$ and $\hat{y}_2$ and the optimal sample position $\hat{x}$ needs to be found. Intuitively, $\hat{x}$ should be at the center between the two samples in a unique minimum. However, the Cramér-von Mises distance yields the same result for all $\hat{x}$ between the two original samples. When replacing the distance measure with the 2-Wasserstein distance, the optimal solution to the above problem is unique and lies in the center of $\hat{y}_1$ and $\hat{y}_2$.

Plugging the Wasserstein distance into (9) instead of the Cramér-von Mises distance yields

$$\tilde{W} = \frac{1}{V}\sum_{k=1}^{V}\min_{t_{ij}}\sum_{i=1}^{N}\sum_{j=1}^{M} t_{ij}\left(\underline{v}_k^\top \hat{\underline{x}}_i - \underline{v}_k^\top \hat{\underline{y}}_j\right)^2 \quad . \tag{12}$$

This quantity $\tilde{W}$ is also known as the sliced Wasserstein distance [8]. It has for example been successfully employed for texture synthesis [17], generative neural networks [18], [19] or interpolation between measures [20]. It makes use of the fact that one-dimensional optimal transport can be calculated efficiently without explicitly solving the linear program (3).

In this case, the optimal transport plan between two DMs does not depend on the absolute sample positions, but only on their weights and their order on the real line [1]. This means, that the transport plan between two DMs with M and N components with weights $1/M$ and $1/N$ can be calculated by sorting the samples on the real line and filling the transport matrix sequentially as detailed in Algorithm 1. To calculate the sliced Wasserstein distance for fixed sample positions, this transport matrix can stay the same for all projections, as long as the projected samples are sorted. With the orderings $\sigma(i)$ and $\tau(j)$, this simplifies (12) to

$$\tilde{W} = \frac{1}{V}\sum_{k=1}^{V}\sum_{i=1}^{N}\sum_{j=1}^{M} t_{ij}\left(\underline{v}_k^\top \hat{\underline{x}}_{\sigma(i)} - \underline{v}_k^\top \hat{\underline{y}}_{\tau(j)}\right)^2 \tag{13}$$

without the explicit minimization.

III. Optimizing Projected Positions

While the sliced Wasserstein distance is a well established substitute for the Wasserstein distance, it has not been used as an objective function for optimal sample placement or DM reduction. To find the optimal sample positions, the sliced Wasserstein distance in (13) needs to be minimized with respect to the positions. This leads to a very similar problem to the location-allocation problem. However, due to the one-dimensional projections, the optimal transport plan in each direction can be calculated easily using algorithm 1. Given this transport plan, the optimal projected sample positions can be found with (7).

Overall this means that the optimal sample positions for each projection can be computed efficiently. The goal now is to find sample positions in the original high-dimensional space, such that their projections are close to the optimal positions for all projection directions. This can be written as a least-squares problem

$$\min_{\underline{x}} \|\mathbf{R}\underline{x} - \underline{b}\|^2 \quad , \tag{14}$$

where the positions of all samples are stacked in the vector $\underline{x}$.

Algorithm 1 Calculation of the optimal transport plan between two equally weighted, one-dimensional Dirac mixtures with M and N samples respectively. It is assumed that the samples are already sorted by position.

```
function TransportPlan($M$, $N$)
    $T^{M\times N} \leftarrow 0.0$
    $n,\ m,\ sum,\ a \leftarrow 0$
    for $i$ from $1$ to $M+N-1$ do
        if $\frac{n}{N} < \frac{m}{M}$ then
            $t_{mn} \leftarrow \frac{n}{N} - sum$
            $a \leftarrow t_{mn}$
            $n \leftarrow n+1$
        else
            $t_{mn} \leftarrow \frac{m}{M} - sum$
            $a \leftarrow t_{mn}$
            $m \leftarrow m+1$
        end if
        $sum \leftarrow sum + a$
    end for
return $T$
end function
```

For each of the V directions $\underline{v}_k$, the optimal projected positions $\underline{X}_k^* = (x_{1,k}^*, \ldots, x_{N,k}^*)^\top$ are calculated according to (7). The results are collected in the right-hand-side vector of the least-squares problem $\underline{b}$

$$\underline{b} = \begin{bmatrix}\underline{X}_1^* & \underline{X}_2^* & \ldots & \underline{X}_V^*\end{bmatrix}^\top \quad . \tag{15}$$

The matrix R on the left-hand side is a collection of matrices $\mathbf{R}_k$ that are also calculated for each direction

$$\mathbf{R} = \begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2 & \ldots & \mathbf{R}_V\end{bmatrix}^\top \quad . \tag{16}$$

Each $\mathbf{R}_k$ projects the vector of samples $\underline{x}$ onto direction $\underline{v}_k$ and sorts the projected samples according to their positions. Therefore it consists of the projection matrix

$$\mathbf{V}_k = \begin{bmatrix}\underline{v}_k^\top & & \\ & \ddots & \\ & & \underline{v}_k^\top\end{bmatrix} \tag{17}$$

and the permutation matrix $\mathbf{P}_k$ that sorts $\underline{x}^\top \underline{v}_k$ as

$$\mathbf{R}_k = \mathbf{P}_k \mathbf{V}_k \quad . \tag{18}$$

It is clear that the permutation matrix depends on the optimal sample positions and vice versa. To break this dependency, an iterative algorithm alternating between two steps is proposed. First, the sample positions are optimized by solving the least-squares problem. Then, the new permutation matrix is calculated from the found positions as described in Algorithm 2. As an initial guess for the matrix, some random sample positions are used. The algorithm terminates either when a maximum number of iteration is reached or the changes of the sample positions is small enough. In Fig. 2, the sparse structure of these matrices is shown. It can be seen

Algorithm 2 Proposed iterative algorithm for Dirac mixture reduction with the M given sample positions Y and initial approximation X with N samples using V directions.

```
function DIRACREDUCTION($Y$, $X$, $V$, $iters$)
    $W \leftarrow$ TRANSPORTPLAN(M, N)
    $R \leftarrow [\,], b \leftarrow [\,]$
    for $i$ from 1 to $iters$ do
        for $k$ from 1 to $V$ do
            $v_k \leftarrow$ sampleUnitVector()
            concatenate($b, NW$sort$(Y^\top v_k)$)
            $P_k \leftarrow$ permutationMatrix($X^\top v_k$)
            $V_k \leftarrow$ blockdiag($v_k^\top, N$)
            concatenate($R, P_k V_k$)
        end for
        $X \leftarrow (R^\top R)^{-1} R b$
    end for
    return $X$
end function
```

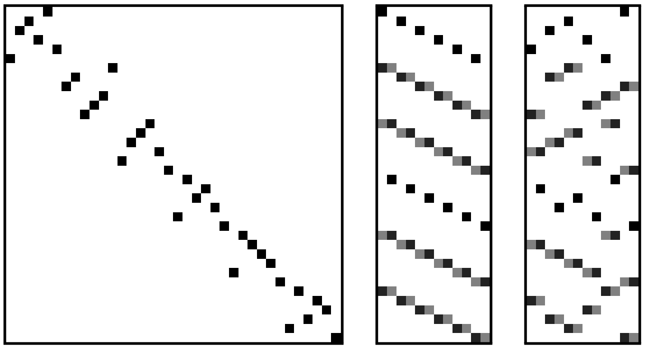

(a) (b) (c)

Fig. 2: Sparsity structure of (a) the permutation matrix $\mathbf{P}$, (b) projection matrix $\mathbf{V}$ and (c) overall matrix $\mathbf{R}$ of a problem with $N = 6$ samples and $V = 6$ projections.

that there are no dependencies between the different projection directions.

This can be used to reformulate the problem as a recursive least-squares problem, incorporating each direction one by one. This means that the complete matrix $\mathbf{R}$ does not have to be computed at once, but only one $\mathbf{R}_k$ at a time. The recursive least-squares algorithm needs some initialization for the covariance matrix $\mathbf{C}_0$, which is typically chosen to be large, for example $\mathbf{C}_0 = 100\,\mathbf{I}$. The weight matrix for each direction is chosen to be the identity matrix. Hence, the gain only depends on the initial covariance and left hand side matrices $\mathbf{R}_k$

$$\mathbf{K}_k = \mathbf{C}_k \mathbf{R}_k^\top \left(\mathbf{I} + \mathbf{R}_k \mathbf{C}_k \mathbf{R}_k^\top\right)^{-1} \; . \tag{19}$$

The updates for the sample position and covariance are the standard equations for recursive least square filtering

$$\underline{x}_{k+1} = \underline{x}_k + \mathbf{R}_k \left(\underline{b}_k - \mathbf{R}_k \underline{x}_k\right) \; , \tag{20}$$

$$\mathbf{C}_{k+1} = (\mathbf{I} - \mathbf{K}_k \mathbf{R}_k)\mathbf{C}_k \; . \tag{21}$$

The formulation above can be broken down even further, by separating $\mathbf{R}_k$ into its projection and permutation part from (18). The permutation matrix matches each of the projected sample positions with one entry from the right-hand side. It can easily be inverted and the inverse applied to the right hand side vector instead to get the permuted vector $\underline{b}'_k$. This leaves the projection operation for the left-hand side, which is identical for each sample. As there are no dependencies between the samples now, the updates to their positions can be performed independently for each sample

$$\underline{x}_{i,k+1} = \underline{x}_{i,k} + \mathbf{K}_{i,k}\left(\underline{b}'_k - \underline{v}_k^\top \underline{x}_{i,k}\right) \; . \tag{22}$$

By taking a closer look at the covariance matrix $\mathbf{C}_k$, it can be seen, that it is a block-diagonal matrix with the same $D \times D$ block $\mathbf{\Sigma}_k$ for each sample on the diagonal. This means, that is sufficient to only calculate and save one of these blocks for all samples. The gain from (22) becomes

$$\mathbf{K}_{i,k} = \mathbf{G}_k = \mathbf{\Sigma}_k \mathbf{v}_k \left(1 + \underline{v}_k^\top \mathbf{\Sigma}_k \underline{v}_k\right)^{-1} \tag{23}$$

and the covariance update

$$\mathbf{\Sigma}_{k+1} = \left(\mathbf{I} - \mathbf{G}_k \underline{v}_k^\top\right) \mathbf{\Sigma}_k \; . \tag{24}$$

A. Computational Complexity

As all of the involved algorithms are iterative and showed a similar time to convergence in the practical experiments, in the following the time complexity per iteration is investigated. The number of given samples and approximation samples is summarized in the variable $n = M + N$.

In the original Cooper-ATL algorithm, an optimal transport problem is solved in each iteration. This is equivalent to solving a min-cost-flow problem, which can be done in $O(n^3)$ [21] or $O(n^3 \log(n))$ [9] depending on the algorithm used. The calculation of the optimal positions, given the transport plan runs in $O(nD)$ time by multiplying each of the $O(n)$ entries in the transportation matrix with the corresponding D-dimensional given sample and summing up these products.

In the PCD-based method, no linear program needs to be solved, instead the transport plan is approximated by the one-dimensional projections. In each iteration, n samples are projected and sorted for each of V directions in $O(Vn\log(n))$. As the given samples are not changing between iterations, they actually need to be projected only once, which has a noticeable effect when there are many given samples and few approximation samples, but will be neglected for this analysis. If the given Dirac mixture is equally weighted, the transport plan needs to be calculated only once for the entire algorithm in $O(n)$, otherwise this needs to be repeated for each direction. The optimal positions are then calculated by averaging the best positions in each projection in $O(VDn)$. Overall the dominating cost in the Cooper-ATL lies in finding the solution to the optimal transport problem. In the PCD-based method this is replaced by a sliced version, that can be calculated in $(Vn\log(n))$, where V somewhat controls the quality of

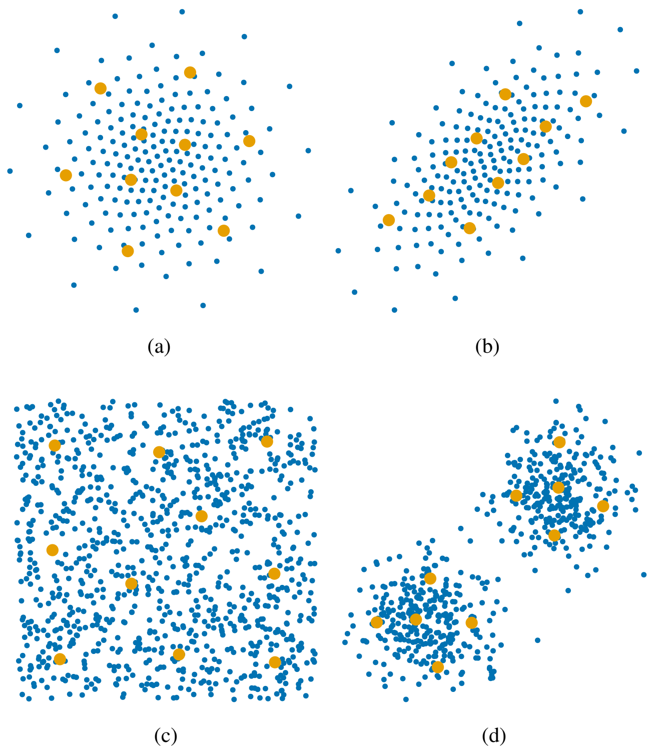


Fig. 3: Some example results for Dirac mixture reduction to 10 samples with the proposed method. (a) 200 samples LCD approximation of default Gaussian (b) 200 samples LCD approximation of correlated Gaussian (c) 1000 samples from uniform distribution (d) 800 samples from Gaussian mixture.

the approximation. On the other hand, the calculation of the optimal positions gets slightly more complex for the proposed method, as the average over all directions needs to be taken.

IV. Evaluation

The proposed algorithm and the Cooper-ATL were implemented in Julia using JuMP [22] to model and solve the occurring linear program. The code including the examples presented in this paper is made available on GitHub[1].

Some example results of reducing different given densities can be seen in Fig. 3. This demonstrates that the proposed method works well for many different distributions.

The proposed method was compared to the Cooper-ATL and k-means clustering. The k-means algorithm is a well established and fast algorithm to split a dataset into clusters and calculated the cluster centroid. It was included in the comparison to evaluate the performance of such clustering algorithms for DM reduction. Each of the algorithms was run for 25 iterations with 100 different random starting positions. The average, minimum, and maximum values of Wasserstein and sliced Wasserstein distance were recorded for all starting positions Fig. 4. As would be expected, the Cooper-ATL yields the smallest final Wasserstein distance on average. The sliced Wasserstein distance outperforms k-means in terms of minimization of the actual Wasserstein distance. This experiment also illustrates the dependency of the final solution on the starting position as there is a significant difference between the maximum and minimum values of the final Wasserstein distance. This can be observed for all three algorithms, but is most prominent with k-means.

In addition to the distances, the final sample positions were also recorded and the best solutions are shown in Fig. 5. As the k-means algorithm has no limit to the number of points in one cluster it covers the support of the given density more or less evenly without respecting the sample density. The other two methods yield visually very similar results and take the increased sample density in the center of the distribution into account. Both give a good approximation of the original DM, with the proposed sliced Wasserstein variant being about ten times faster in this example as it does not need to solve a linear program in each iteration.

V. Conclusion and Outlook

In this paper, a new method for Dirac mixture reduction based on PCDs was proposed. It was first shown how DM reduction can be reinterpreted as the location-allocation problem, known from optimal transport theory. As this leads to an optimization problem that is computationally expensive to solve, an approximate solution algorithm utilizing PCDs was derived. By replacing the Cramér-von Mises distance in the original formulation of PCDs with the Wasserstein distance, the well-known sliced Wasserstein distance between the given and the reduced DM distribution was obtained as an objective function.

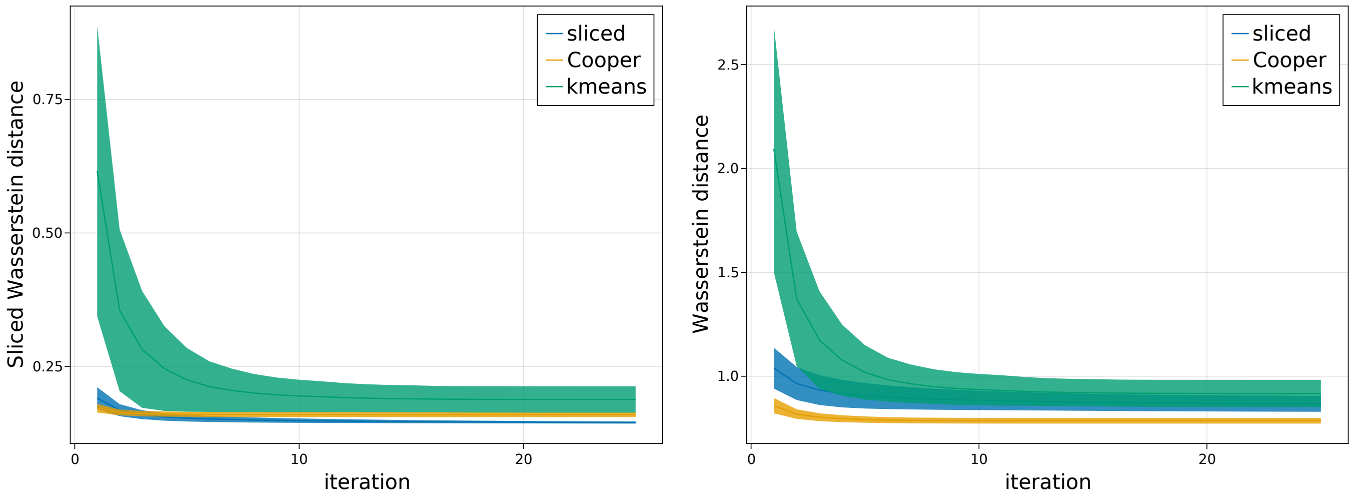
The minimization of the sliced Wasserstein distance with respect to the new sample position was formulated as a least-squares problem, that can be solved with an iterative algorithm. The complexity of this new method is shown to be less than when optimizing the true Wasserstein distance, while still yielding very good results.

However, the results of the proposed method are highly dependent on the initial guess for the new sample positions. Therefore, some techniques that might help the algorithm find a good solution more consistently, should be investigated. For example, it could be advantageous to introduce additional heuristics as was done for the Cooper-ATL. Using a sophisticated initialization scheme for the initial sample positions or permutation matrix itself could also yield more consistent results.

There are also some variants of the sliced Wasserstein distance, like the max-Wasserstein distance, that takes the maximum Wasserstein distance in any direction instead of the average, and also generalized versions [23], [24]. These seem to give improved results for some applications and scenarios and could be worthwhile to investigate for Dirac mixture reduction.

While the proposed method can easily be extended to work with weighted Dirac mixtures or to increase the number of

[1] https://github.com/KIT-ISAS/DiracMixtureReduction



(a)

(b)

Fig. 4: (a) Sliced Wasserstein distance and (b) Wasserstein distance after each iteration of the least-squares, Cooper-ATL and k-means algorithms. The border of the shaded area denotes the minimum and maximum value obtained in each iteration.

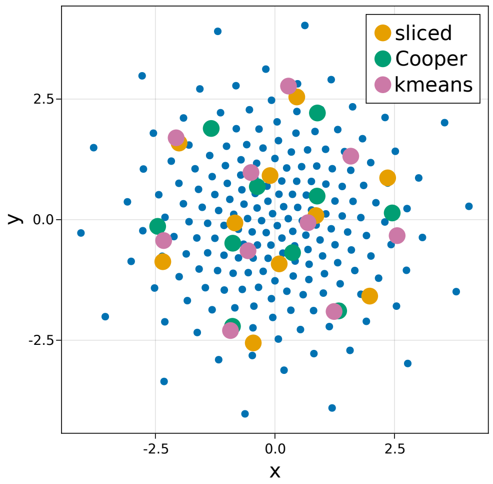


Fig. 5: Comparison of the final positions for k-means, Cooper-ATL and least-squares algorithms with the smallest Wasserstein distance out of 100 restarts with random initial positions.

samples instead of reducing them, the result is not a good approximation of the underlying continuous density anymore. This can immediately be seen, when trying to double the number of samples of a uniform distribution. Two of the new samples would be located at each of the original sample positions, minimizing the transport cost to zero. This is not the desired result of keeping the uniform sample distribution. Some kind of continuous interpolation between samples needs to be used to get around this effect.

Overall the approximation of densities through one-dimensional projections seems to be a promising approach, that is worthy of further investigation.

VI. Acknowledgement

This work was supported by the Competence Center Karlsruhe for AI Systems Engineering (CC-KING, https://www.ai-engineering.eu) sponsored by the Ministry of Economic Affairs, Labour, and Tourism Baden-Württemberg.

References

[1] H. D. Sherali and F. L. Nordai, “NP-Hard, Capacitated, Balanced p-Median Problems on a Chain Graph with a Continuum of Link Demands,” *Mathematics of Operations Research*, Feb. 1988. Publisher: INFORMS.

[2] S. Kullback and R. A. Leibler, “On Information and Sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951. Publisher: Institute of Mathematical Statistics.

[3] U. D. Hanebeck and V. Klumpp, “Localized Cumulative Distributions and a Multivariate Generalization of the Cramér-von Mises Distance,” in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, (Seoul), pp. 33–39, IEEE, Aug. 2008.

[4] U. D. Hanebeck, “Optimal Reduction of Multivariate Dirac Mixture Densities,” *at - Automatisierungstechnik*, vol. 63, pp. 265–278, Apr. 2015. Publisher: Oldenbourg Wissenschaftsverlag.

[5] D. Frisch, K. Li, and U. D. Hanebeck, “Optimal Reduction of Dirac Mixture Densities on the 2-Sphere,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1282–1287, 2020.

[6] K. Li, F. Pfaff, and U. D. Hanebeck, “Hyperspherical Dirac Mixture Reapproximation,” *arXiv:2110.10411 [cs, eess, stat]*, Oct. 2021. arXiv: 2110.10411.

[7] U. D. Hanebeck, “Deterministic Sampling of Multivariate Densities based on Projected Cumulative Distributions,” in *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, (Princeton, NJ, USA), pp. 1–6, IEEE, Mar. 2020.

[8] J. Rabin, G. Peyré, J. Delon, and M. Bernot, “Wasserstein Barycenter and Its Application to Texture Mixing,” in *Scale Space and Variational Methods in Computer Vision* (A. M. Bruckstein, B. M. ter Haar Romeny, A. M. Bronstein, and M. M. Bronstein, eds.), vol. 6667, pp. 435–446, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. Series Title: Lecture Notes in Computer Science.

[9] O. Pele and M. Werman, “Fast and robust Earth Mover’s Distances,” in *2009 IEEE 12th International Conference on Computer Vision*, pp. 460–467, Sept. 2009. ISSN: 2380-7504.

[10] C. Frogner, C. Zhang, H. Mobahi, M. Araya-Polo, and T. Poggio, “Learning with a Wasserstein Loss,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’15, (Cambridge, MA, USA), p. 2053–2061, MIT Press, 2015.

[11] J. Brimberg, P. Hansen, and N. Mladenovi, “A Survey of Solution Methods for the Continuous Location-Allocation Problem,” *International Journal of Operations Research*, vol. 5, p. 12, 01 2008.

[12] L. Cooper, “Heuristic Methods for Location-Allocation Problems,” *SIAM Review*, vol. 6, no. 1, pp. 37–53, 1964. Publisher: Society for Industrial and Applied Mathematics.

[13] Z. M. Zainuddin and S. Salhi, “A perturbation-based heuristic for the capacitated multisource Weber problem,” *European Journal of Operational Research*, vol. 179, pp. 1194–1207, June 2007.

[14] M. Luis, S. Salhi, and G. Nagy, “A guided reactive GRASP for the capacitated multi-source Weber problem,” *Computers & Operations Research*, vol. 38, pp. 1014–1024, July 2011.

[15] D. Frisch and U. D. Hanebeck, “Deterministic Sampling on the Circle Using Projected Cumulative Distributions,” in *Proceedings of the 25th International Conference on Information Fusion (Fusion 2022)*, (Linköping, Sweden), July 2022.

[16] M. Rowland, J. Hron, Y. Tang, K. Choromanski, T. Sarlos, and A. Weller, “Orthogonal Estimation of Wasserstein Distances,” in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics* (K. Chaudhuri and M. Sugiyama, eds.), vol. 89 of *Proceedings of Machine Learning Research*, pp. 186–195, PMLR, 16–18 Apr 2019.

[17] E. Heitz, K. Vanhoey, T. Chambon, and L. Belcour, “A Sliced Wasserstein Loss for Neural Texture Synthesis,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Nashville, TN, USA), pp. 9407–9415, IEEE, June 2021.

[18] J. Wu, Z. Huang, D. Acharya, W. Li, J. Thoma, D. P. Paudel, and L. Van Gool, “Sliced Wasserstein Generative Models,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Long Beach, CA, USA), pp. 3708–3717, IEEE, June 2019.

[19] S. Kolouri, C. E. Martin, and G. K. Rohde, “Sliced-Wasserstein Autoencoder: An Embarrassingly Simple Generative Model,” *ArXiv*, p. 26, 2018.

[20] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, “Sliced and Radon Wasserstein Barycenters of Measures,” *Journal of Mathematical Imaging and Vision*, vol. 51, pp. 22–45, Jan. 2015.

[21] P. Dvurechensky, A. Gasnikov, and A. Kroshnin, “Computational Optimal Transport: Complexity by Accelerated Gradient Descent Is Better Than by Sinkhorn’s Algorithm,” *arXiv:1802.04367 [cs, math]*, June 2018. arXiv: 1802.04367.

[22] I. Dunning, J. Huchette, and M. Lubin, “JuMP: A Modeling Language for Mathematical Optimization,” *SIAM Review*, vol. 59, pp. 295–320, Jan. 2017.

[23] S. Kolouri, K. Nadjahi, U. Simsekli, R. Badeau, and G. Rohde, “Generalized Sliced Wasserstein Distances,” in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[24] F.-P. Paty and M. Cuturi, “Subspace Robust Wasserstein Distances,” in *Proceedings of the 36th International Conference on Machine Learning*, pp. 5072–5081, PMLR, May 2019. ISSN: 2640-3498.