The Wasserstein Distance as a Dissimilarity Measure for Mass Spectra with Application to Spectral Deconvolution

Szymon Majewski[1]
Institute of Mathematics, Polish Academy of Sciences, Warsaw, Poland
smajewski@impan.pl

Michał Aleksander Ciach[1]
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Warsaw, Poland
m_ciach@student.uw.edu.pl

Michał Startek
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Warsaw, Poland

Wanda Niemyska
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Warsaw, Poland

Błażej Miasojedow
Institute of Mathematics, Polish Academy of Sciences, Warsaw, Poland

Anna Gambin
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Warsaw, Poland

Abstract

We propose a new approach for the comparison of mass spectra using a metric known in the computer science under the name of Earth Mover's Distance and in mathematics as the Wasserstein distance. We argue that this approach allows for natural and robust solutions to various problems in the analysis of mass spectra. In particular, we show an application to the problem of deconvolution, in which we infer proportions of several overlapping isotopic envelopes of similar compounds. Combined with the previously proposed generator of isotopic envelopes, IsoSpec, our approach works for a wide range of masses and charges in the presence of several types of measurement inaccuracies. To reduce the computational complexity of the solution, we derive an effective implementation of the Interior Point Method as the optimization procedure. The software for mass spectral comparison and deconvolution based on Wasserstein distance is available at `https://github.com/mciach/wassersteinms`.

2012 ACM Subject Classification Applied computing → Chemistry

Keywords and phrases Mass spectrometry, Tandem mass spectrometry, Wasserstein distance, Earth mover's distance, Similarity measure, Jaccard score, Tanimoto similarity

Digital Object Identifier 10.4230/LIPIcs.WABI.2018.25

Funding This work was partially supported by Polish National Science Center grants 2014/12/W/ST5/00592 and UMO-2017/26/D/ST6/00304.

[1] Both authors contributed equally to this work.

1 Introduction

Mass Spectrometry (MS) is one of the main analytical techniques of modern proteomics and metabolomics, which allows for identification and quantification of molecular compounds. In the first step, the particles are ionized; next, they are separated in an electromagnetic field according to their mass to charge ratio (m/z), and finally, transferred to a detector. The detected signal, usually proportional to the number of ions, is plotted against the corresponding m/z value on a *mass spectrum*. A pair of detected m/z value and the corresponding signal intensity is called a *peak*. The signal intensity is often referred to as *ion current* [11, 3].

The m/z value can be used to infer the chemical composition of molecules (see e.g. [1]), but it does not give information about its chemical structure. To gain insight into the latter, several measurement steps are performed in a technique called Tandem Mass Spectrometry (Tandem MS). After each step, a range of m/z value is selected, and ions from that range are subjected to fragmentation before the next measurement. The mass spectrum obtained from the n-th measurement is referred to as an MS^n spectrum.

Even though the MS^1 spectrum is recorded prior to any fragmentation, a single compound can give rise to several peaks. This is due to the natural occurrence of *isotopes* – atoms with the same number of electrons and protons, but different numbers of neutrons. Molecules which differ only in their isotopic compositions are termed *isotopologues*. A group of peaks corresponding to isotopologues of a single molecule is referred to as an *isotopic envelope* (c.f. Fig. 1).

Tandem MS can be used to identify the molecule under study. There are two main approaches to this task: *de novo* sequencing and database search. The first one strives to identify the elemental composition and/or structure of the molecule purely based on the mass spectrum of fragments. The second one searches a database of mass spectra obtained from known molecules to find the most similar one [16, 24, 25].

To be able to search for a similar spectrum, either a similarity or a distance measure needs to be employed. There are two main groups of such measures. The first one relies on the number of *matching peaks*. Two peaks are said to match if their m/z values differ by less than a given threshold. An example of such measure is the Jaccard score, equal to the number of matching peaks divided by the number of distinct peaks in both spectra. The second group of measures takes into account both the location and the intensities of peaks. An example of such measure is the Euclidean distance or the correlation coefficient [16, 24].

Both groups are similar in the sense that they compare peaks with the same m/z value. As a consequence, they are highly sensitive to even the slightest differences in chemical formulas. For example, *apigenin* ($C_{15}H_{10}O_5$) and *quercetin* ($C_{15}H_{10}O_7$) are two molecules which differ by two oxygen atoms (see Fig. 1). Even though this difference is relatively small compared to the overall atom count, the MS^1 spectra contain no matching peaks. Consequently, the discussed measures do not detect any similarity between these molecules. Some approaches make a preprocessing of spectra to infer an optimal pairwise matching of peaks before computing the similarity [10].

Our contribution. In the present work, we propose a new measure which quantifies both the differences in intensities and m/z values of peaks in a continuous way. As such, the measure is more robust to changes in chemical formulas than the most common measures based on peak matching. The measure is based on the concept of transporting the ion current between the spectra. The distance between spectra is equal to the total distance in the m/z domain covered by the current. This allows to express the distance between spectra in Daltons. This

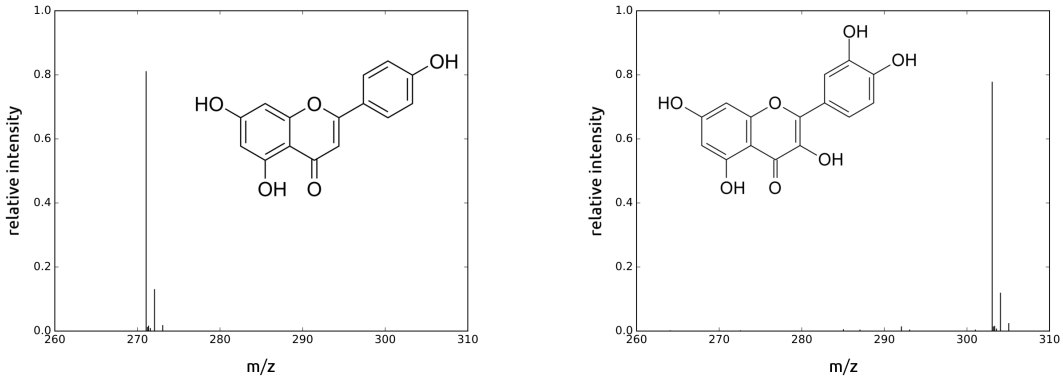

Figure 1 Molecular structures and MS^1 spectra of apigenin (left) and quercetin (right) showing their isotopic envelopes. Peak intensities have been normed to sum to 1. The mass spectra have been downloaded from the MassBank database (MassBank IDs: TY000164, TY000119).

distance is known in the field of probability theory as the (first) Wasserstein metric, and in the field of image processing as the Earth Mover's distance. Under certain assumptions, it can be computed in time linear in the number of distinct peaks in both spectra.

The Wasserstein distance allows to more accurately reflect the differences in chemical compositions of the molecules; for example, the distance between the MS^1 spectra from Fig. 1 is equal to 31.48 Da, while the difference in their masses is equal to 32.19 Da. Apart from quantifying the dissimilarity, the computed transport of ion current allows to match corresponding peaks in the compared spectra, which aids in the detection of differences in elemental composition and chemical structure (see Fig 2).

As a more advanced application of the Wasserstein distance, we show how it can be used to efficiently solve the problem of mass spectral deconvolution, i.e. the problem of separating overlapping isotopic envelopes.

Structure of the article. This article is structured as follows. In the next section, we introduce the formal definition of the Wasserstein metric. We also give some examples of the distance and the ion current transport between both experimental and *in silico* generated spectra. In Section 3, we give a formal statement of the deconvolution problem and the proposed solution based on the Wasserstein metric. Next, we show the results of computational experiments on *in silico* generated spectra to assess the performance of the solution. The results show that the Wasserstein distance allows for precise deconvolution in the presence of measurement errors even when the isotopic envelopes show considerable overlap.

2 The Wasserstein distance

In this section we introduce the first Wasserstein distance for discrete probability measures with finite support on $\mathbb{R}^d$. It is known in the computer science community as the Earth Mover's Distance, and was successfully used in a variety of applications including image processing [20]. In this article, we assume that the mass spectra are normalized so that the peak intensities sum to 1. This allows to interpret them as discrete probability measures.

Let $\mu = \sum_{i=1}^{n} w_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m} v_j \delta_{y_j}$ be two discrete probability distributions, where w_i and v_i are sets of weights, x_i and y_i are points in $\mathbb{R}^d$, and δ_{x_i} is the Dirac delta centered at x_i. Suppose that those measures describe the distribution of some mass, i.e. there is a

mass of w_i at point x_i. Then, the Wasserstein distance can be defined as the minimal cost of transporting all the mass given by μ onto the mass given by ν. An example of such transport is given by Fig. 2.

Let $\gamma(i,j)$ denote the amount of mass transported from the point x_i to y_j. This function is referred to as a *transport plan*. Since all the mass from x_i needs to be transported somewhere, we have $\sum_{j=1}^{m} \gamma(i,j) = w_i$. On the other hand, the mass at y_j needs to be filled completely, implying that $\sum_{i=1}^{n} \gamma(i,j) = v_j$. This, combined with the non-negativity of the transport plan, means that $\gamma(i,j)$ can be interpreted as the joint probability distribution of the two measures.

Let $\Gamma(\mu,\nu)$ denote the set of all possible transport plans between the measures μ and ν. Then, the Wasserstein distance between the two measures, $W_1(\mu,\nu)$, is equal to the total distance traversed by the mass under the optimal transport plan:

$$W_1(\mu,\nu) = \min_{\gamma\in\Gamma(\mu,\nu)} \sum_{i=1}^{n}\sum_{j=1}^{m} \gamma(i,j)||x_i - y_j||.$$

It is easy to see that the function W_1 defined above is a metric on the space of discrete probability measures with finite support. Furthermore, the above construction can be extended to more general measures, underlying spaces and cost functions. For details we refer our reader to [21].

In the applications to mass spectrometry data, we will be mainly interested in the W_1 distance on the real line. This distance has the following important representation:

▶ **Lemma 1** (Proposition 2.17 in [21]). *Let μ and ν be two discrete probability measures with finite support on real line $\mathbb{R}$, and let F_μ and F_ν be their cumulative distribution functions. Then, we have:*

$$W_1(\mu,\nu) = \int_{\mathbb{R}} |F_\mu(x) - F_\nu(x)|dx$$

Based on the above Lemma, we can easily compute the distance $W_1(\mu,\nu)$ for finite discrete probability measures corresponding to normalized mass spectra. A common way of representing such a spectrum is a peak list, i.e. a list of pairs (x_i, p_i) such that x_i are in increasing order and represent m/z values of peaks with intensities p_i.

Algorithm 1, adapted from [12], shows how to compute W_1 for two such lists of peaks. It is based on the observation that the absolute difference between the cumulative distribution functions of mass spectra, $|F_\mu - F_\nu|$, is a step function, and therefore it is easily integrable. Furthermore, the value $F_\mu(x) - F_\nu(x)$ is equal to the surplus of the intensity in μ relative to ν, which needs to be transported through the point x.

Note that the runtime of Algorithm 1 is $\mathcal{O}(n+m)$: in each iteration of the main loop either i or j is incremented, no index variable will ever exceed the length of the corresponding list, and the algorithm terminates when both indices have reached the end of their respective lists.

2.1 Some basic properties

To help the reader gain some initial intuitions behind the Wasserstein distance, in this short subsection we discuss some of its qualitative properties when applied to mass spectra. In the next subsection, we illustrate some of the points discussed here by computational experiments.

First, for MS^1 spectra of two molecules, the Wasserstein distance is approximately equal to the absolute mass difference of the molecules. The other main factor that influences the distance in this case is the presence of measurement inaccuracies. Note, however, that

Algorithm 1: Computation of Wasserstein distance between two spectra.

Data: Two lists, N, M, of pairs (x, p), containing the lists of peaks of respective spectra
Result: W_1 distance between given spectra

$i \leftarrow 0$; $j \leftarrow 0$; $ret \leftarrow 0.0$; $\gamma \leftarrow$ an empty transport scheme
$n \leftarrow \text{length}(N); m \leftarrow \text{length}(M)$
while $i < n \vee j < m$ **do**
  $d \leftarrow \min(N[i].p, M[j].p)$
  $ret \leftarrow ret + d \cdot |N[i].x - M[j].x|$
  $N[i].p \leftarrow N[i].p - d$
  $M[j].p \leftarrow M[j].p - d$
  $\gamma(i, j) \leftarrow d$
  if $0 = N[i].p$ **then**
    $i \leftarrow i + 1$
  else
    $j \leftarrow j + 1$
  end
end
The variable ret contains the Wasserstein distance and γ the transport plan.

this influence remains small as long as the inacurracies in intensity measurements are small compared to the corresponding peak intensities.

Usually, some inaccuracy in both the intensity and the mass measurement is present. Naturally, the latter poses a major problem for measures based on peak matching. On the other hand, the Wasserstein distance is not significantly influenced by small mass measurement errors – instead, the imprecise measurement simply gets shifted to match its theoretical counterpart.

The implicit assumption of this metric, which may not be desirable in some applications, is that the mass difference reflects chemical difference. Therefore, two molecules differing by an OH group are assumed to be more similar to each other than two molecules differing by a C_2H_5 group. It is possible to relax this assumption by applying a different metric in the mass domain, say $c(x, y)$, in the definition of the Wasserstein distance:

$$W_c(\mu, \nu) = \min_{\gamma \in \Gamma(\mu,\nu)} \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma(i, j) c(x_i - y_j).$$

An important caveat in this case is that treating all modifications as equivalent may lead to unexpected results – notably, a protein being treated as a single carbon atom with an extremely large modification. Furthermore, using other distances in the mass domain may lead to difficult optimization problems. The use of absolute difference, $|x_i - y_j|$, allows to avoid the optimization in the space of all possible transport plans.

If the two molecules differ by a modification which does not change their fragmentation pattern, then the Wasserstein distance between their MS^2 spectra will not exceed the weight of the modification. This follows from the observation that the modification is present only in some of the fragments, which account to a fraction of the total intensity in the spectrum. Note that this is a highly idealized example, since modifications may significantly change the fragmentation patterns and inflict a greater influence on the Wasserstein distance. In general, however, this distance cannot exceed the mass of the heavier molecule.

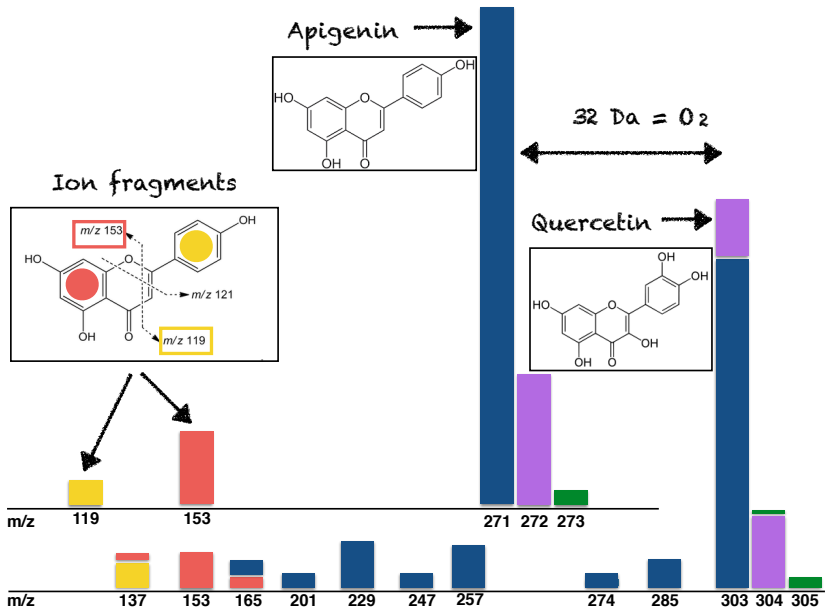

Figure 2 The optimal ion current transport plan for MS^2 spectra of apigenin (top) and quercetin (bottom), fragmented using 30 eV collision energy. The colors on the quercetin mass spectrum correspond to the origin of the transported ion current. The isotopic envelope of quercetin is shifted by 32 Da, i.e. the mass of two oxygen atoms.

Lastly, the structure of the optimal transport plan is highly sensitive to chemical noise, i.e. the presence of unexpected molecules. Recall that all the intensity from one spectrum needs to be used to explain all the intensity of the second spectrum. Therefore, if one of the analyzed spectra contains an additional peak, some of the intensity from the first spectrum needs to be used to explain it. This may lead to global changes in the structure of the optimal transport plan. It follows that in the presence of chemical noise, the Wasserstein distance may not reflect the similarity between the analyzed compounds.

2.2 Case study

To quantitatively analyze the properties of the Wasserstein metric when applied to mass spectral data, we have analyzed two sets of spectra obtained from the MassBank database [5]. In both cases, we have compared the performance of the Wasserstein distance with two other popular approaches: the Euclidean distance and the Jaccard score (i.e. the ratio of matching peaks to the total number of different peaks in both spectra). When analyzing those two measures, the spectra were binned to 0.01 Da resolution to increase the number of matching peaks and decrease their sensitivity to small measurement errors. No binning was performed during the analysis of the Wasserstein metric.

The first test was based on 615 MS^1 ESI-QTOF spectra with positive ionization mode. The goal of comparing MS^1 spectra was to verify the correlation between Wasserstein distance and the difference in mass of the molecules. The spectra have been compared pairwise, resulting in 188805 pairs. These pairs were then used to compute the Spearman's rank correlation between the distance and the absolute difference between masses of the corresponding molecules. The results are summarized in Table 1.

Table 1 Spearman's rank correlations between the Wasserstein distance, Jaccard score, Euclidean distance, and either the absolute mass difference or Tanimoto similarity of chemical structures. M, absolute mass difference; W, Wasserstein distance; J, Jaccard score; E, Euclidean distance; T, Tanimoto similarity; RW, relative Wasserstein distance; RE, relative Euclidean distance (see text).

MS^1 spectra

	M	**W**	**J**	**E**
M	1.00	0.89	-0.07	-0.37
W	–	1.00	-0.08	-0.22
J	–	–	1.00	-0.17
E	–	–	–	1.00

MS^2 spectra

	T	**RW**	**J**	**RE**
T	1.00	-0.41	0.22	-0.24
RW	–	1.00	-0.21	0.43
J	–	–	1.00	-0.11
RE	–	–	–	1.00

As expected, the metics based on peak matching are sensitive to mass differences, and therefore less correlated than the Wasserstein distance. Note that for the Wasserstein and Euclidean distance the correlation is expected to be positive, while for the Jaccard similarity metric it is expected to be negative. Surprisingly, we have found a negative correlation between the mass difference and the Euclidean distance.

The second test was based on MS^2 ESI-QTOF spectra with positive ionization mode. Here, the goal was to investigate the relationship between Wasserstein distance and structural similarity. Note that this distance is sensitive to the fragmentation intensity – two MS^2 spectra obtained for a given molecule can have a large distance if there is a significant difference in the intensity of fragments. To account for that, we have selected a subset of 473 MS^2 spectra for different molecules in which the precursor peak had around 10% relative intensity. This resulted in 111628 pairs of spectra. For each pair of spectra, we have computed the Wasserstein, Jaccard and Euclidean metrics. Next, we have computed the Tanimoto similarity between the structures of the corresponding molecules, based on the Morgan circular fingerprints [17, 19]. The fingerprints have been computed using the RDKit package (`http://www.rdkit.org`), with the radius of 2 and the default set of the feature-based invariants. The results are summarized in Table 1.

Note that the selected set of spectra comes from a diverse set of molecules. In particular, the mean mass is 310 Da, while the standard deviation is 160 Da. This poses a problem for the Wasserstein metric, as a pairs of small molecules will yield small distances regardless of the structural similarity. To account for this, we have divided the distance by the product of masses of the analyzed molecules. Without this correction, the correlation between the Wasserstein metric and the Tanimoto similarity drops to -0.22. This procedure also improved the correlation between the Tanimoto similarity and the Euclidean distance, but not the Jaccard score. We refer to the distances with this correction as *relative* distances.

The detailed relationship between the relative Wasserstein distance and the Tanimoto similarity is depicted in Fig. 3. For comparison, the Figure also shows the relationship between the Tanimoto similarity and the Jaccard score. Note that all compounds with high Tanimoto similarity have small relative Wasserstein distances. However, this relative distance is much more variable for compounds with low similarity. This frequent occurence of molecules with highly divergent structures but similar MS^2 spectra decreases the extent to which the Wasserstein distance correlates with the Tanimoto structural similarity.

The experiments show that the approaches to spectral comparison based on Wasserstein distance outperform the Jaccard score and the Euclidean distance in terms of correlation to mass in MS^1 spectra and to the chemical structure in MS^2 spectra. However, at this moment the Wasserstein distance should be applied only to MS^2 spectra with similar proportions of

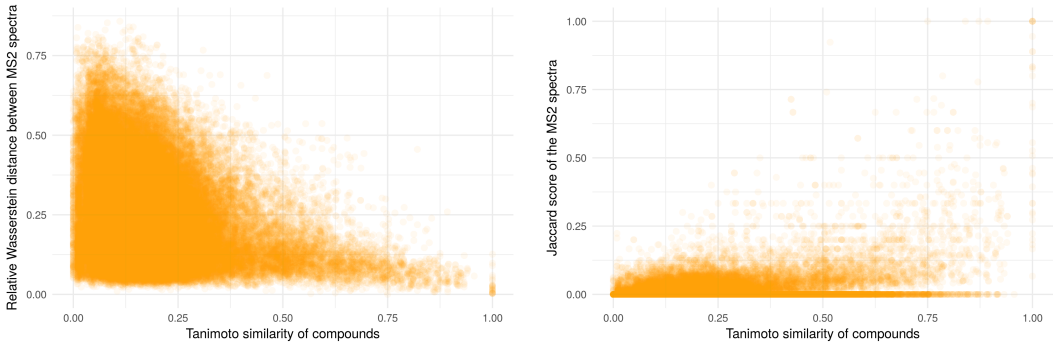

Figure 3 The relationship between MS^2 spectra and the structural similarity according to the relative Wasserstein distance (left) and the Jaccard score (right).

precursor molecules, and preferably obtained from compounds of similar mass. The results so far are optimistic, but more work needs to be done in order to generalize the Wasserstein distance so that it can be applied to a broader class of mass spectra.

3 Mass Spectral Deconvolution

In the mass spectral literature, the term *deconvolution* is used to refer to several problems which usually deal with separating overlapping peaks and/or isotopic envelopes. The authors of [4] define deconvolution as inferring the relative quantities of molecules with overlapping isotopic envelopes. Similar problems have been described in [7, 6, 15]. The term deconvolution is sometimes used as a synonym for *deisotoping*, that is, conversion of isotopic envelopes into single peaks with average m/z value and joint intensity [8, 13]. Other authors have used this term for conversion of m/z values to masses of multiply charged molecules [14, 18, 2].

We propose the following formalization of the mass spectral deconvolution, which encompasses several problems described above.

▶ **Problem 2** (Mass Spectral Deconvolution, MSD). *Let ν be a normalized mass spectrum, and let $\{\mu_i \colon 1 \leq i \leq k\}$ be a collection of normalized mass spectra. Let $d(\nu, \mu)$ be a distance measure between spectra. Let Δ_{k-1} be a $k-1$-dimentional probability simplex. Find a set of weights $p^* \in \Delta_{k-1}$ which minimizes the distance between ν and the convex combination of $\{\mu_i \colon i = 1, 2, \ldots, k\}$:*

$$p^* = \underset{p \in \Delta_{k-1}}{\operatorname{argmin}} \; d\left(\sum_{i=1}^{k} p_i \mu_i, \nu\right) \tag{1}$$

In the above definition, ν is referred to as an *experimental* spectrum, and μ_i are referred to as *theoretical* spectra. Note that neither the distance measure nor the origin of the theoretical spectra is not specified. Depending on the latter, this definition can be reduced to several of the problems mentioned at the beginning of this section. For example, if the theoretical spectra correspond to isotopic envelopes, the solution to MSD can be used for deisotoping, in which case p_i corresponds to the joint intensity of i-th envelope. On the other hand, if μ_i correspond to mass spectra of a single molecule with different charges, the problem reduces to conversion of m/z to mass. Finally, if μ_i are mass spectra from a database, the problem can be reduced to annotation of mass spectrum.

In this section, we propose a solution to the deconvolution problem using the Wasserstein metric as the distance measure, i.e. in (1) we use $d = W_1$. For the sake of clarity, we assume that the goal of MSD is to infer the proportions of compounds with overlapping isotopic

envelopes. Therefore, we assume that the user knows the molecular formulas of compounds which may be found in the analyzed spectrum. The theoretical spectra μ_i will correspond to mass spectra of those compounds predicted by the IsoSpec [26] algorithm. Note however that the proposed method is general, and can be easily extended to other use cases like deisotoping by simply choosing a different set of the theoretical spectra.

3.1 Efficient algorithm for the MSD problem

In what follows we will present an efficient method of solving MSD problem. First, we demonstrate the reduction of the original problem to a linear programming problem. Next, we will show how to use the structure of resulting linear programming problem to efficiently compute one iteration of a standard primal-dual Interior Point Method (IPM) (see for example [9]). In this subsection we present the main ideas behind the solution and state the most important results. The technical details and proofs are relegated to the Appendix.

Let $\nu = \sum_{j=1}^{m_0} w_{0,j}\delta_{x_{0,j}}$ be the experimental spectrum and for $i = 1, 2, \ldots, k$ let $\mu_i = \sum_{j=1}^{m_i} w_{i,j}\delta_{x_{i,j}}$ be the i-th theoretical spectrum. Denote the set of all support points from the empirical and theoretical spectra by $\mathcal{S} = \{x_{i,j} : 1 \leq j \leq m_i, 0 \leq i \leq k\}$ and let $n = |\mathcal{S}|$.

Let $\boldsymbol{s} = s_1 < s_2 < \cdots < s_n$ be the vector of ordered elements of $\mathcal{S}$. Notice that the cumulative distribution functions of μ_i's and ν are constant on intervals $[s_j, s_{j+1})$. For $1 \leq j \leq n-1$ let $f_{i,j}$ and g_j be the values of the cdfs of μ_i and ν respectively on the interval $[s_j, s_{j+1})$, and set $f_{i,n} = 1 = g_n$. Let $d_j = s_{j+1} - s_j$ be the length of the j-th interval.

Denote by F the $k \times n$ matrix with entries $F[i, j] = f_{i,j}$ for any $1 \leq i \leq k$ and $1 \leq j \leq n$, I_k an identity matrix of size k, and J_n an $(n-1) \times n$ matrix equal to the identity matrix of size n without the last row. We define

$$A = \begin{bmatrix} -J_n & -J_n & 0 \\ F & -F & -I_k \end{bmatrix}.$$

Finally, let c be a vector of length $2n + k$, such that $c = (g, -g, 0_k)$, where 0_k is the vector of zeros of length k, and $b = (-d, 0_k)$ be a vector of length $n - 1 + k$. The following Lemma, proved in the Appendix, states that MSD can be reduced to linear programming.

▶ **Lemma 3.** *The following dual linear programming problems:*

$$\begin{array}{lll} \min\limits_{x} & x^T c & \\ \text{s.t.} & Ax = b & \\ & x \geq 0 & \end{array} \qquad \begin{array}{lll} \max\limits_{y} & y^T b & \\ \text{s.t.} & A^T y + z = c & \\ & z \geq 0 & \end{array} \tag{2}$$

are feasible. Furthermore, for any solution (x_*, y_*, z_*) of the above problem, the vector of the last k elements of y_* belongs to the set of solutions of MSD.

We propose to solve the above linear program using IPM, while using the structure of our linear programming problem to significantly decrease the time and memory cost of each iteration. In general, IPM for linear programming solves both primal and dual problems simultaneously, by solving a cleverly chosen nonlinear approximation of those problems using the Newton's Method. For an overview of IPM we refer our reader to [9] and references therein. In Algorithm 2 we present the pseudocode for the general scheme of IPM for the dual problem (2). In what follows, x_t, y_t, z_t are t-th iterates of variables x, y, z from the problem, while $X_t = \mathrm{diag}(x_t)$, $Z_t = \mathrm{diag}(z_t)$ are diagonal matrices.

A triple (x_t, y_t, z_t) is called an ϵ-feasible ϵ-solution if it is both primal-dual feasible and optimal up to ϵ tolerance. Since the computational cost of each iteration is dominated by

Algorithm 2: General scheme of a primal-dual Interior Point Method.

Data: Matrix A and vectors b, c defining dual linear problems. A starting point (x_0, y_0, z_0) and error tolerance $\epsilon > 0$.

Result: an ϵ-feasible ϵ-solution of the linear problem

Set $t = 0$ **repeat**

compute centrality $\mu_t = \langle x_t, z_t \rangle / (2n + k)$ compute primal residual $r_p^t = b - Ax_t$ and dual residual $r_d^t = c - A^T y_t - z_t$ choose scaling factor σ_t (see Appendix) find direction (d_x, d_y, d_z) by solving the system of linear equations:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z_t & 0 & X_t \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} r_p^t \\ r_d^t \\ \sigma_t \mu_t \mathbf{1} - X^t z_t \end{bmatrix} \tag{3}$$

find $\alpha_p \in (0, 1]$ such that $x_{t+1} = x_t + \alpha_p d_x > 0$ find $\alpha_d \in (0, 1]$ such that $z_{t+1} = z_t + \alpha_d d_z > 0$ and take $y_{t+1} = y_t + \alpha_d d_y$ $t = t + 1$

until *triple* (x_t, y_t, z_t) *is an* ϵ*-feasible* ϵ*-solution*;

solving the equation (3), we focus on this part and relegate to the Appendix the discussion about choosing the stopping condition, the starting point (x_0, y_0, z_0) and the scaling factors σ_t.

The solution of the equation (3) can be obtained by solving the normal equation

$$AZ_t^{-1} X_t A^T d_y = b + AZ_t^{-1}(X_t r_d^t - \sigma_t \mu_t \mathbf{1}).$$

After solving the normal equation d_x, d_z can be obtained using formulas:

$$d_z = z = r_d^t - A^T d_y, \quad d_x = -x_t + Z_t^{-1}(\sigma_t \mu_t \mathbf{1} - X_t d_z).$$

The computational cost of single step of IPM is dominated by solving the normal equation, which is of order $\mathcal{O}((k(n-1)^3)$. However, thanks to the specific structure of matrix A, for any v, w we can compute Av, $A^T v$ and solve an equation $AZ_t^{-1} X_t A^T v = w$ efficiently. The detailed derivation of efficient algorithm, tailored to deconvolution problem is given in the Appendix. This allows us to perform one step of IPM efficiently, i.e the cost of single step is of order $\mathcal{O}(k^3 + k \sum_{i=1}^{k} m_i + n)$.

▶ **Lemma 4.** *One step of IPM for problem defined in Lemma 3 can be computed in time* $\mathcal{O}(k^3 + k \sum_{i=1}^{k} m_i + n)$ *and memory* $\mathcal{O}(k^2 + n)$.

Note that the for the given error tolerance ϵ, the IPM needs $\mathcal{O}(\sqrt{2n + k} \log(\epsilon^{-1}))$ iterations to find an ϵ-accurate solution.

3.2 Computational experiments

We have performed several computational experiments to illustrate the performance of the proposed solution to MSD, and to analyze its robustness to various kinds of distorsions occuring in MS measurements. In contrast to the previous case studies, in this section we use *in silico* generated spectra. This allows us to precisely control the signal-to-noise ratio, and to rigorously estimate the error of the method.

Our main goal is to demonstrate the applicability of the Wasserstein distance to MSD in case of noisy experimental spectra. There are several sources of noise in mass spectrometric measurements, among others: *(i)* precision of the intensity measurement, *(ii)* precision of the m/z measurement, *(iii)* resolving power, i.e. the ability to detect peaks with similar masses,

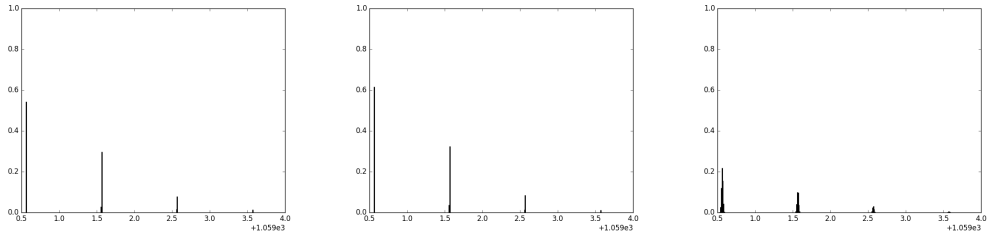

Figure 4 An illustration of the simulated measurement inaccuracies based on a theoretical spectrum of bradykinin ($C_{50}H_{73}N_{15}O_{11}$). Left: clean spectrum. Middle: noise in the intensity domain. Right: noise in the mass domain. The apparent change in intensity in the right spectrum is caused only by blurring the peaks and binning afterwards.

(iv) chemical noise, i.e. presence of unexpected molecules in a spectrum [3]. In this section, we focus mostly on the first three types of noises, i.e. low resolving power and/or precision. The first step of all our experiments was to generate the isotopic envelopes of selected molecules by the IsoSpec algorithm [26]. These envelopes form the set of the theoretical spectra. The experimental spectrum was obtained by taking a convex combination of the latter. Finally, the experimental spectrum was distorted in the following manner:

- Gaussian noise has been added to the logarithm of the peak intensity, and the result has been exponentiated,
- Each peak has been replaced by the density function of the normal distribution,
- The resulting intensity distribution has been binned.

Both Gaussian noises had a standard deviation of 0.01. For binning of the mass spectrum, we have assumed two resolving powers of the spectrometer: 0.001 Da and 0.01 Da. An example of the result of this procedure is depicted in Fig. 4. A spectrum without these distortions is referred to as *clean*, while the distorted one as *noisy*.

The performance of our approach to MSD has been quantified by the Root Mean Square Error (RMSE) between the original and inferred proportions of different isotopic envelopes. Note that the absolute prediction error for any variable does not exceed RMSE; moreover, if RMSE for deconvolution of n variables is equal ϵ, then it cannot happen that the absolute error exceeds $\epsilon/\sqrt{n}$ on every variable at the same time.

We have performed three tests, inspecting the method's sensitivity to the number of overlapping envelopes, molecular mass of deconvolved molecules, and their charge. The first test is based on random molecules. The next two are based on simulated proteins composed of *averagine* – a model aminoacid with molecular formula $C_4 \cdot 9384\, H_7 \cdot 7583\, N_1 \cdot 3577\, O_1 \cdot 4773\, S_0 \cdot 0417$ and average molecular mass of 111.1254 Da [22]. In the first test, we have inspected both clean and noisy spectra. In test two and three, only noisy spectra were analyzed. To check the standard deviation of the prediction error, the tests were replicated, with noise added independently in each replicate. Below we present each test in detail.

Test no. 1 – increasing number of molecules. This test is based on 17 randomly chosen isobars consisting of carbon, oxygen, hydrogen, nitrogen and sulfur, each one with the nominal mass of 30 000 Da. The experimental spectra with a range of interfering isobars were constructed by gradually extending a subset of those molecules. This procedure was replicated 20 times, resulting in 340 experimental spectra. The results are shown on Fig. 5. The prediction error is very low and stable for less than 5 isobars, suggesting that the ratios of particular elements in the deconvolved molecules have no considerable influence on the method's performance.

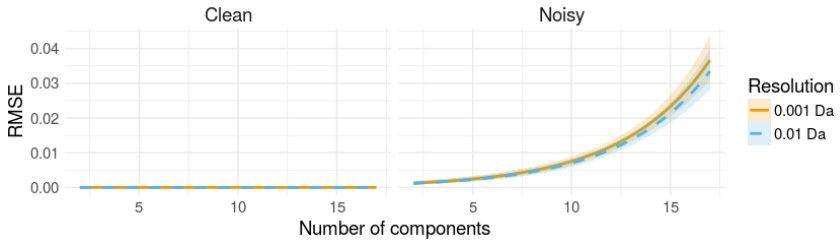

Figure 5 The performance of MSD method for increasing number of deconvolved molecules. The solid line represents the average RMSE over 20 repetitions. The ribbon represents the standard deviation of the error.

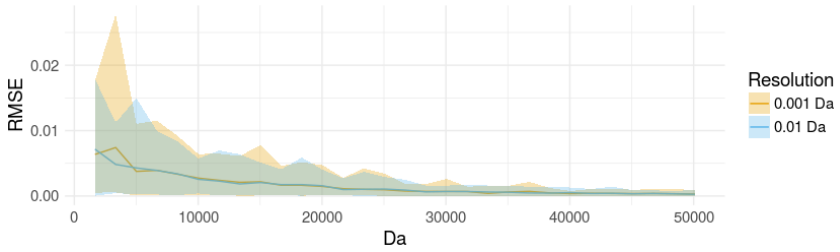

Figure 6 The performance of MSD method for increasing mass of deconvolved molecules. The solid line represents the average RMSE over 50 repetitions. The ribbon represents the standard deviation of the error.

Test no. 2 – increasing mass of molecules. In this test, we consider overlapping isotopic evelopes of two types of proteins: singly charged protein consisting of n units of averagine, and doubly charged protein consisting of $2n$ units, where the values of n were selected so that the average m/z ratio of proteins spans the range from 1,500 Da to 45,000 Da. For any given n, the isotopic envelopes of the two corresponding proteins were mixed in proportions 0.8 and 0.2 respectively. The procedure was replicated 50 times. The outcome of this experiment is presented in Fig. 6.

Test no. 3 – increasing charge of molecules. In this test, we consider the following four proteins based on averagine: $C_{1482}H_{2328}N_{408}O_{444}S_{12}$, $C_{1482}H_{2329}N_{408}O_{444}S_{12}$, $C_{1482}H_{2330}N_{408}O_{444}S_{12}$ and $C_{1481}H_{2341}N_{408}O_{444}S_{12}$, mixed in proportions 0.3, 0.5, 0.1 and 0.1, respectively. The first three molecules differ by one hydrogen atom, resulting in partially overlapping isotopic envelopes. The fourth molecule is an isobar of the second one, as one carbon has been replaced by 12 hydrogens. All molecules have been equally charged, with the charge varying from 1 to 10. This yields a sequence of deconvolution problems with increasing difficulty, because the peaks become more densely packed while the resolution stays constant. For each charge, 50 replicates were performed. The results are presented in Fig. 7.

Computational experiments show that our approach is able to deconvolve complex spectra even in the presence of measurement inaccuracies. Even for 17 isobars the RMSE does not exceed 0.05, which implies that the prediction error does not exceed 0.0125 on all 17 variables simultaneously. However, it must be noted that our approach to MSD is expected to be sensitive to chemical noise, because the Wasserstein metric requires that all the intensity of the experimental spectrum is explained. Therefore, our solution to MSD should be applied to mass spectra of highly purified compounds.

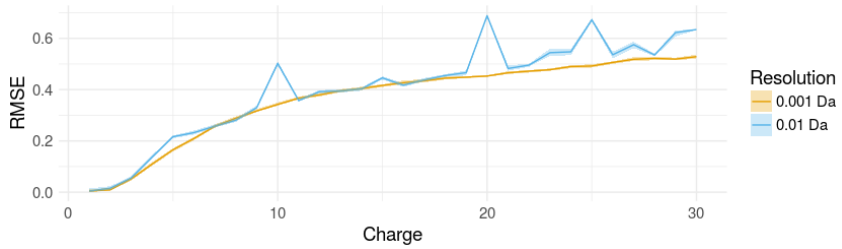

Figure 7 The performance of MSD method for increasing charge of deconvolved molecules. The solid line represents the average RMSE over 50 repetitions. The ribbon represents the standard deviation of the error.

4 Discussion and conclusions

In this work, we have presented a new method of comparing mass spectra. The method is based on the first Wasserstein metric, which is a well-established and well-studied metric in both probability theory and image processing field. Compared to the current approaches for spectral comparison, it is more robust to differences in chemical composition and measurement errors. In the MS^2 spectra of similar compounds, the Wasserstein distance reflects differences in both chemical structure and fragmentation intensities.

We have proposed a new formalization of the Mass Spectral Deconvolution (MSD) problem, which encompasses separating of overlapping isotopic envelopes, deisotoping, and decharging. We have shown that the Wasserstein distance can be used to effectively solve this problem in the presence of measurement inaccuracies.

The proposed solution for MSD works for a wide range of m/z values and multiply charged ions. Furthermore, it is not limited to a single class of compounds like peptides or metabolites. The theoretical isotopic envelopes can be either predicted *in silico* or measured experimentally.

The implementation of the Wasserstein metric for mass spectral comparison and deconvolution is available at `https://github.com/mciach/wassersteinms`.

References

1 S. Böcker and K. Dührkop. Fragmentation trees reloaded. *Journal of Cheminformatics*, 8(1):5, 2016.

2 S. Cappadona, P. R. Baker, P. R. Cutillas, A. JR. Heck, and B. van Breukelen. Current challenges in software solutions for mass spectrometry-based quantitative proteomics. *Amino acids*, 43(3):1087–1108, 2012.

3 I. Eidhammer, K. Flikka, L. Martens, and S-O. Mikalsen. *Computational methods for mass spectrometry proteomics*. John Wiley & Sons, 2008.

4 F. Lermyte et al. Conformational Space and Stability of ETD Charge Reduction Products of Ubiquitin. *J. Am. Soc. Mass Spectrom.*, Aug 2016.

5 H. Hisayuki et al. Massbank: a public repository for sharing mass spectral data for life sciences. *Journal of Mass Spectrometry*, 45(7):703–714, 2010.

6 K. Xiao et al. Accurate and Efficient Resolution of Overlapping Isotopic Envelopes in Protein Tandem Mass Spectra. *Sci Rep*, 5:14755, Oct 2015.

7 M.M Koek et al. Quantitative metabolomics based on gas chromatography mass spectrometry: status and perspectives. *Metabolomics*, 7(3):307–328, Sep 2011.

8 N. Jaitly et al. Decon2ls: An open-source software package for automated processing and visualization of high resolution mass spectrometry data. *BMC Bioinformatics*, 10(1):87, 2009.

9 J. Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.

10 M. E. Hansen and J. Smedsgaard. A new matching algorithm for high resolution mass spectra. *J. of Am. Soc. Mass Spectrom.*, 15(8):1173–1180, 2004.

11 C. E. Housecroft and E. C. Constable. *Chemistry: An introduction to organic, inorganic and physical chemistry*. Pearson education, 2010.

12 Jedrzej Jablonski and Anna Marciniak-Czochra. Efficient algorithms computing distances between radon measures on r. *arXiv preprint arXiv:1304.3501*, 2013.

13 Q. Kou, L. Xun, and X. Liu. Toppic: a software tool for top-down mass spectrometry-based proteoform identification and characterization. *Bioinformatics*, 32(22):3495–3497, 2016.

14 M. Mann, C. K. Meng, and J. B. Fenn. Interpreting mass spectra of multiply charged ions. *Analytical Chemistry*, 61(15):1702–1708, 1989.

15 J. Meija and J.A. Caruso. Deconvolution of isobaric interferences in mass spectra. *J. of Am. Soc. Mass Spectrom.*, 15(5):654–658, 2004.

16 S. Neumann and S. Böcker. Computational mass spectrometry for metabolomics: identification of metabolites and small molecules. *Analytical and Bioanalytical Chemistry*, 398(7-8):2779–2788, 2010.

17 Nina Nikolova and Joanna Jaworska. Approaches to measure chemical similarity–a review. *Molecular Informatics*, 22(9-10):1006–1026, 2003.

18 B. B. Reinhold and V. N. Reinhold. Electrospray ionization mass spectrometry: Deconvolution by an entropy-based algorithm. *J. of Am. Soc. Mass Spectrom.*, 3(3):207–215, 1992.

19 David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

20 Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, Nov 2000.

21 F. Santambrogio. *Optimal transport for applied mathematicians*. Springer, 2015.

22 Michael W. Senko, Steven C. Beu, and Fred W. McLafferty. Determination of monoisotopic masses and ion populations for large biomolecules from resolved isotopic distributions. *Journal of the American Society for Mass Spectrometry*, 6(4):229–233, 1995.

23 R. J. Vanderbei et al. *Linear programming*. Springer, 2015.

24 K. X. Wan, I. Vidavsky, and M. L. Gross. Comparing similar spectra: from similarity index to spectral contrast angle. *J. of Am. Soc. Mass Spectrom.*, 13(1):85–88, 2002.

25 Ş. Yilmaz, E. Vandermarliere, and L. Martens. *Methods to Calculate Spectrum Similarity*, pages 75–100. Springer New York, New York, NY, 2017.

26 M. K. Łącki et al. IsoSpec: Hyperfast Fine Structure Calculator. *Anal. Chem.*, 89(6):3272–3277, Mar 2017.

A Appendix

A.1 Derivation of linear program

In this subsection we show how the MSD problem defined in (1) can be reduced to linear programming in the case when the distance measure is the Wasserstein metric W_1. We also prove Lemma 3.

Recall that our problem is to find:

$$p_* = \operatorname*{argmin}_{p \in \Delta_{k-1}} W_1 \left(\sum_{i=1}^{k} p_i \mu_i, \nu \right) \tag{4}$$

where μ_i, ν are discrete probability measures with finite support.

First, we show that the MSD problem can be restated as weighted L_1 regression on the probability simplex Δ_{k-1}. Using Lemma 1 we can write:

$$\operatorname*{argmin}_{p \in \Delta_{k-1}} W_1 \left(\sum_{i=1}^{k} p_i \mu_i, \nu \right) = \operatorname*{argmin}_{p \in \Delta_{k-1}} \int_{\mathbb{R}} \left| \sum_{i=1}^{k} p_i F_{\mu_i}(x) - F_\nu(x) \right| dx. \tag{5}$$

Recall that $\mathcal{S}$ denotes the set of points from theoretical and empirical spectra, and that $(s_i)_{i=1}^n$ are elements of $\mathcal{S}$ ordered increasingly. Note that for $x < s_1$ and $x \geq s_n$, the function under the integral on the RHS of (5) is zero. At the same time, the function is constant on intervals $[s_i, s_{i+1})$. Let $1 \leq j \leq n-1$ let $f_{i,j}$ and g_j be the values of the cdf on the interval $[s_j, s_{j+1})$ of μ_i and ν respectively, and set $f_{i,n} = 1 = g_n$. For $1 \leq j \leq n-1$ let $d_j = s_{j+1} - s_j$ be the length of the interval. We write:

$$\int_{\mathbb{R}} \left| \sum_{i=1}^{k} p_i F_{\mu_i}(x) - F_\nu(x) \right| dx = \sum_{j=1}^{n-1} d_j \left| \sum_{i=1}^{k} p_i f_{i,j} - g_j \right|,$$

and we reduce the optimization problem (4) to weighted L_1 regression on probability simplex:

$$p_* = \operatorname*{argmin}_{p \in \Delta_{k-1}} \sum_{j=1}^{n-1} d_j \left| \sum_{i=1}^{k} p_i f_{i,j} - g_j \right| \tag{6}$$

Now, we apply a well known technique of representing weighted L_1 regression as linear programming problem (see e.g. [23]). Let us introduce dummy variables t_j, such that $t_j \geq |\sum_{i=1}^m p_i f_{i,j} - g_j|$. With this notation the problem (6) is equivalent to minimizing linear function $\sum_{i=1}^{n-1} d_j t_j$. For any j the inequality $t_j \geq |\sum_{i=1}^m p_i f_{i,j} - g_j|$ can be represented by two linear inequalities, $t_j \geq \sum_{i=1}^m p_i f_{i,j} - g_j$ and $t_j \geq -\sum_{i=1}^m p_i f_{i,j} + g_j$. To also take into account the fact that vector $(p_i)_{i=1}^k$ needs to belong to the probability simplex, we just need to add inequality constraints $p_i \geq 0$ and equality constraint $\sum_{i=1}^k p_i = 1$. By rewriting the

latter as two inequality constraints, we end up with the following linear program:

$$\begin{aligned} \min_{p,t} \quad & d^T t \\ \text{s.t.} \quad & -t_j + \sum_{i=1}^{m} p_i f_{i,j} \leq g_j \\ & -t_j - \sum_{i=1}^{m} p_i f_{i,j} \leq -g_j \\ & 1 \leq \sum_{i=1}^{m} p_i \leq 1 \\ & p_i \geq 0. \end{aligned} \tag{7}$$

The minimized function in (7) does not depend itself on p, however variable p appears in constrains. From the construction of (7) it follows that for any feasible pair (p, t) we have:

$$d^T t \geq \sum_{i=1}^{n-1} d_j \left| \sum_{i=1}^{k} p_i f_{i,j} - g_j \right|$$

with equality holds if and only if $t_j = |\sum_{i=1}^{k} p_i f_{i,j} - g_j|$. Therefore for any solution p_* of problem (6) there exists t_* such that (p_*, t_*) is a solution of (7). It also follows that for any solution (p_*, t_*) of (7), vector p_* is a solution of (6). Furthermore, since (6) is a problem of optimizing continuous convex function on a compact set, at least one solution exists.

We have thus reduced MSD problem (4) to a Linear Programming problem. Denote by F the $k \times n$ matrix with entries $F[i, j] = f_{i,j}$ for any $1 \leq i \leq k$ and $1 \leq j \leq n$, I_k an identity matrix of size k, and J_n an $(n-1) \times n$ matrix equal to the identity matrix of size n without the last row. Ussing definitions of vectors g and d, we can rewrite the problem (7) in a simplified form:

$$\begin{aligned} \max_{t} \quad & -d^T t \\ \text{s.t.} \quad & \begin{bmatrix} -J_n^T & F^T \\ -J_n^T & -F^T \\ 0 & -I_k \end{bmatrix} \begin{pmatrix} t \\ p \end{pmatrix} \leq \begin{pmatrix} g \\ -g \\ 0_k, \end{pmatrix} \end{aligned}$$

where 0_k is a zero vector of length k. We bring to our reader's attention that the constrain $\sum_{i=1}^{k} p_i = 1$, splits into two inequality constraints $\sum_{i=1}^{k} p_i \leq 1$ and $-\sum_{i=1}^{k} p_i \leq -1$, which was included in the above program, since the last row of matrix F^T contains ones.

Let us recall definitions of $c = (g, -g, 0_k)$, $b = (-d, 0_k)$ and

$$A = \begin{bmatrix} -J_n & -J_n & 0 \\ F & -F & -I_k \end{bmatrix}.$$

Using the above notation, and adding the slack variables z to replace inequality constrains by equality constrains, we rewrite the problem (7) as

$$\begin{aligned} \max_{y} \quad & y^T b \\ \text{s.t.} \quad & A^T y + z = c \\ & z \geq 0. \end{aligned} \tag{8}$$

This summarizes the reduction of the original minimization problem to linear programming. We end this section with the proof of Lemma 3.

Proof of Lemma 3. From the above discussion it follows that the feasible region of the problem (8) is nonempty. Furthermore, note that the problem given by (6) is bounded from below. This, combined with the inequality:

$$y^T b \leq -\sum_{j=1}^{n-1} d_j \left| \sum_{i=1}^{k} p_i f_{i,j} - g_j \right|,$$

means that the maximization problem from Lemma 3 has a solution. From the duality theory for linear programs it follows that the minimization problem is feasible as well and the duality gap is zero.

Finally, from the above discussion it follows as well that if (x_*, y_*, z_*) is a solution of dual problems in Lemma 3, then the last k elements of y_* form a vector in Δ_{k-1} that is a solution of the initial MSD problem. ◀

A.2 Interior Point Method

We remind our reader that our goal is to solve dual linear problems from Lemma 3. In the Section 3.1 we presented a general scheme for a primal-dual Interior Point Method for the case when primal problem has only equality constraints, as is the case for the problem of interest to us. In this section we discuss the details of the Algorithm 2, and prove our main claim presented in Lemma 4 by presenting an efficient way of computing one iteration of an IPM.

We first address the issues of initial conditions (x_0, y_0, z_0), stopping criterion and choosing the scaling factor σ_t. The IPM does not require the starting point or the iterates to be in feasible region. The only requirement is that all elements of vectors x_t, z_t are positive for $t \geq 0$. Hence we can choose almost arbitrary starting point, although in practice it is beneficial to choose x, z such that they are not to close to zero. We propose the following choice of x_0: for $1 \leq i \leq n-1$ we take $(x_0)_i = (x_0)_{n+i} = d_i/2$, and $(x_0)_n = 2/3$, $(x_0)_{2n} = 1/3$. We also let $(x_0)_{2n+i} = 1/3$ for $1 \leq i \leq k$. It is straightforward to check, that such x_0 satisfies $Ax_0 = b$, and $x_0 > 0$. Finding a point (y_0, z_0) that is dual feasible and $z_0 \geq 1$ is straightforward using the connection of the dual problem to L_1 regression described in (6). We choose a uniform vector $p \in \Delta_{k-1}$ and t_j equal to $|\sum_{i=1}^{k} p_i f_{i,j} - g_j| + 1$ for all $j \leq n-1$. Taking $y_0 = (t, p)$ and $z_0 = c - A^T y_0$ we get a dual feasible point with $z_0 \geq 1$. For now we just note that this construction can be easily computed using subroutines for computing $F^T v$ and $A^T v$ for given vectors v of appropriate size, and vector operations on vectors of size $\mathcal{O}(n+k)$. The efficient multiplication by F^T and A^T is described later.

The stopping criterion of the main loop of the IPM is triple (x_t, y_t, z_t) begin and ϵ-feasible ϵ-solution. The residuals r_p^t, r_d^t measure how far the triple (x_t, y_t, z_t) is from feasibility, while the duality gap $\langle x_t, z_t \rangle$ measures how far the point is from optimality. We say a point is an ϵ-solution it the duality gap is smaller than a given ϵ. For a point to be ϵ-feasible, the norms of the residuals have to be smaller than ϵ, that is $\|r_p^t\|_2 < \epsilon$ and $\|r_d^t\|_2 < \epsilon$. In practice usually different $\epsilon_o, \epsilon_p, \epsilon_d$ are chosen based on the problem, and the stopping criterion is $\langle x_t, z_t \rangle \leq \epsilon_o$, $\|r_p^t\|_2 < \epsilon_p$ and $\|r_d^t\|_2 < \epsilon_d$. We also note, that to prevent infinite loops practical algorithms stop after reaching a set maximal amount of iterations. From the perspective of our analysis, the most important fact is that checking the stopping criterion is of order $\mathcal{O}(n+k)$.

Choosing the scaling factor can be done in a variety of ways. The simplest method is to choose $\sigma_k = \gamma \in (0, 1)$. More sophisticated approaches like predictor-corrector method first find the Newton direction $(\hat{d}_x, \hat{d}_y, \hat{d}_z)$ for the most optimistic $\hat{\sigma}_t = 0$. Then the actual scaling factor σ_t is chosen based on how much reduction in duality gap could be achieved

while going in the direction $(\hat{d_x}, \hat{d_y}, \hat{d_z})$. Again, for more details on methods of choosing the scaling factor we refer to [9] and references therein. We just note that there exist methods for choosing the scaling factor that are both practical and lead to theoretical guarantees on the number of iterations necessary for an IPM to converge. Furthermore for the state of the art methods the cost of computing the scaling factor is the same as the cost of solving the system of linear equations for (d_x, d_y, d_z), and is actually done by solving a system of equations with the same RHS. As will be clear from what follows, the cost of solving this system of linear equations does not depend on the RHS.

The above discussion justifies our claim, that the cost of computing one step of an IPM is dominated by the cost of finding the solution of the system:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z_t & 0 & X_t \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} r_p^t \\ r_d^t \\ \sigma_t \mu_t \mathbf{1} - X^t z_t, \end{bmatrix} \tag{9}$$

where $r_p^t = b - Ax_t$ and $r_d^t = c - A^T y_t - z_t$.

As we previously noted, a common technique for solving the system of equations (9) is to reduce it to solving the normal equation $\Sigma d_y = r$, where:

$$\Sigma = A Z_k^{-1} X_k A^T$$

and

$$r = b + A(Z_k)^{-1}(X_k r_d^k - \sigma_k \mu_k \mathbf{1}).$$

This equation is arrived at, after applying a blockwise Gaussian elimination on the linear system (9). Given a solution d_y of the normal equation, we can compute d_x, d_z from equations $d_z = r_d^k - A^T d_y$ and $d_x = -x_k + (Z_k)^{-1}(\sigma_k \mu_k \mathbf{1} - X_k d_z)$.

Therefore as we have noted in Section 3.1 one iteration of IPM can be computed efficiently, if we can efficiently compute Av, $A^T v$ and solve $\Sigma v = w$ for v, w vectors of appropriate length, and $\Sigma = AHA^T$ where H is a diagonal matrix with positive elements on the diagonal. We devote the rest of this section to presenting and analyzing methods for those three computational problems, that take advantage of the specific structure of matrix A.

We first observe that matrix F is related to a sparse matrix. For simplicity we denote $m = \sum_{i=1}^{k} m_i$, where m_i are the sizes of theoretical spectra.

▶ **Lemma 5.** *Let U denote an upper-triangular $n \times n$ matrix, such that $U[i,j] = 1$ for $i \leq j$ and $U[i,j] = 0$ for $i > j$. Then there exists a sparse $m \times n$ matrix W, with m nonzero entries such that $F = WU$. Furthermore sparse representations of matrices W and W^T can be constructed in $\mathcal{O}((n+m)\log(n+m))$ time.*

Proof. Let $s_1 \leq \ldots \leq s_n$ be the ordered point from set $\mathcal{S}$. For any theoretical measure μ_i, we can represent μ_i as $\sum_{j=1}^{\mathcal{S}} w_{i,j}\delta_{s_j}$, whith only m_i nonzero elemets $w_{i,j}$ for each i. Take matrix W, such that $W[i,j] = w_{i,j}$. Then the matrix W is sparse and has a total of $\sum_{i=1}^{k} m_i$ nonzero elements. It is straightforward to check that $F = WU$, since for any i, j we have $WU[i,j] = \sum_{l \leq j} w_{i,l} = f_{i,j}$. In other words, the cdf of distribution μ_i on interval $[s_j, s_{j+1})$ is equal to the sum of probability masses in μ_i with x coordinate less or equal to s_{j+1}.

To finish the proof, we need to show how to construct a sparse representation of W and W^T. In a sparse representation of a matrix, we represent each row of W as a list of nonzero elements, i.e. $(j, w_{i,j})$ for j such that $w_{i,j} > 0$. The matrix W is represented as list of rows. This is a classic representation of a sparse matrix that allows to compute

Wv in time $\sum_{i=1}^{k} m_i$ for any vector $v \in \mathbb{R}^n$. Constructing the sparse representations of W and W^T is very easy, and needs to be done only once before the first iteration of the IPM. As previously let $\nu = \sum_{j=1}^{m_0} w_{0,j}\delta_{x_{0,j}}$ be the experimental spectrum and for $i = 1, 2, \ldots, k$ let $\mu_i = \sum_{j=1}^{m_i} w_{i,j}\delta_{x_{i,j}}$ be the i-th theoretical spectrum. We construct a list of triples $(x_{i,j}, i, w_{i,j})$ and sort it with respect to the first element. Then it is enough to pass once through this sorted list. Each triple with $i > 0$ corresponds to one element of the matrix W. Since we pass through the elements in increasing order of x, we can remember how many different x we have seen so far. For the triple (x, i, w), if we have seen l different x so far, then we know that we have $W[i, l] = w$, and we simply add (l, w) to i-th list in the representation of W and (i, w) to l-th list in the representation of W^T. The total cost of this construction is $\mathcal{O}((n + m) \log(n + m))$, due to the sorting of elements, and the memory needed for storing both representations is $\mathcal{O}(m)$. ◀

▶ **Lemma 6.** *For any vector $v \in \mathbb{R}^N$ and $w \in \mathbb{R}^{n+m-1}$ the products Av and $A^T w$ can be computed in $\mathcal{O}(n + k + m)$ time and using additional $\mathcal{O}(n + k)$ memory.*

Proof. The only nontrivial part of both of those operations is computing Fv or $F^T v$ for some vector v of appropriate length, which can be done efficiently thanks to the representation $F = WU$ given by Lemma 5. To compute $w = Fv$, we first compute $u = Uv$ which can be done in time $\mathcal{O}(n)$ by computing suffix sums of vector v, without the need to explicitly store matrix U. We need $\mathcal{O}(n)$ memory to store the result of this operation. Next, we have $w = Wu$. Thanks to the sparse representation, this multiplication can be done in $\mathcal{O}(m)$ time and we need $\mathcal{O}(k)$ memory for storing the result. Similarly, to compute $w = F^T v$, we first multiply v by W^T and then multiply the result by U^T which corresponds to computing prefix sums, and does not require explicit construction of U. As for multiplication by F we need $\mathcal{O}(n + k + m)$ time and $\mathcal{O}(n + k)$ memory for those operations. ◀

We are left with the task of solving a system of linear equations $\Sigma v = w$, where $\Sigma = AHA^T$ for a diagonal matrix H with positive elements on the diagonal. To prove that this can be done efficiently, we first prove.

▶ **Lemma 7.** *For any diagonal matrix G of size $n \times n$ the matrix FGF^T can be computed in $\mathcal{O}(m(k + m) + n)$ time and using $\mathcal{O}(m^2)$ memory.*

Proof. It is straightforward to check, that for $i, j \leq n$ we have $UDU^T[i, j] = \sum_{l=max_{i,j}}^{n} G[l, l]$. We have, also, $FGF^T = W(UDU^T)W^T$. Therefore if we compute the suffix sums of the diagonal of G in time and space $\mathcal{O}(n)$, we can compute $UDU^T[i, j]$ in constant time for any i, j when we need it. We denote $\alpha_{i,j} = UDU^T[i, j]$. Now let us choose $i, j \leq n$. We have:

$$FGF^T[i, j] = \sum_{p \leq n} \sum_{q \leq n} w_{i,p}\alpha_{p,q}w_{j,q}$$

Given the sparse representations of rows W_i, W_j the above sum could be computed in time $\mathcal{O}(m_i m_j)$, but there is a faster way. Notice that for $q \geq p$ we have $\alpha_{p,q} = \alpha_{q,q}$, and write:

$$\sum_{1 \leq p \leq q \leq n} w_{i,p}\alpha_{p,q}w_{j,q} = \sum_{1 \leq p \leq q \leq n} w_{i,p}\alpha_{q,q}w_{j,q} \tag{10}$$

The above sum can be computed in $\mathcal{O}(m_i + m_j)$. Let L_i, L_j be the lists containing the sparse representations of W_i, W_j. We first compute the suffix sums of $\alpha q, q w_{q,j}$ for $(q, \cdot) \in L_j$, which can be done in $\mathcal{O}(m_j)$ since list L_j is ordered by column number q. Then for any $(p, w_{i,p}) \in L_i$ we add to the result the suffix sum of $\alpha q, q w_{q,j}$ for the smallest $q \geq p$ such

that $(q, \cdot) \in L_j$. Since lists L_i, L_j are ordered by column numbers p, q, this can be done in $\mathcal{O}(m_i + m_j)$ time in standard way. Therefore (10) can be computed in $\mathcal{O}(m_i + m_j)$.

Similarly, the sum:

$$\sum_{1 \le q < p \le n} w_{i,p} \alpha_{p,q} w_{j,q} = \sum_{1 \le q < p \le n} w_{i,p} \alpha_{p,p} w_{j,q}$$

can be computed in $\mathcal{O}(m_i + m_j)$. Therefore after we compute suffix sums of the diagonal of G in time and memory $\mathcal{O}(n)$, the cell (i, j) of matrix FGF^T can be computed in $\mathcal{O}(m_i + m_j)$ time. We have

$$\sum_{1 \le i,j \le n} m_i + m_j = (2k-1)m$$

and therefore the whole matrix FGF^T can be computed in $\mathcal{O}(km + n)$ time and $\mathcal{O}(k^2 + n)$ memory. ◀

We can now prove that the normal equation can be solved efficiently, no matter the right hand side.

▶ **Lemma 8.** *Let $r \in \mathbb{R}^{n+k-1}$ be a vector and let H be a $(2n+k) \times (2n+k)$ diagonal matrix, with positive elements on the diagonal. Then the matrix AHA^T has full rank, and equation $AHA^T v = r$ can be found in $\mathcal{O}(k^3 + km + n)$ time and $\mathcal{O}(k^2 + n)$ space.*

Proof. First we present a usefull decomposition of matrix AHA^T. Suppose that $H = \operatorname{diag}(H_1, H_2, H_3)$ where H_1, H_2, H_3 are diagonal matrices of sizes n, n, k respectively. Then using the definition of A we can write:

$$AHA^T = \begin{bmatrix} J_n(H_1 + H_2)J_n^T & J_n(H_2 - H_1)F^T \\ F(H_2 - H_1)J_n^T & F(H_1 + H_2)F^T + H_3 \end{bmatrix} \tag{11}$$

We will use block LDU decomposition for the right hand side of the above equation. For a given matrix $B = \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix}$ with $B_{1,1}$ invertible the block LDU decomposition of B is:

$$B = \begin{bmatrix} I & 0 \\ B_{2,1}B_{1,1}^{-1} & I \end{bmatrix} \begin{bmatrix} B_{1,1} & 0 \\ 0 & B_{2,2} - B_{2,1}B_{1,1}^{-1}B_{1,2} \end{bmatrix} \begin{bmatrix} I & B_{1,1}^{-1}B_{1,2} \\ 0 & I \end{bmatrix}$$

We note, that $J_n(H_1 + H_2)J_n^T$ is a diagonal $(n-1) \times (n-1)$ matrix with positive elements on the diagonal, and is therefore invertible and easy to invert numerically. For brevity, we denote $K = J_n(H_1 + H_2)J_n^T$ and $L = J_n(H_2 - H_1)$ and $G = H_1 + H_2 - L^T K^{-1} L$. Using the block LDU decomposition for (11) we get:

$$AHA^T = \begin{bmatrix} I_{n-1} & 0 \\ FL^T K^{-1} & I_k \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & FGF^T + H_3 \end{bmatrix} \begin{bmatrix} I_{n-1} & K^{-1}LF^T \\ 0 & I_k \end{bmatrix} =: LMR$$

From above representation it follows, that AHA^T has full rank. Indeed, it is a product of square matrices L, M, R, and it is obvious that L and R have full rank. Furthermore M also has full rank. To see why, observe that G is a diagonal matrix with positive elements. This follows from the fact that elementwise we have $G \ge H_1 + H_2 - (H_2 - H_1)^2/(H_1 + H_2)$ (with equality on all elements except the right bottom row). On the other hand $H_1 + H_2 - (H_2 - H_1)^2/(H_1 + H_2) = 4H_1H_2/(H_1 + H_2)$ which is a diagonal matrix with positive elements. Therefore FGF^T is nonnegative definite, and since H_3 is positive definite, we conclude that

$FGF^T + H_3$ is positive definite. Since K is positive definite as well, M has full rank and so does AHA^T.

For given vector r we would like to solve the equation $LMRv = r$. An easy way to do this, is first solve equation $Lv_1 = r$, then solve equation $Mv_2 = v_1$ and lastly $Rv_3 = v_2$. Then we have $LMRv_3 = LMv_2 = Lv_1 = r$. We can therefore work with each matrix L, M, R separately.

Solving $Lv_1 = r$ is trivial. Let us assume that $r = (r_1, r_2)$, where r_1 is of length $n - 1$ and r_2 has length k. Then:

$$v_1 = \begin{pmatrix} r_1 \\ r_2 - FL^T K^{-1} r_1 \end{pmatrix}$$

which is easy to compute, since the cost of multiplying by $L^T K^{-1}$ is $\mathcal{O}(n)$, and we can efficiently multiply vector by F thanks to Lemma 6. Solving the linear equation $Rv_3 = v_2$ is equally easy, since we can efficiently multiply by F^T as well.

The only thing left is solving $Mv_2 = v_1$. Assume $v_1 = (u_1, u_2)$ where u_1 is of size $n - 1$ and u_2 is of size k. Let $P = FGF^T + H_3$. Then P is a positive definite matrix of size $k \times k$ that we can compute in $\mathcal{O}(km + n)$ thanks to Lemma 7. Let v' be a solution of $Pv' = u_2$, that we find using standard methods in time $\mathcal{O}(k^3)$ and space $\mathcal{O}(k^2)$. Then $v_2 = (u_1^T, v'^T)^T$ is the solution of equation $Mv_2 = v_1$.

Summing up, the cost of finding a solution to the equation $AHA^T v = r$ is $\mathcal{O}(k^3 + km + n)$, where $\mathcal{O}(k^3)$ is the cost of inverting a $k \times k$ matrix, $\mathcal{O}(km + n)$ is the cost of creating this matrix, and all other operations have cost linear in n, k, m. For computing the $k \times k$ matrix we need $\mathcal{O}(k^2)$ memory, but all other operations can be done using additional memory linear in n, k. ◀

Proof of Lemma 4. For each iteration we need to solve a finite amount of normal equations (maybe more than one, depending on our mechanism of choosing scaling factor) and additionally perform a finite amount of multiplications Av, $A^T v$, Fv, $F^T v$. Since the other operations can be done in time and memory linear in n, k, m, we conclude that one iteration of a primal-dual Interior Point Method can be done in $\mathcal{O}(k^3 + km + n)$ time and $\mathcal{O}(k^2 + n)$ memory. ◀