Snapshots of modern mathematics
from Oberwolfach

№ 8/2017

Computational Optimal Transport

Justin Solomon

Optimal transport is the mathematical discipline of matching supply to demand while minimizing shipping costs. This matching problem becomes extremely challenging as the quantity of supply and demand points increases; modern applications must cope with thousands or millions of these at a time. Here, we introduce the computational optimal transport problem and summarize recent ideas for achieving new heights in efficiency and scalability.

1 Introduction

The mathematical field of *optimal transport* (OT) continues to reach new levels of elegance and sophistication, but the basics are motivated by a practical problem in economics. The main idea is to find the best way to transport something from one or more sources to a distribution of targets.

Suppose we are in charge of distribution for a small chain of gas stations. The company has 4 refineries and 5 gas stations. For convenience, we'll number the refineries from 1 to 4 and use the index i to refer to a refinery: $i = 1$ refers to the first refinery and $i = 4$ refers to the fourth. Similarly, we'll use the index j to refer to an individual gas station, from $j = 1$ to $j = 5$.

We are given the following information:

- Refinery i can produce s_i gallons of gasoline per day; s_i is the *supply* provided by refinery i.
- Gas station j sells d_j gallons of gasoline per day; d_j is the *demand* needed by station j.
- It *costs* C_{ij} dollars to transport a gallon of gas from refinery i to station j. For example, C_{ij} may be large if refinery i is far away from station j.

For simplicity, we will assume that supply meets demand. That is, we do not produce extra gas, and the gas stations fill all their customers' tanks. Mathematically, this assumption is expressed that by ensuring that the sum of the s_i's over i equals the sum of the d_j's over j, in formulas $\sum_i s_i = \sum_j d_j$.

Our goal is to develop a *transportation plan* that determines how much gas to ship from each refinery to each station. More specifically, a transportation plan is a matrix $\boldsymbol{T}$ of values T_{ij} that tells us how much gas we should ship from refinery i to station j. For example, if $T_{24} = 100$, then we will ship 100 gallons from refinery 2 to gas station 4.

Not all matrices $\boldsymbol{T}$ make sense as transportation plans. At the very least, the transportation plan $\boldsymbol{T}$ must satisfy the following criteria:

- We cannot ship a negative amount of gas. Hence, $T_{ij} \geq 0$ for all refineries i and stations j.
- The supply s_i of every refinery i is depleted. That is,

  $$\sum_j T_{ij} = s_i, \text{ for all refineries } i \in \{1, 2, 3, 4\}.$$

  Therefore, the left-hand side of the equality is the total gas shipped from refinery i to *all* stations j. Concretely, the sum of all the elements of the i-th row of gives s_i.
- The demand d_j of every gas station j is met. That is,

  $$\sum_i T_{ij} = d_j, \text{ for all gas stations } j \in \{1, 2, 3, 4, 5\}.$$

  In other words, the j-th column of T_{ij} sums to d_j.

Example. *The following matrix is an example of a transportation plan:*

$$T = \begin{array}{r} \\ \textit{Refinery 1} \\ \textit{Refinery 2} \\ \textit{Refinery 3} \\ \textit{Refinery 4} \end{array} \begin{array}{c} \begin{array}{ccccc} \textit{Station 1} & \textit{Station 2} & \textit{Station 3} & \textit{Station 4} & \textit{Station 5} \end{array} \\ \begin{pmatrix} 1 & 3 & 2 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1.5 & 1.5 \\ 0 & 0 & 0 & 3 & 6 \end{pmatrix} \end{array}.$$

Here, the supply vector $\boldsymbol{s} = (s_i)$ *is* $s := (6, 5, 4, 9)$ *and the demand vector* $\boldsymbol{d} = (d_i)$ *is* $d := (6, 4, 2, 4.5, 7.5)$*. These vectors prescribe the row and column sums of the transportation plan, respectively. To link with our previous notation,* $T_{21} = 5$ *above indicates that refinery 2 sends 5 gallons of gasoline to station 1. Can you check that the supply is depleted and the demand is met?*

There are many ways to develop a transportation plan satisfying these requirements, called *constraints* in the mathematical literature. For instance, each refinery could send out 5 small trucks of gas, evenly splitting the load. But, this plan likely would not be cost effective! Instead, we should attempt to fulfill the demand of a station with gasoline from a nearby refinery to reduce costs. One way to formalize this idea is to seek a transportation plan that *minimizes* the total cost to the company, where

$$\text{Total cost} = \sum_{ij} \Bigg[\underbrace{C_{ij}}_{\text{Cost of shipping from } i \text{ to } j} \times \underbrace{T_{ij}}_{\text{Amount shipped from } i \text{ to } j} \Bigg].$$

Here the sum is taken over all possible combinations of indices i and j.

Combining this cost function with the constraints above yields an instance of the classic *Monge–Kantorovich optimal transport (OT) problem* [11, 13]:

minimize	$\sum_{ij} C_{ij} T_{ij}$	[Total cost]
subject to	$T_{ij} \geq 0$ for all refineries i and stations j	[Positive mass]
	$\sum_j T_{ij} = s_i$ for all refineries i	[Supply depleted]
	$\sum_i T_{ij} = d_j$ for all gas stations j.	[Demand met]

In words, OT seeks to minimize the total cost of moving products from suppliers to purchasers given that all demand is met and all supplies are depleted.

Example. *Continuing the previous example, suppose the cost matrix is:*

$$C = \begin{array}{c} \\ \textit{Refinery 1} \\ \textit{Refinery 2} \\ \textit{Refinery 3} \\ \textit{Refinery 4} \end{array} \begin{array}{c} \begin{array}{ccccc} \textit{Station 1} & \textit{Station 2} & \textit{Station 3} & \textit{Station 4} & \textit{Station 5} \end{array} \\ \begin{pmatrix} 7 & 4 & 1 & 2 & 3 \\ 5 & 1.2 & 2.5 & 7 & 0.1 \\ 9 & 6 & 7 & 8 & 6 \\ 2 & 2 & 1 & 5 & 6.5 \end{pmatrix} \end{array}.$$

For example, $C_{25} = 0.1$ *expresses that it costs \$0.10 to ship a gallon of gasoline from refinery 2 to station 5. The cost of the transport plan in the previous example is*

$$7\times1 + 4\times3 + 1\times2 + 5\times5 + 6\times1 + 8\times1.5 + 6\times1.5 + 5\times3 + 6.5\times6 = \$127.$$

The optimal plan can be shown to be

$$T_{\text{optimal}} = \begin{array}{c} \scriptstyle\textit{Refinery 1} \\ \scriptstyle\textit{Refinery 2} \\ \scriptstyle\textit{Refinery 3} \\ \scriptstyle\textit{Refinery 4} \end{array} \overset{\begin{array}{ccccc} \scriptstyle\textit{Station 1} & \scriptstyle\textit{Station 2} & \scriptstyle\textit{Station 3} & \scriptstyle\textit{Station 4} & \scriptstyle\textit{Station 5} \end{array}}{\begin{pmatrix} 0 & 0 & 1.5 & 4.5 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 1.5 & 0 & 0 & 2.5 \\ 6 & 2.5 & 0.5 & 0 & 0 \end{pmatrix}},$$

with cost

$$1\times1.5 + 2\times4.5 + 0.1\times5 + 6\times1.5 + 6\times2.5 + 2\times6 + 2\times2.5 + 1\times0.5 = \$52.50.$$

This new plan yields a savings of \$127.00 − \$52.50 = \$74.50 *for our gas station company over the old plan. In the following sections, we will describe methods that find an optimal plan for these types of problems.*

2 The Optimal Transport Problem

Many variations of the basic OT problem could be proposed, expressing many possible situations that might be encountered by our chain of gas stations. For instance, we could attempt to maximize profits rather than minimize costs. Or, we could introduce inequalities to deal with the case when there is an oversupply of gasoline or when it is impossible to meet the demand with the gasoline we have in stock. If we are shipping television sets rather than gasoline, we may add *integrality* constraints to avoid an optimal solution whereby a television must be cut in half. Each of these changes throws a wrench into the basic machinery of OT but can be addressed within the same basic framework.

The problem of optimal transport is as old as commerce itself, and it comes as no surprise that theoretical and applied mathematicians alike have dedicated considerable attention to its analysis. A version of the problem in which every source can ship to only one target was proposed by Gaspard Monge in 1781. Centuries later, Leonid Kantorovich received the 1975 Nobel Prize in Economics for his study of the more general version of the problem laid out in the introduction. Since the invention of high-speed computing technology, transport problems appear in shipping and operations, computational data analysis, and countless disciplines in between.

In its most basic form, OT can be viewed as a tool for comparing *histograms* or *distributions*. A histogram is simply a list of nonnegative values that sum to one, such as $\{0.2, 0.3, 0.3, 0.1, 0.1\}$. In the gas station example, the problem can be written in terms of histograms by dealing in *percentages* of the total amount of distributed gallons, rather than in gallons themselves.

The OT problem takes as input two histograms s and d. The *transportation cost* of transforming s into d is then defined as

$$\mathrm{OT}(s,d;C) := \begin{cases} \text{minimum of} & \sum_{ij} C_{ij}T_{ij} & \text{[Total cost]} \\ \text{subject to} & T_{ij} \geq 0 \text{ for all } i,j & \text{[Positive mass]} \\ & \sum_j T_{ij} = s_i \text{ for all } i & \text{[Supply depleted]} \\ & \sum_i T_{ij} = d_j \text{ for all } j. & \text{[Demand met]} \end{cases} \tag{1}$$

Here the minimum is taken over all possible transportation plans $\boldsymbol{T}$. The gas station example can be notated $\mathrm{OT}(s,d;C)$.

Intuitively, $\mathrm{OT}(s,d;C)$ measures the dissimilarity between s and d. For example, if the transportation cost of moving supplies to the gas stations is small, then relatively few gallons of gas have to travel very far to reach their final destinations. Contrastingly, if $\mathrm{OT}(\cdot)$ is large, then *any* strategy used to transport the gasoline from the refineries to the stations is going to cost the company serious money. This occurs because $\mathrm{OT}(\cdot)$ is obtained by minimizing the cost. If the minimum cost is very high, all other strategies will be even more expensive.

In some cases, $\mathrm{OT}(\cdot)$ truly can be providing us with a notion of *distance* between histograms s and d. Rather than thinking of refineries and gas stations, suppose we run a chain of bookstores. A popular textbook is carried at every store; store i has supply s_i of the book on its shelves. At the end of the day, the chain collects demands d_j representing the number of copies of that book ordered to be picked up at bookstore j. If C_{ij} contains the cost of shipping a book from bookstore i to bookstore j, then $\mathrm{OT}(s,d;C)$ is the minimum cost to the chain of bookstores to redistribute their books so that every order is satisfied.

Unlike the gas station–refinery example, here indices i and j *both* represent bookstores (as opposed to i representing a gas station and j representing a refinery); that is, s and d are histograms *over the same space.* In examples like these, $\mathrm{OT}(\cdot)$ can satisfy the following "distance-like" properties:

Non-negativity. Let's assume that shipping books always incurs a cost, so no shipment is free. Mathematically, this condition is equivalent to assuming $C_{ij} > 0$ whenever $i \neq j$. Furthermore, let's assume that it is free to leave a book at the same store, that is, $C_{ii} = 0$ for all i. Then, $\mathrm{OT}(s,d;C) \geq 0$ with equality exactly when $s = d$. In other words, moving books never *earns* us money, and any time that supply and demand do not align exactly, the cost of shipment is strictly positive.

Symmetry. When it costs the same to ship from bookstore i to bookstore j as it does to ship from bookstore j to bookstore i for all pairs (i,j), we say that $\boldsymbol{C}$ is a *symmetric* matrix that satisfies $C_{ij} = C_{ji}$ for all pairs (i,j).

In this case, for any s, d we know $\mathrm{OT}(s, d; C) = \mathrm{OT}(d, s; C)$. In words, after shipping out books from s to d, we can ship back from d to s for the same cost.

Triangle inequality. This final property is somewhat more involved. Suppose $\boldsymbol{C}$ is a *metric matrix*, that is, it satisfies the following two properties:

1. The cost of not shipping is zero: $C_{ii} = 0$ for all i.
2. The entries of $\boldsymbol{C}$ satisfy the triangle inequality for all (i, j, k) triplets:

$$C_{ij} + C_{jk} \geq C_{ik}.$$

   In words, shipping from i to j and then from j to k is never cheaper than shipping directly from i to k.

Then, $\mathrm{OT}(\cdot)$ satisfies the *triangle inequality*. Formally, for three histograms s, d, and d', we have

$$\mathrm{OT}(s, d'; C) + \mathrm{OT}(d', d; C) \geq \mathrm{OT}(s, d; C).$$

We invite the reader to see [5] for a careful proof of this property. It expresses the intuitive idea that it is economical to ship directly from source to target rather than stopping through an intermediate station.

The observation that $\mathrm{OT}(\cdot)$ satisfies the properties above lies the tip of a large mathematical iceberg. Continuing to modify our analogy slightly, suppose instead of moving books that we are given piles of sand to transport; for example, maybe we are planning how to move sand from stockpiles onto a network of highways after a snowstorm. Eventually it might become impractical to describe the problem in terms of individual grains of sand; instead, we could seek a *continuum* theory of transport that describes the motion of sand similarly to flow of water through a set of pipes. Starting from this new analogy, the most general language for these problems is given by the mathematical theory of optimal transport, and the smooth analog of $\mathrm{OT}(\cdot)$ is called the *Wasserstein distance*. See [18] for details.

3 Modern Applications

It is not obvious how to solve the $\mathrm{OT}(\cdot)$ problem in equation (1) even when operating with just 4 refineries and 5 gas stations, but with some effort one can formulate "brute force" algorithms enumerating all reasonable transportation plans and taking the best. These algorithms, however, are too slow for modern applications of optimal transport, which might need to match millions of suppliers and consumers. In this section, we sketch a few of these applications to motivate the need for fast, practical optimal transport algorithms that *scale* to huge data sets.

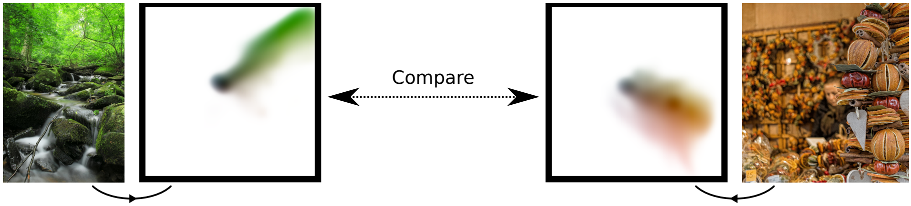

Figure 1: Two images might be considered visually similar if they have similar histograms of colors; optimal transport is used to compute distances between color histograms.

Supply chain management. Perhaps the most obvious application of optimal transport is in the world of *supply chain management*, in which companies seek to minimize shipping costs while moving products around the world. While road networks in the 18th century might have been simple enough to allow for solving transport problems by hand, nowadays advanced computational machinery is needed to contend with the complexities of modern business logistics. For instance, as of May 2017, Amazon.com sends products from 383 facilities to over 54 million Amazon Prime members alone [9, 10]. Carefully optimizing the cost of shipping products to a huge network of customers – over a network of millions of roads – can save money and energy while supplying customers with the products they demand.

Computer vision and image processing. Digital images are now essential to our daily life. Each image we look at contains extremely detailed and complex information, encoded using millions of pixel colors. Further complicating matters, websites like Flickr and Google Images must contend with not just a single image but *billions* of them. By one estimate, 657 billion photos are uploaded to the internet every year [8], and this number is likely to increase in the future. Algorithms must be invented that can quickly navigate databases of photos to pick out interesting ones or search for matches.

One simple way to work with an image is to treat it as *histogram* of features, a compact description of the interesting and salient features it contains. For instance, in artistic endeavors, images might be reduced to their *color palettes*. Color palettes are histograms that count how many pixels in an image are of each color. For example, a photograph of a leafy tree might have a lot of green pixels while a photograph of a tomato will have many red ones. As illustrated in Figure 1, these palettes are easier to compare than the full set of pixel colors.

Optimal transport can then be used to ask how similar two images are based own how similar their color palettes – or histograms – are. This approach was pioneered in the early computer vision literature, where optimal transport comes under the name "earth mover's distance" [14].

Shape matching. Modern software for *geometry processing* provides tools for understanding and comparing complex shapes. As input, on the consumer's end, 3D scanners can quickly obtain the shape of a human in front of a camera. Similarly, in the medical imaging world, magnetic resonance imaging technology (also known as MRI) allows us to look inside the human body at the shape of the brain and other organs. Once we obtain a shape, the task of *shape matching* seeks to transfer information from one shape to another. For instance, in brain imaging we might wish to transfer the painstaking work of labeling the individual folds on a brain from a previously-labeled patient to a new scan, by matching the bends and twists of their grey matter surfaces.

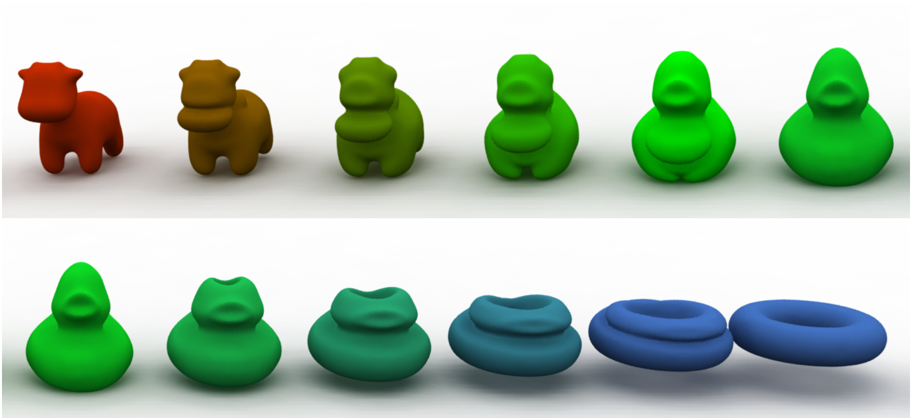

Figure 2: Shape interpolation computed using optimal transport.

Figure 2 illustrates one application of shape matching, enabled by modern algorithms for optimal transport. Here, we take three volumes – a cow, a duck, and a torus – and match these volumes using optimal transport. Once this matching is computed, we can interpolate between the geometries of the different shapes. While the interpolation in this example is somewhat artificial, machinery for this task can be used in computer graphics to mix and match geometric objects to model new shapes.

4 Computational Optimal Transport

While mathematicians in the 18th century may have been satisfied by solving transport problems between five or ten facilities, the examples above illustrate why we need high-speed, accurate algorithms for optimal transport that scale to the demands of modern "big data." Recent computer systems can optimize matchings between thousands or millions of facilities in a fraction of a second, enabling new applications never dreamed of in the early days of transport as a mathematical discipline.

Modern research at the intersection of computer science and mathematics seeks to design exact and approximate algorithms for optimal transport that are as fast as possible. Here we outline a few of the big ideas in computational optimal transport, which enable the applications in §3.

4.1 Linear Programming

The basic Monge–Kantorovich optimal transport problem (1) is an example of a *linear program*. That is, it is an optimization problem whose objective is linear in the unknown variables T_{ij}, with linear inequality and equality constraints. This is a classic problem in the worlds of optimization and operations research, for which there exist many solution algorithms. Among these we find, for example, the famous *simplex algorithm* developed by Dantzig in the 1940s [6].

A popular approach to the optimal transport linear program is the *Hungarian algorithm*, christened in 1955 by Harold Kuhn in honor of work by Hungarian mathematicians Dénes Kőnig and Jenő Egerváry [12]. This algorithm iteratively improves an estimate of the transportation matrix T_{ij}, refining the estimate until it reaches an output that minimizes the objective function and satisfies the constraints that T_{ij} transports all supply to all demand. For n sources of supply or demand (e.g. $n = 10$ would indicate that we match 10 refineries to 10 gas stations), this algorithm takes on the order of n^3 steps. This means that matching even 100 objects to each other using the Hungarian algorithm can take on the order of 100^3, that is one million, steps! Another popular method called the *auction algorithm* [3] uses an analogy to bidding to solve for T, but can suffer from similar scaling issues.

While these are are well-known historical algorithms that can be used to solve transportation problems, they work best for small or medium sized problems. They are designed for the most difficult possible case: solving transportation problems exactly, with no assumptions on the cost C_{ij}.

4.2 Entropic Regularization

Jumping forward several decades, interest in applications of optimal transport was renewed in the computational community motivated in large part by research in machine learning [4]. This breakthrough research, inspired by the large-scale transport applications suggested in §3, proposed solving a *regularized* version of optimal transport.

Regularizing a problem means that you modify the problem slightly to make it easier or more tractable. In the context of optimal transport, the Monge–

Kantorovich objective function $\sum_{ij} C_{ij}T_{ij}$ in equation (1) is replaced with a new one that contains a second term:

$$\sum_{ij}\Big[\underbrace{C_{ij}T_{ij}}_{\text{transport cost}} + \gamma \underbrace{T_{ij}\log T_{ij}}_{\substack{\text{entropic}\\ \text{regularizer}}}\Big]. \tag{2}$$

When $\gamma = 0$, we recover the original Monge–Kantorovich problem (1). As γ increases, however, the optimization objective begins to favor minimizing the second term, which we recognize as a contribution that equals minus the *entropy* of $\boldsymbol{T}$. Eliminating double negatives, large γ values increase the entropy of $\boldsymbol{T}$.

In more detail, the entropy of the matrix $\boldsymbol{T}$ is defined as

$$H(\boldsymbol{T}) := -\sum_{ij} T_{ij}\log T_{ij}.$$

Notice the negative sign, so we can understand the objective function in (2) as adding $-\gamma H(\boldsymbol{T})$ to the transport problem. Entropy is a measure of *disorder*, which increases in the matching $\boldsymbol{T}$ as γ is increased. This manifests itself as increased *fuzziness* in $\boldsymbol{T}$, from an extremely sharp matching when $\boldsymbol{T} = 0$ to a blurrier one for larger regularizing coefficients γ.

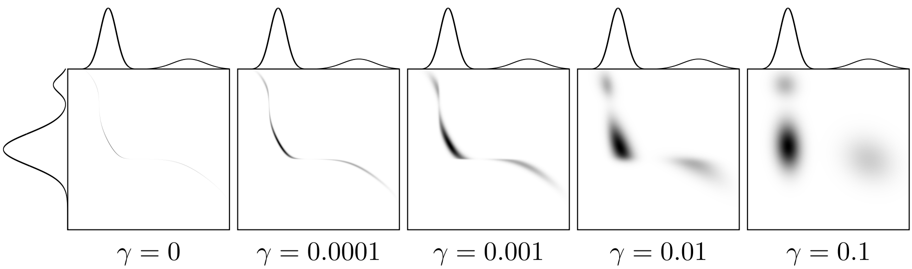

Figure 3: Entropy-regularized transportation plans for different values of γ. Here, s_i is plotted on the vertical (y) axis as a function of i, and d_j is plotted on the horizontal (x) axis. The entropy-regularized plan $\boldsymbol{T}$ for cost $C_{ij} = |i-j|^2$ is colored from white to black in each box.

Figure 3 shows the effect of increasing γ. Here, we solve optimal transport between two one-dimensional distributions, illustrated as functions on the x and y axes of a box. The matrix $\boldsymbol{T}$ is shown in black-and-white inside the box. When $\gamma = 0$, the nonzero elements of the matching $\boldsymbol{T}$ occupy a tiny sliver of the matrix $\boldsymbol{T}$. As γ increases, the matching preserves some of its structure but does not give as clear a map. From a broader perspective, as γ gets larger and larger, $\boldsymbol{T}$ deviates from solving the original problem more and more.

A question comes natural at this point: Why sacrifice the accuracy of $\boldsymbol{T}$ by increasing γ from zero? In exchange for the imprecision of solving the "wrong" problem (2), we gain a theoretical property leading to extremely fast algorithms. To this end, define

$$K_{ij} := e^{-C_{ij}/\gamma}. \tag{3}$$

This definition does not make sense when $\gamma = 0$, underscoring the need to regularize our problem. Then, it can be shown, using the technique of Lagrange multipliers, that optimizing the regularized function (2) subject to the constraints in problem (1) yields the condition

$$T_{ij} = u_i K_{ij} v_j, \tag{4}$$

for unknown vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. This property is critical. While there are a quadratic number of elements in the unknown matrix $\{T_{ij}\}_{i,j=1}^n$, now we have only a linear number of unknowns $\{u_i, v_i\}_{i=1}^n$. In other words, we now have to compute $2n$ values instead of n^2, a much smaller problem size!

Next, inserting equation (4) into the constraints of the transport problem (1) gives the following two relationships:

$$\begin{aligned} \textstyle\sum_j T_{ij} = s_i \quad &\Longrightarrow u_i \textstyle\sum_j K_{ij} v_j = s_i \Longrightarrow \boxed{u_i = \frac{s_i}{\sum_j K_{ij} v_j}} \\ \textstyle\sum_i T_{ij} = d_j \quad &\Longrightarrow v_j \textstyle\sum_i K_{ij} u_i = d_j \Longrightarrow \boxed{v_j = \frac{d_j}{\sum_i K_{ij} u_i}} \end{aligned}$$

These equations show you how to compute $\boldsymbol{u}$ from $\boldsymbol{v}$ and vice versa. The only remaining issue is that we know neither $\boldsymbol{u}$ nor $\boldsymbol{v}$!

The key insight used in entropic optimal transport [4], originally proposed for different applications by Sinkhorn in the 1960s [15], is to use these relationships anyway. The end result, called *Sinkhorn's Algorithm* and illustrated in Algorithm 1, estimates $\boldsymbol{u}$ and $\boldsymbol{v}$ jointly by first approximating $\boldsymbol{u}$ from $\boldsymbol{v}$, then $\boldsymbol{v}$ from $\boldsymbol{u}$, then $\boldsymbol{u}$ from $\boldsymbol{v}$, and so on until both $\boldsymbol{u}$ and $\boldsymbol{v}$ converge to a final value. Given very weak conditions, it can be shown that this algorithm will reach the correct values for $\boldsymbol{u}$ and $\boldsymbol{v}$ regardless of the initial guess.

Algorithm 1 is remarkable for a number of reasons. In five lines of code, it extracts an approximation of the solution to transport problems, which otherwise must be solved using relatively complex and specialized techniques. For practical problems, it typically takes relatively few iterations for $\boldsymbol{u}$ and $\boldsymbol{v}$ to converge; this is particularly true when the regularizing coefficient γ is large. The elegance and simplicity of this algorithm arguably inspired a much larger audience of practitioners to experiment with transport in their target applications.

$$
\begin{array}{ll}
\textbf{function } \textsc{Sinkhorn}(s, d; C, \gamma) & \\
\quad u, v \leftarrow \mathbb{1} & \textit{// Initialize the unknown vectors } \boldsymbol{u} \textit{ and } \boldsymbol{v} \\
\quad \boldsymbol{K} \leftarrow e^{-\boldsymbol{C}/\gamma} & \textit{// Kernel matrix} \\
\quad \textbf{for } i = 1, 2, 3, \ldots & \textit{// Loop until } \boldsymbol{u} \textit{ and } \boldsymbol{v} \textit{ converge} \\
\quad\quad \boldsymbol{u} \leftarrow s \oslash \boldsymbol{K} v & \\
\quad\quad \boldsymbol{v} \leftarrow d \oslash \boldsymbol{K}^\top u & \\
\quad \textbf{return } T_{ij} := u_i K_{ij} v_j & \textit{// Apply equation (4)}
\end{array}
$$

Algorithm 1: Sinkhorn iteration for entropy-regularized transport. The operation $\oslash$ denotes element-wise division.

4.3 Structured Instances

Entropic regularization makes transport easier by approximating the solution rather than solving for full precision. While regularization and approximation lead to fast generic algorithms, an alternative approach to speeding up transport is to *add structure*. Rather than developing algorithms for all possible scenarios, we can develop specialized "optimal transport for X" algorithms, where the structure of the application "X" reveals context-specific ways to make transport faster. Concretely, we typically put assumptions on the structure of the cost matrix $\boldsymbol{C}$ in the transport problem (1). Here, we mention a few such approaches.

Graphs. For transport over a road network, one way to add structure is to assume that C_{ij} is the *shortest path* from node i to node j in a network, or graph. This assumption provides a connection to literature in graph theory, where the optimal transport problem comes under the name *minimum-cost flow* and can be solved using network flow algorithms from graph theory.

Fluid flow. In 2000, Benamou and Brenier published a landmark research paper connecting optimal transport problems to *fluid flow*, when the costs C_{ij} come from distances in space [2]. In their language, if s and d are supported in space and C_{ij} is the squared distance $C_{ij} = \|x_i - x_j\|^2$, then $\mathrm{OT}(\cdot)$ equals the minimum amount of kinetic energy it takes to transform s into d by flowing them like fluids. This is not only a remarkable theoretical observation but leads to a totally different class of algorithms for solving transport problems based on methods for fluid dynamics. Figure 4 illustrates an example of a flow p_t from s to d, transitioning from $t = 0$ to $t = 1$.

Semidiscrete transport. Another special case of transport appears when transporting from a continuum to a set of discrete points. For instance, perhaps we wish to approximate a shaded image with a stippling of black-and-white points [7]. These problems are known as *semidiscrete* problems, since the target

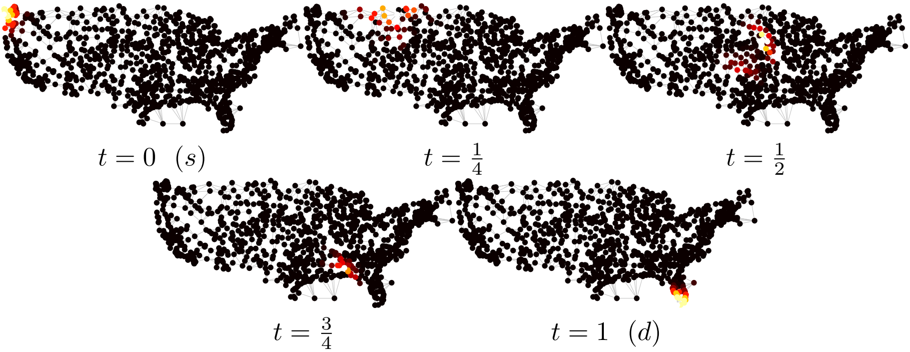

Figure 4: At time $t = 0$ a population is located in the northwest, and at time $t = 1$ it is in the southeast. Optimal transport is used to explain how the population moves diagonally along the map.

of transport is discrete. Algorithms for these problems involve constructions like *power diagrams* which divide up a region into pieces which all transport to the same point; see [1] for an overview.

5 Conclusion

Optimal transport is a unique mathematical discipline for many reasons. For one, the theory and practice of optimal transport are moving in lockstep: Theoretical advances in transport are improving the ways we solve transport problems in practice, while insights from the computational world have proven valuable in theory. Both abstract and applied instances of this problem remain active research topics.

Many open problems remain in the world of computational optimal transport. Some questions that remain to be addressed include:

- What is the best way to solve transport problems in the presence of stochasticity (randomness) or uncertainty?
- What are the best ways to incorporate optimal transport as a piece of larger algorithms for applications in machine learning and other fields?
- Do current transport algorithms approach theoretically optimal behavior, or do techniques exist that can enable fast processing of even larger datasets?

Future work aside, computational optimal transport already has proven itself valuable for countless end users. Enabled by new mathematics, faster algorithms and developments in computer hardware, transport uniquely benefits from mathematical and engineering insight while providing a toolbox for geometry, logistics, and matching.

Acknowledgments

J. Solomon acknowledges funding from an MIT Skoltech Seed Fund grant ("Boundary Element Methods for Shape Analysis") and from the MIT Research Support Committee ("Structured Optimization for Geometric Problems"), as well as Army Research Office grant W911NF-12-R-0011 (Smooth Modeling of Flows on Graphs).

Image credits

Fig. 1 created by the author for this publication.

Fig. 2 and Fig. 3 reprinted with permission from [16].

Fig. 4 reprinted with permission from [17].

References

[1] F. Aurenhammer, *Power diagrams: properties, algorithms and applications*, SIAM Journal on Computing **16** (1987), no. 1, 78–96.

[2] J. D. Benamou and Y. Brenier, *A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem*, Numerische Mathematik **84** (2000), no. 3, 375–393.

[3] D. Bertsekas and D. Castanon, *The auction algorithm for the transportation problem*, Annals of Operations Research **20** (1989), no. 1, 67–96.

[4] M. Cuturi, *Sinkhorn distances: Lightspeed computation of optimal transport*, Advances in Neural Information Processing Systems (NIPS) 26, 2013, pp. 2292–2300.

[5] M. Cuturi and D. Avis, *Ground metric learning*, The Journal of Machine Learning Research **15** (2014), no. 1, 533–564.

[6] G. Dantzig, *A history of scientific computing*, ACM, 1990, pp. 141–151.

[7] F. De Goes, K. Breeden, V. Ostromoukhov, and M. Desbrun, *Blue noise through optimal transport*, ACM Transactions on Graphics (TOG) **31** (2012), no. 6, 171.

[8] R. Eveleth, *How many photographs of you are out there in the world?*, The Atlantic (2015).

[9] MWPVL International, *Amazon global fulfillment center network*, 2017, http://www.mwpvl.com/html/amazon_com.html, visited on October 24, 2017.

[10] C. Isidore, *Amazon Prime now reaches nearly half of U.S. households*, CNN Tech (2016).

[11] L. Kantorovich, *On the transfer of masses (in russian)*, Doklady Akademii Nauk **37** (1942), no. 2, 227–229.

[12] H. Kuhn, *The Hungarian method for the assignment problem*, Naval Research Logistics Quarterly **2** (1955), no. 1-2, 83–97.

[13] G. Monge, *Mémoire sur la théorie des déblais et des remblais*, Histoire de l'Académie Royale des Sciences (1781), 666–704.

[14] 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** (2000), no. 2.

[15] R. Sinkhorn, *A relationship between arbitrary positive matrices and doubly stochastic matrices*, Annals of Mathematical Statistics **35** (1964), 876–879.

[16] J. Solomon, F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas, *Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains*, ACM Transactions on Graphics (Proc. SIGGRAPH 2015) **34** (2015), no. 4, 66:1–66:11.

[17] J. Solomon, R. Rustamov, L. Guibas, and A. Butscher, *Continuous-flow graph transportation distances*, arXiv:1603.06927v1, 2016.

[18] C. Villani, *Topics in optimal transportation*, American Mathematical Society, 2003.

Justin Solomon *is an X-Consortium Career Development Assistant Professor of Electrical Engineering and Computer Science at MIT.*

Mathematical subjects
Numerics and Scientific Computing

Connections to other fields
Computer Science

DOI
10.14760/SNAP-2017-008-EN

Snapshots of modern mathematics from Oberwolfach are written by participants in the scientific program of the Mathematisches Forschungsinstitut Oberwolfach (MFO). The snapshot project is designed to promote the understanding and appreciation of modern mathematics and mathematical research in the interested public worldwide. It started as part of the project “Oberwolfach meets IMAGINARY” in 2013 with a grant by the Klaus Tschira Foundation. The project has also been supported by the Oberwolfach Foundation and the MFO. All snapshots can be found on www.imaginary.org/snapshots and on www.mfo.de/snapshots.

Mathematisches Forschungsinstitut
Oberwolfach gGmbH
Schwarzwaldstr. 9 – 11
77709 Oberwolfach
Germany

Director
Gerhard Huisken

Mathematisches Forschungsinstitut Oberwolfach
Member of the Leibniz Leibniz Association
Klaus Tschira Stiftung gemeinnützige GmbH KTS
oberwolfach FOUNDATION
IMAGINARY open mathematics