Double Continuum Limit of Deep Neural Networks

Sho Sonoda [*1] **Noboru Murata** [1]

Abstract

The continuum limit is an effective method for modeling complex discrete structures such as deep neural networks to facilitate their interpretability. The continuum limits of deep networks are investigated with respect to two directions: width and depth. The width continuum limit is a limit of the linear combination of functions, or a continuous model of the shallow structure. We can understand that what shallow networks do is the ridgelet expansion of their approximating functions. The depth continuum limit is a limit of the composition of functions, or a continuous model of the deep structure. We can understand that what deep networks do is to transport mass to decrease a certain potential functional $\mathcal{F}$ of the data distribution. A discretization method can potentially replace the backpropagation. Specifically, we can synthesize a deep neural network from broken line approximation and numerical integration of a double continuum model, without backpropagation. In this study-in-progress, we have developed the ridgelet transform for potential field, and synthesized an autoencoder without backpropagation. In this paper, we review recent developments of the width and depth continuum limits, introduce our results, and present future challenges.

1. Depth Continuum Limit

The continuum limit with respect to depth is a recently developed technique. It is a continuum analogy of the deep neural network, formulated as a formal limit of the composition of functions as below:

$$\phi_L \circ \cdots \circ \phi_1(x) \;\to\; \phi_{t=T}(x), \tag{1}$$

where $\phi_\ell : H_\ell \to H_{\ell+1}$ $(\ell = 1, \ldots, L)$ with feature vector space H_ℓ is a feature map defined by the ℓ-th hidden layer, and $\phi_t : \mathbb{R}^m \to \mathbb{R}^m$ $(t \in [0, T])$ is a continuous analog of ϕ_ℓ's. In the following subsections, we identify ϕ_t as a transport map with *depth as time variable t*. We then obtain a finite deep neural network as a *broken line approximation* of trajectory $t \mapsto \phi_t(x)$.

The dynamical system viewpoint is not new to recurrent neural networks (Seung, 1998). These days, distribution-based formulations of deep neural networks are successful. For example, they are generative density estimators (Bengio et al., 2013), variational autoencoder (Kingma & Welling, 2014), reverse diffusion process (Sohl-Dickstein et al., 2015) and adversarial generative networks (Goodfellow et al., 2014). In shrinkage statistics, the expression of "transport map" $x + f(x)$ is known as Brown's representation of posterior (George et al., 2006). Liu & Wang (2016) analyzed it from a Bayesian viewpoint, apart from deep learning, and proposed kernel Stein discrepancy. Recently, we have observed that some kinds of convolutional networks can also be regarded as transport maps. Specifically, the skip connection structure $x+f(x)$ used in highway networks (Srivastava et al., 2015) and ResNet (He et al., 2016) is formally understood as a transport map.

1.1. Wasserstein Gradient Flow of Deep Network

Consider a transport map defined by a velocity vector field

$$\partial_t \phi_t(x) = \nabla V_t(\phi_t(x)), \quad x \in \mathbb{R}^m \tag{2}$$

with time-dependent potential function $V_t : \mathbb{R}^m \to \mathbb{R}$, where ∇ denotes the gradient operator on $\mathbb{R}^m$. When t is small, it has an "explicit" expression

$$\phi_t(x) = x + t\nabla V_t(x) + o(t^2) \quad \text{as} \quad t \to 0. \tag{3}$$

Generally, (2) is rewritten as an integral equation

$$\phi_t(x) = x + \int_0^t \nabla V_s(\phi_s(x)) \mathrm{d}s,$$

however the explicit solution is rarely tractable.

An important example of (2) is the Gaussian denoising autoencoder (DAE), where Alain & Bengio (2014) determined that $\phi_t(x) = x + t\nabla \log \mu_t(x) + o(t^2)$ as $t \to 0$ with data distribution μ_t. See Sonoda & Murata (2016) for transport theoretic reformulation of the DAE.

[*]Equal contribution [1]Waseda University, Tokyo, Japan. Correspondence to: Sho Sonoda <sho.sonoda@aoni.waseda.jp>.

Presented at the ICML 2017 Workshop on Principled Approaches to Deep Learning, Sydney, Australia, 2017.

Associated with a transport map, we consider data distributions. We write by μ_0 the initial state of the data distribution, or the probability density function of the input data x; and by μ_t its *pushforward* $\phi_{t\sharp}\mu_0$, or the probability density function of the "feature vector" $\phi_t(x)$. That is, $x \sim \mu_0$ and

$$\phi_t(x) \sim \mu_t (:= \phi_{t\sharp}\mu_0.)$$

Recall that a mass transportation is a change of variables, thus the pushforward measure μ_t satisfies the following equation

$$\mu_t(\phi_t(x))\,|\nabla\phi_t(x)| = \mu_0(x), \quad t \le 0,\, x \in \mathbb{R}^m. \tag{4}$$

As time t evolves, the data distribution μ_t changes according to the *continuity equation*

$$\partial_t \mu_t(x) = -\nabla \cdot [\mu_t(x) \nabla V_t(x)], \quad x \in \mathbb{R}^m \tag{5}$$

where $\nabla\cdot$ denotes the divergence operator on $\mathbb{R}^m$. The proof is simply to calculate the change of variables formula (4). This result is intuitively reasonable because the term ∇V in (2) corresponds to a flux from the viewpoint of hydrodynamics.

The continuity equation (5) has no explicit solution either, except for some special cases such as heat equation ($V_t = -\log \mu_t$). According to Otto calculus (Villani, 2009, Ex.15.10), the solution μ_t coincides with a trajectory of the *Wasserstein gradient flow*

$$\frac{\mathrm{d}}{\mathrm{d}t}\mu_t = -\mathsf{grad}\, \mathcal{F}[\mu_t], \tag{6}$$

with respect to a potential functional $\mathcal{F}$ that satisfies the following equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}[\mu_t] = \int_{\mathbb{R}^m} V_t(x)[\partial_t \mu_t](x)\mathrm{d}x.$$

Here grad denotes the gradient operator on L^2-Wasserstein space $W_2(\mathbb{R}^m)$.

The L^2-Wasserstein space $W_2(\mathbb{R}^m)$ is a functional manifold, or the family of probability density functions on $\mathbb{R}^m$ equipped with an infinite-dimensional Riemannian metric called the L^2-Wasserstein metric. While (6) is an ordinary differential equation on the space $W_2(\mathbb{R}^m)$ of probability density functions, (5) is a partial differential equation on Euclidean space $\mathbb{R}^m$. Hence, we use different time derivatives $\frac{\mathrm{d}}{\mathrm{d}t}$ and ∂_t.

The Wasserstein gradient flow (6) possesses a distinct advantage that the potential functional $\mathcal{F}$ does not depend on time t. In the following subsections, we will see that both the Boltzmann and Renyi entropy are examples of $\mathcal{F}$. The Wasserstein gradient flow facilitate the *interpretability* of deep neural networks because we can understand a deep network as a *transport map ϕ_t that transports mass to decrease the quantity $\mathcal{F}[\mu_t]$ of the data distribution.*

1.2. Example: Gaussian DAE

Sonoda & Murata (2016) determined that data distribution μ_t of the Gaussian DAE evolves according to the *backward heat equation*

$$\partial_t \mu_t(x) = -\triangle \mu_t(x), \quad \mu_{t=0} = \mu_0$$

and concluded that the feature map of the Gaussian DAE is equivalent to a transport map that *decreases* the entropy $\mathcal{H}[\mu] := -\int \mu(x)\log\mu(x)\mathrm{d}x$ of the data distribution:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mu_t = -\mathsf{grad}\,\mathcal{H}[\mu_t], \quad \mu_{t=0} = \mu_0,$$

where $\triangle$ denotes the Laplacian on $\mathbb{R}^m$. This is immediate because when $\mathcal{F} = \mathcal{H}$, then $V = -\log\mu_t$ and thus

$$\mathsf{grad}\,\mathcal{H}[\mu_t] = \nabla\cdot[\mu_t \nabla \log\mu_t] = \nabla\cdot\left[\mu_t \frac{\nabla\mu_t}{\mu_t}\right] = \triangle\mu_t,$$

which means (5) reduces to the backward heat equation.

1.3. Example: Renyi Entropy

Similarly, when $\mathcal{F}$ is the Renyi entropy

$$\mathcal{H}^\alpha[\mu] := \int_{\mathbb{R}^m} \frac{\mu^\alpha(x) - \mu(x)}{\alpha - 1}\mathrm{d}x,$$

then $\mathsf{grad}\,\mathcal{H}^\alpha[\mu_t](x) = \triangle\mu_t^\alpha(x)$ (see (Villani, 2009, Ex.15.6) for the proof) and thus (5) reduces to the *backward porous medium equation*

$$\partial_t \mu_t = -\triangle \mu_t^\alpha. \tag{7}$$

Open Question

Can we relate a potential functional $\mathcal{F}$ and an existing training procedure of deep learning? What is the best discretization strategy of a transport map?

2. Width Continuum Limit

The continuum limit with respect to width was developed in the 1990s. It is a continuum analogy of the shallow neural network, formulated as a formal limit of the linear combination of functions as below

$$\sum_{j=1}^n c_j \sigma(a_j \cdot x - b_j), \quad (a_j, b_j, c_j) \in \mathbb{R}^m \times \mathbb{R} \times \mathbb{C}$$
$$\to \int_{\mathbb{R}^m\times\mathbb{R}} c(a,b)\sigma(a\cdot x - b)\mathrm{d}\lambda(a,b), \tag{8}$$

which is also known as the *integral representation* of a neural network. Here $\sigma : \mathbb{R} \to \mathbb{C}$ is an activation function, $c(a,b)$ is a continuous analog of c_j, and

$\mathrm{d}\lambda(a,b)$ is an appropriate measure. Typically, σ is either Gaussian, sigmoidal function or ReLU; and $\mathrm{d}\lambda(a,b)$ is either the Lebesgue measure $\mathrm{d}a\mathrm{d}b$ or a Borel measure $|a|^{-(m-1)}\mathrm{d}a\mathrm{d}b$.

The integral representation theory is introduced by many authors (Poggio & Girosi, 1990; Mhaskar & Micchelli, 1992; Leshno et al., 1993; Barron, 1993; Girosi et al., 1995; Murata, 1996; Candès, 1998; Rubin, 1998) to investigate *how shallow networks work.*; and further developed by Donoho (2002); Le Roux & Bengio (2007); Kůrková (2012); Sonoda & Murata (2017). These studies are conducted from the standpoint of linear algebra. That is, they regard c_j and $\sigma(a_j \cdot x - b_j)$ as coefficients and basis functions respectively.

Today, we can understand that what shallow networks do is *the ridglet expansion of an integrable function* f. Note that it refers only to *what they can potentially do* and not to *what they actually do*.

Linear algebra was appropriate in the age of shallow networks. However, it lacks considerations of depth, and thus it is inadequate to explain *why deep networks perform better than shallow networks.* The depth continuum limit made a breakthrough by introducing dynamical system viewpoint and going beyond *what they actually do.*

2.1. Ridgelet Analysis

Ridgelet analysis is a well organized framework of the integral representation theory. The *ridgelet transform* $\mathcal{R}_\rho f(a,b)$ of an integrable function $f \in L^1(\mathbb{R}^m)$ with respect to a Schwartz function $\rho : \mathbb{R} \to \mathbb{C}$ is defined as

$$\mathcal{R}_\rho f(a,b) := C(a,b) \int_{\mathbb{R}^m} f(x)\overline{\rho(a \cdot x - b)}\mathrm{d}x, \qquad (9)$$

for every $(a,b) \in \mathbb{R}^m \times \mathbb{R}$, where $C(a,b)$ is an appropriate normalizing constant.

We say that ρ and σ are *admissible* when the integral $\int_{-\infty}^{\infty} \overline{\widehat{\rho}(\zeta)}\widehat{\sigma}(\zeta)|\zeta|^{-m}\mathrm{d}\zeta$ exists and not zero. Here $\widehat{\cdot}$ denotes the Fourier transform. A typical choice of ρ is a derivative of Gaussian function.

When ρ and σ are addmissible, then the reconstruction formula

$$\int_{\mathbb{R}^m \times \mathbb{R}} [\mathcal{R}_\rho f(a,b)]\sigma(a \cdot x - b)\mathrm{d}\lambda(a,b) = f(x), \qquad (10)$$

holds for every $f \in L^1(\mathbb{R}^m)$. That is, if we plug the ridgelet transform $\mathcal{R}_\rho f(a,b)$ in place of the coefficient $c(a,b)$ in (8), the integral representation network behaves as $f(x)$. Because the integral $\int_{\mathbb{R}^m \times \mathbb{R}}$ is an idealized limit of a finite sum $\sum_{j=1}^n$, the reconstruction formula represents the *universal approximation property* of neural networks.

It is intriguing that, in general, there exist infinitely many different ρ's that are admissible with the same activation function σ (see § 6 of Sonoda & Murata (2017) for example). It means that there are infinitely many different coefficients $c(a,b)$ that results in the same function $f(x)$. The backpropagation implicitly choose ρ without control, probably depending on the initial parameters, network structure and optimization algorithms.

2.2. Discretization Methods

A constructive discretization method can potentially replace the backpropagation. That is, by numerically integrating the reconstruction formula with a finite sum:

$$\text{LHS of (10)} \approx \sum_{j=1}^{n} c_j \sigma(a_j \cdot x - b_j), \qquad (11)$$

we can "synthesize" a neural network that approximates $f(x)$ without using backpropagation.

The backpropagation results in the so-called *black box* network in the sense that no one knows how the trained network processes information, because the training result is simply a local minimizer of a loss function that lacks control on the network parameters. In contrast, the discretization method could provide a *white box* network because the training result converges to a unique limit without any loss of the parameter controllability. The development of a discretization method with theoretical guarantees such as an error bound and a convergence guarantee is our important future work.

Today we have many discretization strategies: regular grid (frame) and atomic decomposition (Donoho, 1999), Monte Carlo integration (Sonoda & Murata, 2014), random feature expansion and/or kernel quadrature (Bach, 2017). Probabilistic numerics (Briol et al., 2016) is an emerging field that aims to unify these methods. Mhaskar (1996) estimated the approximation error as $O(n^{-s/m})$ with the number n of hidden units, input dimension m, and smoothness parameter (Sobolev order) s.

Open Question

Can we really replace backpropagation with discretization?

3. Double Continuum Limit

The double continuum limit is the width continuum limit of the depth continuum limit. In other words, it reduces to the ridgelet analysis of a transport map:

$$\int_{\mathbb{R}^m \times \mathbb{R}} \mathcal{R}_\rho[\mathsf{id} + t\nabla V](a,b)\sigma(a \cdot x - b)\mathrm{d}\lambda(a,b) \qquad (12)$$

where id denotes the identity map. Technically, the ridgelet transform is defined for integrable functions. Hence, we

consider a transport map with compact support.

3.1. Ridgelet Transform of Potential Vector Field

We present an integration-by-parts formula for the vector ridgelet transform. Let $K \subset \mathbb{R}^m$ be a compact set with smooth boundary ∂K. Given that a smooth scalar potential V is supported in K, the ridgelet transform of potential vector field ∇V is calculated by

$$\mathcal{R}_\rho[\nabla V](a,b) = -a\,\mathcal{R}_{\rho'}[V](a,b). \tag{13}$$

The proof is straitforward as below:

$$\begin{aligned}\mathcal{R}_\rho[\nabla V](a,b) &= C(a,b)\int_K \nabla V(x)\overline{\rho(a\cdot x-b)}\mathrm{d}x \\ &= C(a,b)\left[\int_{\partial K} V(x)\overline{\rho(a\cdot x-b)}n(x)\mathrm{d}S\right. \\ &\qquad \left. - a\int_K V(x)\overline{\rho'(a\cdot x-b)}\mathrm{d}x\right] \\ &= 0 - a\,\mathcal{R}_{\rho'}[V](a,b).\end{aligned}$$

The LHS of (13) denotes a vector ridgelet transform defined by element-wise mapping, whereas the RHS consists of a scalar ridgelet transform. We can understand the RHS given that the network shares common knowledge among element-wise tasks.

3.2. Example: Autoencoder

As the most fundamental transport map, we consider a smooth "truncated" autoencoder $\mathsf{id}_{r,\varepsilon}$. Denote by $\mathbb{B}^m(z;r)$ a closed ball in $\mathbb{R}^m$ with center z and radius r. We assume that $\mathsf{id}_{r,\varepsilon}$ is (1) smooth, (2) equal to the identity map id when it is restricted to $\mathbb{B}^m(r)$, and (3) truncated to be supported in $\mathbb{B}^m(r+\varepsilon)$ with a small positive number $\varepsilon > 0$. Let $V_{r,\varepsilon}$ be a smooth function that satisfies

$$V_{r,\varepsilon}(x) := \begin{cases} \frac{1}{2}|x|^2 & x \in \mathbb{B}^m(0;r), \\ (\text{smooth map}) & x \in \mathbb{B}(0;r+\varepsilon)\setminus\mathbb{B}(0;r), \\ 0 & x \notin \mathbb{B}^m(0;r+\varepsilon), \end{cases}$$

and let

$$\mathsf{id}_{r,\varepsilon} := \nabla V_{r,\varepsilon}.$$

Note that we can construct $\mathsf{id}_{r,\varepsilon}$ and $V_{r,\varepsilon}$ by using mollifiers, and thus such maps exist.

The ridgelet transform of the truncated autoencoder is given by

$$\mathcal{R}_\rho[\mathsf{id}_{r,\varepsilon}](a,b) \approx -KC(a,b)a\overline{\rho'(-b)} \quad \text{as} \quad \varepsilon \to 0 \tag{14}$$

with a certain constant K. See supplementary for the proof.

Therefore, by numerically integrating the integral representation

$$-K\int_{\mathbb{R}^m\times\mathbb{R}} a\overline{\rho'(-b)}\sigma(a\cdot x-b)\mathrm{d}a\mathrm{d}b \approx \mathsf{id}_{r,0}, \tag{15}$$

we can obtain an autoencoder *without backpropagation*.

Figure 1 depicts an autoencoder on $\mathbb{R}^2$ realized by numerically integrating (15). Note that we omitted calculating the normalizing coefficients and rescaled values instead.

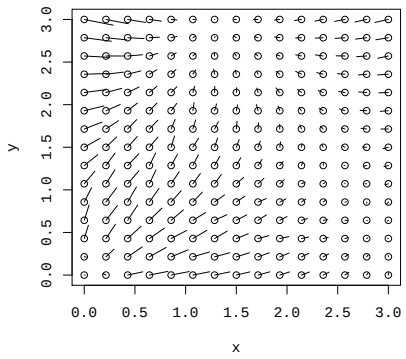

Figure 1. Autoencoder on $\mathbb{R}^2$ realized by discretizing the ridgelet transform of the truncated identity map, without backpropagation.

Open Question

According to Mhaskar (1996), the approximation error is estimated by the Sobolev order s of the transport map $\mathsf{id} + t\nabla V$. Can we determine any trade-off relation between the smoothness s and depth t? Can we estimate the generalization error?

4. Conclusion

We have provided an overview of depth (1) and width (8) continuum limits of neural networks, and developed the double continuum limit (12). We have introduced the ridgelet transform (13) for potential vector fields, and synthesized an autoencoder (15) without backpropagation. As suggested in the Wasserstein gradient flow (6), we expect that what a deep neural network does corresponds to a ridgelet transform $\mathcal{R}_\rho[\mathsf{id} + t\nabla V]$ of a transport map $\mathsf{id} + t\nabla V$ that decreases a functional $\mathcal{F}[\mu_t]$ of the data distribution μ_t. With respect to the double continuum limit, the development of discretization algorithms in collaboration with probabilistic numerics, estimation of the generalization error, are important topics for our future research.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful and constructive comments. This work is supported by the Waseda University Grant for Special Research Projects Number 2017S-119.

References

Alain, Guillaume and Bengio, Yoshua. What Regularized Auto-Encoders Learn from the Data Generating Distribution. *JMLR*, 15:3743–3773, 2014.

Bach, Francis. On the Equivalence between Kernel Quadrature Rules and Random Feature Expansions. *JMLR*, 18:1–38, 2017.

Barron, Andrew R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

Bengio, Yoshua, Yao, Li, Alain, Guillaume, and Vincent, Pascal. Generalized denoising auto-encoders as generative models. In *NIPS2013*, pp. 899–907, 2013.

Briol, François-Xavier, Oates, Chris. J., Girolami, Mark, Osborne, Michael A., and Sejdinovic, Dino. Probabilistic Integration: A Role for Statisticians in Numerical Analysis? 2016.

Candès, Emmanuel Jean. *Ridgelets: theory and applications*. PhD thesis, Standford University, 1998.

Donoho, David L. Emerging applications of geometric multiscale analysis. *Proceedings of the ICM, Beijing 2002*, I:209–233, 2002.

Donoho, David Leigh. Tight frames of k-plane ridgelets and the problem of representing objects that are smooth away from d-dimensional singularities in $\mathbb{R}^n$. *Proceedings of the National Academy of Science of the United States of America (PNAS)*, 96(5):1828–1833, 1999.

George, Edward I., Liang, Feng, and Xu, Xinyi. Improved minimax predictive densities under Kullback?Leibler loss. *Annals of Statistics*, 34(1):78–91, 2006.

Girosi, Federico, Jones, Michael, and Poggio, Tomaso. Regularization Theory and Neural Networks Architectures. *Neural Computation*, 7(1):219–269, 1995.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative Adversarial Nets. In *NIPS2014*, pp. 2672–2680, 2014.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep Residual Learning for Image Recognition. In *CVPR*, pp. 770–778, 2016.

Kingma, Diederik P. and Welling, Max. Auto-Encoding Variational Bayes. In *ICLR2014*, pp. 1–14, 2014.

Kůrková, Věra. Complexity estimates based on integral transforms induced by computational units. *Neural Networks*, 33:160–167, 2012.

Le Roux, Nicolas and Bengio, Yoshua. Continuous Neural Networks. In *AISTATS2008*, pp. 404–411, 2007.

Leshno, Moshe, Lin, Vladimir Ya., Pinkus, Allan, and Schocken, Shimon. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

Liu, Qiang and Wang, Dilin. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. In *NIPS2016*, pp. 1–9, 2016.

Mhaskar, H. N. Neural Networks for Optimal Approximation of Smooth and Analytic Functions. *Neural Computation*, 8:164–177, 1996.

Mhaskar, H.N and Micchelli, Charles A. Approximation by superposition of sigmoidal and radial basis functions. *Advances in Applied Mathematics*, 13(3):350–373, 1992.

Murata, Noboru. An integral representation of functions using three-layered betworks and their approximation bounds. *Neural Networks*, 9(6):947–956, 1996.

Poggio, Tomaso and Girosi, Federico. Networks for approximation and learning. *Proceedings of the IEEE*, 78 (9):1481–1497, 1990.

Rubin, Boris. The Calderón reproducing formula, windowed X-ray transforms, and radon transforms in L^p-spaces. *Journal of Fourier Analysis and Applications*, 4 (2):175–197, 1998.

Seung, Sebastian H. Learning continuous attractors in recurrent networks. In *NIPS1997*, pp. 654–660, 1998.

Sohl-Dickstein, Jascha, Weiss, Eric, Maheswaranathan, Niru, and Ganguli, Surya. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *ICML2015*, volume 37, pp. 2256–2265, 2015.

Sonoda, Sho and Murata, Noboru. Sampling hidden parameters from oracle distribution. In *ICANN2014*, pp. 539–546, 2014.

Sonoda, Sho and Murata, Noboru. Decoding Stacked Denoising Autoencoders. 2016.

Sonoda, Sho and Murata, Noboru. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43 (2):233–268, 2017.

Srivastava, Rupesh Kumar, Greff, Klaus, and Schmidhuber, Jürgen. Highway Networks. In *ICML 2015 Workshop on Deep Learning*, 2015.

Villani, Cédric. *Optimal Transport: Old and New*. Springer-Verlag Berlin Heidelberg, 2009.