Wasserstein Information Geometry for Learning from Data

Guido Montúfar
montufar@math.ucla.edu

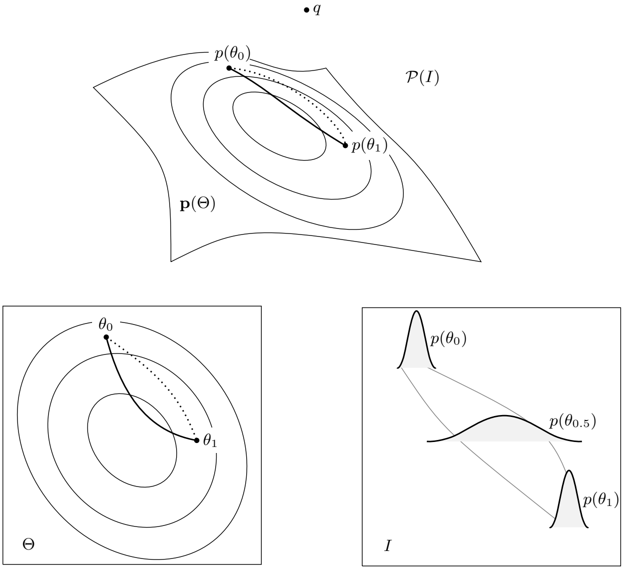
Tutorial at Geometry and Learning From Data, IPAM, March 2019

UCLA

1 Introduction

2 Information Geometry

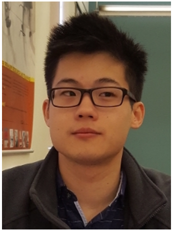
3 Wasserstein Information Geometry



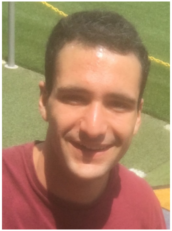
Parameter Space - Function Space - Data Space - Loss

- Develop approaches that can integrate the geometry of function space and geometry of data in learning
- Loss functions, optimization, and regularization techniques based on Information Geometry and Wasserstein Geometry

Wuchen Li



Alex Tong Lin



Yonatan Dukler

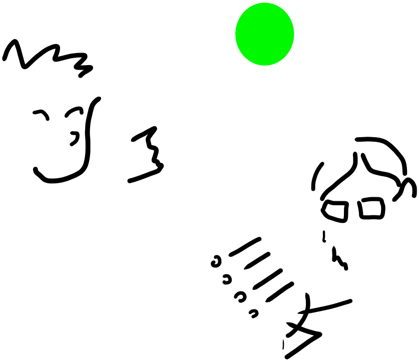
Motivation

- Information geometry derives geometric structures on the parameter space of a statistical model by pulling back structures from the space of probability distributions. The methods of information geometry have been successful in statistics and machine learning. However, in their current form they do not incorporate the geometry of sample space.
- Wasserstein geometry incorporates the geometry of sample space. Has been useful in specific applications, such as image retrieval and implicit generative models. However, current approaches do not integrate differential structures and natural gradients for parametrized models. Full potential in machine learning has yet to be developed.
- Wasserstein Information Geometry is about developing synergies between the two fields!

What is learning / from data?

- Learning: Try to obtain a general behavior based on data / examples
- Design: Try to obtain a general behavior based on expert knowledge

Data → Knowledge → Specifications → Hypotheses



“If you show a picture to a 3 yr and ask if there is a tree in it, you will likely get the correct answer. If you ask a 30 yr for the definition of a tree, you will likely get an inconclusive answer.” [AMMIL12]

Consider the problem of classifying handwritten 1s and 5s.

- We could try to write a list of properties characterizing a 1 (e.g., symmetric, straight, vertical), and try to obtain a classifier analytically.
- Obtaining such specifications may be very difficult in general, making the design approach unfeasible.
- However, data / examples should reveal a lot about this, and we might be able to obtain a good empirical solution.
- By learning from data, we can take an “end to end” approach that automatically selects the task relevant aspects.

- Nonetheless, the more meaningful priors we can incorporate, the faster and better we can expect to solve the problem.

  E.g., compositionality in deep nets / convolutional networks (model selection) / pre-trained filters (transfer learning) / implicit regularization / parametrization (resnet, centering tricks) / optimization methods (SGD, batch norm)
- At this, the geometry of the

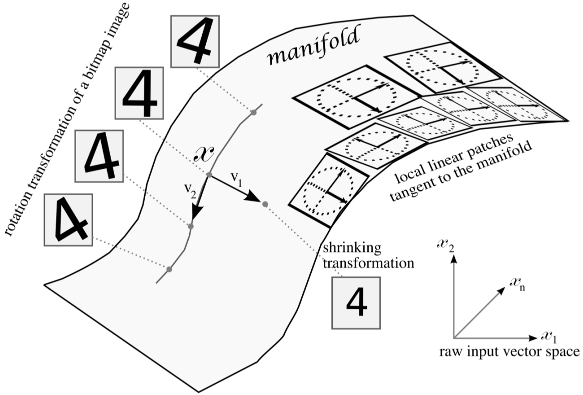
  data, models, parametrization, loss function,

  plays a key role

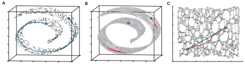
Geometry of data

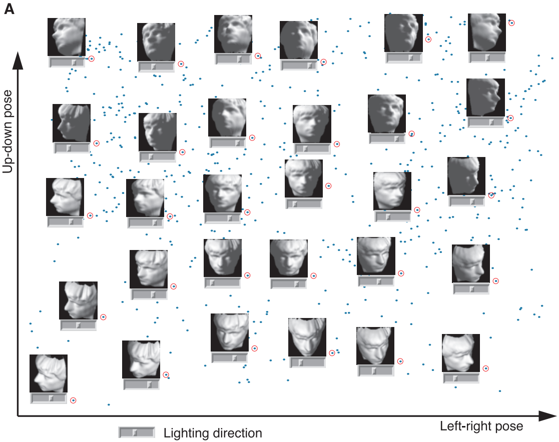
Manifold hypothesis

- Data is non uniformly distributed and is concentrated on lower dimensional sets.
- High dimensional data can be represented in a much lower dimensional feature space.
- In most cases, the relationship is non linear.
- The geometry of the data can be exploited for learning.

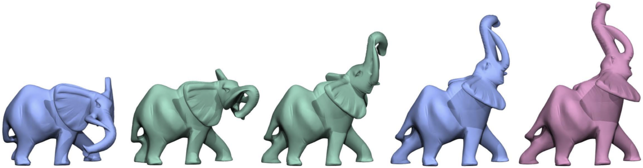


Deep Learning [Ben09, Fig. 4]

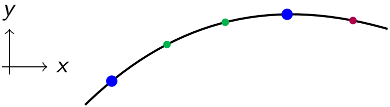


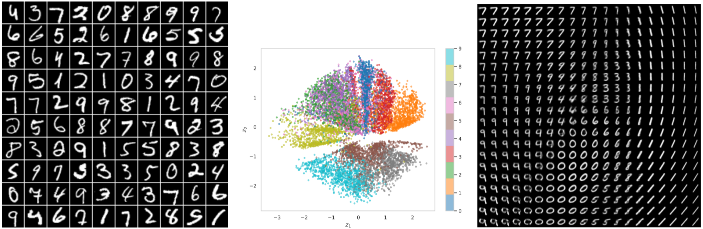


Manifold learning / Isomap [TdSL00]



Shape Space [KMP07, Fig. 1]

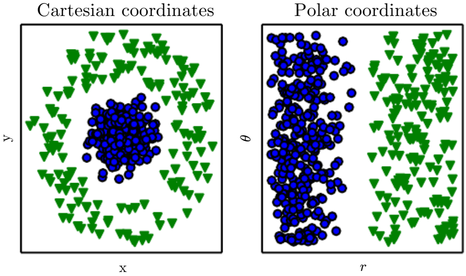




Variational Autoencoders / Information Bottlenecks [BM18]
Input → compressed task-sufficient representation → output

Representation Learning

- Performance of Machine Learning algorithms depends heavily on how data is represented.
- For many tasks, its difficult to know what features should be extracted, plus there may be many.
- Representation learning is about using ML to learn not only the map from representation to output, but also the representation.

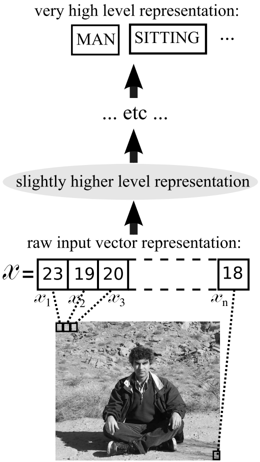


Representation learning / Deep learning [GBC16, Fig.]

Deep Learning

- Many of the factors of variation can be identified only using very sophisticated understanding of the data.
- Deep learning seeks to automatically discover such abstractions, from the lowest level features to the highest level concepts.
- Deep models introduce representations that are expressed in terms of other simpler representations.

[GBC16, p 5]



very high level representation:

MAN SITTING ...

... etc ...

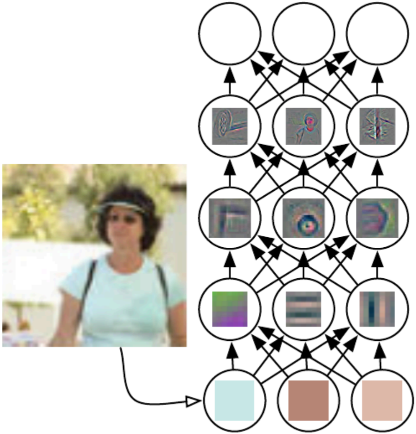
slightly higher level representation

raw input vector representation:

x = 23 19 20 18

x_1 x_2 x_3 x_n

[Ben09]



[ZF13]

Learning by minimizing a loss function

- Learning problems are often formulated as

$$\min_{\theta} L(\theta)$$

where L is an (empirical) loss, and θ parametrizes our hypotheses.

- Keep in mind

$$L_{\text{training data}} \qquad vs \qquad L_{\text{population}}$$

Examples:

$L(\theta) = -\sum_i p_\theta(x^{(i)})$ (likelihood) $\quad p_\theta(x) = \theta^x(1-\theta)^{x-1}$ (Bernoulli),

$L(\theta) = \sum_i (f_\theta(x^{(i)}) - y^{(i)})^2$ (MSE) $\quad f_\theta(x) = \sigma(\theta^\top x)$ (perceptron)

Function space and learning

Statistical learning

- The complexity of learning is typically described in terms of the geometry of the hypothesis space / function space

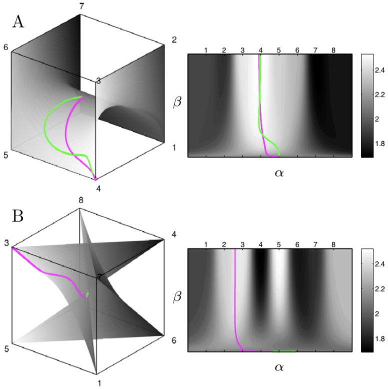
  (e.g., DOFs, VC dimension, complexity measures)
- Complexity measures for approximation and estimation depend on the input space via the geometry of hypothesis space.
- It is desirable to take the geometry of data space more directly into consideration

  (e.g., TDA, Topological DL, Geometric DL, Adversarial training)

  → Talk to Nina Otter!

Function space and optimization

If the optimization domain is convoluted, the optimization problem will typically be convoluted as well.



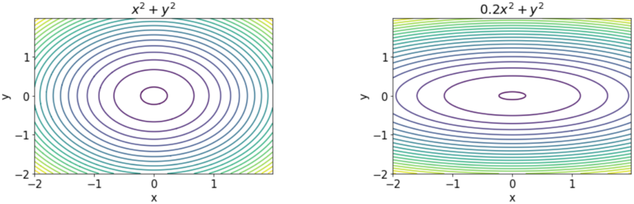
Selection criteria for neuromanifolds of stochastic dynamics [AMR13]

Geometry of function space

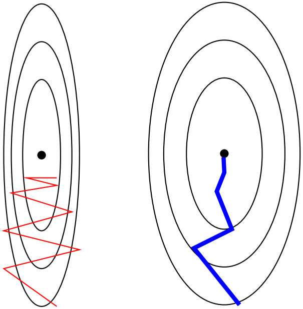
- How should we measure the distance between two hypotheses?
- Often we consider parametric models and work over the parameter space.
- How should this reflect in optimization / estimation?

Loss function $\leftrightarrow$ model $\leftrightarrow$ parametrization

Parametrization and optimization

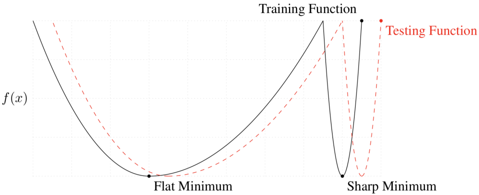


Optimization landscape / gradient optimization can be affected by the parametrization.



Momentum, Accelerated (Nesterov) moment, RMSprop, Adam, Feature normalization, Batch normalization

Optimization landscape and generalization

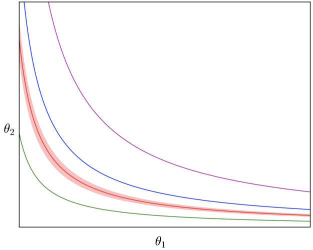


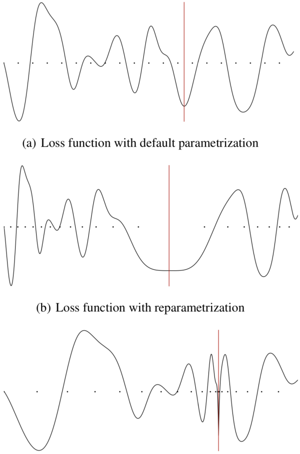
Large-batch methods tend to converge to sharp minimizers of the training function ... and tend to generalize less well. Generalization and sharp minima [KMN+16]

See Chaudhari, Soatto, Osher

→ Discuss with Hui Yin!

Optimization landscape and generalization





(a) Loss function with default parametrization

(b) Loss function with reparametrization

(c) Loss function with another reparametrization

Spectral radius and trace of the Hessian can be manipulated without actually changing the behavior of the function. Sharp minima can generalize [DPBB17]

(Non) Identifiability

Deep networks are usually not identifiable, meaning that several parameters represent the same function.

$$f(x;\theta) = \sigma(W_l\sigma(W_{l-1}\sigma(\cdots\sigma(W_1x)))$$

Weight space symmetry

- The latent variables are interchangeable, such that $f(\cdot;\theta) = f(\cdot;\theta_\pi)$, where θ_π is θ with permuted indices.
- Additionally, we might have scaling symmetries of the form $W_2\sigma(W_1x) = \frac{1}{c}W_2\sigma(cW_1x)$. For ReLUs this creates hyperbolas of equivalent local minima.

Geometry of parameter space / Parametrization invariance

- Often we are not interested in the parameter $\theta \in \Theta$ but rather in the hypothesis $p_\theta \in \mathcal{P}(\mathcal{X})$.
- We seek for p_θ by minimizing a loss function of the form

  $$L(\theta) = L(p_\theta),$$

  meaning that it depends on θ only through the corresponding distribution p_θ.
- Parametrization invariance can be useful. Define the geometry on Θ based on the geometry that is defined on $\mathcal{P}(\mathcal{X})$.
- Taking the steepest descent with respect to function space (instead of an arbitrary parameter space) can help against vanishing / exploding gradients.
- Question: how to define the geometry on $\mathcal{P}(\mathcal{X})$?

Video loss function on parameter space / function space

Geometry of parameter space

- Information Geometry uses the Fisher metric. But the Fisher is oblivious to the geometry of data space.
- Want: Keep perspective of using geometry of function space, but incorporate the geometry of data space.

Geometry of data and function space

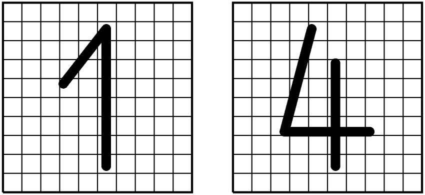
- How should the geometry of data space enter into the geometry of function space?
- We may be able to choose function spaces which are less complex, depending on the data geometry
- Symmetry / invariance / continuity with respect to certain variability in the input

Using the geometry of the data

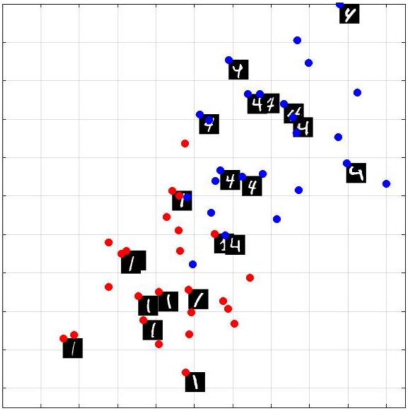
- Representation learning (unsupervised feature learning, deep learning, autoencoders, graphical models)
- Topological data analysis (dimensionality reduction, metric independence, persistent homology) (handcrafted features)
- Hand crafted features
- Geometric Deep Learning (data defined on non-Euclidean domains, convolutions on graphs / manifolds) (semi hand crafted features) IPAM New DL Techniques 2018, M Bronstein Tutorial

The curse of dimensionality

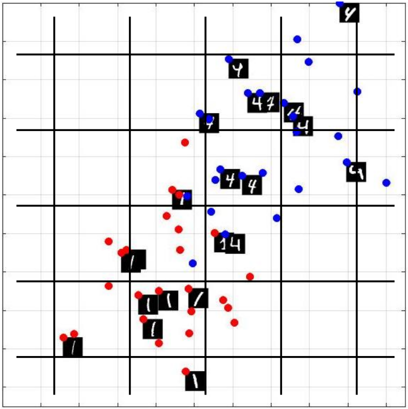
- Consider a simple problem of distinguishing handwritten versions of the digits ‘1’ and ‘4’.



- Each image is a point in a high dimensional space.



- Divide each coordinate into intervals. For a new point x, return the average y for training points in the same box.



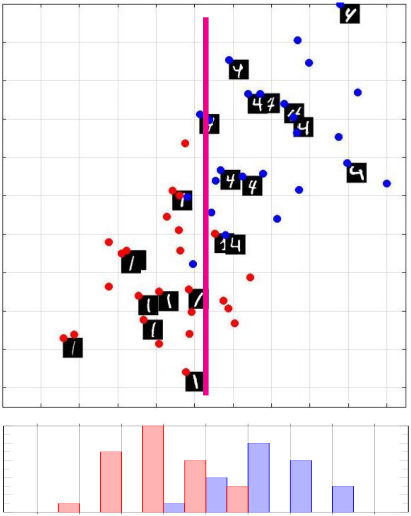
- By increasing the number K of intervals for each variable, we can obtain a more precise specification. Leads to K^d cells, which is exponential in the dimensionality of the input space.
- Specifying the mapping requires an exponential number of examples! This phenomenon is called curse of dimensionality.
- If we only have/can process a limited amount of data, increasing the dimensionality of the space rapidly leads to very sparse data, and the above gives a very poor representation.

- Approaches based on neural networks can be much less susceptible to the curse of dimensionality. These techniques are able to exploit two important properties of real data:

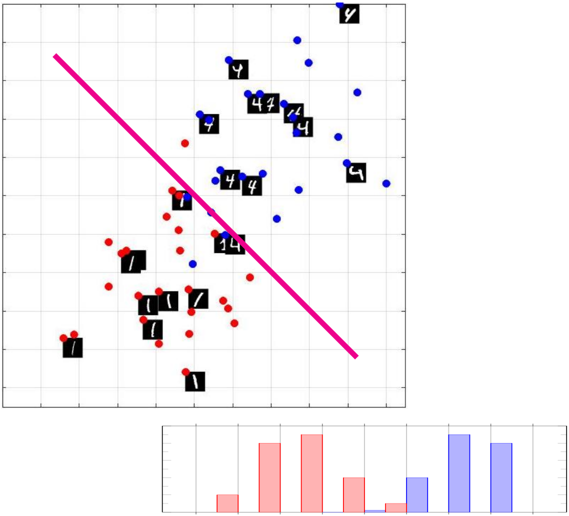
1. The input variables are generally correlated in some way, so that the data points do not fill out the entire input space, but tend to be restricted to a sub-space of lower dimension.
2. For most mappings of practical interest, the output varies smoothly with the input. Thus it is possible to infer the output values at intermediate points where no data is available, by a process similar to interpolation.

[Bis]

- We could consider lower dimensional features.



- We could consider lower dimensional features.



1 Introduction

2 Information Geometry

3 Wasserstein Information Geometry

What is Information Geometry?

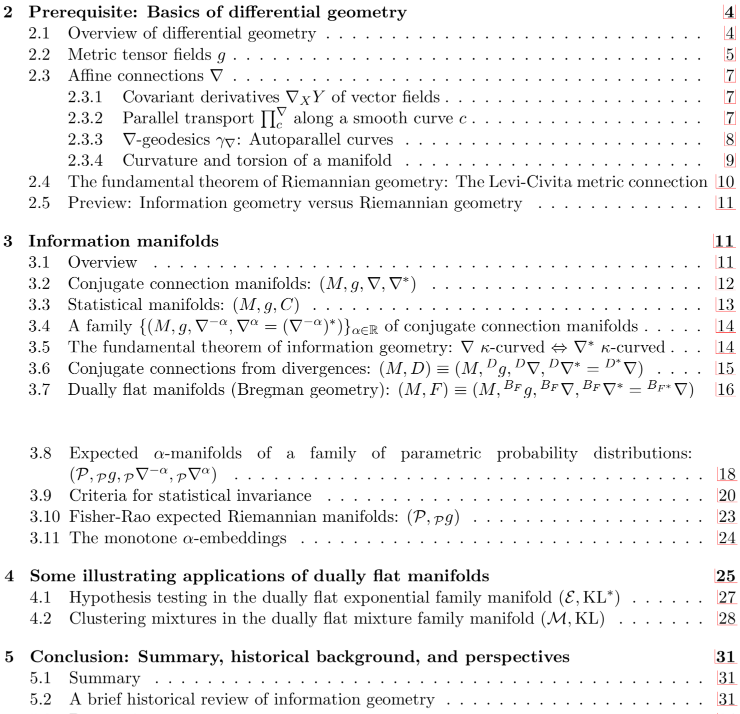
- Information geometry is a branch of mathematics that applies the techniques of differential geometry to the field of probability theory.
- This is done by taking probability distributions as the points of a Riemannian manifold.
- The Fisher information metric provides the Riemannian metric.

Conferences

- IGAIA, GSI, TGSI, ...

Resources

- Methods of Information Geometry, Amari and Nagaoka
- Information Geometry and Its Applications, Amari
- Information Geometry, Ay, Jost, Le, Schwachhöfer
- Information Geometry Springer Journal (since 2018)
- An elementary introduction to information geometry, Nielsen
- ...



2 Prerequisite: Basics of differential geometry 4
2.1 Overview of differential geometry 4
2.2 Metric tensor fields g 5
2.3 Affine connections ∇ 7
2.3.1 Covariant derivatives $\nabla_X Y$ of vector fields 7
2.3.2 Parallel transport $\prod_c^{\nabla}$ along a smooth curve c 7
2.3.3 ∇-geodesics γ_∇: Autoparallel curves 8
2.3.4 Curvature and torsion of a manifold 9
2.4 The fundamental theorem of Riemannian geometry: The Levi-Civita metric connection 10
2.5 Preview: Information geometry versus Riemannian geometry 11

3 Information manifolds 11
3.1 Overview 11
3.2 Conjugate connection manifolds: (M, g, ∇, ∇^*) 12
3.3 Statistical manifolds: (M, g, C) 13
3.4 A family $\{(M, g, \nabla^{-\alpha}, \nabla^{\alpha} = (\nabla^{-\alpha})^*)\}_{\alpha \in \mathbb{R}}$ of conjugate connection manifolds 14
3.5 The fundamental theorem of information geometry: ∇ κ-curved $\Leftrightarrow$ ∇^* κ-curved . . . 14
3.6 Conjugate connections from divergences: $(M, D) \equiv (M, {}^D g, {}^D\nabla, {}^D\nabla^* = {}^{D^*}\nabla)$ 15
3.7 Dually flat manifolds (Bregman geometry): $(M, F) \equiv (M, {}^{B_F} g, {}^{B_F}\nabla, {}^{B_F}\nabla^* = {}^{B_{F^*}}\nabla)$ 16

3.8 Expected α-manifolds of a family of parametric probability distributions: $(\mathcal{P}, {}_{\mathcal{P}}g, {}_{\mathcal{P}}\nabla^{-\alpha}, {}_{\mathcal{P}}\nabla^{\alpha})$ 18
3.9 Criteria for statistical invariance 20
3.10 Fisher-Rao expected Riemannian manifolds: $(\mathcal{P}, {}_{\mathcal{P}}g)$ 23
3.11 The monotone α-embeddings 24

4 Some illustrating applications of dually flat manifolds 25
4.1 Hypothesis testing in the dually flat exponential family manifold $(\mathcal{E}, \mathrm{KL}^*)$ 27
4.2 Clustering mixtures in the dually flat mixture family manifold $(\mathcal{M}, \mathrm{KL})$ 28

5 Conclusion: Summary, historical background, and perspectives 31
5.1 Summary 31
5.2 A brief historical review of information geometry 31

- Fisher metric
- Natural gradient

Fisher metric

We consider the space of probability distributions as a Riemannian manifold. What should be the correct metric?

- Consider a probability model $\{p(\theta) \colon \theta \in \Theta\}$. We assume that the parametrization $\theta \mapsto p(\theta)$ is smooth and locally injective.
- At each point $\theta \in \Theta$ we have a matrix given by

$$G(\theta) = \mathbb{E}_{p(\theta)}\left[\nabla \log p(\theta) \nabla \log p(\theta)^\top\right].$$

If $\Theta \subseteq \mathbb{R}^d$, this is $d \times d$ real symmetric matrix.

Example: Discrete simplex

- Consider as our model the set of all probability distributions on a finite set $I = \{1, \dots, n\}$.
- Each probability distribution is a vector $p = (p(1), \dots, p(n))$ with $p(i) \geq 0$ and $\sum_{i=1}^{n} p(i) = 1$.
- Standard $n-1$ simplex in $\mathbb{R}^n$. We can parametrize it as

$$p(i) = \theta_i, \quad \text{for } i = 1, \dots, n-1, \quad p(n) = 1 - \sum_{i=1}^{n-1} \theta_i.$$

- Then the Fisher matrix is given by

$$G_{ij}(\theta) = \sum_k p(k) \frac{\delta_i(k)}{p(k)} \frac{\delta_j(k)}{p(k)} = \frac{\delta_{i,j}}{p(i)}.$$

Interpretation of Fisher metric

- Information an observation carries about a parameter
- Curvature of the log likelihood
- Optimal variance of an estimator
- Invariant Riemannian metric

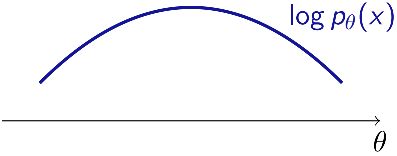
Fisher Information

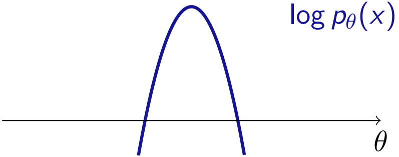
Information that an observable random variable X carries about unknown parameter θ of a distribution that models X.

$$G(\theta) = \mathbb{E}_{p_\theta}[\nabla \log p_\theta(x) \nabla \log p_\theta(x)^\top]$$

- $p_\theta(x)$ is the *likelihood* function of θ given observation x.
- $\nabla \log p_\theta(x)$ is the *score*, which measures how sensitively the model depends on θ at the current θ.

- If $x \sim p_\theta(x)$, $\mathbb{E}_{p_\theta}[\nabla \log p_\theta] = 0$, and G is the variance of the score, positive semi-definite.
- Negative expectation of the Hessian of the log likelihood. Curvature of the log likelihood. Low value at shallow maximum. High value at sharp maximum.


$\log p_\theta(x)$
θ


$\log p_\theta(x)$
θ

Invariant Riemannian metric

The Fisher metric is uniquely characterized (up to scaling) by being invariant under all sufficient statistics.

Chentsov, Campbell, Lebanon, Ay, Jost, ...

- Finite set $[n] := \{1, \dots, n\}$. Probability simplex

$$\Delta_{n-1} := \{(p_i)_i \in \mathbb{R}^n \colon p_i \geq 0, \sum_{i\in[n]} p_i = 1\}$$

- Tangent space $T_p\Delta^\circ_{n-1} \subset T_p\mathbb{R}^n_+ = \langle \partial_1, \dots, \partial_n \rangle$

$$u = \sum_{i\in[n]} u_i \partial_i \quad \text{with} \quad \sum_{i\in[n]} u_i = 0$$

- A metric g_p is an inner product on $T_p\Delta^\circ_{n-1}$ at each p
- The *Fisher metric* on Δ°_{n-1} is given by

$$g^{(n)}_p(u, v) = \sum_{i\in[n]} \frac{u_i v_i}{p_i}, \quad \text{for all } u, v \in T_p\Delta^\circ_{n-1}$$

- Consider two Riemannian manifolds $(\mathcal{E}, g)$, $(\mathcal{E}', g')$, and a smooth embedding $f \colon \mathcal{E} \to \mathcal{E}'$.
- The push-forward through f is

$$f_* \colon \quad T_p\mathcal{E} \to T_{f(p)}\mathcal{E}'; \quad \sum_i u_i \partial_i \mapsto \sum_j \sum_i u_i \frac{\partial f_j(p)}{\partial_i} \partial'_j.$$

- The pull-back of g' through f is

$$(f^* g')_p(u, v) := g'_{f(p)}(f_* u, f_* v), \quad \text{for all } u, v \in T_p\mathcal{E}.$$

- The embedding f is an *isometry* (g is *invariant* under f) iff

$$g_p(u, v) = (f^* g')_p(u, v), \quad \text{for all } p \in \mathcal{E} \text{ and } u, v \in T_p\mathcal{E}.$$

Embeddings by Markov maps

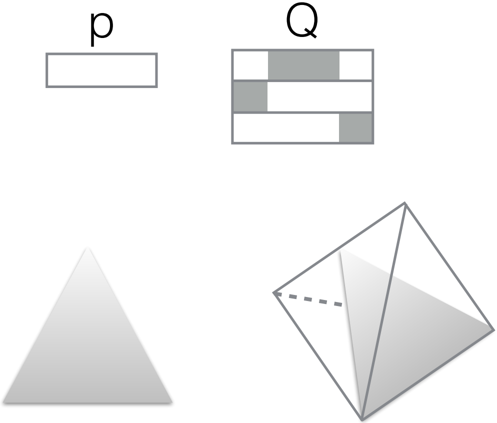
- A *Markov map* is a map of the form

$$f\colon \quad \mathbb{R}^m_+ \to \mathbb{R}^n_+; \quad p \mapsto p \cdot Q,$$

  where $Q \in \mathbb{R}^{m\times n}_{\geq 0}$ is a *row-partition matrix*, meaning that there is a partition $\dot\cup_{i=1}^m A_i = [n]$ with $\sum_{j\in A_{i'}} Q_{ij} = \delta_{ii'}$.

- This defines an embedding $f\colon \Delta^\circ_{m-1} \to \Delta^\circ_{n-1}$.

Embeddings by Markov maps



The result of Chentsov

Characterization of Fisher metric on the probability simplex via invariance under a class of natural statistical embeddings

Theorem 1 (Chentsov '72)

- *Let $g^{(m)}$ be a Riemannian metric on Δ°_{m-1} for $m \in \{2, 3, \dots\}$, with every embedding by a Markov map an isometry. Then there is a constant $C > 0$ such that*

$$g_p^{(m)}(u, v) = C \sum_i \frac{u_i v_i}{p_i}. \qquad (1)$$

- *Conversely, for any $C > 0$, (1) defines Riemannian metrics for which every embedding by a Markov map is an isometry.*

Geometry and Estimation I

It turns out that the curvature of the model also affects the estimation problem.

- Let $M = \{p(x, \xi)\}$ be a statistical model specified by parameter ξ.
- We observe N independent data points $D = \{x_1, \dots, x_N\}$ generated from $p(x, \xi)$ and want to know ξ.
- This is a problem of estimation. An estimator is a function

$$\hat{\xi} = f(x_1, \dots, x_N).$$

- The estimation error is $e = \hat{\xi} - \xi$. The bias of the estimator is

$$b(\xi) = \mathbb{E}[\hat{\xi}] - \xi.$$

The estimator is unbiased when $b(\xi) = 0$ and it is asymptotically unbiased when $\lim_{N \to \infty} b(\xi) = 0$.

Geometry and Estimation II

- It is expected that a good estimator is *consistent*, meaning that it converges to the true parameter as N tends to infinity,

$$\lim_{N\to\infty} \hat{\xi} = \xi.$$

- The accuracy is measured by the error covariance matrix

$$V = \mathbb{E}\left[(\hat{\xi} - \xi)(\hat{\xi} - \xi)^\top\right].$$

- The Cramér-Rao theorem gives a bound on the accuracy:

Theorem 2 (Cramér '46, Rao '45)

For an asymptotically unbiased estimator $\hat{\xi}$, it holds that

$$V \geq \frac{1}{N} G^{-1},$$

where G is the Fisher information matrix.

Example MLE

- The Maximum Likelihood Estimator (MLE) is the maximizer of the likelihood,

$$\hat{\xi}_{MLE} = \mathrm{argmax}_{\xi} \prod_{i=1}^{N} p(x^{(i)}, \xi).$$

- The MLE is asymptotically unbiased and its error covariance satisfies

$$V_{MLE} = \frac{1}{N} G^{-1} + O\left(\frac{1}{N^2}\right),$$

  attaining the Cramér-Rao bound asymptotically. It is said to be *Fisher efficient* (first-order efficient).

Natural gradient

What is the natural gradient?

- A type of gradient descent method
- Generally applicable to optimization over probability models
- Defined as the gradient times the inverse of the Fisher matrix of the model

Steepest descent I

The natural gradient is motivated as a way to obtain the steepest descent direction in the set of distributions.

- Let $\Theta = \{\theta \in \mathbb{R}^n\}$ be the parameter space, on which a function $L(\theta)$ is defined.
- When Euclidean, the square length of an increment $d\theta$ of θ is

$$|d\theta|^2 = \sum_i (d\theta_i)^2.$$

- In general, when the coordinate system is nonorthonormal,

$$|d\theta|^2 = \sum_{i,j} g_{ij}(\theta) d\theta_i d\theta_j.$$

The matrix is called Riemannian metric tensor, and it depends on θ. In the Euclidean case it reduces to $g_{ij}(\theta) = \delta_{i,j}$.

Steepest descent II

- The steepest descent direction of $L(\theta)$ at θ is defined as the $d\theta$ that minimizes $L(\theta + d\theta)$ with fixed $|d\theta|$, i.e.,

$$\begin{aligned} \min \quad & L(\theta + d\theta) \\ \text{s.t.} \quad & |d\theta|^2 = \epsilon^2. \end{aligned}$$

Theorem 3

The steepest descent direction of $L(\theta)$ in a Riemannian space is

$$-\tilde{\nabla} L(\theta) = -G(\theta)^{-1} \nabla L(\theta),$$

where $G = (g_{ij})$ is the Riemannian metric and

$$\nabla L(\theta) = \left(\frac{\partial L}{\partial \theta_1}, \dots, \frac{\partial L}{\partial \theta_n} \right)^\top$$

is the ordinary gradient. Note $\tilde{\nabla}$ is just the contravariant form of ∇.

Proof.

Put $d\theta = \epsilon a$ and minimize

$$\begin{aligned} \min \quad & L(\theta + d\theta) = L(\theta) + \epsilon \nabla L(\theta)^\top a, \\ \text{s.t.} \quad & |a|^2 = \sum_{ij} g_{ij}(\theta) a_i a_j = 1. \end{aligned}$$

Lagrange

$$\begin{aligned} & \frac{\partial}{\partial a_i} \left\{ \nabla L(\theta)^\top a - \lambda a^\top G a \right\} = 0 \\ & \Rightarrow \nabla L(\theta) = 2\lambda G a \\ & \Rightarrow a = \frac{1}{2\lambda} G^{-1}(\theta) \nabla L(\theta), \end{aligned}$$

where λ is determined by the constraint. □

Fewer iterations

- In many applications, the natural gradient seems to require far fewer iterations than the ordinary gradient.
- This makes it a potentially attractive alternative to the regular gradient method.

Parametrization invariance

- So, the natural gradient is the steepest descent on a Riemannian manifold.
- An important aspect is that the Riemannian metric comes from the space of distributions.
- Under these assumptions, the flow (of distributions) defined by the natural gradient is invariant with respect to smooth invertible reparametrizations of the distributions.
- Note, the matrix G still depends on the specific parametrization that we choose.

Example

Of course, the metric can look different depending on our parametrization.

- Consider an exponential family

$$p_\theta(x) = \exp(\sum_j \theta_j f_j(x) - \psi(\theta))$$

  The functions f_j are “observables” or sufficient “statistics” which define the model and its specific parametrization.

- Any choice of $f_1, \ldots, f_n$ with the same span produces the same set of distributions.
- The Fisher metric is given by

$$G(\theta) = \mathbb{E}_{p_\theta}\left[\nabla \log(p_\theta) \nabla \log(p_\theta)^\top\right] = \mathrm{cov}_{p_\theta}(f)$$

$$\nabla \log(p_\theta(x)) = \nabla(\sum_j \theta_j f_j(x) - \psi(\theta)) = f(x) - \mathbb{E}_{p_\theta}[f]$$

Example I

Natural gradient can help against the vanishing / exploding gradient problem.

- Consider as a loss function the negative log likelihood

  $$L(\theta) = -\sum_i \log(p_\theta(x^{(i)})),$$

  For simplicity consider only one example $x^{(1)}$ and write x.
- For the ordinary gradient we have

  $$\begin{aligned} -\nabla L(\theta) &= \nabla \log(p_\theta(x)) \\ &= (f(x) - \mathbb{E}_{p_\theta}[f]) \end{aligned}$$

Example II

- For the natural gradient we have

$$
\begin{aligned}
-\tilde{\nabla} L(\theta) &= G(\theta)^{-1} \nabla \log(p_\theta(x)) \\
&= G(\theta)^{-1} (f(x) - \mathbb{E}_{p_\theta}[f]) \\
G(\theta) &= (f - f p_\theta^\top 1) \operatorname{diag}(p) (f - f p_\theta^\top 1)^\top
\end{aligned}
$$

Challenges

- For large models (with many parameters), computing the natural gradient is impractical due to the large size of the Fisher information matrix.
- This is addressed through various approximations to make it easier to compute, store, invert than the exact Fisher. (e.g Le Roux et al., 2008; Ollivier, 2015; Grosse and Salakhudinov, 2015; Martens and Grosse, 2015)
- We will consider proximal methods and affine restrictions on dual variables.

Loss functions

- Often we formulate a learning problem in terms of a probability distribution.
- What should be the loss function to be used here?
- In particular, should the discrepancy between probability distributions have something to do with the geometry of the data on which they are defined?

1. Introduction

2. Information Geometry

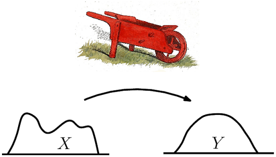
3. Wasserstein Information Geometry

Motivation

We can use the Wasserstein distance as a vehicle to define

- Geometry of data space (e.g., distance between images)
- Loss functions for parameter estimation in parametrized models, which capture the geometry of the data space (WGANs, WWGANs)
- Riemannian structure on function space and natural gradients which incorporate the geometry of the data space

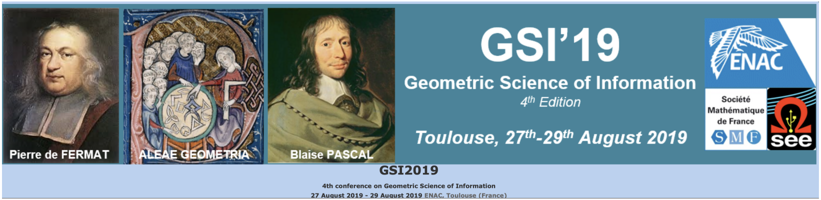
Optimal transport



- Mapping formulation: Monge problem (1781): Monge-Ampére equation
- Statical formulation: Kantorovich problem (1940): Linear programming
- Dynamical formulation: Density optimal control (Nelson, Lafferty, Gangbo, Otto, Villani, Chow, Zhou, Osher)
- In recent times in relation to information geometry: Amari, Karakida, Malago, Pistone

Announcement

Special session on Wasserstein Information Geometry at GSI 2019


Pierre de FERMAT
ALEAE GEOMETRIA
Blaise PASCAL
GSI'19
Geometric Science of Information
4th Edition
Toulouse, 27th-29th August 2019
ENAC
Société Mathématique de France
SMF
see

GSI2019

4th conference on Geometric Science of Information

27 August 2019 - 29 August 2019 ENAC, Toulouse (France)

Wasserstein distance

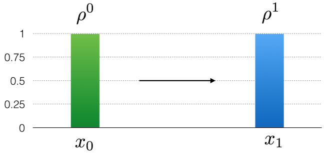
- Consider a metric space $(\mathcal{X}, d_{\mathcal{X}})$ and the set $\mathcal{P}_p(\mathcal{X})$ of densities with finite p-th moment.
- The Wasserstein-p distance of a pair $\mathbb{P}_0$, $\mathbb{P}_1 \in \mathcal{P}_p(\mathcal{X})$ is

$$W_{p,d_{\mathcal{X}}}(\mathbb{P}_0, \mathbb{P}_1) = \inf_{\Pi} \left\{ \left(\mathbb{E}_{(X,Y)\sim\Pi} d_{\mathcal{X}}(X, Y)^p \right)^{\frac{1}{p}} \right\},$$

  where Π is a joint distribution of (X, Y) with marginals $X \sim \mathbb{P}_0$, $Y \sim \mathbb{P}_1$.
- Note W_p depends on the *ground metric* $d_{\mathcal{X}} \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

Why optimal transport?

Optimal transport provides a particular distance among histograms which relies on the ground metric d on sample space.



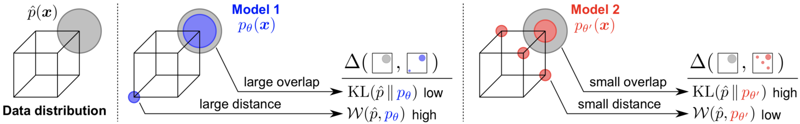
If $X_0 \sim \rho^0 = \delta_{x_0}$, $X_1 \sim \rho^1 = \delta_{x_1}$,

$$W(\rho^0, \rho^1) = \inf_{\pi \in \Pi(\rho^0, \rho^1)} \mathbb{E}_{(X_0, X_1) \sim \pi} c(X_0, X_1) = c(x_0, x_1)$$

$$\mathrm{TV}(\rho^0, \rho^1) = \int_\Omega |\rho^0(x) - \rho^1(x)| dx = 2$$

$$\mathrm{KL}(\rho^0 \| \rho^1) = \int_\Omega \rho^0(x) \log \frac{\rho^0(x)}{\rho^1(x)} dx = \infty.$$

Wasserstein Loss

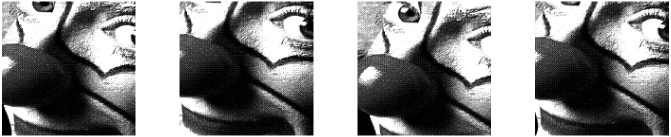

$\hat{p}(\boldsymbol{x})$
Data distribution
Model 1
$p_\theta(\boldsymbol{x})$
large overlap
large distance
$\Delta(\square, \square)$
$\mathrm{KL}(\hat{p} \,\|\, p_\theta)$ low
$\mathcal{W}(\hat{p}, p_\theta)$ high
Model 2
$p_{\theta'}(\boldsymbol{x})$
small overlap
small distance
$\Delta(\square, \square)$
$\mathrm{KL}(\hat{p} \,\|\, p_{\theta'})$ high
$\mathcal{W}(\hat{p}, p_{\theta'})$ low

Wasserstein training of RBMs [MMC16, Fig. 1]

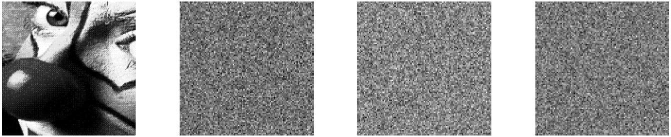
Wasserstein Loss

Samples in the training set

Samples from p^1 (nearby points in sample space)



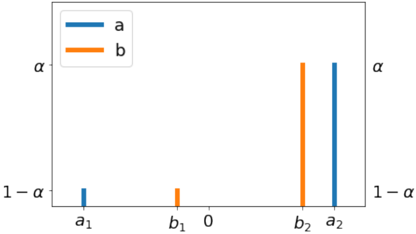
Samples from p^2 (arbitrary locations in sample space)



Wasserstein Loss

- Wasserstein is more “continuous” than KL.

Distance between two distributions



  - $W_2(a,b)^2 = (1-\alpha)(a_1-b_1)^2 + \alpha(a_2-b_2)^2$
  - $KL(a\|b) = +\infty$ (because no overlap between distributions)
  - Euclidean$(a,b) = (a_1-b_1)^2 + (a_2-b_2)^2$
  - $L_2(a,b)^2 = +\infty$ (because integrating over all of $\mathbb{R}$)
- So when $a^{(k)} \to b$, then we have convergence under the W_2 and Euclidean metric, but not others.
- But the Euclidean metric overemphasizes the distance between a_1 and b_1, which should be weighted less.

Wasserstein metric

Video Dynamical OT

Density manifold

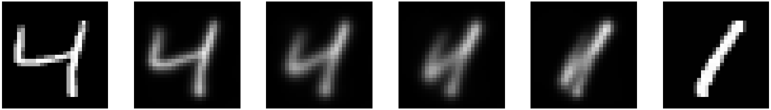
Optimal transport has an optimal control reformulation that gives rise to a Riemannian metric:

$$\inf_{\rho_t} \int_0^1 g_W(\partial_t \rho_t, \partial_t \rho_t) dt = \int_0^1 \int_\Omega (\nabla \Phi_t, \nabla \Phi_t) \rho_t dx dt,$$

under the dynamical constraint, i.e., the continuity equation

$$\partial_t \rho_t + \nabla \cdot (\rho_t \nabla \Phi_t) = 0, \quad \rho_0 = \rho^0, \quad \rho_1 = \rho^1.$$

Here, $(\mathcal{P}(\Omega), g_W)$ forms an infinite-dimensional Riemannian manifold[1].

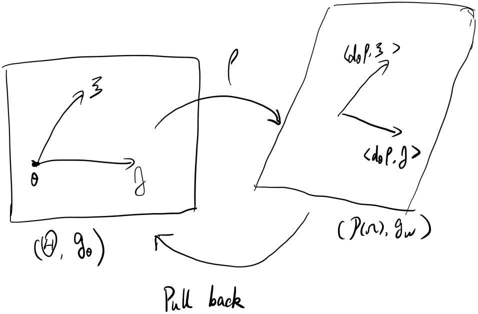


[1] Lafferty 1988, Otto 2001.

Density submanifold

If we have a parametrized model ρ_θ, $\theta \in \Theta$, we define the metric g_θ on Θ as the pull-back of the metric g_W on $\mathcal{P}(\Omega)$,

$$g_\theta(\xi, \eta) = g_W(d\rho_\theta(\xi), d\rho_\theta(\eta)), \qquad \text{for } \xi,\ \eta \in T_\theta\Theta.$$



ξ
θ
η
(Θ, g_θ)
ρ
$\langle d_\theta \rho, \xi \rangle$
$\langle d_\theta \rho, \eta \rangle$
$(\mathcal{P}(\Omega), g_W)$
Pull back

We call (Θ, g_θ) a Wasserstein statistical manifold.

Wasserstein matrix

Write $g_\theta(\xi, \eta) = \xi^T G_W(\theta)\eta$, where $G_W(\theta) \in \mathbb{R}^{d\times d}$.
The Wasserstein matrix is given by

$$G_W(\theta)_{ij} = \Big(\nabla_{\theta_i}\rho(\cdot, \theta), (-\Delta_\rho)^{-1}\nabla_{\theta_j}\rho(\cdot, \theta)\Big)_{L_2},$$

where Δ_{ρ_θ} is the weighted elliptic operator.

Consider the probability space $(\mathcal{P}(\Omega), g)$ with metric tensor g, and a smoothly parametrized probability model ρ_θ with parameter $\theta \in \Theta$. Then the pull-back G of g is given by

$$G(\theta) = \Big(\nabla_\theta \rho_\theta, g(\rho_\theta)\nabla_\theta \rho_\theta\Big).$$

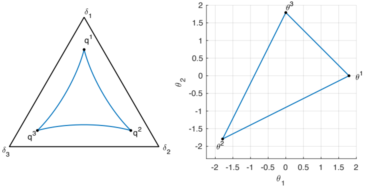
(i) If $g_\theta = -(\Delta_{\rho_\theta})^{-1}$, with Δ_{ρ_θ} being the weighted elliptic operator, then $G(\theta)$ is the Wasserstein metric tensor, given by

$$G_W(\theta)_{ij} = \Big(\nabla_{\theta_i} \rho_\theta, (-\Delta_{\rho_\theta})^{-1}\nabla_{\theta_j} \rho_\theta\Big),$$

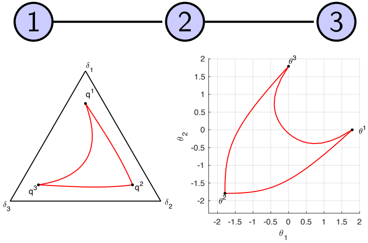
(ii) If $g_\theta = \frac{1}{\rho_\theta}$, then $G(\theta)$ is the Fisher-Rao metric tensor, given by

$$G_{FR}(\theta)_{ij} = \Big(\nabla_{\theta_i} \rho_\theta, \frac{1}{\rho_\theta}\nabla_{\theta_j} \rho_\theta\Big).$$

Geodesics

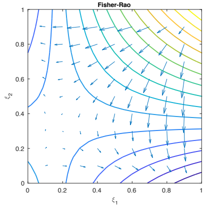


Fisher

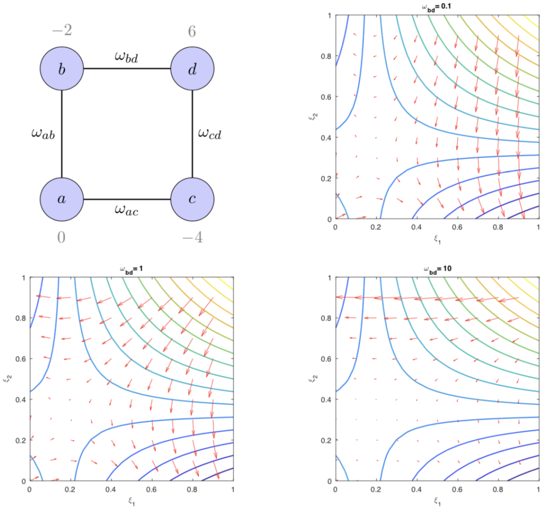


Wasserstein

Gradient Flows



Fisher

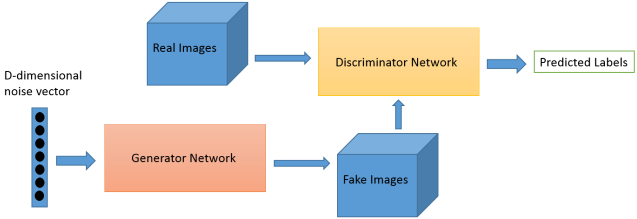


Wasserstein

GANs

Generative Adversarial Networks (GANs)

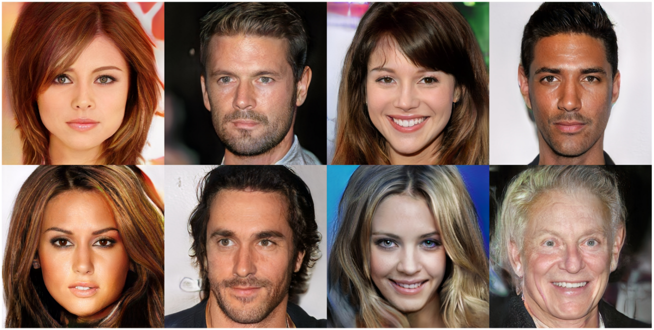
- Generative Adversarial Networks (GANs) are a way to mimic a probability distribution. Given training data, they can construct samples that look like the training data.



[slide A. Lin]

What can GANs do?

GANs can create new celebrity faces:



[KALL18]

[slide A. Lin]

What can GANs do?

GANs can do image superresolution:

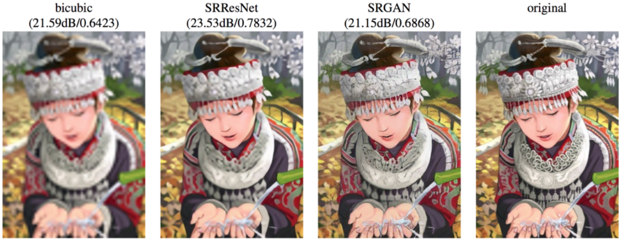
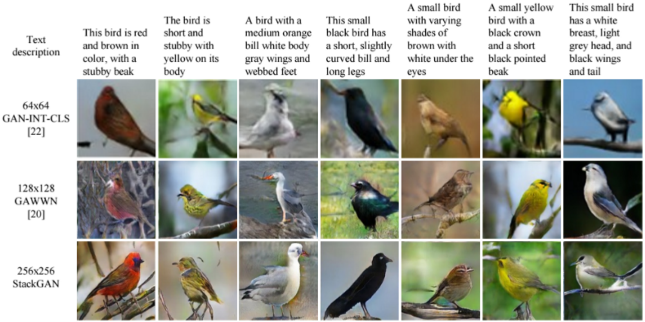


Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

[LTH$^+$16]

What can GANs do?

GANs can do text-to-image synthesis:



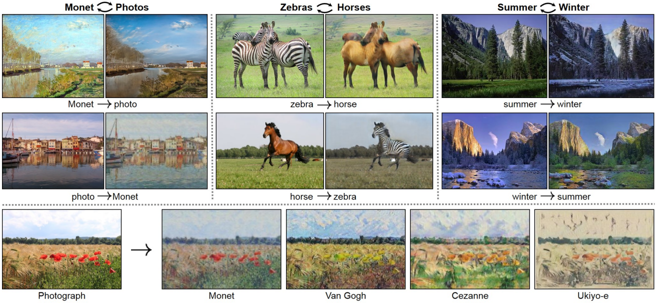
Text description
This bird is red and brown in color, with a stubby beak
The bird is short and stubby with yellow on its body
A bird with a medium orange bill white body gray wings and webbed feet
This small black bird has a short, slightly curved bill and long legs
A small bird with varying shades of brown with white under the eyes
A small yellow bird with a black crown and a short black pointed beak
This small bird has a white breast, light grey head, and black wings and tail
64x64 GAN-INT-CLS [22]
128x128 GAWWN [20]
256x256 StackGAN

Figure 3. Example results by our proposed StackGAN, GAWWN [20], and GAN-INT-CLS [22] conditioned on text descriptions from CUB test set. GAWWN and GAN-INT-CLS generate 16 images for each text description, respectively. We select the best one for each of them to compare with our StackGAN.

[ZXL$^+$16]

What can GANs do?

GANs can do image-to-image translation:



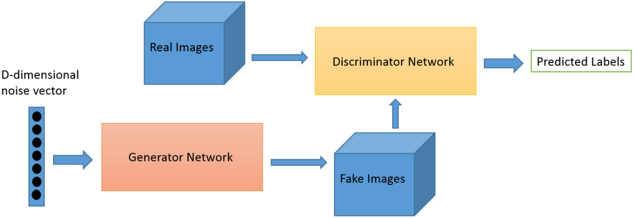
[ZPIE17]

What are GANs, mathematically?

- GANs consist of two networks: D – the discriminator, and G – the generator.
- The (initially proposed) performance function for GANs is:

$$\max_{\omega} \min_{G_\theta} \; \mathbb{E}_{x\sim \text{real}}[\log D_\omega(x)] + \mathbb{E}_{z\sim N}[\log 1 - D_\omega(G_\theta(z))]$$

where N is the normal distribution.



[Sky18]

Different performance functions for GANs

Many other performance functions are available:

- Standard GAN [GPM+14]:

$$\max_D \min_G \ \mathbb{E}_{x\sim \text{real}}[\log D(x)] + \mathbb{E}_{z\sim N}[\log 1 - D(G(z))]$$

- (Standard) WGAN [ACB17]:

$$\max_D \min_G \mathbb{E}_{x\sim \text{real}}[D(x)] - \mathbb{E}_{z\sim N}[D(G(z))] \text{ (and clip the weights of } D\text{)}$$

- WGAN-GP [GAA+17]:

$$\max_D \min_G \mathbb{E}_{x\sim \text{real}}[D(x)] - \mathbb{E}_{z\sim N}[D(G(z))] + \lambda \cdot \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\| - 1)^2]$$

- DRAGAN [KAHK17]:

$$\max_D \min_G \ \mathbb{E}_{x\sim \text{real}}[\log D(x)] + \mathbb{E}_{z\sim N}[\log 1 - D(G(z))] \\ + \lambda \cdot \mathbb{E}_{x\sim \text{real}, \delta\sim N_d(0,cI)}[\|\nabla_x D(x+\delta)\| - k]^2$$

Wasserstein of Wasserstein Loss for Learning Generative Models

Wasserstein Loss

Given a probability model $\{\mathbb{P}_G \colon G \in \Theta\} \subseteq \mathcal{P}_p(\mathcal{X})$ and a data distribution $\mathbb{P}_r \in \mathcal{P}_p(\mathcal{X})$, we find a hypothesis by minimizing

$$\inf_G W_{p,d_{\mathcal{X}}}(\mathbb{P}_G, \mathbb{P}_r).$$

This depends on a choice of the ground metric $d_{\mathcal{X}}$ on the sample space $\mathcal{X}$.

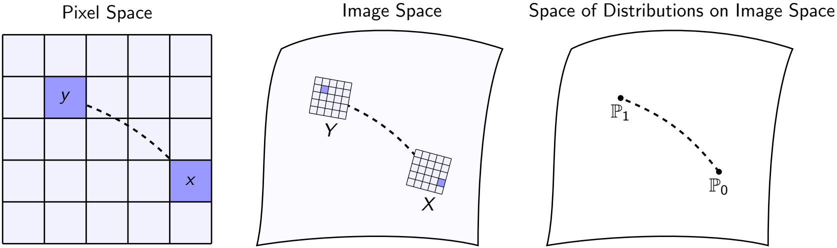
Wasserstein ground metric

The Wasserstein distance is known to be effective for images. Motivated by this, we introduce a Wasserstein ground metric

$$d_{\mathcal{X}}(X, Y) := W_{q,d_\Omega}(X, Y) = \inf_{\pi} \left\{ \left(\mathbb{E}_{(x,y)\sim\pi} d_\Omega(x, y)^q \right)^{\frac{1}{q}} \right\}.$$

An image $X \in \mathcal{X}$ is viewed as a histogram over pixels $x \in \Omega$.

The pixel ground metric $d_\Omega \colon \Omega \times \Omega \to \mathbb{R}_+$ assigns distances to pairs of pixels.

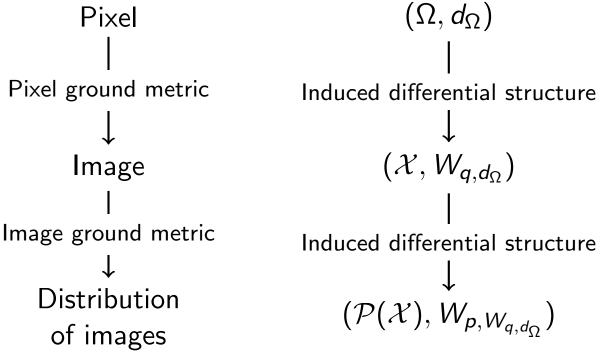


Pixel Space

Image Space

Space of Distributions on Image Space

Wasserstein of Wasserstein loss [DLLM19, Fig. 3]



Pixel

Pixel ground metric

Image

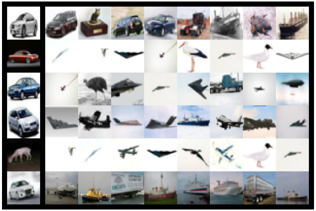
Image ground metric

Distribution of images

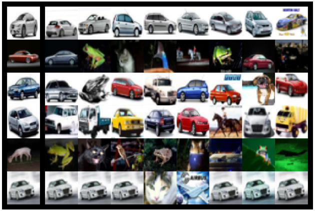
$$\begin{tikzcd}
(\Omega, d_\Omega) \arrow[d, "\text{Induced differential structure}"] \\
(\mathcal{X}, W_{q,d_\Omega}) \arrow[d, "\text{Induced differential structure}"] \\
(\mathcal{P}(\mathcal{X}), W_{p,W_{q,d_\Omega}})
\end{tikzcd}$$

Wasserstein ground metric

L^2 (Euclidean) ground metric



Wasserstein-2 ground metric



Source and nearest images from the CIFAR-10 dataset.

Duality and computation

The linear programming computation is unfeasible.
We use a Kantorovich duality formulation with Lipschitz-1 condition.

Theorem 4 (Duality of Wasserstein of Wasserstein)

The Wasserstein-1 loss function over Wasserstein-2 ground metric has the following equivalent formulation:

$$W_{1,W_{2,d_\Omega}}(\mathbb{P}_G, \mathbb{P}_r) = \sup_{f \in C(\mathcal{X})} \Big\{ \mathbb{E}_{X\sim\mathbb{P}_G} f(X) - \mathbb{E}_{X\sim\mathbb{P}_r} f(X) \colon \\ \int_\Omega \|\nabla_x \delta_X f(X)(x)\|^2_{d_\Omega} X(x) dx \leq 1 \Big\},$$

where ∇_x is the gradient operator in pixel space Ω and δ_X is the L^2 gradient in image space $\mathcal{X}$.

Pixel discretization

Proposition 5 (Wasserstein gradient on discrete image space)

Given a pixel space graph $\mathcal{G}$, the gradient of $f \in C^1(\mathcal{X})$ is

$$\operatorname{grad} f(X) = L(X)\nabla_X f(X),$$

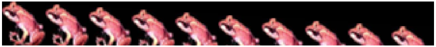
where ∇_X is the Euclidean gradient operator, and $L(X) \in \mathbb{R}^{n\times n}$ is the weighted Laplacian matrix defined as

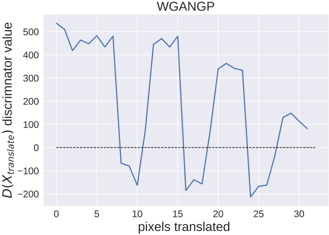
$$L(X)_{ij} = \begin{cases} \frac{1}{2}\sum_{k\in N(i)} \omega_{ik}(\frac{X_i}{d_i} + \frac{X_k}{d_k}) & \textit{if } i = j; \\ -\frac{1}{2}\omega_{ij}(\frac{X_i}{d_i} + \frac{X_j}{d_j}) & \textit{if } j \in N(i); \\ 0 & \textit{otherwise}. \end{cases}$$

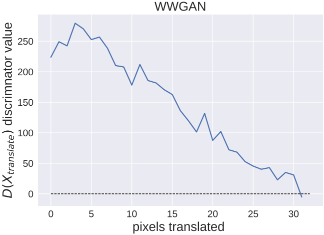
Moreover, the 1-Lipschitz condition on $(\mathcal{X}, W)$, $\|\operatorname{grad} f(X)\|_W \leq 1$, is equivalent to

$$\nabla_X f(X)^\top L(X)\nabla_X f(X) \leq 1.$$

Stability to natural data variability

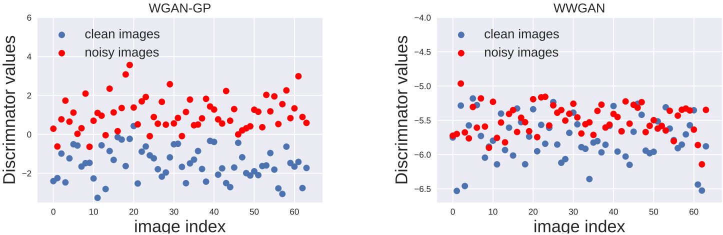






Discriminator for CIFAR-10 images translated continuously.
Both discriminators were trained to reach an FID value of 40.

Stability to noise



Discriminator values on CIFAR-10 images with RGB salt and pepper noise 15% of the pixels.

- WW allows us to incorporate a meaningful geometry in sample space and train generative models that are more in line with the natural variability of the data.

Wasserstein Natural Gradient

Natural gradient

- We can use Wasserstein geometry not only design distances in sample space and useful loss functions for learning, but also to develop optimization methods.

Consider the natural gradient

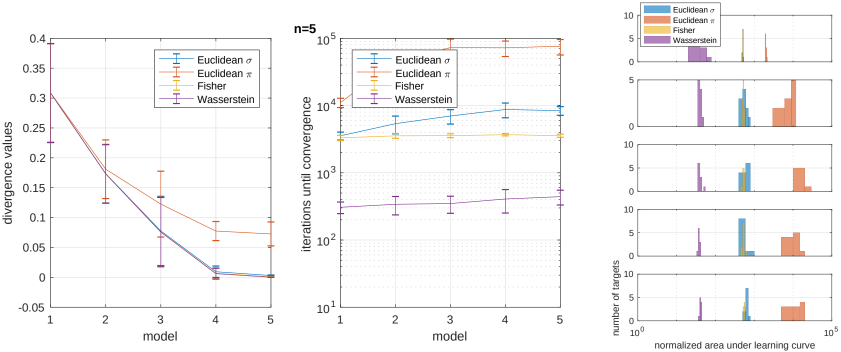
$$\theta' = \theta + \alpha \tilde{\nabla} F(\theta),$$

where

$$\tilde{\nabla} F(\theta) = G^{-1}(\theta) \nabla F(\theta).$$

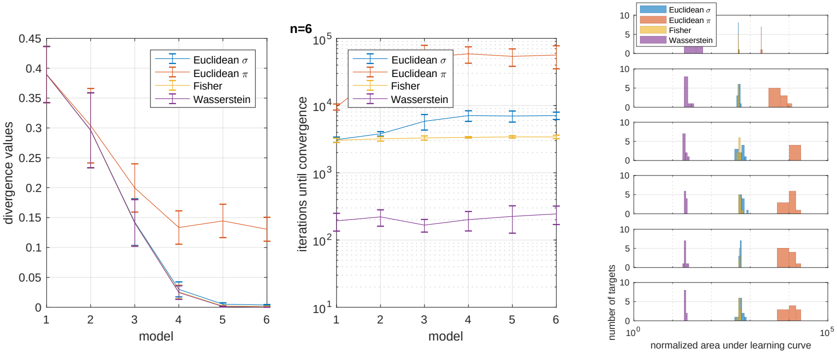
Here G is the matrix that defines the inner product on the tangent space of the probability model. While this is usually taken to be the Fisher information matrix, we can use the Wasserstein matrix.

MLE



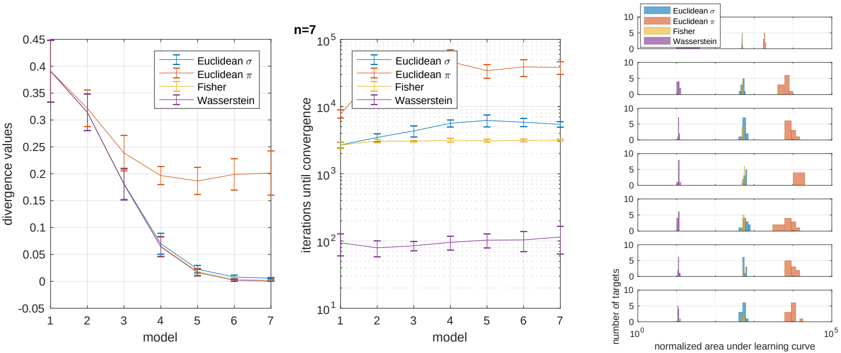
Euclidean, Fisher, Wasserstein gradients on MLE for discrete hierarchical models

MLE



Euclidean, Fisher, Wasserstein gradients on MLE for discrete hierarchical models

MLE



Euclidean, Fisher, Wasserstein gradients on MLE for discrete hierarchical models

Wasserstein proximal of GANs

Wasserstein Natural Gradient for GANs

- In GANs, we can utilize the reparameterization trick and some more Taylor expansions, to arrive at the following update scheme:

$$\theta^{k+1} = \operatorname{argmin}_\theta L(\theta) + \frac{1}{2h}\mathbb{E}_{z\sim N}\|g_\theta(z) - g_{\theta^k}(z)\|_2^2$$

  which we call the Relaxed Wasserstein Proximal (RWP).

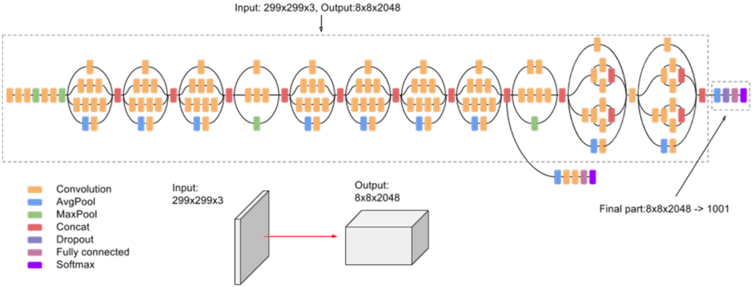
- This resembles a proximal operator:

$$\operatorname{prox}_\lambda(f)(x) = \operatorname{argmin}_y \left\{ f(y) + \frac{1}{2\lambda}\|x-y\|_2^2 \right\}$$

- We use this to gain better speed and stability in GANs.

Fréchet Inception Distance

- To measure convergence, we use a quantitative measure: the Fréchet Inception Distance
- It compares statistics at the last pooling layer of Inception v3:



[Goo]

The RWP Algorithm

- The RWP algorithm is meant as a drop-in regularizer. You train your GAN in the usual way with a slight modification:
- The algorithm is as follows:
  - Sample real data $\{x_i\}_{i=1}^B$ and noise data $\{z_i\}_{i=1}^B$
  - Do

$$\omega^{k+1} \leftarrow \text{Optimizer}_\omega \Big\{ \text{Loss}(D_{\omega^k}(\{x_i\}_{i=1}^B),\ D_\omega(G_\theta(\{z_i\}_{i=1}^B))) \Big\}$$

  - Sample noise data $\{z_i\}_{i=1}^B$
  - Do

$$\theta^{k+1} \leftarrow \text{Optimizer}_\theta \Big\{ Loss(G_\theta(\{z_i\}_{i=1}^N)) + \frac{1}{B} \sum_{i=1}^{B} \| G_\theta(z_i) - G_{\theta^k}(z_i) \|^2 \Big\}$$

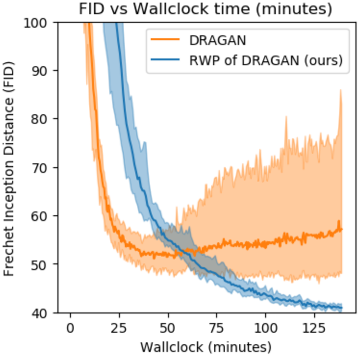
  - Repeat until convergence.

RWP of DRAGAN on CIFAR-10

- The performance function of DRAGAN is,

$$\max_D \min_G \ \mathbb{E}_{x\sim\text{real}}[\log D(x)] + \mathbb{E}_{z\sim N}[\log 1 - D(G(z))]$$
$$+ \lambda \cdot \mathbb{E}_{x\sim\text{real},\delta\sim N_d(0,cI)}[\|\nabla_x D(x+\delta)\| - k]^2$$

- Results with and without RWP regularization:

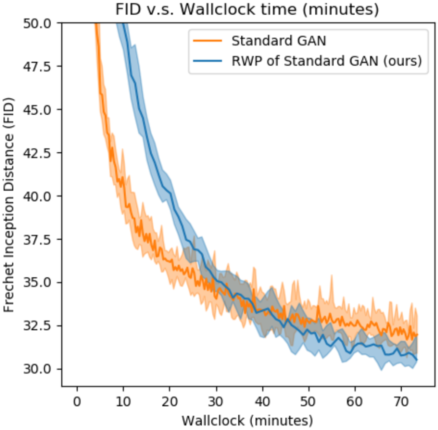


RWP of Standard GANs on CIFAR-10

- The performance function of Standard GANs is,

$$\max_D \min_G \ \mathbb{E}_{x\sim \text{real}}[\log D(x)] + \mathbb{E}_{z\sim N}[\log 1 - D(G(z))]$$

- Results with and without RWP regularization:

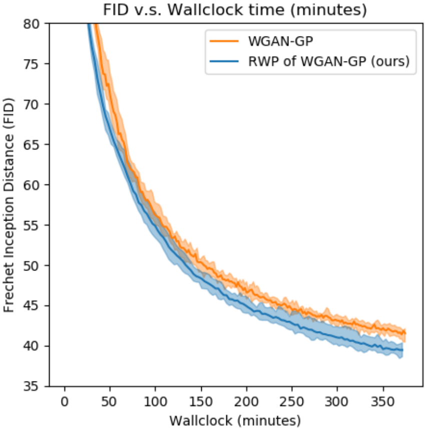


RWP of WGANGP on CIFAR-10

- The performance function of WGANGP is,

$$\max_D \min_G \mathbb{E}_{x\sim \text{real}}[D(x)] - \mathbb{E}_{z\sim N}[D(G(z))] + \lambda \cdot \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\| - 1)^2]$$

- Results with and without RWP regularization:

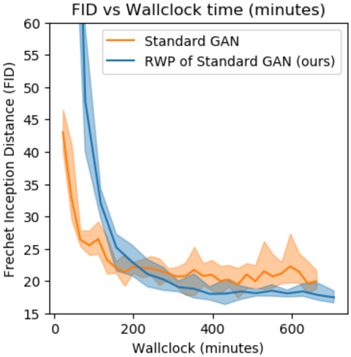


RWP of Standard GANs on CelebA

- The performance function of Standard GANs is,

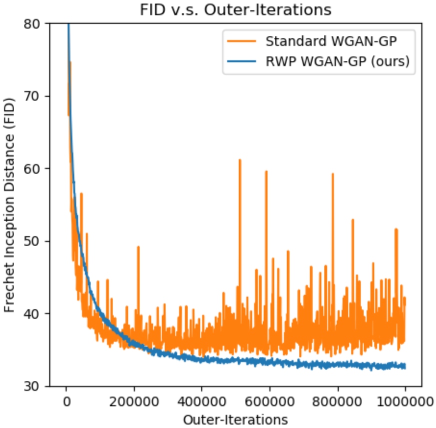
$$\max_{D} \min_{G} \ \mathbb{E}_{x\sim \mathrm{real}}[\log D(x)] + \mathbb{E}_{z\sim N}[\log 1 - D(G(z))]$$

- Results with and without RWP regularization:

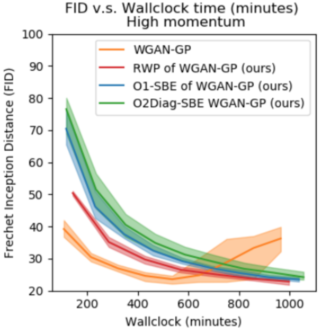


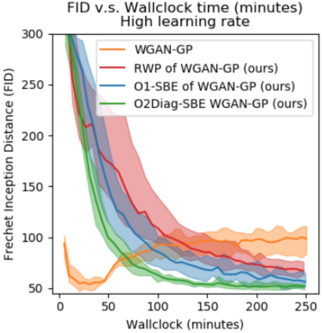
RWP of Standard GANs on CelebA for 1 million updates

- Results with and without RWP regularization:



Stability to Hyperparameters

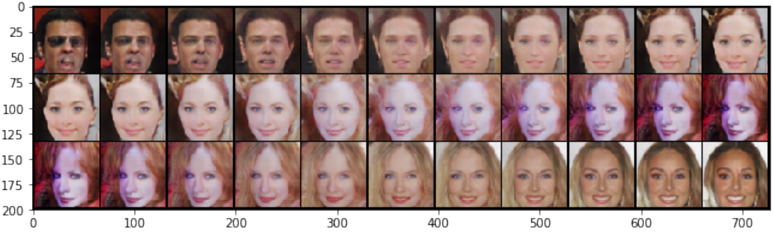




The Wasserstein proximal improves the training by providing a lower FID when the learning rate is high. The results are based on the CelebA dataset.

Latent Space Walk

- Latent space walk of Standard GANs with RWP on CelebA:



Affine Natural Proximal Gradients

Natural gradient

- The natural gradient

$$\theta^{k+1} = \theta^k - hG(\theta^k)^{-1}\nabla_\theta F(\theta^k)$$

  requires $G(\theta)$ and its inverse at each iteration.
- This is difficult in high dimensional parameter spaces.
- We develop an alternative approach based on the proximal method and approximations of the proximity term

Natural proximal operators

- The proximal operator refers to

$$\theta^{k+1} = \mathrm{Prox}_{hF}(\theta^k) = \arg\min_{\theta} \; F(\theta) + \frac{D(\theta, \theta^k)}{2h},$$

- D penalizes the distance from the current point. Choose Riemannian distance

$$D(\theta, \theta^k) = \inf_{\theta(t)} \left\{ \int_0^1 \dot{\theta}(t)^\top G(\theta(t))\dot{\theta}(t) dt \colon \theta_0 = \theta, \; \theta_1 = \theta^k \right\}$$

- h adjusts the strength. When h is infinity, the proximal operator returns the global minimizer of F.

Quadratic approximation

- Consider the local approximation of the Riemannian distance

$$\tilde{D}(\theta, \theta^k) = \Big(\rho_\theta - \rho_{\theta^k}, g(\rho_{\tilde{\theta}})(\rho_\theta - \rho_{\theta^k})\Big).$$

- Express $\tilde{D}$ in terms of its Legendre dual:

$$\frac{1}{2}\tilde{D}(\theta, \theta^k) = \sup_{\Phi\colon \Omega \to \mathbb{R}} (\Phi, \rho_\theta - \rho_{\theta^k}) - \frac{1}{2}\Big(\Phi, g(\rho_{\tilde{\theta}})^\dagger \Phi\Big).$$

(Maximizer $\Phi = g(\rho_{\tilde{\theta}})(\rho_\theta - \rho_{\theta^k})$ recovers above)

Affine space restriction

Now we restrict the dual variable to an affine space

$$\mathcal{F}_\Psi = \left\{ \Phi(x) = \sum_{j=1}^{n} \xi_j \psi_j(x) = \xi^\top \Psi(x) \colon \xi \in \mathbb{R}^n \right\},$$

where $\xi = (\xi_j)_{j=1}^n$ is a parameter vector and $\Psi = (\psi_j)_{j=1}^n$ collects a choice of basis functions $\psi_j \colon \Omega \to \mathbb{R}$.

Theorem 6 (Affine space approximation)

Given a basis Ψ, the proximity term $\tilde{D}$ within the affine function space $\mathcal{F}_\Psi = \{\xi^\top \Psi \colon \xi \in \mathbb{R}^n\}$ is given by

$$\tilde{D}_\Psi(\theta, \theta^k) = (\mathbb{E}_\theta[\Psi] - \mathbb{E}_{\theta^k}[\Psi])^\top \Big(\Psi, g(\rho_\theta)^\dagger \Psi\Big)^\dagger (\mathbb{E}_\theta[\Psi] - \mathbb{E}_{\theta^k}[\Psi]).$$

(i) *For the Wasserstein metric, we have*

$$\tilde{D}^W_\Psi(\theta, \theta^k) = (\mathbb{E}_\theta[\Psi] - \mathbb{E}_{\theta^k}[\Psi])^\top \Big(\mathfrak{C}^W(\tilde{\theta})\Big)^{-1} (\mathbb{E}_\theta[\Psi] - \mathbb{E}_{\theta^k}[\Psi]),$$

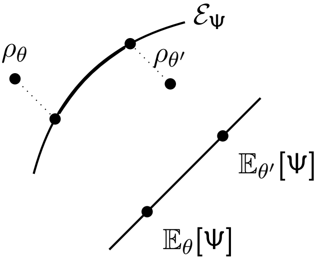
where $\mathfrak{C}^W(\tilde{\theta}) = \mathbb{E}_{\tilde{\theta}}[\sum_l \Big(\partial_l \Psi\Big)\Big(\partial_l \Psi\Big)^\top]$.

(ii) *For the Fisher-Rao metric, we have*

$$\tilde{D}^{FR}_\Psi(\theta, \theta^k) = (\mathbb{E}_\theta[\Psi] - \mathbb{E}_{\theta^k}[\Psi])^\top \Big(\mathfrak{C}^{FR}(\tilde{\theta})\Big)^{-1} (\mathbb{E}_\theta[\Psi] - \mathbb{E}_{\theta^k}[\Psi]),$$

where $\mathfrak{C}^{FR}(\tilde{\theta}) = \mathbb{E}_{\tilde{\theta}}[\Big(\Psi(x) - \mathbb{E}_{\tilde{\theta}}[\Psi]\Big)\Big(\Psi(x) - \mathbb{E}_{\tilde{\theta}}[\Psi]\Big)^\top]$.

Interpretation



- Intuitively, the metric between two distributions is measured along a chosen set of statistics.
- If Ψ is the sufficient statistics of an exponential family $\mathcal{E}_\Psi$, then we are measuring local distances of MLE projections onto $\mathcal{E}_\Psi$, whose dual parameters are $\mathbb{E}_\theta[\Psi]$ and $\mathbb{E}_{\theta'}[\Psi]$.

Example 7 (Order-1 approximation)

For the metric approximation with the (linear) space of linear functions, $\mathcal{F}_1 = \Big\{ \Phi(x) = a^\top x + b \colon a \in \mathbb{R}^m,\ b \in \mathbb{R} \Big\}$, we have:

(i)
$$\tilde{D}_1^W(\theta, \theta^k) = (\mathbb{E}_\theta[x] - \mathbb{E}_{\theta^k}[x])^\top (\mathbb{E}_\theta[x] - \mathbb{E}_{\theta^k}[x]).$$

(ii)
$$\tilde{D}_1^{FR}(\theta, \theta^k) = (\mathbb{E}_\theta[x] - \mathbb{E}_{\theta^k}[x])^\top \Big(\mathbb{E}_{\tilde{\theta}} \Big[(x - \mathbb{E}_{\tilde{\theta}} x)(x - \mathbb{E}_{\tilde{\theta}} x)^\top \Big] \Big)^{-1} (\mathbb{E}_\theta[x] - \mathbb{E}_{\theta^k}[x]).$$

Example 8 (Order-2 approximation)

For the space of quadratic functions,
$\mathcal{F}_2 = \left\{ \Phi(x) = \frac{1}{2} x^\top Q x + a^\top x + b \colon\, Q \in \mathbb{R}^{m \times m},\ a \in \mathbb{R}^m,\ b \in \mathbb{R} \right\}$,
we have:

(i)

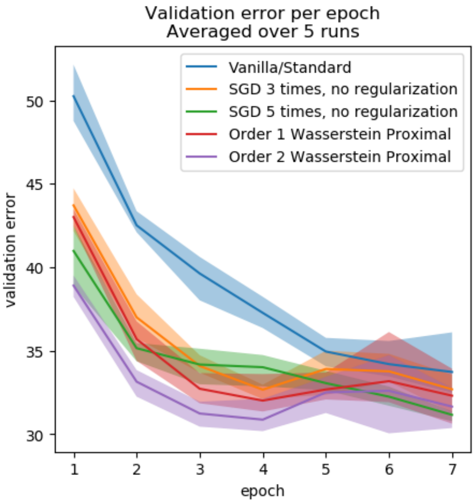
$$\tilde{D}_2^W(\theta, \theta^k) = \left(\mathbb{E}_\theta \left[\begin{smallmatrix} x \\ \frac{x \otimes x}{2} \end{smallmatrix} \right] - \mathbb{E}_{\theta^k} \left[\begin{smallmatrix} x \\ \frac{x \otimes x}{2} \end{smallmatrix} \right] \right)^\top \mathbb{E}_{\tilde{\theta}} \left[\begin{smallmatrix} I_m & x^\top \otimes I_m \\ x \otimes I_m & I_m \otimes x x^\top \end{smallmatrix} \right]^{-1} \left(\mathbb{E}_\theta \left[\begin{smallmatrix} x \\ \frac{x \otimes x}{2} \end{smallmatrix} \right] - \mathbb{E}_{\theta^k} \left[\begin{smallmatrix} x \\ \frac{x \otimes x}{2} \end{smallmatrix} \right] \right).$$

(ii)

$$\tilde{D}_2^{FR}(\theta, \theta^k) = \left(\mathbb{E}_\theta \left[\begin{smallmatrix} x \\ \frac{x \otimes x}{2} \end{smallmatrix} \right] - \mathbb{E}_{\theta^k} \left[\begin{smallmatrix} x \\ \frac{x \otimes x}{2} \end{smallmatrix} \right] \right)^\top \left(\mathfrak{C}^{FR}(\tilde{\theta}) \right)^{-1} \left(\mathbb{E}_\theta \left[\begin{smallmatrix} x \\ \frac{x \otimes x}{2} \end{smallmatrix} \right] - \mathbb{E}_{\theta^k} \left[\begin{smallmatrix} x \\ \frac{x \otimes x}{2} \end{smallmatrix} \right] \right),$$

where

$$\mathfrak{C}^{FR} = \mathbb{E}_{\tilde{\theta}} \left[\left(\left[\begin{smallmatrix} x \\ \frac{x \otimes x}{2} \end{smallmatrix} \right] - \mathbb{E}_{\tilde{\theta}} \left[\begin{smallmatrix} x \\ \frac{x \otimes x}{2} \end{smallmatrix} \right] \right) \left(\left[\begin{smallmatrix} x \\ \frac{x \otimes x}{2} \end{smallmatrix} \right] - \mathbb{E}_{\tilde{\theta}} \left[\begin{smallmatrix} x \\ \frac{x \otimes x}{2} \end{smallmatrix} \right] \right)^\top \right].$$



Test error on CIFAR-10 classification

References

- Natural gradient via optimal transport [LM18a]
- Wasserstein of Wasserstein for learning generative models [DLLM19]
- Wasserstein proximal of GANs [LLOM18]
- Affine natural proximal learning [LLM19]
- Ricci curvature for parametric statistics via optimal transport [LM18b]

References I

M. Arjovsky, S. Chintala, and L. Bottou.
Wasserstein GAN.
ArXiv e-prints, January 2017.

Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin.
Learning From Data.
AMLBook, 2012.

Nihat Ay, Guido Montúfar, and Johannes Rauh.
Selection criteria for neuromanifolds of stochastic dynamics.
In Yoko Yamaguchi, editor, *Advances in Cognitive Neurodynamics (III)*, pages 147–154. Springer-Verlag, 2013.

References II

Y. Bengio.
Learning Deep Architectures for AI. Foundations and Trends in Machine Learning, V2(1).
Now Publishers, 2009.

Pradeep Kumar Banerjee and Guido Montufar.
The variational deficiency bottleneck.
ArXiv, 2018.

Yonatan Dukler, Wuchen Li, Alex Tong Lin, and Guido Montufar.
Wasserstein of wasserstein loss for learning generative models.
Preprint, 2019.

References III

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio.
Sharp minima can generalize for deep nets.
In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1019–1028, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville.
Improved training of wasserstein gans.
CoRR, abs/1704.00028, 2017.

References IV

I Goodfellow, Y Bengio, and A Courville.
Deep Learning.
MIT Press, 2016.
http://www.deeplearningbook.org.

Google.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio.
Generative Adversarial Networks.
ArXiv e-prints, June 2014.

Naveen Kodali, Jacob D. Abernethy, James Hays, and Zsolt Kira.
How to train your DRAGAN.
CoRR, abs/1705.07215, 2017.

References V

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen.
Progressive growing of GANs for improved quality, stability, and variation.
In *International Conference on Learning Representations*, 2018.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang.
On large-batch training for deep learning: Generalization gap and sharp minima.
CoRR, abs/1609.04836, 2016.

Martin Kilian, Niloy J. Mitra, and Helmut Pottmann.
Geometric modeling in shape space.
ACM Transactions on Graphics (SIGGRAPH), 26(3):#64, 1–8, 2007.

References VI

Wuchen Li, Alex Tong Lin, and Guido Montufar.
Affine natural proximal learning.
2019.

Alex Tong Lin, Wuchen Li, Stanley Osher, and Guido Montufar.
Wasserstein proximal of gans.
2018.

Wuchen Li and Guido Montufar.
Natural gradient via optimal transport.
Information geometry, 1(2):181–214, 2018.

Wuchen Li and Guido Montufar.
Ricci curvature for parametric statistics via optimal transport.
2018.

References VII

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi.
Photo-realistic single image super-resolution using a generative adversarial network.
CoRR, abs/1609.04802, 2016.

Grégoire Montavon, Klaus-Robert Müller, and Marco Cuturi.
Wasserstein training of restricted boltzmann machines.
In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3718–3726. Curran Associates, Inc., 2016.

Frank Nielsen.
An elementary introduction to information geometry.
CoRR, abs/1808.08271, 2018.

References VIII

SkyMind.
A beginner's guide to generative adversarial networks (gans), 2018.

Joshua B. Tenenbaum, Vin de Silva, and John C. Langford.
A global geometric framework for nonlinear dimensionality reduction.
Science, 290(5500):2319, 2000.

M. D. Zeiler and R. Fergus.
Visualizing and understanding convolutional networks.
Technical Report arXiv:1311.2901 [cs.CV], NYU, 2013.

References IX

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros.
Unpaired image-to-image translation using cycle-consistent adversarial networks.
In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris N. Metaxas.
Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks.
CoRR, abs/1612.03242, 2016.