preprint number
June 2019

Notes on numerical CY metrics

Michael R. Douglas,
Subramanian Lakshminarasimhan and Yidi Qi

Department of Physics and Simons Center for Geometry and Physics, Stony Brook University

Abstract

We try ML inspired ansatzes for computing Ricci-flat Kähler metrics.

1 Introduction

Brief introduction to CY metrics and their uses in string theory and mathematics.

Briefly survey work on numerical metrics and HYM connections, including Headrick and Wiseman, Donaldson, our work and the Penn work, and the recent work using decision forests [**?**].

Related ML work, for example 2003.03828 by Chrysos et al.

The main line of work has been to represent the metric using an embedding by holomorphic sections of an ample line bundle, leading to

$$K = \log \sum_{I,\bar{J}} h_{I,\bar{J}} s^I \bar{s}^{\bar{j}} \tag{1}$$

where s^I is a basis of N holomorphic sections. This gives us an N^2 real dimensional family of metrics parameterized by the hermitian matrix $h_{I,\bar{J}}$. One then does integrals over M by Monte Carlo. This setup can be used in various ways, but conceptually the simplest is to choose a measure of how close the metric is to Ricci flatness and numerically optimize it, as done in [**?**]. This is easy to program (as we will review below), especially compared with methods that use explicit coordinate patches or a grid.

Let us focus on the case of M a hypersurface $f = 0$ in a weighted projective space W. The homogeneous coordinates will be Z^i with degrees n_i. Now a holomorphic section is simply a weighted homogeneous polynomial, so this ansatz is easy to work with mathematically. Physically one can define a holomorphic section as a ground state of a magnetic Schrödinger

operator, *i.e.* a state in the lowest Landau level. Thus, in the terminology of numerical methods, this is a spectral method. A spectral method expands continuous functions in terms of the low lying eigenfunctions of some operator, such as the translation operator (leading to a Fourier basis) or a Laplacian. The upper cutoff on eigenvalue corresponds to a fixed lower cutoff on the length scale, and thus spectral methods are usually not well adapted to problems with multiple scales.

In our problem, let us denote the degree of the polynomial (equivalently of the line bundle) by k. An expansion in polynomials of degree k can represent functions with up to k nodes, so with variations on length scales roughly down to $1/k$. However, by adjusting moduli it is easy to produce CY metrics with multiple scales – for example, we might have a small resolved cycle in a large bulk manifold. Thus we would like a more flexible ansatz than Eq. (1) but one which is still easy to program.

One option would be to use a higher degree line bundle and a subset of its sections, keeping more sections in regions with more variation. Mathematically, this amounts to restricting the rank of the matrix $h_{I\bar{J}}$. This could be done by imposing the form $h = U^\dagger \lambda U$ with λ a diagonal matrix of restricted rank. Let us denote the space of metrics of the form Eq. (1) with a degree k embedding and in which $\operatorname{rank} h \leq D$ as $\mathcal{G}_{k,D}$. It has real dimension ND (check!).

1.1 Feed-forward networks

In the years since [**?**], there has been tremendous progress in machine learning. This has had many applications in physics [**?**], including many works which use feed-forward networks (also called multi-layer perceptrons, or simply neural networks) to represent high dimensional functions.

Let us briefly review the definition of a feed-forward network. It is a parameterized space of functions

$$F_w : \mathcal{X} \to \mathcal{Y}, \tag{2}$$

with an input $x \in \mathcal{X} \cong \mathbb{R}^D$ and an output $y \in \mathcal{Y} \cong \mathbb{R}^{D'}$ (we will generally take $D' = 1$). It is the composition of a series of functions or layers indexed $0, 1, \ldots, d$. The number of layers $d + 1$ is the depth.

The parameters or weights w are a series of matrices $\{w_{(0)}, w_{(1)}, \ldots, w_{(d)}\}$ which enter

the following expression:

$$y_j(w;x) = \sum_{i_d=1}^{D_d} w^{(d)}_{j,i_d} \theta(z^{i_d}_{(d-1)}) \tag{3}$$

$$\vdots$$

$$z^{i_2}_{(1)} = \sum_{i_1=1}^{D_1} (w_{(1)})^{i_2}_{i_1} \theta(z^{i_1}_{(0)}) \tag{4}$$

$$z^{i_1}_{(0)} = \sum_{i=1}^{D} (w_{(0)})^{i_1}_{i} x^i \tag{5}$$

$$\tag{6}$$

The "activation function" $\theta(x)$ is a nonlinear function, for which a popular choice is the "ReLU" function

$$\theta_{ReLU}(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} . \tag{7}$$

The term "unit" is sometimes used to denote a sum followed by a single application of θ, so this network will have $D_1 + \ldots + D_d$ units.

It has been shown that feed-forward networks can approximate arbitrary functions. This is the case even for depth two ($d = 1$) [**?**], but in this case one can need an exponentially large number of units, as would be the case for simpler methods of interpolation (the "curse of dimension"). By using more layers, one can gain many advantages – complicated functions can be represented with many fewer units [], and local optimization techniques are much more effective []. How exactly this works is not well understood theoretically and there are many interesting observations and hypotheses as to how these advantages arise.

A basic application of this for physics and statistics is to use a feed-forward network with a single output to represent a family P_w of probability distributions on $\mathcal{X}$ parameterized by the weights w, by taking the output y to be the probability density function.[1] The expectation value of a function $\mathcal{O}(x)$ in the distribution P_w is then

$$\mathbb{E}_w[\mathcal{O}] \equiv \frac{\int d\mu(x)\ y(w;x)\, \mathcal{O}(x)}{\int d\mu(x)\ y(w;x)}. \tag{8}$$

Here $d\mu(x)$ is a measure on $\mathcal{X}$, which mathematically is required to turn the function $y(w;x)$ on $\mathcal{X}$ into a density on $\mathcal{X}$. Given coordinates x^i on $\mathcal{X}$, this could be Lebesgue measure, but we are free to make other choices.

[1] A different and also very common choice for representing a probability distribution over a finite set S is to use a network with a separate output y_a for each option $a \in S$, and take the final layer to be the "softmax" $P(a) = \exp y_a / \sum_{b \in S} \exp y_b$ to get a normalized probability distribution $P(a)$. This has its own advantages but is not as relevant for our present purposes.

In a high dimensional configuration space $\mathcal{X}$, one often does these integrals by sampling, choosing a set of N_p points $x_{(i)} \in \mathcal{X}$ and taking

$$d\mu(x) = \frac{1}{N_p} \sum_{i=1}^{N_p} \delta(x - x_{(i)}). \tag{9}$$

The sample points $x_{(i)}$ could be distributed according to an *a priori* probability measure Dx on $\mathcal{X}$, or we could adapt the sampling to prefer regions with large measure (importance-based sampling). Many options have been developed in statistics.

An analogous idea could be used to get a parameterized family of quantum wave functions, in this case taking all of (x, w, y) to be complex, and taking

$$\mathbb{E}_w[\mathcal{O}] \equiv \frac{\int d\mu(x)\ |y(w;x)|^2 \mathcal{O}(x)}{\int d\mu(x)\ |y(w;x)|^2}. \tag{10}$$

One would also have to choose an appropriate $\theta(x)$. Some works using these wave functions include [].

1.2 Multilayer holomorphic embeddings

The idea we will pursue in this work is to use Eq. (3) to define a subspace of sections. Thus we take the input space $\mathcal{X}$ to be the ambient weighted projective space, and we choose $\theta(x)$ to be a nonlinear homogeneous holomorphic function. The simplest choice and the one we will use is to take

$$\theta(x) = x^2. \tag{11}$$

We then take the successive layers Eq. (5), Eq. (4), *etc.* to define subspaces of a space of sections. To get a real valued Kähler potential, we replace Eq. (3) with

$$K(w;Z) = \log \sum_{i_d=1,\bar{j}_d=1}^{D_d} h^{(d)}_{i_d,\bar{j}_d} \theta(z^{i_d}_{(d-1)}) \theta(\bar{z}^{\bar{j}_d}_{(d-1)}) \tag{12}$$

where $z_{(d-1)}$ and the previous layers are as above.

This construction gives us a class of metrics for each choice of depth and layer widths $D_1, \ldots, D_d$, corresponding to a restricted class of embeddings with degree $k = 2^d$.

[count the number of parameters!]

1.3 Generalizations to other manifolds

For hypersurfaces in weighted projective space, we need to modify the construction. The simplest option is to take the inputs to be a basis of all sections of the same degree as the defining function f. Another option is not to have a single feed-forward structure but rather a DAG of layers connected in all ways consistent with homogeneity. One might also

generalize the activation function, allowing two layers with outputs z and z' to send the product zz' to the next layer.

Next let us consider toric varieties. In physics terms, a toric variety is the moduli space of a $U(1)^r$ gauge theory with four supercharges (say, a $(2,2)$ sigma model). There are many equivalent mathematical definitions; the closest to the physics is that it is the symplectic (or GIT) quotient of $\mathbb{C}^N$ by a $U(1)^r$ action. Thus a toric manifold M is specified by an $N \times r$ integral matrix Σ of $U(1)^r$ charges and a moment map, a real-valued function on the Lie algebra of $U(1)^r$. If it is a smooth manifold, it will have $b^2 = r$, and the fields (the coordinates on $\mathbb{C}^N$) are sections of line bundles with $c_1(\mathcal{L})$ given by the corresponding row of Σ (say this in a more mathematical way). We can then multiply coordinates and take weighted sums in any way that preserves the grading, to get a nonlinearly defined subspace of the space of sections of some $\mathcal{L}$. All of these constructions should be unified in some representation theoretic way.

A basic example is to take all of the monomials in the defining polynomial, *i.e.* a basis of sections of $\mathcal{N}$, project to a subspace, and take the sum of the squares of the resulting functions. Thus we are comparing the representing power of $\mathrm{Sym}^2 H^0(\mathcal{L})$ with $\sigma(P \cdot H^0(\mathcal{L})$ for some projector P.

How can we compare the general ability to represent functions of a basis and a subset? We can define the associated kernels, which here are the Bergman kernels. Given a metric on $\mathcal{L}$ and a volume form, we can define the inner product on sections and the corresponding Bergman kernel

$$\rho(z, \bar{z}') = \sum_i s(z)\bar{s}(\bar{z}'). \tag{13}$$

We can then look at $\rho_1 - \rho_2$ and show that it has small norm in some sense. This will probably not be the case, but ρ_2 is parameterized and what we really care about is how well the best ρ_2 can approximate ρ_1.

1.4 Supervised learning, sampling and data

So far our discussion has not had much to do with machine learning. Let us briefly review a simple ML problem, mostly because we intend to adapt ML software to do our computations.

In supervised learning, we have a data set of N_{data} items, each of which is an input-output pair (x_n, y_n). These are supposed to be drawn from a probability distribution $\mathcal{P}$ on $\mathcal{X} \times \mathcal{Y}$. The goal is to choose the function Eq. (2) from $\mathcal{X}$ to $\mathcal{Y}$ which best describes the general relation $\mathcal{P}$ between input and output, in the sense that it minimizes some definition of the expected error (an objective or "loss" function). The procedure of making this choice given the data set (and perhaps limited "prior" information about $\mathcal{P}$) is called training the network.

A simple choice of objective function is the mean squared error,

$$\mathcal{E} = \mathbb{E}_{\mathcal{P}} \left[(f_w(x) - y)^2 \right]. \tag{14}$$

If we estimate this by evaluating it on our data set, we get the training error

$$\mathcal{E}_{train} = \frac{1}{N_{data}} \sum_{n=1}^{N_{data}} (f_w(x_n) - y_n)^2. \tag{15}$$

A standard ML training procedure is the following. We start with an FFN as in Eq. (3), with the weights initialized to random values – in other words, we draw the w from some distribution independent of the data. A common choice is for each matrix element $w_{i_m}^{(m),i_{m+1}}$ to be an independent Gaussian random variable with mean zero and variance $1/\sqrt{D_m}$. This choice is made so that the expected eigenvalues of the weight matrix are order D_m^0.

The next step is to minimize Eq. (15) as a function of the weights. A simple algorithm for this is gradient descent, a stepwise process in which the weights at time $t+1$ are derived from those at t as

$$w(t+1) = w(t) - \epsilon(t) \frac{\partial \mathcal{E}_{train}}{\partial w}\bigg|_{w=w(t)}. \tag{16}$$

While this will only find a local minimum, it works better for these problems than one might have thought. One trick for improving the quality of the result is to make the step size $\epsilon(t)$ decrease with time, according to a "learning schedule" chosen empirically to get good results for the task at hand.

An improvement on this procedure is "stochastic gradient descent" or SGD. This is much like Eq. (16) except that instead of evaluating the training error $\mathcal{E}_{train}$ on the full data set, one evaluates it on a subset or "batch," with the batch varied from one step to the next so that their union covers the full data set. This was originally done for computational reasons but it also turns out to produce a noise term with beneficial properties, for example in helping to escape local minima.

Finally, once the optimization is deemed to have converged, one judges the results by estimating Eq. (14). This estimate must be made by using an independent data set from that used in training as otherwise we are rewarding our model for matching both signal and noise.[2] However in most applications we do not have any direct access to $\mathcal{P}$, rather we only have an empirical data set. Thus one starts by dividing the full data set into disjoint "training" and "testing" subsets, evaluates Eq. (15) on the training set for training, and then evaluates the sum of errors over the testing set to estimate $\mathcal{E}$. The final model can be very accurate, surprisingly so when compared to expectations from standard statistical theory. Let us cite [] as a few papers which study these theoretical questions.

While our problem is not one of supervised learning, it will be useful to phrase it in terms as similar as possible, so that we can most easily use ML software. The workflow of the supervised learning task involves defining a set of data points (x_n, y_n) which are independent of the weights, repeated evaluation of the network at each x_n to get a prediction $f(x_n)$ for the corresponding y_n, and optimization of an objective function which is a sum of terms

[2] In classification problems, one often uses networks with many more parameters than data points and which can completely fit the dataset, so that the minimum of $\mathcal{E}_{train}$ is zero! In this case $\mathcal{E}_{train}$ is clearly a poor estimate for $\mathcal{E}$.

which each depend on a single data point. The network is normally defined by concatenating layers, such as multiplication by a weight matrix (a fully connected layer), application of an activation function, and so on. These layers are implemented in associated software libraries, such as Keras for Tensorflow. As we explain next, while we will have to implement some new layers for our problem, otherwise our workflow is the same.

[There is also a related unsupervised learning problem: take the inputs to be a nontrivial embedding, say all the sections at degree $k > 1$, and learn a lower dimensional representation. The exact relation is not polynomial, for example we might take $(Z^i)^k \to Z^i$, so it is not clear how interesting it is.]

2 Implementation

We first list the geometric quantities entering the computation. There is the holomorphic three-form

$$\Omega = \dots \tag{17}$$

and the associated volume form

$$d\mu_\Omega \equiv \mathcal{N}_\Omega \Omega \wedge \bar{\Omega}. \tag{18}$$

The normalization $\mathcal{N}_\Omega$ will be described shortly. This volume form depends on the complex structure but is independent of the Kähler form and thus the embedding we use to represent M.

The Kahler form and associated volume element are

$$\omega_g = \partial_i \partial_{\bar{j}} K \tag{19}$$

$$d\mu_g \equiv \omega_g^3 = \det \omega_g = \det_{i,\bar{j}} \partial_i \partial_{\bar{j}} K. \tag{20}$$

Now the integral $\int_M \omega_g^3$ is a topological invariant, so one can choose the normalization of $d\mu_\Omega$ to make $\eta = 1$ for the Ricci flat metric.

We first describe the general implementation, which follows the approach taken in [] with the main difference being the use of the feed-forward network Eq. (12).

We start by sampling points on M for doing integrals, using the procedure of [**?**]. This sample will play the role of the input dataset in a supervised learning task. It will be naturally distributed according to the Fubini-Study volume element and thus (using a theorem of Shiffman-Zelditch)

$$\lim_{N_{data}\to\infty} \frac{1}{N_{data}} \sum_i f(x_i) = \int_M \omega_{FS}^3 f(x), \tag{21}$$

where the Fubini-Study metric ω_{FS} and volume element is the special case of Eq. (19) with $k = 1$ and a fixed hermitian metric $h_{i\bar{j}}$. Given such a metric, to sample from M we first choose two points a and b on the ambient WP space[3] from a normal distribution with covariance $h_{i\bar{j}}$. We then find the points on the intersection of the line $\lambda a + \rho b$ with M, in other words

[3]More details on computations on WP can be found in the Penn group papers.

the choices of λ/ρ for which $f(\lambda a + \rho b) = 0$. This equation will have $\deg f$ solutions (with multiplicity) and for our purposes (which do not look at correlations between points) we can simply add all of them to the data set as inputs x_n, evaluating the variable y_n for each.

By reweighting, we define an empirical approximation to $d\mu_\Omega$, which is

$$d\mu_{\Omega,\mathcal{D}} = \sum_{i\in\mathcal{D}} \frac{|\Omega(x_i)|^2}{\omega_{FS}(x_i)^3}. \tag{22}$$

The objective function measures the deviation of the metric from Ricci flatness. The simplest such measure is

$$\eta = \frac{\omega_g^3}{\Omega\wedge\bar{\Omega}} \tag{23}$$

The "least squares error" for Ricci flatness is then

$$\begin{aligned}
\mathcal{E} &= \int_M \Omega\wedge\bar{\Omega}\,(\eta-1)^2 && (24)\\
&= \int_M \omega_{FS}^3 \frac{\left(\omega_g^3 - \Omega\wedge\bar{\Omega}\right)^2}{\omega_{FS}^3\,\Omega\wedge\bar{\Omega}} && (25)\\
&= \int_M \omega_{FS}^3 \frac{(f_w(x_i)-y_i)^2}{y_i} && (26)
\end{aligned}$$

where

$$\begin{aligned}
y_i &= \frac{\Omega\wedge\bar{\Omega}}{\omega_{FS}^3} && (27)\\
f_w(x_i) &= \frac{\omega_g^3}{\omega_{FS}^3}. && (28)
\end{aligned}$$

The point of writing it this way is that y_i is independent of the weights, so we can also consider it as part of the "data set," the target value for f_w evaluated at the point x_i. An ML package will provide common objective functions such as least squared error and we will be able to use this one directly.

Note that we can choose a different function of η which has the same optimum yet has a statistics/information theory interpretation. For example, if we consider a volume form on the CY as a probability density, it is natural to look at the KL divergence:

$$\mathrm{KL}\left(d\mu_\Omega | d\mu_g\right) = -\int_M d\mu_\Omega \log\eta. \tag{29}$$

This will also have its minimum at the Ricci-flat metric. Still, it may be less convenient for optimization, so we compute with the least squares definition.

Thus we need to construct layers which implement Eq. (12), the two derivatives which produce $g_{i\bar{j}}$, and the nonlinear definition of f_w in terms of this. Now optimization requires

evaluating the gradient of f_w and this is done by the backpropagation algorithm built into the ML package. Thus there is already a built-in ability to compute derivatives, the question is how to use it to compute the three derivatives implicit in the definition grad $\mathcal{E}$. This should be possible because we can chain gradient computations, for example see [1].

Now given that the inputs to the network are the coordinates Z^a of a point x_i, the derivatives with respect to Z can be obtained by interpolating an additive layer $Z \to Z + b$ and taking the b derivative. Presumably, we can combine these to get the required second and third derivatives.

2.1 Adaptive method

We could vary the dataset to lower the variance of the objective function. Interesting to work out but not clear it is worth implementing.

3 Results

Implement the $d = 1$ and $d = 2$ versions of this algorithm with various widths. Compare to the original results and also to the rank restricted version. We should take the hint from ML and produce a test set of data points in addition to our training set in order to evaluate Eq. (26) (in the existing work one uses a different functional and the same dataset).

4 Neural Tangent Kernel

Given a loss function $\mathcal{L}$, we define the NTK on $M \times M$ as

$$K_{NTK}(z, z') = \sum_w \Pi(w, w') \left\langle \left. \frac{\partial \mathcal{L}}{\partial w} \right|_z \left. \frac{\partial \mathcal{L}}{\partial w'} \right|_{z'} \right\rangle. \tag{30}$$

where the expectation value is over the initialization distribution for the weights w. Our loss function is the integral of a density on M, so the NTK will be a bilocal density,

$$K_{NTK}(z, z') = d\mu_\Omega(z) d\mu_\Omega(z') \hat{K}_{NTK}(z, z'). \tag{31}$$

Let's use the loss function Eq. (29), then

$$\left. \frac{\partial \mathcal{L}}{\partial w} \right|_z = d\mu_\Omega \omega^{-1} \left. \frac{\partial \omega}{\partial w} \right|_z \tag{32}$$

$$= d\mu_\Omega \omega^{i\bar{j}} \partial_i \bar{\partial}_{\bar{j}} \left. \frac{\partial K}{\partial w} \right|_z . \tag{33}$$

If we take the linear ansatz Eq. (1), then $w \cong h$ and this becomes

$$\left.\frac{\partial \mathcal{L}}{\partial w}\right|_z = d\mu_\Omega \omega^{i\bar{j}} \partial_i \bar{\partial}_{\bar{j}} \left.\frac{\partial}{\partial w} \log h_{I\bar{J}} s^I \bar{s}^{\bar{J}}\right|_z \tag{34}$$

$$= d\mu_\Omega \omega^{i\bar{j}} \partial_i \bar{\partial}_{\bar{j}} \left.\frac{s^I \bar{s}^{\bar{J}}}{h_{I\bar{J}} s^I \bar{s}^{\bar{J}}}\right|_z . \tag{35}$$

This is already nonlinear in h, and the derivatives will make it more nonlinear. So, even if h is Gaussian initialized, it will not be easy to compute the expectation.

Let the initialization covariance be

$$\left\langle h_{I\bar{J}} h_{K\bar{L}} \right\rangle = P_{I\bar{L}} P_{K\bar{J}}, \tag{36}$$

then this determines a kernel

$$K_P(z, \bar{z}') = P_{I\bar{K}} s^I(z) \bar{s}^{\bar{K}}(\bar{z}'). \tag{37}$$

If the sum Π over w factorizes into a product of holomorphic and antiholomorphic, then it defines another kernel K_Π, which could be the same as K_P. This gets us down to

$$\hat{K}_{NTK}(z, z') = \omega^{i\bar{j}} \partial_i \bar{\partial}_{\bar{j}} \left|_z \omega^{k\bar{L}} \partial_k \bar{\partial}_{\bar{L}}\right|_{z'} |K_\Pi(z, \bar{z}')|^2 \left\langle e^{-K(z,\bar{z})} e^{-K(z',\bar{z}')} \right\rangle. \tag{38}$$

To compute this expectation value, we can write the inverses as (fill in details!)

$$\left\langle e^{-K(z,\bar{z})} e^{-K(z',\bar{z}')} \right\rangle = \int_0^\infty dt \int_0^\infty dt' \left\langle e^{-ths\bar{s}|_z - t'hs\bar{s}|'_z} \right\rangle \tag{39}$$

$$= \int_0^\infty dt \int_0^\infty dt' \ \exp -\frac{1}{2} PP \left(ts(z)\bar{s}(\bar{z}) + t's(z')\bar{s}(\bar{z}')\right)^2 \tag{40}$$

$$= \int_0^\infty dt \int_0^\infty dt' \ \exp -\frac{1}{2} \left(t^2 |K_P(z,\bar{z})|^2 + 2tt' |K_P(z,\bar{z}')|^2 + (t')^2 |K_P(z',\bar{z}')|^2\right)$$

$$\sim \det \begin{pmatrix} |K_P(z,\bar{z})|^2 & |K_P(z,\bar{z}')|^2 \\ |K_P(z,\bar{z}')|^2 & |K_P(z',\bar{z}')|^2 \end{pmatrix}^{-1/2} \tag{41}$$

This would be doable, but the problem is that ω^{-1} in Eq. (34) also depends on h in an even more nonlinear way. It is probably better to choose a loss function with simpler dependence on h. Still, all formulas for the curvature require inverting the metric. We may be better off with a "first order" formalism which includes the inversion as an explicit condition.

Integration over h is a "random Kähler geometry" problem in the sense of Klevtsov, Zelditch *et al.* Suppose this works, presumably the NTK will be a natural geometric kernel. We probably want to set $P = \Pi$ and minimize the dependence on prior data.

Can we guess the result? We make predictions from the NTK as

$$y_{pred}(x) = \vec{K}_{NTK}(x, x_i) \cdot (K_{NTK}(x_i, x_j))^{-1} \cdot \vec{y}_j. \tag{42}$$

So, if we had an orthonormal basis in which to expand the holomorphic volume form, we could fit it. However the goal here is not so much to fit the holomorphic volume form, rather it is to fit it in terms of a Kähler volume form. Thus it would be more useful to have a K_ω with simple dependence on ω and h, such that the minimum of

$$\mathcal{L} = \sum_i \left(d\mu_\Omega(x_i) - \sum_j K_\omega(x_i, x_j) d\mu_\Omega(x_j) \right)^2 \tag{43}$$

is at the Ricci flat ω. A natural candidate or at least building block for this K_ω is the Q operator of section 4 of Donaldson's 2005 paper, which is projection on the orthonormal basis of functions $s^I \bar{s}^{\bar{J}}/hs\bar{s}$. If we take h to be the $d\mu_{\Omega,\mathcal{D}}$ balanced metric, then this should be a reproducing kernel (check! also how do we orthonormalize the basis?). Conversely, optimizing $\mathcal{L}$ should lead us to the balanced metric. We can then use Eq. (1) to obtain K.

If this works, then since our multilayer ansatz is a nonlinear subspace of the general space of sections, then we just need to restrict the optimization to this space. Now we have an explicit formula for $hs\bar{s}$, but what does it mean to "project on a nonlinear subspace" ? Does it mean to accept the final layer defining the vector s as defining the subspace?

The upshot would be that the explicit computation of a loss function like Eq. (29) or even Eq. (26) may be an unneeded complication – since we have the explicit basis of sections we can construct our own kernel which will also have optimality properties for the Ricci flat metric. Or at least for the balanced metric, which if we are primarily interested in the Ricci flat metric, will bring in another $1/k^2$ error.

5 Analogies to information geometry, and other generalizations

An amusing question is whether there is any simple interpretation of the double derivative $\partial^2/\partial w \partial \bar{w}$ of either objective function Eq. (26) or Eq. (29). This is somewhat like a Kähler metric on the space of metrics and possibly has this interpretation.

Follow this up and discuss other relations between Kähler geometry, Hessian geometry and information geometry.

Possible analogs in quantum information? We can think of Ω as a wave function whose norm squared gives the probability. Is there an analogous definition for ω_g^3, perhaps involving a density matrix? The expression Eq. (12) gives us e^K as a density matrix for the LLL states, but it's not clear how to think of the Monge-Ampere expression Eq. (20).

5.1 General Riemannian geometries

We can generalize the concept to embed a manifold M into a high dimensional space $\mathbb{R}^N$ using a neural network. Then, rather than define a single function K, we can pull back the Euclidean metric to get a metric on M, and optimize a geometric condition on it. What class of embeddings can we get this way?

references: Roweis and Saul 2000, Hauser and Ray 2017 ?

Start with a two layer network and ReLU activation. We first need a primary embedding of M into some $\mathbb{R}^n$. Each point x will have a neighborhood which is linearly embedded, and these will make up regions bounded by planes $M_{ij}x^j = 0$. This representation will give a PL metric and we need to either smooth it, or define a Ricci flat PL metric. Either way we will need to find all the planes and their intersections. Also the condition that this is an embedding looks nontrivial.

A Math review

A first reference is GSW volume II chapters 15 and 16, especially §15.3, 15.4 and 15.5 on the Hodge decomposition and Dolbeault cohomology. After that, a classic math reference is Griffiths and Harris, Principles of Algebraic Geometry. This needs serious study, but §0.2 on complex manifolds and Dolbeault cohomology and §0.7 on Kähler metrics are not hard.

To go deeper into the subject one should study toric geometry and toric hypersurfaces. The classic physics paper is Witten's hep-th/9301042, and a standard math introduction is [2]. There are a lot of string theory PhD theses which work this out, e.g. Bouchard hep-th/0609123.

Let us first give the computation of the volume of $\mathbb{CP}^n$ and a hypersurface M defined as $f = 0$ in this space. These volumes are topological quantities, meaning that they are invariant under continuous deformations of the metric. In fact if we take the Fubini-Study metric

$$K = \frac{i}{2\pi} \log \sum_i^{n+1} |Z^i|^2 \tag{44}$$

or its restriction to M, normalized as we just did, the volumes will be integers.

The formula Eq. (19) expresses the Kähler metric as a total derivative of the Kähler potential. On reviewing Dolbeault cohomology, one sees a closely related formula

$$\omega = \partial\bar{\partial}K \tag{45}$$

for the $(1,1)$-form ω, the Kähler form. This can be used to construct the volume form,

$$\sqrt{\det g} = \det \omega = \frac{1}{n!}\omega^n. \tag{46}$$

(see p.31 of GH; it would be nice to check the normalizations in a symbolic algebra system). Furthermore, since $\partial\omega = 0$, we have

$$\omega^n = \partial(\bar{\partial}K \wedge \omega^{n-1}). \tag{47}$$

Now if K were a single valued function, Eq. (45) would imply that the integral of ω over any two-cycle would vanish. It is not, but this argument tells us that varying K by a function, $K \to K + f$, does not change the integral of ω. The same holds for the volume form by Eq. (47).

Next, let us see that for K given by Eq. (44),

$$\int_\Sigma \omega \in \mathbb{Z}, \tag{48}$$

for any two-cycle Σ. Furthermore there is a two-cycle for which this integral is 1. The integral only depends on the homology class of the cycle. In fact $\mathbb{CP}^n$ has a unique generator of H^2, as can be seen by embedding $\mathbb{CP}^1$ into it as $(z^1, z^2) \to (z^1, z^2, 0, \ldots, 0)$, the fact that $\mathbb{CP}^1$ is topologically a two-sphere, and the $SU(n+1)$ symmetry of $\mathbb{CP}^n$ which takes all such embeddings into each other. Thus we can just compute this integral on $\mathbb{CP}^1$. You should do this exercise.

A simpler way to do this integral and understand why it is quantized, is to break up $\mathbb{CP}^n$ into patches, and express the integral as a sum of terms involving the differences of K between different patches. In mathematics this comes from a relation between ω and an associated gauge field and line bundle $\mathcal{L}$. In other words, we can postulate a $U(1)$ gauge field A, with curvature $F = dA$, such that

$$\omega = F. \tag{49}$$

Then, the integral of F over a basis of homology two-cycles, is by definition the first Chern class of the line bundle. One writes $c_1(\mathcal{L}) = 1$, and in the math literature this $\mathcal{L}$ is called $\mathcal{O}(1)$. You can read about this in the references, but the simplest physical construction of F is the following: we consider a Dirac monopole at the origin in $\mathbb{R}^3$, and identify $\mathbb{CP}^1 \cong S^2$ with the sphere of unit radius centered at the origin. Then, A is the monopole field, and the fact that the integral Eq. (48) is quantized is just the Dirac quantization condition (where we set the charge of the electron to 1).

This is related to our problem as follows. We can relate the coordinates Z^i to solutions of the Schrödinger equation in the monopole magnetic field on $\mathbb{CP}^1$,

$$\psi_a^{(i)} = Z_a^i \exp -\frac{1}{2}|Z_a|^2. \tag{50}$$

Here $a = 1, 2$ labels the patches with $Z^a = 1$, and ψ_a is the wave function in patch a. Z_a^i is the value of the coordinate Z^i in that patch. In fact these solutions are ground states in the lowest Landau level.

Similarly, if one put k monopoles at the origin, one would get a solution with magnetic field k times as large. This defines the line bundle $\mathcal{O}(k)$, with $c_1 = k$.

Now, the math fact which I won't explain in detail, is that the numbers c_1 are also related to the number of zeroes of a section of the bundle. In the case at hand they are equal, for example Z^i has a single zero in the patch $a \neq i$ and no zero in the patch $a = i$. Similarly the section $(Z^i)^k$ of $\mathcal{O}(k)$ has k zeroes. In fact one can turn all the problems of integrating wedge products of these ω's, into counting zeroes of simultaneous equations. This is called intersection theory in the mathematics and in the simple case at hand works as follows: we

can associate $\mathcal{O}(1)$ with a dual two-cycle Σ, the generator of H^2. Then the powers ω^k are associated with dual $2k$-cycles which each generate their own H^{2k}. This implies that

$$\int_{\mathbb{CP}^n} \omega^n = 1 \tag{51}$$

and thus the volume of $\mathbb{CP}^n$ in the Kähler metric derived from Eq. (44) is $1/n!$.

One can check this directly by writing the coordinates as

$$Z^i = r_i e^{i\theta_i} \tag{52}$$

upon which the condition $1 = \sum |Z^i|^2$ becomes $1 = \sum r_i^2$. Thus $\mathbb{CP}^n$ is closely related to the sphere S^{2n+1} – it is obtained by quotienting by the overall phase, $\theta_i \to \theta_i + \epsilon$. This is a circle with constant volume, so the two volumes are just related by an overall 2π (I think).

Next, let us consider M, the hypersurface $f = 0$ in $\mathbb{CP}^{n+1}$, where f is a degree k homogeneous polynomial. The volume form on M is $\omega^n/n!$ where ω is the restriction of ω as above to this surface. To do this restriction in practice, one must embed the n-dimensional cotangent space T^*M in the $n+1$-dimensional cotangent space $T^*\mathbb{CP}^{n+1}$. Now the homogeneous coordinates Z^i are also sections of a line bundle $\mathcal{L}$, defined by restricting $\mathcal{O}(1)$ to M. However we cannot use them all as coordinates in a patch as they are redundant. In terms of the cotangent bundle we can express this redundancy as

$$T^*\mathbb{CP}^n \cong T^*M \oplus \mathcal{N}^*_{\mathbb{CP}^{n+1}} M, \tag{53}$$

where $\mathcal{N}^*_{\mathbb{CP}^{n+1}} M$ is the conormal bundle, in other words the dual to the normal bundle in which vectors normal to M live. While the normal vector to M depends on the metric, one can choose a section of the conormal bundle which does not. It is a one-form which vanishes if we contract it with a tangent vector on M. Thus it is just the one-form df, since the condition that a vector v is tangent to M is $v^i \partial_i f = 0$.

Thus, to restrict ω and the metric to M, we want to choose a local set of coordinates in which one of the coordinates Z^i is replaced by f, and change basis from $(dZ^1, dZ^2, \ldots, dZ^{n+1})$ to $(dZ^1, dZ^2, \ldots, df, \ldots, dZ^{n+1})$, and omit df. To restrict the volume form we just need the Jacobian of the resulting matrix (with df omitted). Writing the new coordinates as $\hat{Z}^i$, one can see that this matrix is (for $n = 2$ and replacing $\hat{Z}^3$),

$$\begin{pmatrix} d\hat{Z}^1 & d\hat{Z}^2 & df \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{\partial f}{\partial Z^1} & \frac{\partial f}{\partial Z^2} & \frac{\partial f}{\partial Z^3} \end{pmatrix} \begin{pmatrix} dZ^1 & dZ^2 & dZ^3 \end{pmatrix}. \tag{54}$$

To reexpress ω in the new frame, one must invert this matrix to get dZ^i in terms of the new frame. One can then omit the ∂f and $\bar{\partial} \bar{f}$ components.

As we discussed, we want to choose subpatches of each patch on $\mathbb{CP}^{n+1}$ on which the various components $|\partial f/\partial Z^i|$ are maximized, and then omit this $\hat{Z}^i$ in that patch. Thus this relation will take a slightly different form in each patch.

One can even carry out the argument in terms of intersection theory to get the volume of M in the restriction of the Fubini-Study metric. Now, the Kähler form ω from Eq. (44) is the curvature of $\mathcal{O}(1)$, which is just obtained by restriction. To do an integral over M in intersection theory, one can wedge the form with the curvature of the normal bundle. This is degree f as is explained in the "adjunction formulas" in §1.1 of GH, pages 145–147 in my copy. Thus we have the volume

$$\frac{1}{n!}\int_M \omega^n = \frac{\text{degree } f}{n!} \tag{55}$$

$$= \frac{n+2}{n!} \qquad \text{if degree } f = n+2 \tag{56}$$

as is the case for a Ricci flat manifold. So the volumes for T^2, K3 and the quintic should be 3, 2 and 5/6 respectively.

References

[1] https://github.com/tensorflow/tensorflow/blob/r2.0/tensorflow/python/ops/gradients_i

[2] http://home.ustc.edu.cn/ hanzr/pdf/Introduction%20to%20Toric%20Varieties-Fulton.pdf