Learning from Data&Prior

Carlos C. Rodríguez

405 Main Street,
New York, NY10044.

Abstract

Data is prior information and prior information is more data. Meaningful data is always a two part object. The observable (arbitrary) label and the (unobservable) label for the probability distribution that generated the observed label. A meaningful data space is always a DataTheory space containing objects (x, p). We identify the main category of these spaces by defining the morphisms to be *sufficient* transformations $(x, p) \to (x', p')$, such that from (x', p') we can recover $(\tilde{x}, p)$ where $\tilde{x}$ and x come from the same unboservable p. The only known measures of separation between (unormalized) distribtuions that are invariant under sufficient transformations were identified by Chentsov and Amari [Va] as the one parameter class of δ-information deviations. The δ-deviations allow us to assign a scalar $\mathcal{A}(t, \eta)$ to a pair of distributions where $t = t(y)$ is interpreted as the true distribution of labels and $\eta = \eta(p)$ as a prior over the model. The invariant scalar $\mathcal{A}(t, \eta)$ measures the information in η about the true distribution t given the model. The critical points (t, η) of the invariant action produce the most ignorant prior distributions given the model and a guess at the true distribution. When a sample of data is observed, the true distribution is replaced by the empirical producing new ways for processing the observations that include and extend maximum likelihood and bayesian inference. Finally the general theory is applied to the simple logistic regression model, producing new targets and penalties that are shown to outperform the standard methods in simulations using TensorFlow.

1 Labels and Data

Labels are chosen by us. If they can be changed, what remains after the changes contains important information about the category of objects that the labels are labeling. Today, topology, geometry and vector spaces are defined in these category theory terms. We get topology if we demand the changes to be continuous bijections, geometry if we demand isometries and vector spaces if we demand the relabelings to be linear.

In principle any symbol could be used as a label. The symbol by itself is meaningless. The meaning is in the many. The meaning is in the relation of

the symbols among themselves. This symbol as opposed to all other possible symbols labeling objects around it. That is what fixes the meaning.

There is no shortage of labeling systems. Numerical labels are often useful, not only because there are infinitely many to choose from but also for their internal structures. However, one system of labels is singled out when thinking about data. The system of labels consisting of probability distributions over a given data space. Probability distributions provide a mechanism for encoding the relations among the observed data labels.

An element (labeled as) x of a "data" space, by itself is not data. It is meaningless. The meaning is provided by assuming that this label x was the result of an hypothetical probabilistic sampling from the data space.

The basic data-object of interest has two sides: an observable label x on one side, and an unobservable label p on the reverse side: $px \equiv (x, p)$. A choice of range for these (x, p) data-objects provide concrete and objective prior information.

Let $(x, p) \in S$ and call S the statistical DataTheory space. The space S is a subset of $\mathcal{X} \times \mathcal{P}$ the set $\mathcal{X}$ of all possible data labels x cartesian product $\mathcal{P}$, the set of all possible p, probability distributions over $\mathcal{X}$.

There is often more assumed structure. The space S could be taken as a submanifold of $\mathcal{X} \times \mathcal{P}$. For example, think of $\mathcal{X}$ as a manifold (e.g. a sphere) and $\mathcal{P}$ as embedded in the Hilbert space of wave functions.

The S-Category of DataTheory spaces, can be simply defined by taking the morphisms to be standard coordinate changes (for both x and p separate) enlarged by more general sufficient transformations $(x, p) \to (x', p')$. Where by "sufficient" we mean that from (x', p') we can recover $(\tilde{x}, p)$ in the sense that $\tilde{x}$ and x both come with the same hidden side p. This is just the canonical Fisher notion of statistical sufficiency.

Now think of the elements of S as another set of labels. A probability distribution over S will fix a meaning for these new labels. Notice,

$$P(x, p) = P(x|p)P(p)$$

A likelihood $P(x|p)$ (e.g. $p(x)$) and a prior $P(p)$. Recall that "," really means the logical "AND", which is commutative. Thus,

$$P(x, p) = P(p, x) = P(p|x)P(x)$$

A posterior $P(p|x)$ and the evidence $P(x)$. Bayes theorem follows,

$$P(p|x) = \frac{1}{P(x)} P(x|p)P(p)$$

When p ranges over a Riemannian manifold with finite volume, there is a notion of equally likely p given by the uniform distribution over this finite volume. There is nothing subjective about this, once you choose S.

1.1 Fisher Information

Spaces of probability distributions are clearly not closed under addition and multiplication by scalar. They are not vector spaces. Nevertheless, there are canonical embeddings of probability distributions on Banach spaces that respect the S-Category. For $0 < \delta < 1$ the Banach space $L_{1/\delta}$ of δ powers of measures contains the δ coordinates $l_\delta(p)$ of a probability distribution p with, $l_\delta(p) = p^\delta/\delta$. The Banach space associated to $1-\delta$ is the topological dual of the space associated to δ. The only Hilbert space is the self-dual L_2 associated to $\delta = 0.5$. Fisher information is just the metric induced on the model as it is embedded into L_2. In other words: label the probabilities p in your model with the vector $2\sqrt{p}$ in the Hilbert space L_2 of square integrable functions. The Information metric at p is the matrix $g(p) = (g_{ij}(p))$ with components,

$$g_{ij}(p) = \int \partial_i(2\sqrt{p})\partial_j(2\sqrt{p})\,dx = \int (\partial_i \log p)(\partial_j \log p)\,p\,dx$$

where ∂_i is the partial derivative w.r.t. the ith coordinate vector, and the integrals are over the space of x. Parametric statistical models with smooth parametrizations are Riemannian manifolds with the information metric $g(p)$.

1.2 δ-separation, Entropy, and Duality

Let $l_0(p) = \log(p)$. For $0 \leq \delta \leq 1$ define the δ-separation between (possibly unnormalized) distinct distributions p and q by the positive number $I_\delta(p:q) = I_{1-\delta}(q:p)$ given for $0 < \delta < 1$ by:

$$I_\delta(p:q) = \frac{1}{\delta(1-\delta)} \int [\delta p + (1-\delta)q - p^\delta q^{1-\delta}]\,dx$$

and by the corresponding limit when $\delta \in \{0, 1\}$. Thus,

$$I_0(p:q) = \int \left(q - p + p\log\frac{p}{q}\right) dx = I_1(q:p).$$

The entries $g_{ij}(p)$ of the information matrix at p are also given by the duality product between the coordinates $l_\delta(p) \in L_{1/\delta}$ and the dual coordinates $l_{1-\delta}(p) \in L_{1/(1-\delta)}$ as,

$$g_{ij}(p) = \int (\partial_i l_\delta(p))(\partial_j l_{1-\delta}(p))\,dx$$

The $I_\delta(p:q)$ numbers are invariants of the S-category i.e., they are invariant under sufficient transformations. No other measures of separation between probability distributions, besides these or functions of these, are known to respect the S-category.

1.3 Categorically Sound Objectives

Having observed data arbitrarily labeled $y_1, y_2, \ldots, y_n$ in a background of prior information how should we proceed extracting meaning to best predict unobserved y_{n+1} ? What should we optimize? What should the target be?

In our current deep learning frenzy, Ed Jaynes's words resonate with renewed power:

Maximize Ignorance subject to whatever is assumed to be known!

This is just *Maximum Honesty.* An ethical principle.

The Actions of Ignorance

The risk functionals $\mathcal{A} = \mathcal{A}(t, \eta)$ (see below) defined for a DataTheory space in the S-category, rank the pairs (t, η) according to their information, i.e., separation from ignorance. Here $t = t(y)$ is a (not necessarily normalized) distribution on the space of data labels y and $\eta = \eta(p)$ is a (not necessarily normalized) distribution on the hypothesis space M of possible theories p. We interpret t as the (unknown) true distribution for the data and η as the (unknown) prior distribution on M. Let $\pi = \pi(p)$ be a given fix prior distribution on the hypothesis space M of possible theories p. This π will be taken as a diffuse pre-prior on M. When the information volume $\mathrm{vol}(M)$ is finite, we take $\pi(p) = 1$ as the uniform distribution on M. We always write distributions on the Riemannian manifold M as scalar density fields relative to the invariant Riemannian volume form dp on M.

Denote by $t\pi = P(y, p) = t(y)\pi(p)$, i.e. y and p are chosen independently. First pick $p \in M$ according to the distribution π, then independently choose label y in the data space according to the (true) distribution t. Consider now any other distribution on (y, p). $P(y, p) = p(y)\eta(p) = p\eta$. Then, $I_\delta(p\eta : t\pi)$ measures the δ-separation between the joint distributions $p\eta$ and $t\pi$ and $I_{1-\nu}(\eta : \pi)$ measures the separation between the priors η and π. For $\beta > 0$ define the positive scalar,

$$\mathcal{A}(t, \eta) = \beta \; I_\delta(p\eta : t\pi) + I_{1-\nu}(\eta : \pi) \tag{1}$$

The (unnormalized) pair (t, η) that minimizes the action $\mathcal{A}$ is,

$$\eta(p) = [1 + \beta\nu I_\delta(p : t)]^{-1/\nu} \; \pi(p) \tag{2}$$

$$t^\delta(y) = \int p^\delta(y) \, \eta(p) \, dp \tag{3}$$

The action (1) does not contain derivatives and its optimization is a simple problem in the calculus of variations. Just take derivatives equal to zero as if the functions were real variables. The expressions (2) and (3) pack a considerable amount of information in a short space. Think of (2) as defining a kernel $k(p, t)$ on the dual Banach spaces associated to δ and $1-\delta$. It is a measure of separation

between the probability distributions p and t on the data labels. The kernel (2) is literally the prior distribution of maximum ignorance. The expression (3) provides the δ coordinates of t as the average of the δ coordinates of p or equivalently as the mean kernel. This is remarkably similar to the Reproducing Kernel Hilbert Space (RKHS) embeddings of probability distributions but in Banach spaces instead. The big difference with the RKHS approach is that here the kernel is fixed by the theory as the most ignorant prior given the choice of S. Besides, all the expressions are invariant under sufficient transformations preserving the S-Category. Notice that $t(y)$ is given by (3) as the length of a function measured in the Reproducing Kernel Banach Space associated to δ (δ-RKBS) with kernel (2).

Let us try to unpack the information provided by the optimizer (2). This distribution over M achieves a compromise between spreading the probability mass over M so that $I_{1-\nu}(\eta : \pi)$ is small and concentrating the mass around the δ-projection of t on M so that $I_\delta(p\eta : t\pi)$ is also small. This distribution is indeed most ignorant given only the choice of S.

A simple geometric interpretation of (2) is obtained by noticing that, $I_\delta(p\eta : t\pi) = \int I_\delta(p : t)\,\eta$ so that the minimizer (2) is the one minimizing $I_{1-\nu}(\eta : \pi)$ subject to the constrain that $\int I_\delta(p : t)\,\eta < C$, i.e., subject to the constrain that the expected distance from t (expectation taken w.r.t. the η distribution) is bounded above by some constant C. The positive scalar β is the Lagrange multiplier associated to the inequality constraint. Equivalently we can think of η as the minimizer of $I_\delta(p\eta : t\pi)$ subject to the constrain that $I_{1-\nu}(\eta : \pi) < C$. i.e., this η is the prior distribution that is most ignorant about the data (making $p\eta$ as close as possible to the independent model $t\pi$ in δ-separation) among those close to the spread pre-prior π on M. Notice that when the volume of M is finite we can (and should) take $\pi(p) = 1$ since in the absence of extra prior information besides the choice of S, we should regard all the $p \in M$ as equally likely apriori. The actual location of the (unknown) true distribution t is the only bias and that is in fact what is encoded in the action $\mathcal{A}$.

Figure 1 shows the "true" distribtuion t outside the model M together with the δ-separation between t and a probability distribution $p \in M$. The δ-projection of t onto M is represented by the point $q \in M$ closest to t in δ-separation.

Notice that when the information volume of M is finite so that $\pi(p) = 1$, we have,

$$\begin{aligned} t &= \arg\max_p \eta(p) \\ &= \arg\max_p \left[1 + \beta\nu I_\delta(p : t)\right]^{-1/\nu} \\ &= \arg\min_p I_\delta(p : t) \end{aligned}$$

i.e., the assumed true distribution t maximizes the prior probability. If we constrain $p \in M$ the δ-projection (depicted as q in figure 1) of t on M is the

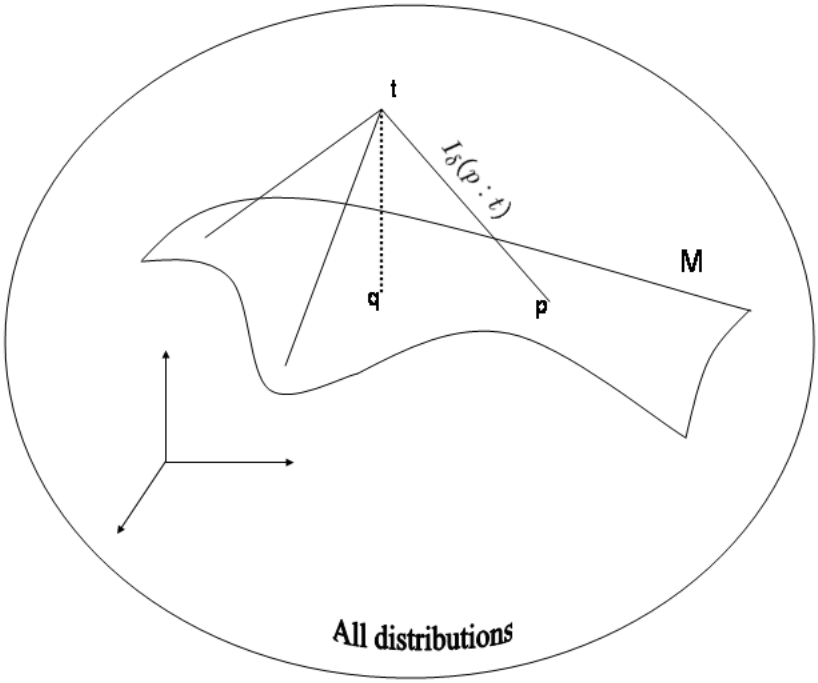


Figure 1: Pictorial representation of the Actions of Ignorance

one with highest prior probability.

Maximum Likelihood, Bayesian Inference and Maximum Honesty

The new way to statistical inference provided by Maximum Honesty is very simple. If data labels $y_1, y_2, \ldots, y_n$ are observed, just plugin the empirical distribution $\hat{t}_n$ instead of t in (2) and predict with (3).

Notice that the Maximum Likelihood Estimator (MLE) $\hat{p} \in M$ is the $\delta = 1$ projection of the empirical onto M.

$$\begin{aligned}
\hat{p} &= \arg\min_{p \in M} I_1(p : \hat{t}_n) \\
&= \arg\min_{p \in M} \int \left[p - \hat{t}_n + \hat{t}_n \log \frac{\hat{t}_n}{p} \right] dy \\
&= \arg\max_{p \in M} \left[\sum_{i=1}^{n} \log p(y_i) - \int p(y)\, dy \right]
\end{aligned}$$

where we have used the (unnormalized) empirical, $\hat{t}_n(y) = \sum_{i=1}^{n} \delta(y - y_i)$ and the fact that $I_1(p : \hat{t}_n) = I_0(\hat{t}_n : p)$.

Let's work out a simple example to help fix the ideas. Take M as the set of unnormalized one dimensional Gaussian distributions. The members of M given by their Lebesgue densities are,

$$p(y) = c \exp\left(\frac{-1}{2\sigma^2} (y - \mu)^2 \right)$$

we have,

$$\int p(y)\, dy = c\sigma \int \exp(-z^2/2)\, dz = ck\sigma.$$

We need to maximize,

$$L(c, \mu, \sigma) = \frac{-1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu)^2 + n \log c - ck\sigma.$$

the sufficient conditions are,

$$\frac{\partial L}{\partial c} = \frac{n}{c} - k\sigma = 0 \quad \Rightarrow \quad c = \frac{n}{k\sigma} \tag{4}$$

$$\frac{\partial L}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^{n} (y_i - \mu) = 0 \quad \Rightarrow \quad \hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{5}$$

$$\frac{\partial L}{\partial \sigma} = \frac{1}{\sigma^3} \sum_{i=1}^{n} (y_i - \mu)^2 - ck = 0 \quad \Rightarrow \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{6}$$

where we used (4) and (5) to obtain (6). Notice that we never needed the fact that $k = \sqrt{2\pi}$. In more complicated problems the trade of a derivative with respect to an extra parameter c for the computation of the exact normalizing constant could be very useful. Hence, the standard MLE is obtained by projecting the unnormalized empirical onto the space of unnormalized model distributions M using $\delta = 1$ i.e., I_1 as the information separation.

δ-MLE

Now repeat the above but with a general δ to obtain $\hat{p}_{1-\delta}$, the $(1-\delta)$-MLE.

$$\hat{p}_{1-\delta} = \arg\max_{p \in M} \left[\sum_{i=1}^{n} p^{\delta}(y_i) - \delta \int p(y)\, dy \right] \tag{7}$$

When M is the space of unnormalized one dimensional Gaussian distributions (as above), the $(1-\delta)$-MLE is obtained as the solution of two nonlinear equations,

$$\mu = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i} \tag{8}$$

$$\sigma^2 = \frac{\sum_{i=1}^{n} w_i (y_i - \mu)^2}{\sum_{i=1}^{n} w_i} \tag{9}$$

where the weights w_i also depend on the unknowns,

$$w_i = \exp\left\{ \frac{-\delta}{2\sigma^2} (y_i - \mu)^2 \right\} \tag{10}$$

Iterative substitution starting from the usual MLE quickly converge to the fix point. The estimators for the mean provide robust alternatives to the standard sample average but the estimators for the variance become more and more biased as δ increases. Recall that even the standard MLE (6) is biased. Notice also that the standard MLEs are recovered in the limit $\delta \to 0$ since in that case $w_i \to 1$.

1.3.1 Direct Posteriors

When the unknown true distribution t is replaced by the unnormalized empirical $\hat{t}_n$ in (2) we get,

$$\hat{\eta}(p) = [1 + \beta\nu I_{\delta}(p : \hat{t}_n)]^{-1/\nu}\ \pi(p) \tag{11}$$

$$\hat{t}^{\delta}(y) = \int p^{\delta}(y)\, \hat{\eta}(p)\, dp \tag{12}$$

we call $\hat{\eta}(p)$ the (unnormalized) *direct posterior* with parameters β, ν and δ. We call (12) the δ-coordinates of the δ-predictive distribution. The special case

when $\delta = 1$, $\nu = 0$ and $\beta = 1$ is singled out as particularly important. By taking the limit when $\nu \to 0$ in (11) with $\delta = 1$ and $\beta = 1$, we get,

$$\hat{\eta}(p) = \exp\left(-n\, I_0(\hat{t}_n : p)\right)\, \pi(p) \tag{13}$$

where we have used the fact that $I_1(p : \hat{t}_n) = I_0(\hat{t}_n : p)$. We noticed that, had we used the normalized empirical instead of the unnormalized version, we would have needed to take $\beta = n$ in order to get the same result. We have,

$$I_0(\hat{t}_n : p) = \int p(y)\, dy - \sum_{i=1}^{n} \log p(y_i) + C \tag{14}$$

If $\int p dy = 1$, i.e. for normalized probability distributions (13), gives the unnormalized direct posterior:

$$\hat{\eta}(p) = p(y_1)p(y_2)\ldots p(y_n)\, \pi(p) \tag{15}$$

that we recognize as the unnormalized posterior distribution when the likelihood is $\prod_{i=1}^{n} p(y_i)$ and the prior is $\pi(p)$. Moreover, with (15) and $\delta = 1$ replaced in (12) we get,

$$\hat{t}(y) = \int p(y)\, (\prod_{i=1}^{n} p(y_i))\, \pi(p)\, dp \tag{16}$$

that we recognize as the standard bayesian unnormalized predictive distribution. In other words with this special choice of parameters the inference is as if we had used Bayes Theorem but we did not! Bayesian Inference was produced automagically as a special case of maximum ignorance. Maximum honesty is more general than Bayesian Inference.

To get simpler formulas we are going to assume normalized distributions for the rest of this section. Now,

$$\hat{t}_n(y) = \sum_{i=1}^{n} \frac{1}{n} \delta(y - y_i)$$

so that for normalized p we have,

$$I_\delta(p : \hat{t}_n) = \frac{1}{\delta(1-\delta)} \left[1 - \sum_{i=1}^{n} (\frac{1}{n})^{1-\delta} p^\delta(y_i)\right]$$

producing the direct posterior:

$$\hat{\eta}(p) = \left[1 + \frac{\beta\nu}{\delta(1-\delta)} \left(1 - \sum_{i=1}^{n} (\frac{1}{n})^{1-\delta} p^\delta(y_i)\right)\right]^{-1/\nu} \pi(p) \tag{17}$$

The $p \in M$ that maximizes the direct posterior probability provides a natural target:

$$
\begin{aligned}
p^* &= \arg\max_{p\in M}\left\{\hat{\eta}(p)\pi(p)\sqrt{\det g(p)}\right\} \\
&= \arg\min_{p\in M}\left\{\frac{1}{\nu}\log\left[1+\frac{\beta\nu}{\delta(1-\delta)}\left(1-\sum_{i=1}^{n}(\frac{1}{n})^{1-\delta}p^{\delta}(y_i)\right)\right]\right. \\
&\qquad\qquad \left. -\log\pi(p)-\frac{1}{2}\log\det g(p)\right\} \qquad (18)
\end{aligned}
$$

If instead of searching for the minimum over all $p \in M$ we constrain to p close to a given p_0 we can heuristically justify a useful penalized optimization analogous to ridge regression. This can be done by assuming that the L_2 coordinate vectors are close so that $4||p^{1/2}-p_0^{1/2}||_2^2 \leq C$ for a small $C > 0$. But, recall that the information metric is the metric induced on M as embedded in L_2. Thus, in a given parameterization of M, for example $M = \{P_w : w \in R^k\}$ the constraint can be implemented as $(w-w_0)^T g(w)(w-w_0) \leq C$. This could be further interpreted as choosing $\pi(p) = \pi(w) = \exp(-\lambda||w-w_0||_w^2)$ as a gaussian prior on M. If we denote by $\mathcal{L}(w)$ the loss function between the curly brackets in (18) the penalized target will be,

$$
\mathcal{L}(w) + \lambda(w-w_0)^T g(w)(w-w_0) \qquad (19)
$$

where $\lambda > 0$ is the Lagrange multiplier associated to the constraint. When the dimension of the manifold M, (i.e. k) is large, we expect the shrinking towards (any) w_0 to behave like Herbert Robbins's empirical Bayes estimators that borrow strength from the different dimensions (when $k \geq 3$) and also help control overfitting.

1.3.2 Deep Teaching for Deep Learners

There is much more that needs to be unpacked from (2) and (3). Without data, from a choice of DataTheory space S and parameters $t, \eta, \pi, \beta, \delta, \nu$ we get a new DataTheory space with new parameters at a higher level of abstraction just by extremizing honesty. We can repeat the process at the higher level and obtain what I call the abstraction sequence. Recall that a probability distribution on data labels y is nothing but a code for the data. An explanation for that data. A theory. A probability distribution on the set of those theories (i.e., a prior) therefore gives an explanation for the explanations, etc. Now bring in the observed labels $y_1, y_2, \ldots, y_n$ at the ground layer 0. Take,

$$
\begin{aligned}
t_0(y) &= \sum_{i=1}^{n} \delta(y - y_i), \quad M_0 = M, \quad S_0 = \mathcal{X} \times M_0, \quad \eta_0(p) = \pi(p) \\
&\Longrightarrow \\
\eta_{i+1}(p) &= \eta(p|t_i), \quad t_{i+1}^{\delta} = \int_{M_0} p^{\delta}(y)\, \eta_{i+1}(p)\, dp, \quad M_{i+1} = \{t_{i+1}\text{'s}\} \\
&\qquad S_{i+1} = \mathcal{X} \times M_{i+1} \\
&\Longrightarrow \quad \ldots \quad \Longrightarrow \quad (S^*, t^*(y), \eta^*(p))
\end{aligned}
$$

where we move from the ground layer-0 to layer-1,..., layer-i, to layer-$i+1$, etc, by maximizing honesty with $\eta(p|t_i)$ given by (2) replacing t with t_i. The idea is to model the outputs of the layers of a deep learning model as an abstraction sequence of this type. Exactly how to do this, at this time, is not clear but the theory of ignorance shows a way forward that needs to be tested.

Can this be useful?

That will depend on how accurately we can approximate $g(p)$ and its determinant. Even though this could be challenging, it is easier than the computation of the inverse $g^{-1}(p)$ which is required for computing Amari's natural gradient. Besides, all the current attempts to implement the natural gradient can be immediately used in (18) and (19).

Some of the current difficulties of deep learning are no doubt due to the lack of categorical invariance of the procedures. When the correct information geometry of the hypothesis space is taken into account, we should expect faster convergence with much less data.

In the following sections, standard logistic regression is used to illustrate the general theory. This corresponds to only one layer of a deep learning network. A sequence of non interacting layers produce block diagonal information matrices which can, in principle, be studied separately one layer at a time. The new penalty terms involving the metric $g(p)$ may help in more general layers than plain logistic regression just like l_1 and l_2 do. For recurrent, convolutional and other types of layers that share parameters more work seems to be needed but promises big rewards. Geometry is our best known antidote to the curse of dimensionality.

2 Logistic Regression

Logistic regression is a special kind of regression where we want to explain labels y with other labels x, called features, using a parametric model labeled with a vector of parameters w, called weights. Supervised learning models are of this

type. Often we are not interested in modeling the features x, only the y in terms of the x's.

The DataTheory space is given by the assumption that, conditionally on w

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

are independent with the probabilities $P(y_i|x_i, w)$ specified as a function of the vector of weights w.

The special case of logistic regression is obtained when the $y_i \in \{0, 1\}$ and instead of labeling the chances for y_i with a number in the constrained interval $[0, 1]$ we monotonically relabel them in the unconstrained real line R, by using the logarithm of the odds of y_i instead. Finally, demand the log odds of y_i to be a linear function of the features x_i.

Let's spell out completely the DataTheory space for logistic regression for n observations with k parameters:

Assume $x_i \in R^{1\times k}$ row k-vector of covariates, $y_i \in \{0, 1\}$, where the distribution of y_i conditionally on x_i and w, is Binary (Bernoulli) Bin(θ_i), with $\theta_i = \theta_i(x_i, w) = P(y_i = 1|x_i, w) = 1 - P(y_i = 0|x_i, w)$ such that,

$$\log \frac{\theta_i}{1 - \theta_i} = x_i w \tag{20}$$

where $w \in R^{k\times 1}$ is the column k-vector of parameters.

Let $x \in R^{n\times k}$ be the n by k design matrix of observed covariates. The above assumptions, define (conditionally on the covariates x) the DataTheory space $S = \{0, 1\}^n \times \mathcal{P}_x^k$ where,

$$\mathcal{P}_x^k = \{P_{w,x} : w \in R^k\}$$

with $P_{w,x} : \{0, 1\}^n \to [0, 1]$ the probability distribution indexed by a k-vector $w \in R^k$ given by, $P_{w,x}(y_1, y_2, \ldots, y_n) = \prod_{i=1}^n P(y_i|w, x_i) = \prod_i \theta_i^{y_i}(1 - \theta_i)^{1-y_i}$ where,

$$\theta_i = (1 + \exp(-x_i w))^{-1}. \tag{21}$$

2.1 Information Metric of Logistic Regression

The Fisher information matrix is given at $p = P_{w,x}$ by,

$$g(w) = E_w \left\{(\nabla \log p(y|x, w))^T (\nabla \log p(y|x, w))\right\}$$

where E_w denotes expectation w.r.t. $P_{w,x}$ and ∇ is the gradient w.r.t. w.

Let's denote by $l = l(w) = \log p(y|x, w)$. Then,

$$l(w) = \sum_{i=1}^{n} \{y_i \log \theta_i + (1 - y_i) \log(1 - \theta_i)\}$$

therefore,

$$\nabla l(w) = \sum_i \left\{ \frac{y_i}{\theta_i} - \frac{1-y_i}{1-\theta_i} \right\} \nabla\theta_i$$

and,

$$\begin{aligned}
\nabla\theta_i &= \frac{e^{-x_i w}}{(1+e^{-x_i w})^2} x_i \\
&= \theta_i(1-\theta_i)x_i \\
&= \frac{x_i}{(1+e^{-x_i w})(1+e^{x_i w})} \\
&= \frac{x_i}{2+e^{x_i w}+e^{-x_i w}} \\
&= \frac{x_i}{2(1+\cosh(x_i w))}
\end{aligned}$$

substituting the second line above into the previous expression for $\nabla l(w)$, we obtain $\nabla l(w) = \sum_i (y_i - \theta_i)\, x_i$. Hence,

$$(\nabla l)^T (\nabla l) = \sum_{i,j} (y_i - \theta_i)(y_j - \theta_j) x_i^T x_j$$

therefore, taking expectations with the probability distribution $P_{x,w}$ and recalling that the y_i are independent for different values of i we get,

$$\begin{aligned}
g(w) &= \sum_{i,j} \mathrm{cov}(y_i, y_j) x_i^T x_j \\
&= \sum_i \mathrm{var}(y_i) x_i^T x_i \\
&= \sum_i \theta_i(1-\theta_i) x_i^T x_i
\end{aligned}$$

thus, using the last line in the above series of expressions for $\nabla\theta_i$, we finally obtain two useful expressions for the information metric,

$$g(w) = \frac{1}{2} \sum_{i=1}^{n} \frac{x_i^T x_i}{1+\cosh(x_i w)} \tag{22}$$

which in matrix form is,

$$g(w) = \frac{1}{2} x^T \,\mathrm{diag}\left(\frac{1}{1+\cosh(xw)} \right) x \tag{23}$$

2.2 The Information Volume of Logistic Regression

The hypothesis space M generated by the logistic regression model is a Riemannian manifold of dimension k with metric tensor given by the information

matrix at each w by the expression (23). The volume form on a Riemannian manifold with metric $g(w)$ is given in the $w = (w^j)$ coordinates by,

$$dp = dV(w) = \sqrt{\det g(w)}\, dw^1 \wedge dw^2 \ldots \wedge dw^k$$

with total (k-dim) volume given by integrating the volume element over M.

$$\mathrm{vol}(M) = \int_M dV = \int_{R^k} \sqrt{\det g(w)}\, dw.$$

When x is of full rank so that $\det x^T x > 0$, we have $\mathrm{vol}(M) < \infty$. When $k = n$, i.e. when x is a square k by k matrix, the computation of the determinant is trivial using expression (23).

$$\det g(w) = 2^{-k} \det(x^T x) \prod_{j=1}^{k} \frac{1}{1 + \cosh(x_j w)} \tag{24}$$

thus,

$$\begin{aligned}
\mathrm{vol}(M) &= 2^{-k/2} |\det x| \int \frac{dw}{\sqrt{\prod_{j=1}^{k}(1 + \cosh(x_j w))}} \\
&= 2^{-k/2} \int \frac{du}{\sqrt{\prod_{j=1}^{k}(1 + \cosh(u_j))}} \qquad (25) \\
&= 2^{-k/2} \left(\int_{-\infty}^{\infty} \frac{dz}{\sqrt{1 + \cosh(z)}} \right)^k \qquad (26) \\
&= 2^{-k/2} \left(\sqrt{2}\pi \right)^k \\
&= \pi^k \qquad (27)
\end{aligned}$$

where in (25) we performed the linear change of variables $u = xw$. (26) is by Fubini's theorem. We state this remarkable result as a theorem.

Theorem 1 *When x is a square of full rank the information volume of the logistic regression model is independent of x and has value π^k where k is the dimension of the manifold.*

When $n > k$, the volume does depend on the design matrix of covariates x. The volume is still finite provided x is of full rank so that $\det x^T x > 0$ but I do not know of an exact formula for the volume.

2.3 An Approximate Lower Bound for $\log \det g(w)$

If needed, shuffle the sequence of x_i's so that (23) is written as the sum of $r \geq 1$ matrices of full rank k plus a reminder matrix A, i.e.,

$$g(w) = \frac{1}{2}\sum_{i=1}^{n} \frac{x_i^T x_i}{1+\cosh(x_i w)} = G_1 + G_2 + \ldots + G_r + A$$

Thus,

$$\begin{aligned}
\log\det g(w) &= \log\det r\left[\frac{1}{r}\sum_{j=1}^{r} G_j + \frac{1}{r}A\right] \\
&= k\log r + \log\det\left[\frac{1}{r}\sum_{j=1}^{r} G_j + \frac{1}{r}A\right] \\
&\approx \log\det\left(\frac{1}{r}\sum_{j=1}^{r} G_j\right) + k\log r \qquad (28) \\
&\geq \frac{1}{r}\sum_{j=1}^{r}\log\det G_j + k\log r \qquad (29)
\end{aligned}$$

where (28) follows from the fact that $\log\det$ is continuous and for $n >> k$ will have $r >> 1$ so that the Frobenius norm $||A/r||^2$ is small. (29) follows from the fact that $\log\det$ is log concave as a function of symmetric positive definite matrices. A simple proof of this well known fact is as follows: Suppose A and B are symmetric positive definite matrices, that $0 \leq \lambda \leq 1$ and that z is a column vector of the dimension of the matrices (say k) then,

$$\exp\left(-z^T[(1-\lambda)A + \lambda B]z\right) = \left[\exp(-z^T A z)\right]^{1-\lambda}\left[\exp(-z^T B z)\right]^{\lambda}$$

hence, integrating both sides over R^k and using Holder's inequality we obtain,

$$\begin{aligned}
\int \quad & \exp\left(-z^T[(1-\lambda)A + \lambda B]z\right)\, dz \leq \\
& \left(\int \exp(-z^T A z)\, dz\right)^{1-\lambda}\left(\int \exp(-z^T B z)\, dz\right)^{\lambda}
\end{aligned}$$

computing the Gaussian integrals we obtain,

$$\det\left((1-\lambda)A + \lambda B\right) \geq (\det A)^{1-\lambda}(\det B)^{\lambda}$$

and the result follows by taking logs. i.e., $\log\det$ is log concave as claimed.

Using the expression (24) for computing $\det G_j$ in (29) we finally get,

$$\log\det g(w) \geq \frac{-1}{r}\sum_{i=1}^{n}\log\left(1+\cosh(x_i w)\right) + C \qquad (30)$$

where C is independent of w. This bound is not very good. It could and should be improved upon. However, in some preliminary experiments it does seem to help.

2.4 Targets for Logistic Regression

The general new targets from the geometric theory of ignorance are given by (19). Let us rewrite them for the specific case of logistic regression where we will take $\pi(p) = 1$ and for simplicity $w_0 = 0$. The target expression to be minimized over $w \in R^k$ is,

$$\mathcal{L}_{\delta,\nu}(w) - \frac{1}{2}\log\det g(w) + \lambda\, w^T g(w) w \tag{31}$$

where,

$$\mathcal{L}_{\delta,\nu}(w) = \frac{1}{\nu}\log\left[1 + \frac{\beta\nu}{\delta(1-\delta)}\left(1 - \sum_{i=1}^{n}(\frac{1}{n})^{1-\delta} P_{w,x}^{\delta}(y_i)\right)\right] \tag{32}$$

and,

$$P_{w,x}^{\delta}(y_i) = \left[\theta_i^{y_i}(1-\theta_i)^{1-y_i}\right]^{\delta}$$

with θ_i as in (21). A systematic evaluation of (31) varying all the parameters needs to be tested on different data sets. This is in principle straight forward but it has not been done yet. The parameter ν in $[0,1]$ controls likelihood robustness, δ in $[0,1]$ controls prior robustness, $\beta > 0$ gives the equivalent number of virtual observations supporting the choice of prior and $\lambda > 0$ controls the radius of the L_2 ball where the search is done.

The case $\nu = 0$ makes the targets additive over the observations and we should expect the standard stochastic gradient descent to work as usual. The case $\nu = \delta = 0$ gives,

$$\begin{aligned} \mathcal{L}_{0,0} &= \frac{\beta}{n}\sum_{i=1}^{n}[y_i\log(\theta_i) + (1-y_i)\log(1-\theta_i)] \\ &= \frac{-\beta}{n}\sum_{i=1}^{n} I(\mathrm{Bin}(y_i) : \mathrm{Bin}(\theta_i)) + C \end{aligned} \tag{33}$$

which is the default standard for logistic regression, the so called "Binary Cross-Entropy" target in TensorFlow.

Using (33,32,30,22) we obtain a target in terms of parameters $a > 0$ and $b > 0$ controlling the relative weights of the three terms,

$$\sum_{i=1}^{n}\log P_{w,x}(y_i) + a\sum_{i=1}^{n}\log(1+\cosh(x_i w)) + b\sum_{i=1}^{n}\frac{(x_i w)^2}{1+\cosh(x_i w)} \tag{34}$$

we notice that this target can be easily defined in TensorFlow as a function of the observed vector y and the predicted vector y_{pred} since,

$$xw = \log\frac{y_{\mathrm{pred}}}{1-y_{\mathrm{pred}}}. \tag{35}$$

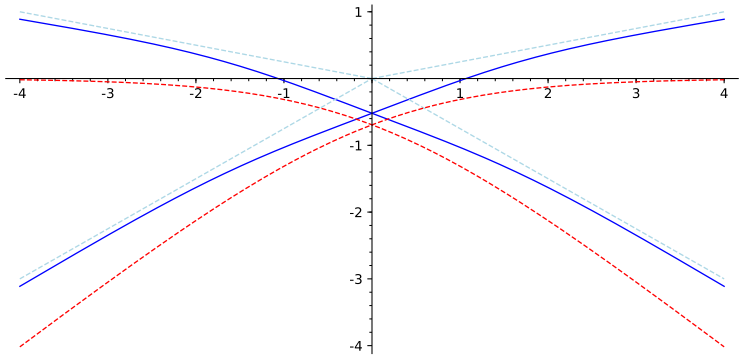


Figure 2: f_0 and f_1 with and without penalty

Let us rewrite (34) as,

$$L(w) = \sum_{i=1}^{n} f_{y_i}(x_i w)$$

where $f_1(z) = z + f_0(z)$. A second order Taylor series expansion about 0 gives,

$$f_0(z) = -(1-a)\log 2 - \frac{1}{2}z - \frac{1}{8}(1 - 2a - 4b)z^2 + o(z^2). \tag{36}$$

We also have that as $|z| \to \infty$,

$$f_0(z) \sim -(1-a)\, z \tag{37}$$

where in (37) we mean that the ratio of the two sides approaches 1 as $|z| \to \infty$. From (36) and (37) we see that f_0 (and similarly $f_1 = z + f_0$) is a quadratic around zero that quickly approaches a straight line as z moves away from zero.

Figure 2 shows f_0 and f_1 as solid (blue) curves, where $2a + 4b = 1$ and $a = 0.25$ so there is no quadratic term in (36). The dashed (red) lines converging to the x-axis show the plain likelihood (i.e., f_0, f_1 with $a = b = 0$) without the new penalties. Figure 2 also shows the linear asymptotes to the solid curves (dashed light blue).

Above and beyond the geometric interpretation for the new penalty terms, Fig 2 provides a possible explanation for why these penalties should work. Without the penalties, the plain likelihood approaches the x-axis assymptotically, making the derivative to approach zero. With the penalties, $f_0(z)$ and $f_1(z)$ approach a straight line with constant derivative. The penalties should work for the same reason that ReLu improves over the sigmoid. The signal from the vanishing derivatives when z is large gets lost and makes the backpropagation algorithm get stuck in the wrong place.

3 TensorFlow Preliminary Tests

We test the performance of some of the new targets for simple logistic regression using Keras with TensorFlow as the backend. This code is available as a Jupyter Colaboratory Notebook.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import wishart

import tensorflow as tf
import tensorflow.keras as K
from tensorflow.keras import backend as B

from numpy import einsum as Ein

import tensorflow.keras.layers as layers
from sklearn.metrics import r2_score


def simul_logist(w_true = np.array([-0.5,0.25,-0.1]),
                 mean_N = 3,
                 size_N = 200,
                 var_xi_df = 50,
                 seed = 123):

  np.random.seed(seed)
  k = len(w_true)
  var_xi_scale = np.identity(k)
  N = 1+np.random.poisson(lam=mean_N,size=size_N)
  var_cov = wishart.rvs(df=var_xi_df,scale=var_xi_scale,size=1)
  # np.corrcoef(var_cov) # to check the correlations

  x_unique = np.random.multivariate_normal(mean=np.zeros(k),
                                           cov=var_cov,size=size_N)
  x = np.repeat(x_unique,N,axis=0)
  theta_true_unique = 1/(1+np.exp(-np.dot(x_unique,w_true)))
  theta_true = np.repeat(theta_true_unique,N)
  y = np.random.binomial(1,theta_true)

  return {'x':x,'y':y,'N':N,
          'theta_true_unique':theta_true_unique,
          'x_unique':x_unique,
          'pars':(w_true,
                  mean_N,
                  size_N,
                  var_xi_df,
                  seed)
              }
```

The above code defines a general purpose data generator for simple logistic regression. To be able to better visualize the accuracy of the inferences we simulate $N[i]$ repeated values of each covariate feature vector x_i. The values $N[i]$ are generated as one plus a Poisson with expected value mean_N. There are size_N different x_i and they are gererated by sampling a multivariate gaussian

distribution centered at the origin with a covariance matrix taken as one sample of a Wishart distribution with unit scale and var_xi_df degrees of freedom. This allows control of the correlations among the dimensions of the covariate vector.

The following code shows how to implement four different targets suggested by the geometric theory of ignorance.

```
#################################################################
#
#    Vol prior and L2(manifold) penalty

eps = 1e-30
def gabloss(y_true,y_pred,glambda=0.5,gb=1.0): # Both penalties:
                                      Lik + a*Vol + b*L2
  bw = K.losses.binary_crossentropy(y_true,y_pred)
  vw = -B.log(y_pred*(1-y_pred))
  wgw = y_pred*(1-y_pred)*B.square(B.log(eps+y_pred) - B.log(eps+1-
                                       y_pred))
  a = glambda/2.
  b = gb*(1-glambda)/4.
  return bw + a*B.mean(vw,axis=-1) + b*B.mean(wgw,axis=-1)

def gloss_vol(y_true,y_pred,glambda=1.): # prior = 0
  bw = K.losses.binary_crossentropy(y_true,y_pred)
  vw = B.log(y_pred*(1-y_pred))
  return bw - glambda*0.5*B.mean(vw,axis=-1)



def gloss(y_true,y_pred,glambda=0.001): # prior = 1 (L2(M))
  bw = K.losses.binary_crossentropy(y_true,y_pred)
  wgw = y_pred*(1-y_pred)*B.square(B.log(eps+y_pred) - B.log(eps+1-
                                       y_pred))
  return bw + glambda*B.mean(wgw,axis=-1)

def glogcosh(y_true,y_pred,glambda=0.1): # prior = 2
  bw = K.losses.binary_crossentropy(y_true,y_pred)
  gw = K.losses.logcosh(y_true,y_true+B.log(y_pred/(1-y_pred)))
  return bw+glambda*gw

#############################################################
```

The first function (*gabloss*) implements (34) by using (35). Notice that when the parameter $gb = 1.0$, the weights are such that $2a + 4b = 1$ and thus, there is no quadratic term in (36). That appears to help when the dimension k is small. The examples presented here should be taken as preliminary. No attempt has been made at optimizing hyperparameters yet. However, the simple examples show that the new terms do improve on the plain likelihood consistently in low dimensions without much trial and error. Here are fairly typical examples implemented by using the following model:

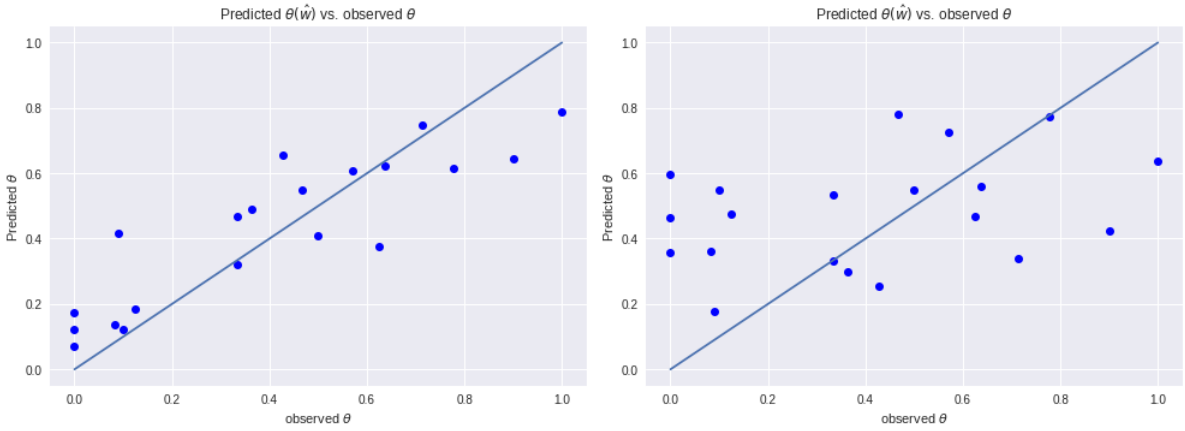
```
#############################################################

EPOCHS = 30
patience = 5
```

```
early_stop = K.callbacks.EarlyStopping(monitor='val_loss', patience
                                       =patience)

def logistic_model(xvals=x_train,learning_rate=0.001,prior=None,
                                       glambda=0.5,gb=1.0):
    n,k = x_train.shape
    model = K.Sequential()
    model.add(layers.Dense(1,input_shape=(k,),
                           activation='sigmoid',
                           use_bias=False))
    optimizer = tf.train.RMSPropOptimizer(learning_rate)
    if prior == None:
      model.compile(optimizer=optimizer,
                    loss='binary_crossentropy',
                    metrics=['accuracy'])
    elif prior == 0:
      model.compile(optimizer=optimizer,
                    loss=gloss_vol,
                    metrics=['accuracy'])

    elif prior == 1:
      model.compile(optimizer=optimizer,
                    loss=gloss,
                    metrics=['accuracy'])

    elif prior == 2: # glogcosh
      model.compile(optimizer=optimizer,
                    loss=glogcosh,
                    metrics=['accuracy'])

    elif prior == 3: # gabloss: Lik+a*vol+b*L2
      model.compile(optimizer=optimizer,
                   loss=gabloss,
                   metrics=['accuracy'])

    model.build((None,xvals.shape[1]))
    print(model.summary())
    return model
#####################################################################
```

Dim $k = 3$

Twenty different θ_i's with an average of 10 observations per θ_i where simulated with the following code:

```
#############################################################
#
#    Get data:
#

sim_train = simul_logist(w_true=[-0.5,0.25,-0.1],
                         var_xi_df = 3,
                         mean_N=10,
                         size_N=20,
                         seed=123)
```



(a) Prior: $R^2 = 0.751$ (b) No Prior: $R^2 = 0.009$

Figure 3: Dim $k = 3$. With and without prior.

```
x_train,y_train,N_train = sim_train['x'],sim_train['y'],sim_train['
                                    N']

pars = sim_train['pars'][:-1]
sim_test = simul_logist(*pars,seed=2)
x_test,y_test,N_test = sim_test['x'],sim_test['y'],sim_test['N']
############################################################
```

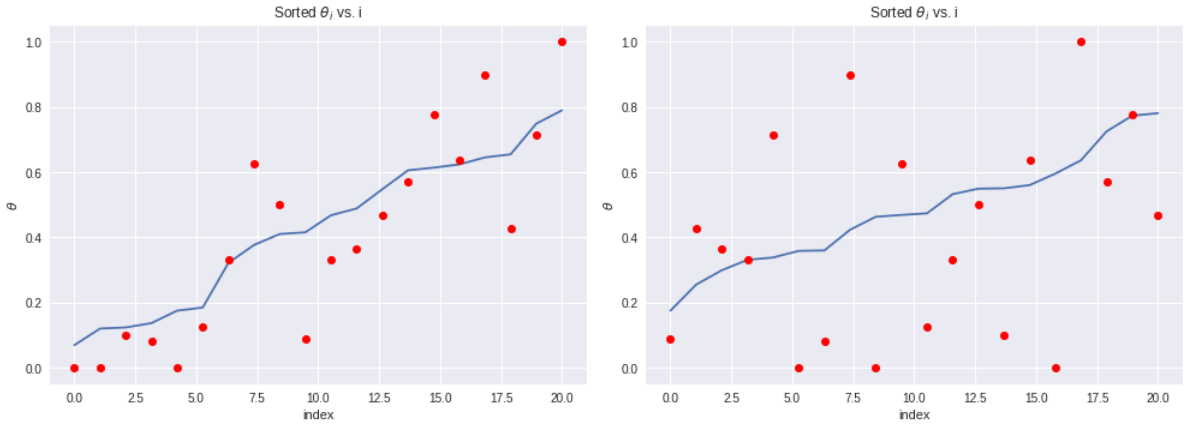
With the new target as:

```
g_prior3 = logistic_model(prior=3,glambda=0.25,gb=0.9)
gh3,gobs3,gpred3 = Fit(g_prior3,sim_train,sim_test)
g_prior3.get_weights()
```

we get a coefficient of determination of $R^2 = 0.751$ showing that 75.1% of the variance of the observed θ_i's is explained by the (logistic) regression on the x_i's. Compare this to an $R^2 = 0.009$ obtained on the same data with the plain likelihood without the new penalties.

Dim $k = 10$

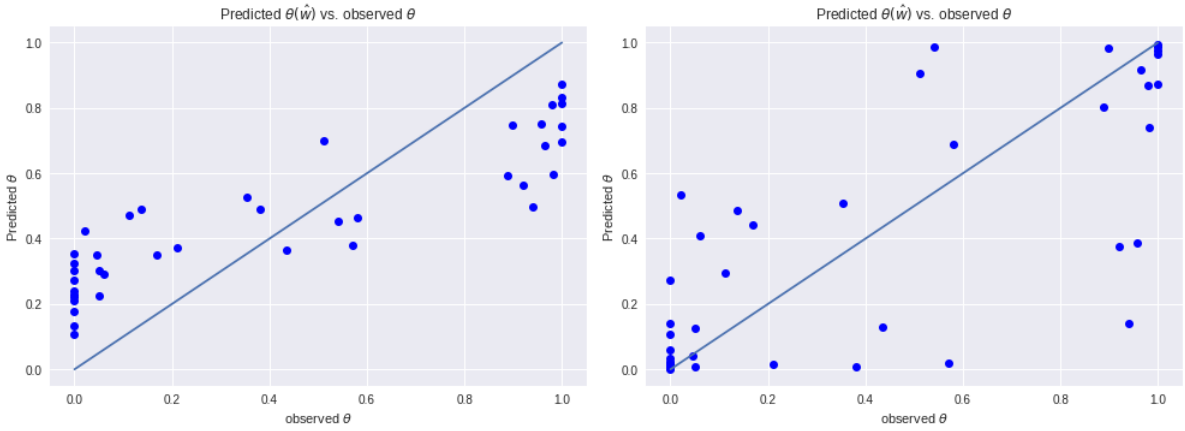
With 40 different θ_i's with an average of 10 observations per θ_i, we get $R^2 = 0.638$ with the prior but only $R^2 = 0.553$ without the prior on the same data.



(a) Prior: $R^2 = 0.751$

(b) No Prior: $R^2 = 0.009$

Figure 4: Dim $k = 3$. Sorted θ_i's.



(a) Prior: $R^2 = 0.638$ (b) No Prior: $R^2 = 0.553$

Figure 5: Dim $k = 10$. With and without prior.

One should expect the gains from using the correct geometry of logistic regression to increase with the dimension of the manifold. However, that is not what these preliminary simulations show. What is observed is that in higher dimensions, it becomes more difficult to make the plain backpropagation algorithm to converge with and without the prior. We have just scratched the surface of the available new targets and a systematic evaluation with hyperparameter tuning will be needed to bring the geometric theory of ignorance to its full practical fruition.

3.1 Logistic Regression is Flat!

Using (23) and the summation convention, write the components of the metric as,

$$g_{ij} = x_{il} d^{ll} x_{lj} \tag{38}$$

where the entries of the diagonal matrix are,

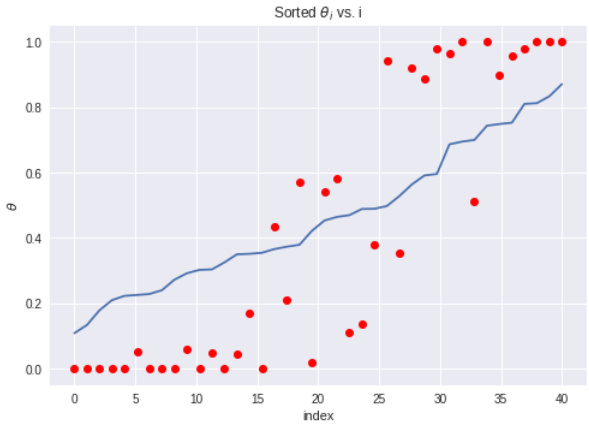
$$d^{ll} = \frac{1/2}{1 + \cosh(x_l w)}. \tag{39}$$

Recall that the components Γ_{ijk} of the Levi-Civita metric connection are,

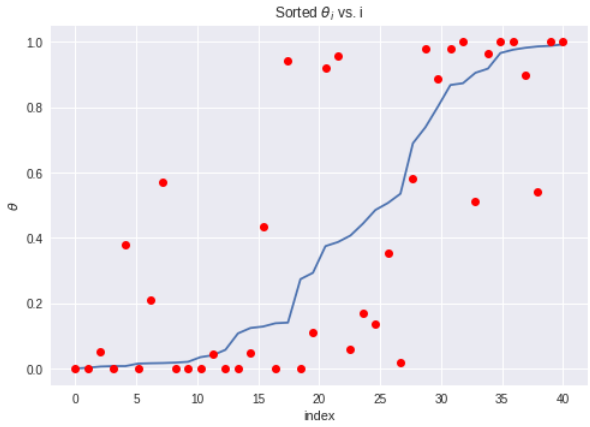
$$2\Gamma_{ijk} = \partial_i g_{jk} + \partial_j g_{ki} - \partial_k g_{ij}. \tag{40}$$

Thus,

$$\Gamma_{ijk} = a^l x_{li} x_{lj} x_{lk} \tag{41}$$



(a) Prior: $R^2 = 0.638$



(b) No Prior: $R^2 = 0.553$

Figure 6: Dim $k = 10$. Sorted θ_i's.

where,

$$a^l = \frac{-\sinh(x_l w}{4(1+\cosh(x_l w))^2}. \tag{42}$$

When $n = k$ with x non singular, we can easily formally write the entries of g^{-1},

$$g^{ij} = x^{il} d_{ll} x^{lj} \tag{43}$$

that are needed to raise the connection indices,

$$\Gamma^i_{jk} = g^{ir}\Gamma_{rjk} \tag{44}$$

but when $n > k$ there is no simple formula for the g^{ij}. Recall that the Ricci curvature scalar R is obtained by contracting the indices of the Riemann curvature tensor R^i_{jkl}, (i.e. taking traces), where,

$$R^i_{jkl} = \partial_k \Gamma^i_{lj} - \partial_l \Gamma^i_{kj} + \Gamma^i_{km}\Gamma^m_{lj} - \Gamma^i_{lm}\Gamma^m_{kj} \tag{45}$$

Sage to the Rescue:

With the help of ig.sage and the the simplificator Sim.sage we get:

```
load("http://omega.albany.edu:8008/sage/sim.sage")
load("http://omega.albany.edu:8008/sage/ig/ig.sage")

def g_logistic_metric(k,n):
    assert n >= k, "n=%d must be at least k = %d " % (n,k)
    for j in range(k): var('w%d'%j)
    w = [var('w%d'%j) for j in range(k) ]
```

```
    x = matrix(SR,n,k)
    for i in range(n):
        for j in range(k):
            x[i,j] = var('x%d%d'%(i,j))
    for i in range(n):
        vars()['x%d'%i] = x[i,:]
    xw = x*matrix(k,1,w)
    diag_vec = vector([1/(1+cosh(xw[i,0])) for i in range(n)])
    Dw = diagonal_matrix(diag_vec)
    gmat = x.transpose()*Dw*x
    g = Metric(coords=w,gmat=gmat)
    return g

g = g_logistic_metric(2,2)
g.get_all()
g.get_Rscalar()
# uncomment the next line to watch the
# largest equation you have ever encountered...

# g.R

Sim(g.R)
```

It simplifies to zero!

Is there a hole?

The logistic regression manifold seems to be flat and of finite volume. If it were periodic it would be like a k-dimensional Torus. There is a boundary though. The boundary is obtained when some of the $\theta_i \in \{0, 1\}$, just like in bitnets.pdf. If we were able to glue the boundaries in the correct way it would make the manifold periodic and there will be a hole.

4 References and Name Dropping

Where does this come from?
Here is a biased personal attempt to explain my path here. First and foremost, I am in debt to Shun-ichi Amari, Edwin Jaynes and Herbert Robbins, the creators of the three main pillars of the geometric theory of ignorance; Amari for Information Geometry, Jaynes for Maximum Entropy, and Herbert Robbins for Empirical Bayes. [Am],[Ro3],[L-S].

A little historical tale: The 1980's and 1990's saw a revival of the ideas of Laplace and Jeffreys in sync with the availability of inexpensive computing and the use of Monte Carlo methods. The sabbatical year that Jaynes spent at St. Johns College in Cambridge ignited a new Bayesian evangelism propelled by Gull, Skilling and later McKay as the main apostles. We are still feeling the effects of their influence today. In the opposite camp at that time with Berkeley as the epicenter, the mathematically minded statisticians were unable to swallow the use of out-of-the-blue prior distributions on arbitrary parameterizations and kept pushing for variants of maximum likelihood. In the mean time, neural nets

came in and went out of fashion a couple of times, justifying the belief that the best thing about neural nets was the label: *Neural Nets*. All that changed a few years ago when the one trick pony of automatic differentiation, combined with better hardware and more data, brought logistic regression to a new level and started the ongoing deep learning revolution. The geometric theory of ignorance promises a new way forward - Bayesianism without cheating that can be applied to deep networks that actually work in practice.

Tong Zhang:

In a 2003 NIPS paper Zhang03.pdf, Tong Zhang showed that the direct δ-posteriors with $\delta < 1$ are robust against incorrect assignments of prior mass away from the true distribution t and that this is not always true for standard bayesian inference, i.e. this is not true when $\delta = 1$. Compare this with Theorem2 in Rodriguez93.pdf.

My papers can be found on my website.

References

[Am] Amari, S.-i.: Differential-Geometrical Methods in Statistics. (Lecture Notes in Statistics, Vol. 28). Springer-Verlag 1985

[I-H] Ibragimov, I., Has'minskii, R.: Statistical estimation. (Applications of Mathematics, Vol. 16). Springer-Verlag 1981

[Je] Jeffreys, H.: Theory of Probability. Oxford University Press 1939

[Ku] Kullback, S.: Information Theory and Statistics. John Wiley, New York 1959

[L-S] Lai, T., Siegmund, D. (eds.): Herbert Robbins Selected Papers. Springer-Verlag 1985

[Ro1] Rodríguez, C.: The metrics induced by the kullback number. In: Skilling, J. (ed.), Maximum Entropy and Bayesian Methods. Kluwer Academic Publishers 1989

[Ro2] Rodríguez, C.: Objective bayesianism and geometry. In: Fougere, P. F. F. (ed.), Maximum Entropy and Bayesian Methods. Kluwer Academic Publishers 1990

[Ro3] Rosenkrantz, R. (ed.): E.T. Jaynes: Papers on Probability, Statistics and Statistical Physics. (Vol. 158). Synthese Library 1983

From Chentsov's obituary:

[Va] Vladimir I Arnol'd et al: Chentsov's obituary. Russ. Math. Surv. 48 161, 1993