Rigorous results in statistical physics of simple machine learning models

C. Gerbelot, A. Abbara and F. Krzakala

ICTP Seminar

17 April 2020

Introduction

Machine learning : black box magic

Learning from data to predict, classify, compute ... Recent success ($\sim$ 2000–ongoing)

Deep learning revolution (2018 Turing Price)

Practical success, limited theoretical understanding

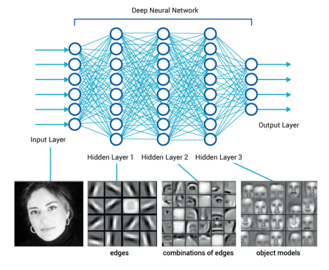


Figure 1: Deep network

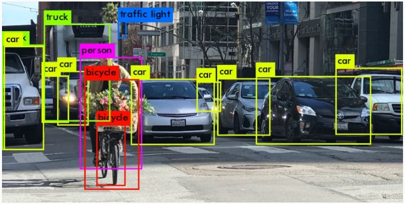


Figure 2: Object recognition

Opening the black box

Go back to a single neuron : the **perceptron** (Rosenblatt, 1958)

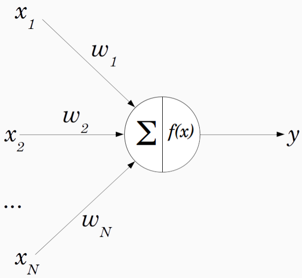


Figure 3: A single neuron

Takes input vector $\mathbf{x} \in \mathbb{R}^N$ and generates

$$y = f(\sum_{i=1}^{N} w_i x_i)$$

Statistical physics of learning

Perceptron extensively studied in statistical physics since the 80's

Optimal storage properties of neural network models

E Gardner† and B Derrida‡

† Department of Physics, Edinburgh University, Mayfield Road, Edinburgh, EH9 3JZ, UK
‡ Service de Physique Theorique, CEN Saclay, F 91191 Gif sur Yvette, France

Received 29 May 1987

Abstract. We calculate the number, $p = \alpha N$ of random N-bit patterns that an optimal neural network can store allowing a given fraction f of bit errors and with the condition that each right bit is stabilised by a local field at least equal to a parameter K. For each value of α and K, there is a minimum fraction $f_{\min}$ of wrong bits. We find a critical line, $\alpha_c(K)$ with $\alpha_c(0) = 2$. The minimum fraction of wrong bits vanishes for $\alpha < \alpha_c(K)$ and increases from zero for $\alpha > \alpha_c(K)$. The calculations are done using a saddle-point method and the order parameters at the saddle point are assumed to be replica symmetric. This solution is locally stable in a finite region of the K,α plane including the line, $\alpha_c(K)$ but there is a line above which the solution becomes unstable and replica symmetry must be broken.

VOLUME 58, NUMBER 9 PHYSICAL REVIEW LETTERS 2 MARCH 1987

Learning of Correlated Patterns in Spin-Glass Networks by Local Learning Rules

Sigurd Diederich and Manfred Opper
Institut für Theoretische Physik, Universität Giessen, D-6300 Giessen, Federal Republic of Germany
(Received 18 November 1986)

Two simple storing prescriptions are presented for neural network models of N two-state neurons. These rules are local and allow the embedding of correlated patterns without errors in a network of spin-glass type. Starting from an arbitrary configuration of synaptic bonds, up to N patterns can be stored by successive modification of the synaptic efficacies. Proofs for the convergence are given. Extensions of these rules are possible.

PACS numbers: 87.30.Gy, 06.50.Mk, 75.10.Hk, 89.70.+c

A few names : H. Sompolinsky (physics, neuroscience), M. Opper (physics computer science), E.Gardner & B.Derrida (physics)

Outline of the talk

Introduction

Statistical physics approach

Penalized linear regression

Rigorous mathematical statement

Extension of the result

Introduction

Introduction to supervised learning

Supervised learning : learning from labeled examples

- Input space $\mathcal{A}$ (ex. $\mathbb{R}^N$)
- Output space $\mathcal{Y}$ (ex. $\{-1, 1\}$ for classification, $\mathbb{R}$ for regression)
- Training set $\mathcal{S}_M = (\mathbf{x}_i, y_i)_{i=1,\ldots,M}$ of (input,output) pairs.

Goal is to estimate a function h $: \mathcal{A} \to \mathcal{Y}$ **to predict outputs of future inputs**

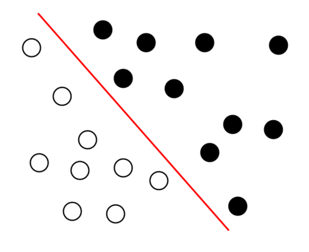


Figure 4: Classification

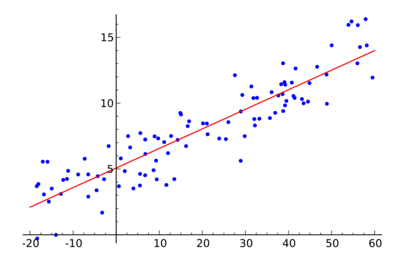


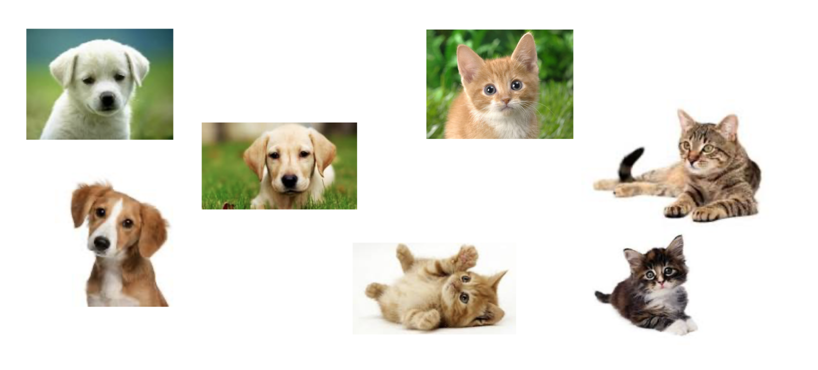
Figure 5: Linear regression

Example : classification

Task: recognize if image is a dog or a cat

$\mathcal{A}$ = set of images (pixels in $\mathbb{R}^N$)

$\mathcal{Y}$ = {cat,dog} = {-1,1}

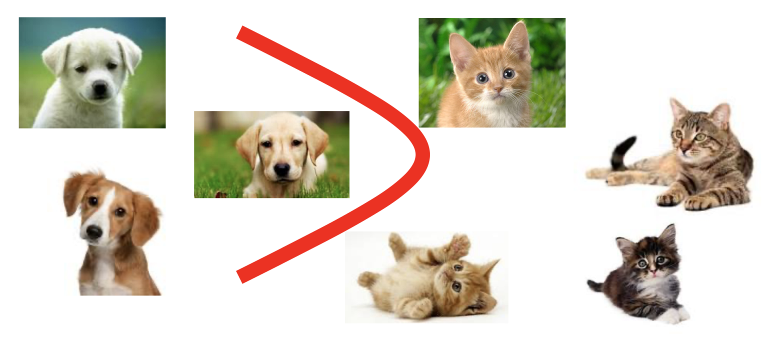


Example : classification

Task: recognize if image is a dog or a cat

$\mathcal{A}$ = set of images (pixels in $\mathbb{R}^N$)

$\mathcal{Y} = \{\text{cat,dog}\} = \{\text{-1,1}\}$



Empirical risk minimization (ERM)

Choose a form for $h(\mathbf{a})$ on a functional space $\mathcal{H}$, and choose a loss function $\mathcal{L}$ to quantify the error made on a given prediction.

Goal

$$h^* \in \arg\min_{\mathcal{H}} \mathbb{E}(\mathcal{L}(h(\mathbf{a}), y) \tag{1}$$

Approximate with empirical risk on a given training set:

$$\frac{1}{M} \sum_{i=1}^{M} \mathcal{L}(h(\mathbf{a}_i), y_i) \tag{2}$$

Add regularisation to prevent overfitting (ML version of constraints):

$$\arg\min_{\mathcal{H}} \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}(h(\mathbf{a}_i), y_i) + \lambda \|h\|_{\mathcal{H}} \tag{3}$$

Example : linear regression

Take linear model and square loss. **x parameter vector**.

$$h(\mathbf{a}) = \mathbf{x}^T\mathbf{a} \quad , \mathcal{L}(h(\mathbf{a}_i), y_i) = \|h(\mathbf{a}_i) - y_i\|_2^2$$

Here $\mathcal{H}$ is just the set of vectors $\theta \in \mathbb{R}^N$, choose any classical Euclidian norm for regularization :

$$\arg\min_{\mathbb{R}^N} \frac{1}{M}\sum_{i=1}^{M}\|\mathbf{x}^T\mathbf{a}_i - y_i\|_2^2 + \lambda\|\mathbf{x}\|_{1,2,\ldots} \tag{4}$$

Simplest is ℓ_2, Ridge regression:

$$\theta^* = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^T\mathbf{y} \tag{5}$$

Probabilistic formulation

Consider a Bayesian approach :
$P(\mathbf{x})$ prior distribution
$P(\mathbf{y}|\mathbf{A},\mathbf{x})$ likelihood distribution
$P(\mathbf{x}|\mathbf{y})$ posterior probability distribution.

From Bayes law

$$P(\mathbf{x}|\mathbf{y}) = \frac{P(\mathbf{y}|\mathbf{A},\mathbf{x})P(\mathbf{x})}{P(\mathbf{y})} \propto e^{\log P(\mathbf{y}|\mathbf{A},\mathbf{x}) + \log P(\mathbf{x})} \tag{6}$$

Maximum a posteriori estimation

$$\mathbf{x}^* \in \arg\max \{\log P(\mathbf{y}|\mathbf{A},\mathbf{x}) + \log P(\mathbf{x})\} \tag{7}$$

Equivalent to ERM with loss $-\log P(\mathbf{y}|\mathbf{A},\mathbf{x})$ and regularization $-\log p(\mathbf{x})$

Statistical physics approach

Statistical physics formulation

$$\mathbf{y} = \mathbf{A}\mathbf{x_0} + \mathbf{w} \tag{8}$$

$$\mathbf{y} \in \mathbb{R}^M,\ \mathbf{A} \in \mathbb{R}^{M\times N},\ \mathbf{x_0} \in \mathbf{R}^N \sim p(\mathbf{x_0})$$
$$\mathbf{w} \in \mathbb{R}^m \sim \mathcal{N}(0, \Delta_0) \text{ is a Gaussian noise.}$$

We know $\mathbf{y}$ and would like to recover $\mathbf{x_0}$.
$\alpha = \frac{M}{N}$ gives the ratio measurements/unknowns.
In this set up, the likelihood is

$$P(\mathbf{y}|\mathbf{A}, \mathbf{x}) = \frac{1}{\sqrt{2\pi\Delta_0}^M} e^{-\frac{(\mathbf{y}-\mathbf{A}\mathbf{x})^2}{2\Delta_0}} = \prod_{\mu=1}^{M} \frac{1}{\sqrt{2\pi\Delta}} e^{-\frac{(y_\mu - \sum_{i=1}^N A_{\mu i} x_i)^2}{2\Delta_0}}. \tag{9}$$

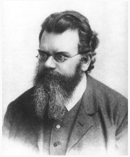
The posterior distribution is

$$P(\mathbf{x}|\mathbf{y}) = \frac{1}{\mathcal{Z}(\mathbf{y})} e^{\sum_{i=1}^{N} \log P(x_i) - \frac{1}{2\Delta_0} \sum_{\mu=1}^{M} (y_\mu - \sum_{i=1}^{N} A_{\mu i} x_i)^2} \tag{10}$$

$$\propto e^{-\beta \mathcal{H}(\mathbf{x})} \tag{11}$$

$\frac{1}{2\Delta_0}$ plays the role of the inverse temperature β, $A_{\mu i}$ is a disordered interaction.

$\rightarrow$ N interacting particles (x_i) in a long range mean-field potential.



Disordered systems and spin glass theory!

Mézard, Parisi, Virasoro: *Spin-glass theory and beyond* 1986

Mézard, Montanari: *Information, physics and computation* 2009

From the **Hamiltonian**

$$\mathcal{H}(x) = -\sum_{i=1}^{N} \log[p(x_i)] + \sum_{\mu=1}^{M} \frac{(y_\mu - \sum_{i=1}^{N} A_{\mu i} x_i)^2}{2\Delta_0}. \tag{12}$$

we can define interesting quantities, such $\frac{1}{N} \log \mathcal{Z}$ the **free energy**.

We focus on the thermodynamic limit $M, N \to \infty$ with $\alpha = \frac{M}{N}$ fixed.

In some cases, the free energy can be computed using
the statistical physics **replica method**.

Statistical physics approach

The replica method

Crash course in replica method

We are interested in $\mathbb{E}_{\mathbf{A},\mathbf{x}_0,\mathbf{w}}(\log \mathcal{Z})$. We will use the replica trick:

$$\lim_{N\to\infty} \frac{1}{N}\mathbb{E}_{\mathbf{A},\mathbf{x}_0,\mathbf{w}}(\log \mathcal{Z}) = \lim_{N\to\infty} \frac{1}{N} \lim_{r\to 0} \frac{\mathbb{E}_{\mathbf{A},\mathbf{x}_0,\mathbf{w}}(\mathcal{Z}^r) - 1}{r}. \tag{13}$$

To compute $\mathcal{Z}^r$, we introduce r copies of the system that we call *replicas* and denote with subscript $a \in \{1, ..., r\}$.

$$\mathbb{E}_{\mathbf{A},\mathbf{x}_0,\mathbf{w}}(\mathcal{Z}^r) = \int \prod_{i=1}^{N} \prod_{a=1}^{r} dx_i^a p(x_i^a)$$
$$\prod_{\mu=1}^{M} \mathbb{E}_{\mathbf{A},\mathbf{x}_0,\mathbf{w}} \frac{1}{\sqrt{2\pi\Delta_0}} e^{-\frac{1}{2\Delta_0} \sum_{a=1}^{r} \left(\sum_{i=1}^{N} A_{\mu i} x_{0,i} + w_\mu - \sum_{i=1}^{N} A_{\mu i} x_i^a\right)^2}. \tag{14}$$

Matrix averaging

A difficult term is

$$\mathbb{E}_{\boldsymbol{F},\boldsymbol{w}}\left[\exp\left\{-\frac{1}{2\Delta_0}\sum_{a=1}^{r}\left[\boldsymbol{F}(\mathbf{x^0}-\boldsymbol{x}^a)+\boldsymbol{w}\right]^T\left[\boldsymbol{F}(\mathbf{x^0}-\boldsymbol{x}^a)+\boldsymbol{w}\right]\right\}\right]. \quad (15)$$

- If **A** is Gaussian i.i.d., everything is Gaussian $\rightarrow$ straightforward averaging [Krzakala et. al. 2012]
- **A** is *rotationally invariant* if its singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\mathbf{T}$ includes orthogonal matrices **U** and **V** that are uniformly sampled from the orthogonal ensemble.

If **A** is rotationally invariant, there is also a way to perform this average [Kabashima et. al. 2014]

Order parameters

In both cases the integral can be written over the following *order parameters*:

$$m^a = \frac{1}{N}\sum_{i=1}^{N} x_i^a x_{0,i} \quad a = 1, 2, \dots r \tag{16}$$

$$Q^n = \frac{1}{N}\sum_{i=1}^{N} (x_i^a)^2 \quad a = 1, 2, \dots r \tag{17}$$

$$q^{ab} = \frac{1}{N}\sum_{i=1}^{N} x_i^a x_i^b \quad a = 1, 2, \dots r. \tag{18}$$

They describe physical quantities: norms and overlaps between vectors. Doing a change of variables, we have an integral on those parameters + a set of conjugate parameters, instead of the $\mathbf{x}^a$.

Replica symmetric ansatz

We need to assume an ansatz on order parameters. The easiest one is *replica symmetry*:
we assume all replicas are "equivalent" and

$$q^{ab} = q \;\forall a \neq b$$
$$m^a = m \text{ and } Q^a = Q \;\forall a.$$

The integral decouples on replicas and becomes

$$\mathbb{E}_{\mathbf{A},\mathbf{x}_0,\mathbf{w}}(\mathcal{Z}^r) = \int dQ\, d\hat{Q}\, dq\, d\hat{q}\, dm\, d\hat{m}\, e^{rN\Phi(Q,q,m,\hat{Q},\hat{q},\hat{m})}. \tag{19}$$

Recall

$$\lim_{N\to\infty} \frac{1}{N}\mathbb{E}_{\mathbf{A},\mathbf{x}_0,\mathbf{w}}(\log \mathcal{Z}) = \lim_{N\to\infty} \frac{1}{N} \lim_{r\to 0} \frac{\mathbb{E}_{\mathbf{A},\mathbf{x}_0,\mathbf{w}}(\mathcal{Z}^r) - 1}{r}. \tag{20}$$

We assume that we can take the limit $N \to \infty$ first. Using the saddle-point method, the integral concentrates around extremal value of Φ, that we call Φ^*.

$$\lim_{N\to\infty} \mathbb{E}_{\mathbf{A},\mathbf{x}_0,\mathbf{w}}(\mathcal{Z}^r) = e^{rN\Phi^*}. \tag{21}$$

$\Phi^*(Q^*, q^*, m^*, \hat{Q}^*, \hat{q}^*, \hat{m}^*)$ is the *free energy* of the system. The saddle-point is described by a set of 6 equations.

Penalized linear regression

Position of the problem

Penalized linear regression

We are interested in the estimator

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^N} \left\{ \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + f(\mathbf{x}) \right\} \tag{22}$$

$$\begin{aligned} \text{where} \quad \mathbf{y} &= \mathbf{A}\mathbf{x}_0 + \mathbf{w} \\ \mathbf{w} \quad \text{i.i.d} \quad &\sim \mathcal{N}(0, \Delta_0) \\ \mathbf{x}_0 &\sim p_{x_0} \end{aligned} \tag{23}$$

$\mathbf{A} \in \mathbb{R}^{M\times N}$ rotationally invariant ($\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ where $\mathbf{U}, \mathbf{V}$ Haar distributed)

f is a convex function

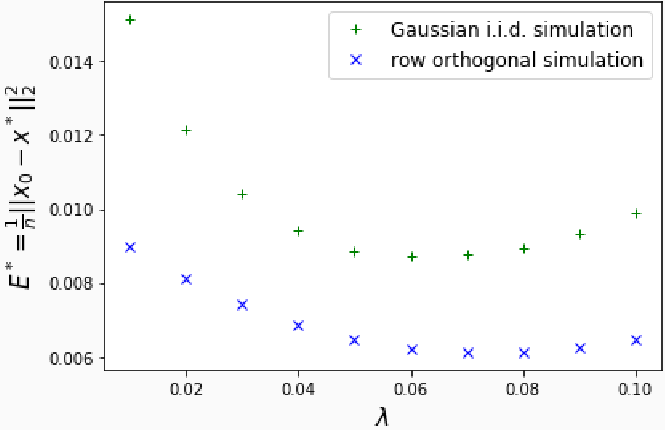
Examples : Ridge, Lasso, Logarithmic Barrier, ...

We want to characterize the mean squared error with respect to the teacher vector :

$$MSE = \mathbb{E}\left[\|\mathbf{x}_0 - \mathbf{x}^*\|_2^2\right]. \tag{24}$$

Simulated result

We can estimate $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\{\frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + f(\mathbf{x})\right\}$ using a convex optimization algorithm. For instance we take $f(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$ and compute the mean squared error, for different values of λ and different types of matrices $\mathbf{A}$ of "small" size (200*100).



Penalized linear regression

Heuristic result

Replica knows it all!

We can analyze this problem with the replica method, using a "prior distribution"

$$p(\mathbf{x}) \propto e^{-f(\mathbf{x})/\Delta_0} \tag{25}$$

inside

$$\mathbb{E}_{\mathbf{A},\mathbf{x}_0,\mathbf{w}}(\mathcal{Z}^r) = \int \prod_{i=1}^{N} \prod_{a=1}^{r} dx_i^a p(x_i^a)$$
$$\prod_{\mu=1}^{M} \mathbb{E}_{\mathbf{A},\mathbf{x}_0,\mathbf{w}} \frac{1}{\sqrt{2\pi\Delta_0}} e^{-\frac{1}{2\Delta_0}\sum_{a=1}^{r}\left(\sum_{i=1}^{N} A_{\mu i} x_{0,i} + w_\mu - \sum_{i=1}^{N} A_{\mu i} x_i^a\right)^2}. \tag{26}$$

The obtained free energy will describe the properties of the estimator $\mathbf{x}^*$.

Notice that the mean squared error

$$\mathbb{E}\left[\|\mathbf{x}_0 - \mathbf{x}^*\|_2^2\right] = \mathbb{E}[\|\mathbf{x_0}\|_2^2] + \mathbb{E}[\|\mathbf{x}^*\|_2^2] - 2\mathbb{E}[\mathbf{x_O} \cdot \mathbf{x}^*] \tag{27}$$

$$= \mathbb{E}[\|\mathbf{x_0}\|_2^2] + q^* - 2m^* \tag{28}$$

can be written in terms of the previously defined order parameters q^*, m^*, and $\mathbb{E}[\|\mathbf{x_0}\|_2^2]$.

Taking the replica saddle-point equations on parameters $Q^*, q^*, m^*, \hat{Q}^*, \hat{q}^*, \hat{m}^* \rightarrow$ we reduce them to only two parameters E^*, V^*.

The replica saddle-point equations describe properties of the estimator $\mathbf{x}^*$, by giving equations about its mean squared error, and variance.

Replica result

The average mean squared error $\mathrm{MSE} = \frac{1}{N}\mathbb{E}\left[\|\mathbf{x}_0 - \mathbf{x}^*\|_2^2\right]$ is given by the fixed point E^* of the equations:

$$V^* = \mathbb{E}\left[\frac{1}{\mathcal{R}_{\mathbf{C}}(-V^*)}\mathrm{Prox}'_{f/\mathcal{R}_{\mathbf{C}}(-V^*)}\left(x_0 + \frac{z}{\mathcal{R}_{\mathbf{C}}\left(-V^*\right)}\sqrt{\left(E^* - \Delta_0 V^*\right)\mathcal{R}'_{\mathbf{C}}\left(-V^*\right) + \Delta_0 \mathcal{R}_{\mathbf{C}}\left(-V^*\right)}\right)\right] \tag{29a}$$

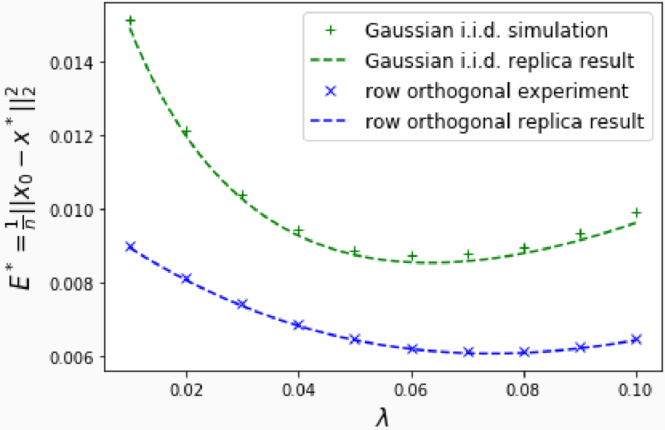
$$E^* = \mathbb{E}\left[\left\{\mathrm{Prox}_{f/\mathcal{R}_{\mathbf{C}}(-V^*)}\left(x_0 + \frac{z}{\mathcal{R}_{\mathbf{C}}\left(-V^*\right)}\sqrt{\left(E^* - \Delta_0 V^*\right)\mathcal{R}'_{\mathbf{C}}\left(-V^*\right) + \Delta_0 \mathcal{R}_{\mathbf{C}}\left(-V^*\right)}\right) - x_0\right\}^2\right], \tag{29b}$$

where $\mathbf{C} = \mathbf{A}^T\mathbf{A}$, $\mathcal{R}_{\mathbf{C}}$ is the R-transform with respect to the spectral distribution of $\mathbf{A}^T\mathbf{A}$, and expectations are over $z \sim \mathcal{N}(0,1)$ and $x_0 \sim p_{x_0}$. Prox is the proximal operator defined as:

$$\forall \gamma \in \mathbb{R}^+, x, y \in \mathbb{R} \quad \mathrm{Prox}_{\gamma f}(y) \equiv \arg\min_x \left\{f(x) + \frac{1}{2\gamma}(x-y)^2\right\}. \tag{30}$$

Replica result vs Simulation

From the replica result, we can numerically solve the equations on E^*, V^* in the case of $f(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$.



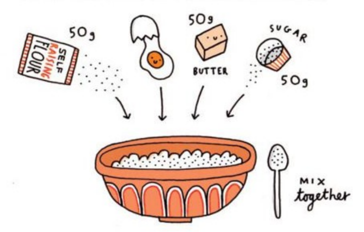
However, the replica result is non-rigorous: several steps are not justified! Inverting the limits, taking $r \to 0$...

The replica result is rigorously proven in some cases, when $\mathbf{A}$ is a Gaussian i.i.d. matrix [Guerra, Talagrand, Barbier]

We would now like to prove the previous replica result, for $\mathbf{A}$ rotationally invariant, with a convex penalty function!

Rigorous mathematical statement

Proof recipe



We will need several ingredients :

- An algorithm that tries to find the estimator $\mathbf{x}^*$, knowing $\mathbf{A}$, $\mathbf{y}$ and f.
- An analytic understanding of this algorithm, that gives us the mean squared error of the estimator (no easy task!)
- A converging sequence of the algorithm.

Then the estimator $\mathbf{x}^*$ would be described by the properties of the fixed point of the algorithm.

Vector Approximate Message Passing...

We will use *message-passing algorithms*: a class of algorithms which have been developed independently in machine learning, and statistical physics!

One of them, Vector approximate message-passing (VAMP) [Rangan et. al. 2019], does exactly what we need, for rotationally invariant matrices.

Choose initial A_{10} and $\mathbf{B}_{10}$

$$\hat{\mathbf{x}}_{1k} = \mathrm{Prox}_{\frac{1}{A_{1k}} f}\left(\frac{\mathbf{B}_{1k}}{A_{1k}}\right) \qquad \hat{\mathbf{x}}_{2k} = (\mathbf{A}^T\mathbf{A} + A_{2k} Id)^{-1}(\mathbf{A}^T y + \mathbf{B}_{2k})$$

$$V_{1k} = \frac{\langle \mathrm{Prox}'_{\frac{1}{A_{1k}} f}\rangle}{A_{1k}} \qquad V_{2k} = \frac{1}{N}\mathrm{Tr}\left[(\mathbf{A}^T\mathbf{A} + A_{2k} Id)^{-1}\right]$$

$$A_{2k} = \frac{1}{V_{1k}} - A_{1k} \qquad A_{1,k+1} = \frac{1}{V_{2k}} - A_{2k}$$

$$\mathbf{B}_{2k} = \frac{\hat{\mathbf{x}}_{1k}}{V_{1k}} - \mathbf{B}_{1k} \qquad \mathbf{B}_{1,k+1} = \frac{\hat{\mathbf{x}}_2^t}{V_{2k}} - \mathbf{B}_{2k}$$

and its State evolution!

This algorithm is analytically tractable $\rightarrow$ we have another set of equations, that follow it step-by-step and give:

- the statistical distribution of the estimators at each iteration
- the mean squared-error between the estimators and $\mathbf{x_0}$.

They are called the **state evolution** equations, and are exact in the asymptotic limit. At the fixed point, the state evolution equations are exactly the same as the replica equations.

Rigorous mathematical statement

Interlude on convex optimization

Interlude on proximal operators : definition

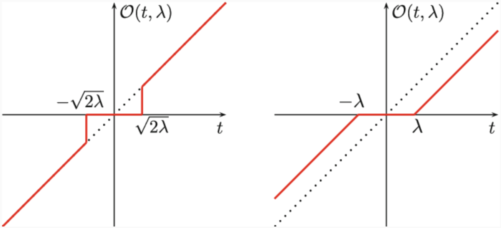
Projection on a set $\mathcal{C}$

$$P_{\mathcal{C}}(\mathbf{y}) = \arg\min_{\mathbf{x}} \left\{ \mathbb{I}_{\mathcal{C}}(\mathbf{x}) + \|\mathbf{x} - \mathbf{y}\|_2^2 \right\}$$

Replace indicator with arbitrary function

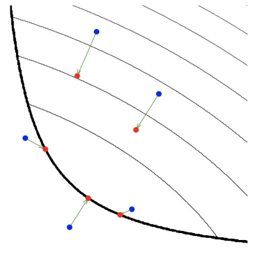
$$\forall \gamma \in \mathbb{R}^+, \mathbf{x}, \mathbf{y} \in \mathcal{X} \quad \mathrm{Prox}_{\gamma f}(\mathbf{y}) = \arg\min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \frac{1}{2\gamma} ||\mathbf{x} - \mathbf{y}||_2^2 \right\}$$

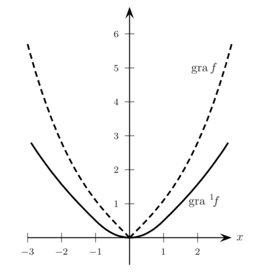
Example : shrinkage (ℓ_2), soft-thresholding (ℓ_1), hard-thresholding (ℓ_0)



Interlude on proximal operators : interpretation

Projection on level sets of f (left)





Moreau envelope (right) : smoothing/regularization of objective

$$M_{\lambda f}(\mathbf{y}) = \inf_{\mathbf{x} \in \mathcal{X}} (f(\mathbf{x}) + \frac{1}{2\lambda} \|\mathbf{x} - \mathbf{y}\|_2^2)$$

Gradient step on Moreau envelope :

$$\text{Prox}_{\gamma f}(\mathbf{x}) = \mathbf{x} - \lambda \nabla M_{\lambda f}(\mathbf{x})$$

Interlude on proximal operators : properties

Proximal enjoy nice properties :

$$\begin{aligned} \text{firm nonexp.} \quad & \langle \mathbf{x} - \mathbf{y}, \mathrm{Prox}_f(\mathbf{x}) - \mathrm{Prox}_f(\mathbf{y}) \rangle \geqslant ||\mathrm{Prox}_f(\mathbf{x}) - \mathrm{Prox}_f(\mathbf{y})||^2 \\ \text{Fixed point} \quad & \mathrm{Prox}_f(\mathbf{x}) = \mathbf{x} \iff \partial f(\mathbf{x}) = 0 \end{aligned}$$

Main idea of proximal algorithms : iteratively apply prox !

Popular algorithm : Douglas-Rachford splitting

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x})$$

$$\mathbf{z}^{k+1} = (1-\rho)\mathbf{z}^k + \rho(2\mathrm{Prox}_{\gamma f} - Id) \circ (2\mathrm{Prox}_{\gamma g} - Id)(\mathbf{z}^k)$$

$$\mathbf{x}^* = \mathrm{Prox}_{\gamma f}(\mathbf{z}^*)$$

Very stable, provably convergent, easy to implement, ...
(ρ is a damping parameter)

VAMP vs Douglas-Rachford

$$
\begin{aligned}
&\textit{Choose initial } A_{10} \textit{ and } \mathbf{B}_{10} \\
\hat{\mathbf{x}}_{1k} &= \mathrm{Prox}_{\frac{1}{A_{1k}} f}\left(\frac{\mathbf{B}_{1k}}{A_{1k}}\right) & \hat{\mathbf{x}}_{2k} &= (\mathbf{A}^T\mathbf{A} + A_{2k} Id)^{-1}(\mathbf{A}^T y + \mathbf{B}_{2k}) \\
V_{1k} &= \frac{\langle \mathrm{Prox}'_{\frac{1}{A_{1k}} f} \rangle}{A_{1k}} & V_{2k} &= \frac{1}{N}\mathrm{Tr}\left[(\mathbf{A}^T\mathbf{A} + A_{2k} Id)^{-1}\right] \\
A_{2k} &= \frac{1}{V_{1k}} - A_{1k} & A_{1,k+1} &= \frac{1}{V_{2k}} - A_{2k} \\
\mathbf{B}_{2k} &= \frac{\hat{\mathbf{x}}_{1k}}{V_{1k}} - \mathbf{B}_{1k} & \mathbf{B}_{1,k+1} &= \frac{\hat{\mathbf{x}}_2^t}{V_{2k}} - \mathbf{B}_{2k}
\end{aligned}
$$

VAMP vs Douglas-Rachford

Choose initial $\mathbf{B}_{10}$, A_1, A_2, V from SE fixed point

$$\hat{\mathbf{x}}_{1k} = \mathrm{Prox}_{\frac{1}{A_1}f}\left(\frac{\mathbf{B}_{1k}}{A_1}\right) \qquad \hat{\mathbf{x}}_{2k} = (\mathbf{A}^T\mathbf{A} + A_2 Id)^{-1}(\mathbf{A}^T y + \mathbf{B}_{2k})$$

$$\mathbf{B}_{2k} = \frac{\hat{\mathbf{x}}_{1k}}{V_1} - \mathbf{B}_{1k} \qquad \mathbf{B}_{1,k+1} = \frac{\hat{\mathbf{x}}_2^t}{V_2} - \mathbf{B}_{2k}$$

Not intended to be used in practice ! For the sake of the proof.

Codable only in teacher student setup

VAMP vs Douglas-Rachford

Remember Douglas-Rachford (take $\rho = 1$) :

$$\mathbf{z}^{k+1} = (2\mathrm{Prox}_{\gamma f} - Id) \circ (2\mathrm{Prox}_{\gamma g} - Id)(\mathbf{z}^k)$$
$$\mathbf{x}^* = \mathrm{Prox}_{\gamma f}(\mathbf{z}^*)$$

VAMP with fixed variances and parameters :

$$\mathbf{B}_2^{t+1} = \left(\frac{1}{V}\mathrm{Prox}_{\frac{1}{A_1}f}\left(\frac{\cdot}{A_1}\right) - Id\right) \circ \left(\frac{1}{V}\mathrm{Prox}_{\frac{1}{2A_2}||\mathbf{y}-\mathbf{A}\mathbf{x}||_2^2}\left(\frac{\cdot}{A_2}\right) - Id\right)(\mathbf{B}_2^t)$$

VAMP is Douglas-Rachford with adaptative parameters
Actually A_1, A_2 are adapted to the curvature of f and g.

Rigorous mathematical statement

Last ingredient

An Important Remark

Now that we have

- the algorithm VAMP that tries to find $\mathbf{x}^*$
- exact equations that describe the mean squared error of VAMP's fixed point estimator

Are we done??

Figure 7: I NEED A CONVERGING TRAJECTORY

Convergence analysis of VAMP

Generate a sequence with the prescription :

$$\begin{aligned} \mathbf{B}_2^{t+1} &= \mathcal{O}_1 \circ \mathcal{O}_2(\mathbf{B}_2^t) \\ \text{where} \quad \mathcal{O}_1 &= \frac{1}{V}\mathrm{Prox}_{\frac{1}{A_1}f}\left(\frac{\cdot}{A_1}\right) - Id \\ \text{and} \quad \mathcal{O}_2 &= \left(\frac{1}{V}\mathrm{Prox}_{\frac{1}{2A_2}||\mathbf{y}-\mathbf{A}\mathbf{x}||_2^2}\left(\frac{\cdot}{A_2}\right) - Id\right) \end{aligned}$$

Find upper bound on the Lipschitz constant of $\mathcal{O}_1 \circ \mathcal{O}_2 \rightarrow$ OK

Can we make it a contraction ?

Forcing the convergence

Enforcing strong convexity and smoothness :

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\min}\ \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + f(\mathbf{x}) + \frac{\lambda_2}{2}\|\mathbf{x}\|_2^2$$

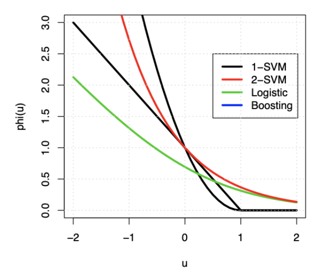
Possibility to enforce convergence for large enough λ_2

- Proof complete for any λ_2 large enough.
- Generalization to any λ_2 with analytical continuation is valid.
- Non-differentiable functions can be approximated.

Extension of the result

Generalized linear estimation

- Keep linear model : $f(\mathbf{a}) = \mathbf{x}^T \mathbf{a}$
- Change the square loss: $\mathcal{L}(\mathbf{x}, \mathbf{y})$



Logistic regression with ℓ_1 penalty

$$\hat{\mathbf{x}}(\gamma, D) = \text{argmin}_{x \in \mathbb{R}^N} \left[-\sum_{\mu=1}^{M} \log \frac{1}{1 + e^{-y_\mu \mathbf{a}_\mu^T \mathbf{x}}} + \gamma \sum_{i=1}^{N} |x_i| \right]$$

Max-margin classifier with ℓ_1 penalty

$$\hat{\mathbf{x}}(\gamma, D) = \text{argmin}_{x \in \mathbb{R}^N} \left[-\sum_{\mu=1}^{M} \max(0, 1 - y_\mu \mathbf{a}_\mu^T \mathbf{x}) + \gamma \sum_{i=1}^{N} |x_i| \right]$$

Extension to the GLM : Generalized-VAMP

Corresponding extension of VAMP $\rightarrow$ Generalized-VAMP

Algorithm 1 VAMP for the SLM

Require: LMMSE estimator $\boldsymbol{g}_2(\boldsymbol{r}_{2k}, \gamma_{2k})$ from (10), denoiser $\boldsymbol{g}_1(\cdot, \gamma_{1k})$, and number of iterations K.

Select initial $\boldsymbol{r}_{10}$ and $\gamma_{10} \geq 0$.
for $k = 0, 1, \dots, K$ **do**
// Denoising
$\widehat{\boldsymbol{x}}_{1k} = \boldsymbol{g}_1(\boldsymbol{r}_{1k}, \gamma_{1k}), \quad \alpha_{1k} = \langle \boldsymbol{g}_1'(\boldsymbol{r}_{1k}, \gamma_{1k}) \rangle$
$\boldsymbol{r}_{2k} = (\widehat{\boldsymbol{x}}_{1k} - \alpha_{1k}\boldsymbol{r}_{1k})/(1 - \alpha_{1k})$
$\gamma_{2k} = \gamma_{1k}(1 - \alpha_{1k})/\alpha_{1k}$
// LMMSE estimation
$\widehat{\boldsymbol{x}}_{2k} = \boldsymbol{g}_2(\boldsymbol{r}_{2k}, \gamma_{2k}), \quad \alpha_{2k} = \langle \boldsymbol{g}_2'(\boldsymbol{r}_{2k}, \gamma_{2k}) \rangle$
$\boldsymbol{r}_{1,k+1} = (\widehat{\boldsymbol{x}}_{2k} - \alpha_{2k}\boldsymbol{r}_{2k})/(1 - \alpha_{2k})$
$\gamma_{1,k+1} = \gamma_{2k}(1 - \alpha_{2k})/\alpha_{2k}$
end for
Return $\widehat{\boldsymbol{x}}_{1K}$.

Algorithm 2 VAMP for the GLM

Require: LMMSE estimators $\boldsymbol{g}_{x2}$ and $\boldsymbol{g}_{z2}$ from (15) or (16), denoisers $\boldsymbol{g}_{x1}$ and $\boldsymbol{g}_{z1}$, and number of iterations K.

Select initial $\boldsymbol{r}_{10}, \boldsymbol{p}_{10}, \gamma_{10} > 0, \tau_{10} > 0$.
for $k = 0, 1, \dots, K$ **do**
// Denoising $\boldsymbol{x}$
$\widehat{\boldsymbol{x}}_{1k} = \boldsymbol{g}_{x1}(\boldsymbol{r}_{1k}, \gamma_{1k}), \quad \alpha_{1k} = \langle \boldsymbol{g}_{x1}'(\boldsymbol{r}_{1k}, \gamma_{1k}) \rangle$
$\boldsymbol{r}_{2k} = (\widehat{\boldsymbol{x}}_{1k} - \alpha_{1k}\boldsymbol{r}_{1k})/(1 - \alpha_{1k})$
$\gamma_{2k} = \gamma_{1k}(1 - \alpha_{1k})/\alpha_{1k}$
// Denoising $\boldsymbol{z}$
$\widehat{\boldsymbol{z}}_{1k} = \boldsymbol{g}_{z1}(\boldsymbol{p}_{1k}, \tau_{1k}), \quad \beta_{1k} = \langle \boldsymbol{g}_{z1}'(\boldsymbol{p}_{1k}, \tau_{1k}) \rangle$
$\boldsymbol{p}_{2k} = (\widehat{\boldsymbol{z}}_{1k} - \beta_{1k}\boldsymbol{p}_{1k})/(1 - \beta_{1k})$
$\tau_{2k} = \tau_{1k}(1 - \beta_{1k})/\beta_{1k}$
// LMMSE estimation of $\boldsymbol{x}$
$\widehat{\boldsymbol{x}}_{2k} = \boldsymbol{g}_{x2}(\boldsymbol{r}_{2k}, \boldsymbol{p}_{2k}, \gamma_{2k}, \tau_{2k}), \quad \alpha_{2k} = \langle \boldsymbol{g}_{x2}'(\dots) \rangle$
$\boldsymbol{r}_{1,k+1} = (\widehat{\boldsymbol{x}}_{2k} - \alpha_{2k}\boldsymbol{r}_{2k})/(1 - \alpha_{2k})$
$\gamma_{1,k+1} = \gamma_{2k}(1 - \alpha_{2k})/\alpha_{2k}$
// LMMSE estimation of $\boldsymbol{z}$
$\widehat{\boldsymbol{z}}_{2k} = \boldsymbol{g}_{z2}(\boldsymbol{r}_{2k}, \boldsymbol{p}_{2k}, \gamma_{2k}, \tau_{2k}), \quad \beta_{2k} = \langle \boldsymbol{g}_{z2}'(\dots) \rangle$
$\boldsymbol{p}_{1,k+1} = (\widehat{\boldsymbol{z}}_{2k} - \beta_{2k}\boldsymbol{p}_{2k})/(1 - \beta_{2k})$
$\tau_{1,k+1} = \tau_{2k}(1 - \beta_{2k})/\beta_{2k}$
end for
Return $\widehat{\boldsymbol{x}}_{1K}$.

Figure 8: Slightly thicker stuff...

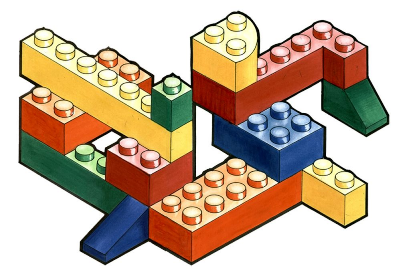
Same proof structure. Convergence analysis requires more sophisticated tools : Lyapunov arguments from dynamical systems.

Conclusion

We proved a statistical physics heuristic result (replica formula) for penalized linear regression, with rotationally invariant matrices.

- This could be done thanks to message-passing algorithms, well-understood in statistical physics, which makes them great theoretical tools.
- We used concepts from convex optimization, random matrix theory..

$\rightarrow$ Exciting problems at the crossroads of several fields! Such "simple" models are the building blocks to understand more complex algorithms.



Thank you :)