

Regularization and Capacity Control

Fabrice Rossi

SAMM
Université Paris 1 Panthéon Sorbonne

2018

Capacity control

Regularization

General setting

Data

- ▶ \mathcal{X} the “input” space and \mathcal{Y} the “output” space
- ▶ D a fixed and unknown distribution on $\mathcal{X} \times \mathcal{Y}$

Loss function

A loss function l is

- ▶ a function from $\mathcal{Y} \times \mathcal{Y}$ to \mathbb{R}^+
- ▶ such that $\forall \mathbf{Y} \in \mathcal{Y}, \quad l(\mathbf{Y}, \mathbf{Y}) = 0$

Model, loss and risk

- ▶ a model g is a function from \mathcal{X} to \mathcal{Y}
- ▶ given a loss function l the risk of g is $R_l(g) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim D}(l(g(\mathbf{X}), \mathbf{Y}))$
- ▶ optimal risk $R_l^* = \inf_g R_l(g)$

Supervised learning

Data set

- ▶ $\mathcal{D} = ((\mathbf{X}_i, \mathbf{Y}_i))_{1 \leq i \leq N}$
- ▶ $(\mathbf{X}_i, \mathbf{Y}_i) \sim D$ (i.i.d.)
- ▶ $\mathcal{D} \sim D^N$ (product distribution)

Empirical risk minimization

- ▶ empirical risk

$$\widehat{R}_l(g, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N l(g(\mathbf{X}_i), \mathbf{Y}_i) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} l(g(\mathbf{x}), \mathbf{y})$$

- ▶ given a class \mathcal{G} define

$$R_{l, \mathcal{G}}^* = \inf_{g \in \mathcal{G}} R_l(g) \text{ and } g_{ERM, l, \mathcal{G}, \mathcal{D}} = \arg \min_{g \in \mathcal{G}} \widehat{R}_l(g, \mathcal{D})$$

What went wrong?

- ▶ if $VCdim(\mathcal{G}) < \infty$
 - 😊 $R_I(g_{ERM,I,\mathcal{G},\mathcal{D}}) \rightarrow R_{I,\mathcal{G}}^*$ (estimation: OK)
 - 😞 $R_{I,\mathcal{G}}^* - R_I^*$ can be large (approximation: KO)
- ▶ if $VCdim(\mathcal{G}) = \infty$
 - 😞 $R_I(g_{ERM,I,\mathcal{G},\mathcal{D}}) - R_{I,\mathcal{G}}^*$ can be large (estimation: KO)
 - 😊 $R_{I,\mathcal{G}}^* \simeq R_I^*$ is possible (approximation: OK)

What went wrong?

- ▶ if $VCdim(\mathcal{G}) < \infty$
 - 😊 $R_l(g_{ERM,l,\mathcal{G},\mathcal{D}}) \rightarrow R_{l,\mathcal{G}}^*$ (estimation: OK)
 - 😞 $R_{l,\mathcal{G}}^* - R_l^*$ can be large (approximation: KO)
- ▶ if $VCdim(\mathcal{G}) = \infty$
 - 😞 $R_l(g_{ERM,l,\mathcal{G},\mathcal{D}}) - R_{l,\mathcal{G}}^*$ can be large (estimation: KO)
 - 😊 $R_{l,\mathcal{G}}^* \simeq R_l^*$ is possible (approximation: OK)

Can we solve this?

Capacity control

Regularization

General idea

- ▶ the VC-dimension gives an idea of the capacity of a class of models
- ▶ to reach $R_{l,\mathcal{G}}^*$ with ϵ with certainty $1 - \delta$, we need $\Theta\left(\frac{VCdim(\mathcal{G}) + \log \frac{1}{\delta}}{\epsilon^2}\right)$ data points
- ▶ we could let the class grow with the data size in such a way that both ϵ and δ could go to zero

Hypotheses

- ▶ infinite data set with $\mathcal{D}_n = ((\mathbf{X}_i, \mathbf{Y}_i))_{1 \leq i \leq n}$
- ▶ $\mathcal{Y} = \{-1, 1\}$ and $l_b(p, t) = \mathbf{1}_{p \neq t}$
- ▶ growing $(\mathcal{G}_j)_{j \geq 1}$ classes of increasing but finite VC dimension
 $VCdim(\mathcal{G}_j) < \infty$
- ▶ asymptotically perfect: $\lim_{j \rightarrow \infty} R_{l_b, \mathcal{G}_j}^* = R_{l_b}^*$
- ▶ $k_n \rightarrow \infty$ et $\frac{VCdim(\mathcal{G}_{k_n}) \log n}{n} \rightarrow 0$

Result

- ▶ define $g_n = \mathcal{G}_{ERM, l, \mathcal{G}_{k_n}, \mathcal{D}_n}$
- ▶ then $R_{l_b}(g_n) \xrightarrow[n \rightarrow \infty]{a.s.} R_{l_b}^*$

Are the hypotheses realistic?

- ▶ yes! There are such model classes!
- ▶ simple example with $\mathcal{X} = [0, 1]$:

$$\mathcal{G}_j = \left\{ g \mid g(X) = \text{sign} \left(a_0 + \sum_{k=1}^j (a_k \cos 2k\pi X + b_k \sin 2k\pi X) \right) \right\}$$

- ▶ $VCdim(\mathcal{G}_j) \leq 2j + 1$ (underlying vector space)
- ▶ use $k_n = n^\alpha$ with $0 < \alpha < 1$
- ▶ many other solutions (radial basis function networks, one hidden layer perceptrons, etc.)

Extensions and limitations

Extensions

- ▶ can be adapted to e.g. $\mathcal{Y} = \mathbb{R}$ with other loss functions
- ▶ bounds on the target values can also be lifted with a similar approach

Limitations

- ▶ classes are **data independent**: they must be chosen beforehand
- ▶ no data adaptation: if the problem is simple, the approximation part might converge too slowly, for instance
- ▶ worst case analysis: the VC-dimension generally overestimates (a lot) the actual capacity of a class of models for the data distribution under study

Central idea

Optimize a compromise between the empirical risk and the complexity of the class

SRM

- ▶ similar hypotheses as before: binary case, infinite data set and asymptotically perfect series of classes
- ▶ global capacity control: $\sum_{j=1}^{\infty} e^{-VCdim(\mathcal{G}^j)} < \infty$
- ▶ capacity penalty: $r(j, n) = \sqrt{\frac{8VCdim(\mathcal{G}^j) \log(en)}{n}}$
- ▶ $j(g) = \inf \{k \mid g \in \mathcal{G}^k\}$
- ▶ define $g_{SRM, n} = \arg \min_{g \in \cup_j \mathcal{G}^j} \left(\widehat{R}_b(g, \mathcal{D}_n) + r(j(g), n) \right)$
- ▶ then $R_{l_b}(g_{SRM, n}) \xrightarrow[n \rightarrow \infty]{a.s.} R_{l_b}^*$

AIC and BIC

- ▶ AIC: $2k - 2 \log \mathcal{L}$, where \mathcal{L} is the likelihood and k the number of parameters
- ▶ BIC: $k \log n - 2 \log \mathcal{L}$
- ▶ notice that the log-likelihood is in general of the form $n \times \log L$, where L is the likelihood for a simple data point
- ▶ thus the per data point penalties are in $\frac{k}{n}$ for AIC and in $\frac{k \log n}{n}$ for BIC
- ▶ in SRM the penalty is in $\frac{\sqrt{k \log n}}{\sqrt{n}}$

- ▶ hypotheses are realistic
- ▶ the trade off between empirical risk and model complexity is now data dependant
- ▶ the model is searched into an class with infinite VC-dimension
- ▶ but
 - ▶ classes are still **data independent**
 - ▶ worst case analysis: the penalty is generally too strong (\sqrt{n} versus n)
 - ▶ this is very costly on a computational point of view
 - ▶ the VC-dimension is quite difficult to compute (frequently bounded above only)
- ▶ **take home message**: replacing ERM by the optimization of a compromise between empirical risk and a capacity measure seems to work

A basic learning framework

1. split the data into \mathcal{D} (learning) and \mathcal{D}' (validation)
2. for each machine learning algorithm \mathcal{A} under study
 - 2.1 for each value θ of the parameters of the algorithm
 - 2.1.1 compute the model using θ on \mathcal{D} , $g_{\mathcal{A},\theta,\mathcal{D}}$
 - 2.1.2 compute $\widehat{R}_l(g_{\mathcal{A},\theta,\mathcal{D}}, \mathcal{D}')$
3. chose the best model g^* among all the models according to $\widehat{R}_l(\cdot, \mathcal{D}')$

ERM view

- ▶ the nested loops build a finite class of models $\mathcal{G}_{\mathcal{D}}$
- ▶ g^* is chosen in $\mathcal{G}_{\mathcal{D}}$ by ERM on \mathcal{D}'
- ▶ works because the class is finite and does not depend on \mathcal{D}' !
- ▶ target risk: $R_{l,\mathcal{G}_{\mathcal{D}}}^*$

Capacity control

Regularization

Regularized Loss Minimization (RLM)

- ▶ many algorithms select a model g in a class \mathcal{G} by minimizing a **Regularized Loss** as follows

$$\arg \min_{g \in \mathcal{G}} (\widehat{A}(g, \mathcal{D}) + \lambda \mathbf{C}(g))$$

- ▶ $\widehat{A}(g, \mathcal{D})$ is a loss (not to be confused with a loss function) which plays a similar role as $\widehat{R}(g, \mathcal{D})$
- ▶ $\mathbf{C}(g)$ is a measure of the regularity of the model g
- ▶ λ is a trade off parameter

CART

- ▶ $\hat{A}(g_T, \mathcal{D}) = \hat{R}_I(g_T, \mathcal{D})$
- ▶ $\mathbf{C}(g_T) = |T|$ (number of leaves)

Structural Risk Minimization

- ▶ $\hat{A}(g, \mathcal{D}) = \hat{R}_{l_b}(g, \mathcal{D})$
- ▶ $\mathbf{C}(g) = \sqrt{VCdim(\mathcal{G})}$ with $g \in \mathcal{G}$ and $\lambda = \sqrt{\frac{8 \log(en)}{n}}$

Ridge regression

- ▶ $\hat{A}(g, \mathcal{D}) = \hat{R}_{l_2}(g, \mathcal{D})$ with $l_2(p, t) = (p - t)^2$ and $g(\mathbf{X}) = \beta_0 + \beta^T \mathbf{X}$
- ▶ $\mathbf{C}(g) = \|\beta\|^2$

With SRM

- ▶ RLM can be seen as an extended SRM
- ▶ the empirical risk can be replaced by an empirical loss
- ▶ the VC-dim based penalty can be replaced by an ad hoc one
- ▶ one specifies directly \mathcal{G} (no need for a structured class of models)

With ERM

- ▶ assume $g^* = \arg \min_{g \in \mathcal{G}} (\hat{A}(g, \mathcal{D}) + \lambda \mathbf{C}(g))$ with $\mu = \mathbf{C}(g^*)$ then g^* is also solution of $\arg \min_{\{g \in \mathcal{G} | \mathbf{C}(g) \leq \mu\}} \hat{A}(g, \mathcal{D})$
- ▶ if both \hat{A} and \mathbf{C} are convex functionals RLM is equivalent to minimizing the loss \hat{A} under a constraint on \mathbf{C} : regularization corresponds to reduced model classes

Impact of the loss

- ▶ in general $\hat{A}(g, \mathcal{D})$ is not the empirical risk
- ▶ can we still provide guarantees with respect to R_l^* for some loss function l ?

Impact of the regularization

- ▶ is the regularization sufficient to ensure some form of learnability?
- ▶ how can we choose λ ?
 - ▶ data size based approaches (as in SRM, AIC, BIC)?
 - ▶ data based approaches (validation)?

Why using a loss?

- ▶ the binary loss function $l_b(p, t) = \mathbf{1}_{p \neq t}$ leads to a very complex optimization problem
- ▶ more generally some loss functions are important from a practical point of view but lead to empirical risks that are more difficult to optimize than others

Consistency

- ▶ in general $\widehat{A}(g, \mathcal{D}) = \widehat{R}_{l'}(g, \mathcal{D})$ for some loss function $l' \neq l$ (frequently up to a transformation of the problem)
- ▶ then we can sometimes ensure that $\widehat{A}(g, \mathcal{D})$ is close to $R_{l'}(g)$
- ▶ but what about $R_l(g)$?

Quadratic relaxation of the binary loss function

- ▶ $\mathcal{Y} = \{-1, 1\}$ and l_b standard binary loss function
- ▶ \mathcal{G} a class of real valued functions
- ▶ empirical risk $\widehat{R}_{l_b}(g, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} \mathbf{1}_{\text{sign}(g(\mathbf{x})) \neq \mathbf{y}}$
- ▶ empirical loss

$$\widehat{A}(g, \mathcal{D}) = \widehat{R}_{l_2}(g, \mathcal{D}) = \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} (g(\mathbf{x}) - \mathbf{y})^2$$

General relaxation for binary classification

Margin based loss

- ▶ $\mathcal{Y} = \{-1, 1\}$ and l_b standard binary loss function
- ▶ \mathcal{G} a class of real valued functions
- ▶ consider $l_\phi(\mathbf{p}, t) = \phi(pt)$ for some function ϕ and $\widehat{A}_\phi(g, \mathcal{D}) = \widehat{R}_{l_\phi}(g, \mathcal{D})$
- ▶ examples
 - ▶ $l_{\text{logi}}(\mathbf{p}, t) = \log(1 + \exp(-pt))$ (logistic loss)
 - ▶ $l_{\text{per}}(\mathbf{p}, t) = \max(0, -pt)$ (perceptron loss)
 - ▶ $l_{\text{hinge}}(\mathbf{p}, t) = \max(0, 1 - pt)$ (hinge loss)
 - ▶ $l_{\text{exp}}(\mathbf{p}, t) = \exp(-pt)$ (exponential loss)
 - ▶ $l_2(\mathbf{p}, t) = (pt)^2 - 2pt + 1$ (because $t \in \{-1, 1\}$)
- ▶ margin interpretation when the decision is $\text{sign}(g(\mathbf{x}))$
 - ▶ $g(\mathbf{x})\mathbf{y} > 0$: correct decision, robust when the product is large
 - ▶ $g(\mathbf{x})\mathbf{y} < 0$: wrong decision, with a “magnitude” proportional to $|g(\mathbf{x})|$

Convex case

- ▶ if ϕ is convex, then minimizing $\widehat{R}_{l_\phi}(g, \mathcal{D}) + \lambda \mathbf{C}(g)$ is probably easier than minimizing $\widehat{R}_{l_b}(g, \mathcal{D})$
- ▶ ϕ is **calibrated** iif
 - ▶ ϕ is convex
 - ▶ ϕ has a derivative in 0
 - ▶ $\phi'(0) < 0$
- ▶ can be extended to the non convex case

Result

- ▶ if ϕ is calibrated then $R_{l_\phi}(g) \rightarrow R_{l_\phi}^*$ implies that $R_{l_b}(g) \rightarrow R_{l_b}^*$
- ▶ in plain English: if we manage to learn with a calibrated surrogate loss, then we learn with respect to the binary loss!

ERM in the binary case

- ▶ is difficult on a computational point of view
- ▶ but the binary loss function can be replaced by any calibrated convex loss: **this is the de facto standard**
- ▶ no adverse consequences asymptotically
- ▶ **however** on a fixed size data set there are differences between loss functions

Consistency

- ▶ using a calibrated convex loss solves the computational aspect
- ▶ but in order to ensure $R_{I_\phi}^*$ can be reached we need \mathcal{G} to be a class of infinite VC-dimension
- ▶ thus we need:
 - ▶ to ensure that sets of the form $\{g \in \mathcal{G} \mid \mathbf{C}(g) \leq \mu\}$ have finite VC-dim
 - ▶ λ can be handled efficiently
- ▶ such results are available for some models, e.g. support vector machines



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

<http://creativecommons.org/licenses/by-sa/4.0/>

Last git commit: 2019-01-12

By: Fabrice Rossi (Fabrice.Rossi@apiacoa.org)

Git hash: baad66f7da2766a14d7a9df74cf06ed054c5d003