A Model Confidence Set extension of grid search in hyper-parameter optimization

Marcell Kujbus, 25/05/2020

Outline for Section 1

1. Why most hyper-parameter optimization techniques fail
   - 1.1 The modeling setup
   - 1.2 Parameters, hyper-parameters
   - 1.3 The goal
2. A Model Confidence Set extension
   - 2.1 Grid search
   - 2.2 Finite and asymptotic termination of the MCS algorithm
   - 2.3 The existence of such a system
3. Loss functions can be extremely flat - an encouraging example
   - 3.1 The CML estimators of an autoregression
   - 3.2 Loss functions in terms of the hyper-parameters
4. The proposed method in a real world application - Covid-19 prediction
   - 4.1 MN-SEIR model
   - 4.2 Results

Assumptions

about learning algorithms

Let

1. $(\Omega, \mathscr{F}, \mathbb{P})$ be a probability space
2. $Y : \Omega \times \mathbb{R}^+ \longrightarrow \mathbb{R}$ be the underlying stochastic process that we aim to model
3. X be the training data set, i.e. a finite set of samples from a common distribution G_Y
4. $\mathscr{A}$ be a learning algorithm, i.e. a functional from X to a function g

Ultimate objective: to a find an algorithm $\mathscr{A}$ that minimizes the expected loss $L_{\mathscr{A}(X)}(y)$ over some samples y having a directly not observable distribution G_Y

Parameters, hyper-parameters

and an illustrative example

An autoregression

Take a $(U_n)_{n\in\mathbb{N}}$ autoregressive process, meaning $U_n = \sum_{i=1}^{\lambda} \theta_i U_{n-i} + \epsilon_n$, where $\theta_{1,\ldots,\lambda} \in \mathbb{R}^+$ and for all $n \in \mathbb{N}$ ϵ_n is standard normally distributed and $\mathbb{E}\epsilon_n\epsilon_m = 0$ for all $n \neq m$.

1. inner optimization with respect to feature parameters
2. bells and whistles in terms of λ : the outer/hyper-parameter optimization

The goal

In general we assume that λ is point in the space Λ spanned by the possible hyper-parameters. Then our goal is to find $\lambda^{(*)}$, such that:

$$\lambda^{(*)} = \operatorname*{argmin}_{\lambda \in \Lambda} \mathbb{E} L_{\mathscr{A}_\lambda(X)}(x) = \operatorname*{argmin}_{\lambda \in \Lambda} \mathbb{E} \Psi(\lambda)$$

$$\lambda^{(*)} \approx \operatorname*{argmin}_{\lambda \in \{\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(n)}\}} \mathbb{E} \Psi(\lambda)$$

The different learning algorithms differ in the way of choosing the trial points $\{\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(n)}\}$.

A commonly accepted, albeit wrong approach

The issue the thesis tries to resolve

It is misleading to choose that hyper-parameter that yields the minimal loss in a single experiment! We have to take random fluctuations coming from finite sampling into consideration!

Outline for Section 2

1. Why most hyper-parameter optimization techniques fail
   - 1.1 The modeling setup
   - 1.2 Parameters, hyper-parameters
   - 1.3 The goal
2. A Model Confidence Set extension
   - 2.1 Grid search
   - 2.2 Finite and asymptotic termination of the MCS algorithm
   - 2.3 The existence of such a system
3. Loss functions can be extremely flat - an encouraging example
   - 3.1 The CML estimators of an autoregression
   - 3.2 Loss functions in terms of the hyper-parameters
4. The proposed method in a real world application - Covid-19 prediction
   - 4.1 MN-SEIR model
   - 4.2 Results

Grid search

The most widely used approach

If Λ is a set indexed by K configuration variables, then the grid search requires that we choose a set of values for each variables $(S^{(1)}, \ldots, S^{(K)})$.
In grid search the set of trials is formed by assembling every possible combination of values, hence $n = \prod_{i=1}^{K} |S^{(i)}|$.

1. Grid search is simple to implement and parallelization is trivial
2. Grid search typically finds better $\lambda^{(*)}$ than purely manual sequential optimization
3. Grid search is reliable in low dimensional spaces (1d, 2d)

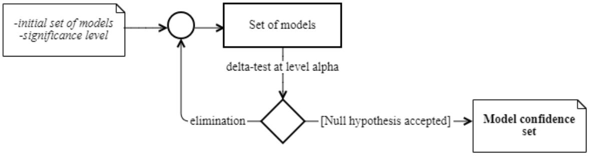
A statistically more robust idea

Model Confidence Set: Hansen et al. 2011

Ingredients:

1. Same trial set: $(S^{(1)}, \ldots, S^{(K)})$
2. A significance level α
3. A user-defined loss function

The algorithm, displayed in a UML Activity diagram



Properties

If the following three conditions hold:

1. $\lim_{n \to \infty} \mathbb{P}(\delta_A = 1 | H_{0,A}) \leq \alpha$, meaning δ is a valid-test
2. $\limsup_{n \to \infty} \mathbb{P}(\delta_A = 1 | H_{1,A}) = 1$, meaning δ has a power of 1 asymptotically
3. $\lim_{n \to \infty} \mathbb{P}(e_A \in P^* | H_{1,M}) = 0$, meaning asymptotically any element thrown out is almost surely not in the superior set,

then:

1. $\liminf_{n \to \infty} \mathbb{P}(P^* \subset \hat{P}_{1-\alpha}) \geq 1 - \alpha$, hence the confidence set nomenclature
2. $\lim_{n \to \infty} \mathbb{P}(i \in P^* | i \notin P^*) = 0$
3. $\lim_{n \to \infty} \mathbb{P}(P^* = \hat{P}_{1-\alpha}) = 1, if |P^*| = 1$

An extended result

Furthermore, if $\mathbb{P}(\delta_A = 1, e_A \in P^*) \leq \alpha)$, meaning there is coherency between the equivalence test and the elimination rule, then the main result holds on a finite sample as well: $\mathbb{P}(P^* \subset \hat{P}_{1-\alpha}) \geq 1 - \alpha$.

The existence of such a coherent system

Hansen et al. (2011) has shown some particular choices of δ equivalence tests and e elimination rules that match the above mentioned criteria. For detailed information, visit the online appendix of this work.

Outline for Section 3

1. Why most hyper-parameter optimization techniques fail
    - 1.1 The modeling setup
    - 1.2 Parameters, hyper-parameters
    - 1.3 The goal
2. A Model Confidence Set extension
    - 2.1 Grid search
    - 2.2 Finite and asymptotic termination of the MCS algorithm
    - 2.3 The existence of such a system
3. Loss functions can be extremely flat - an encouraging example
    - 3.1 The CML estimators of an autoregression
    - 3.2 Loss functions in terms of the hyper-parameters
4. The proposed method in a real world application - Covid-19 prediction
    - 4.1 MN-SEIR model
    - 4.2 Results

Parameters to estimate

in an autoregression

Our modeling assumption are the followings:

1. There is an initial set of orders to experiment on the grid, containing the true value p: $\mathscr{P} = (p, p_1, \ldots, p_{n-1})$. $\mathscr{P}$ is the set of hyper-parameters.
2. For every $i \in \mathscr{P}$,we model the the process assuming the order of the autoregressive process is exactly i. The $(\alpha, \phi_1, \ldots, \phi_i, \sigma^2)^T$ vector is found via an inner optimization, in this case by maximizing the likelihood, that Y follows an autoregressive trend with order i.
3. The loss function is the L_2-norm.

The conditional maximum likelihood approach

Fix $i \in \mathscr{P}$. Let n be the greatest number in $\mathscr{P}$.
Let $\theta_i = (\phi_i, \sigma_i^2)^T$ be the vector of parameters to find, where $\phi_i = [\alpha, \phi_1, \ldots, \phi_i]^T$. Given a single trajectory y of Y up to time t, which parameters maximize the likelihood that given the process is sampled from an $AR(i)$, the actual data is observed?

$$\theta_i^{(*)} = \operatorname{argmax} \prod_{j=n+1}^{T} f_{\theta_i}(y_j | y_{j-1}, \ldots, y_1)$$

The estimators

Definition

Let

1. $x_{t,i} = [1, y_{t-1}, \ldots, y_{t-i}]^T$
2. $X_i = [x_{T,i}, x_{T-1,i}, \ldots, x_{n+1,i}]^T$ be a $((T-n-1) \times (i+1))$ matrix
3. $y = [y_T, y_{T-1}, \ldots, y_{n+1}]^T$ be a $T-n-1$ long vector

Based on the above defined formulas:

1. $\tilde{\phi}_i = (X_i^T X_i)^{-1} X_i^T y$
2. $\tilde{\sigma}_i^2 = \frac{(y - X_i \tilde{\phi}_i)^T (y - X_i \tilde{\phi}_i)}{T-n-1}$

Classify the estimators

with respect to biasedness

Definition

We sat that $\tilde{\phi}_i$ is an unbiased estimator of ϕ_i if $\mathbb{E}\tilde{\phi}_i = \phi_i$

Let the actual ground true order be p.

Modelled order i	$\tilde{\phi}_i$	$\tilde{\sigma}_i^2$
$i < p$	biased	asy. unbiased
$i \geq p$	unbiased	asy. unbiased

Table: Biasedness of the CML estimators of an autoregression in terms of the modelling order

Detailed derivation of these properties are in the paper, open the online appendix to see.

Loss functions associated with the modelled orders

Asymptotic result

$$y - \hat{y} \approx \mathbb{E}(y - \hat{y}) + \mathcal{N}_i(0, Var(y - \hat{y})) = \mathbb{E}(y - \hat{y}) + std(y - \hat{y})\mathcal{N}_i(0, 1)$$
$$L_i \approx \mathbb{E}^2(y - \hat{y}_i) + Var(y - \hat{y}_i)\chi_i^2 + 2\mathbb{E}(y - \hat{y}_i) \cdot std(y - \hat{y}_i)\mathcal{N}_i(0, 1)$$
$$\mathbb{E}L_i \approx \mathbb{E}^2(y - \hat{y}_i) + Var(y - \hat{y})$$

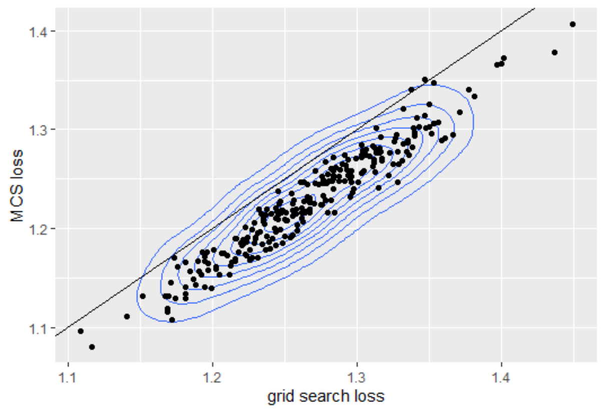
Theorem

$$\forall i \in \mathscr{P}, \ \mathbb{E}L_i = \begin{cases} a^2, & if\ i < p \\ 0 & otherwise \end{cases}.$$

Corollary

The theoretical P^* set for an autoregressive process is asymptotically the hyper-parameter set $A = i \geq p,\ i \in \mathscr{P}$. In the long run, the Model Confidence Set is A at any significance level

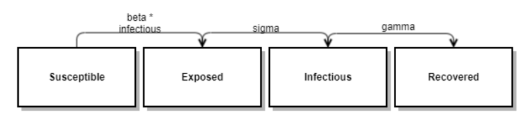
Results



Outline for Section 4

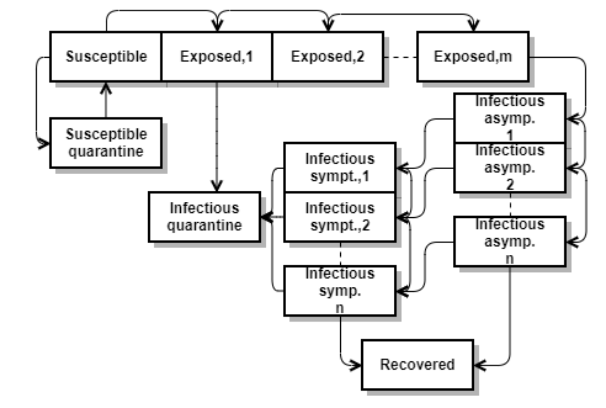
1. Why most hyper-parameter optimization techniques fail
    - 1.1 The modeling setup
    - 1.2 Parameters, hyper-parameters
    - 1.3 The goal
2. A Model Confidence Set extension
    - 2.1 Grid search
    - 2.2 Finite and asymptotic termination of the MCS algorithm
    - 2.3 The existence of such a system
3. Loss functions can be extremely flat - an encouraging example
    - 3.1 The CML estimators of an autoregression
    - 3.2 Loss functions in terms of the hyper-parameters
4. The proposed method in a real world application - Covid-19 prediction
    - 4.1 MN-SEIR model
    - 4.2 Results

Introducing the SEIR model family



Wearing et al. (2005) has shown, that a more realistic approach is to assume that the probability of leaving a class is a function of time spent inside, which is small at first and increasing after the mean infectious/latent period is reached. Lloyd (2001) proposes that a realistic or empirically provable distribution can be obtained by choosing $p(t)$ to be a gamma probability density function, with parameters γ and n (σ and m for the exposed class).

Extend the model to capture contact tracing and isolation



Speaking in terms of differential-equations

The solution of this system describes the epidemic

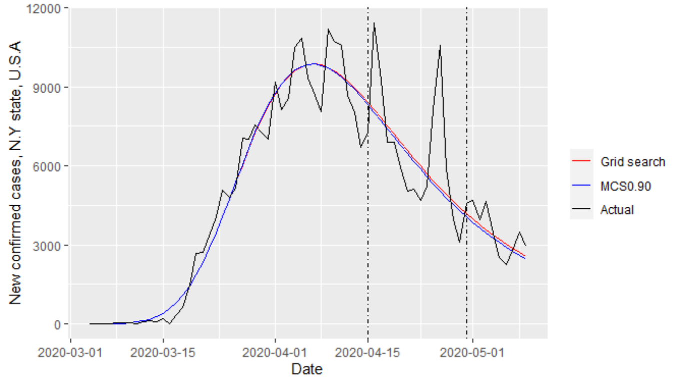
$$
\begin{aligned}
\frac{dk}{dt} &= -\frac{k(t)-k_0}{\lambda} \\
\frac{dS}{dt} &= -\frac{(k(t)bI(t)+qk(t)(1-b)I_S(t))S(t)}{N} + \frac{qk(t)(1-b)S(t-\tau_Q)I_S(t-\tau_Q)}{N} \\
\frac{dS_Q}{dt} &= \frac{qk(t)(1-b)S(t)I_S(t)}{N} - \frac{qk(t)(1-b)S(t-\tau_Q)I_S(t-\tau_Q)}{N} \\
\frac{dE_1}{dt} &= \frac{k(t)b(I(t)-qI_S(t))S(t)}{N} - m\sigma E_1(t) \\
\frac{dE_i}{dt} &= m\sigma E_{i-1}(t) - m\sigma E_i(t), \quad i = 2, \dots, m \\
\frac{dI_{A,1}}{dt} &= m\sigma E_m(t) - n\gamma I_{A,1}(t) - P_{I,1}(t) \\
\frac{dI_{A,i}}{dt} &= n\gamma I_{A,i-1}(t) - n\gamma I_{A,i}(t) - P_{I,i}(t), \quad i = 2, \dots, n \\
\frac{dI_{S,1}}{dt} &= P_{I,1}(t) - (n\gamma + d_I)I_{S,1}(t) \\
\frac{dI_{S,i}}{dt} &= P_{I,i}(t) + n\gamma I_{S,i-1}(t) - (n\gamma + d_I)I_{S,i}(t), \quad i = 2, \dots, n \\
\frac{dQ}{dt} &= \frac{qk(t)bS(t)I_S(t)}{N} + d_I I_S(t) \\
\frac{dR}{dt} &= n\gamma(I_{A,n}(t) + I_{S,n}(t))
\end{aligned}
$$

Limitations of the model

1. homogeneous mixing
2. deterministic approach
3. number of contacts stay low

Results

MCS and grid searched performed statistically indistinguishable



Appendix

The whole paper, all of the code implementations, several figures are all available at the online appendix.