How To Incorporate Prior Knowledge

Paul van Leeuwen (paul2.vanleeuwen@wur.nl), Wageningen Economic Research

26 October 2020

Introduction

Introduction

- Prior knowledge can be of vital significance.
- Takeaways:
  - employing prior knowledge appropriately could lead to
    - improvements that are statistically significant and relevant and
    - better explainability of your model results;
  - try to understand
    - your data-generating process and
    - the underlying mechanism of your algorithm (don't just push buttons).

Analogy To Image Recognition

- Image recognition problems also utilise prior knowledge.
- Transforming the same image still leads to the same object to be classified.
    - Other common transformations are scaling, shifting, and adding noise.

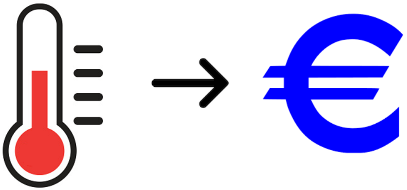
The Practical Example

Introduction

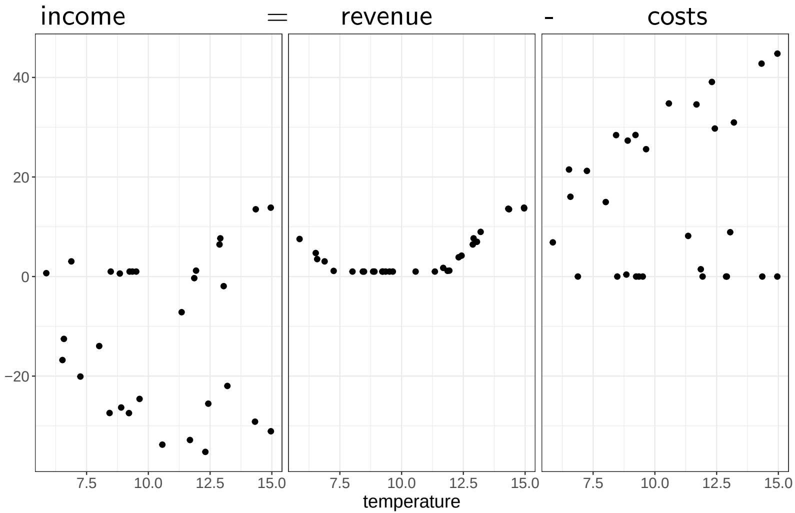
€

Introduction

Introduction



A Visualisation



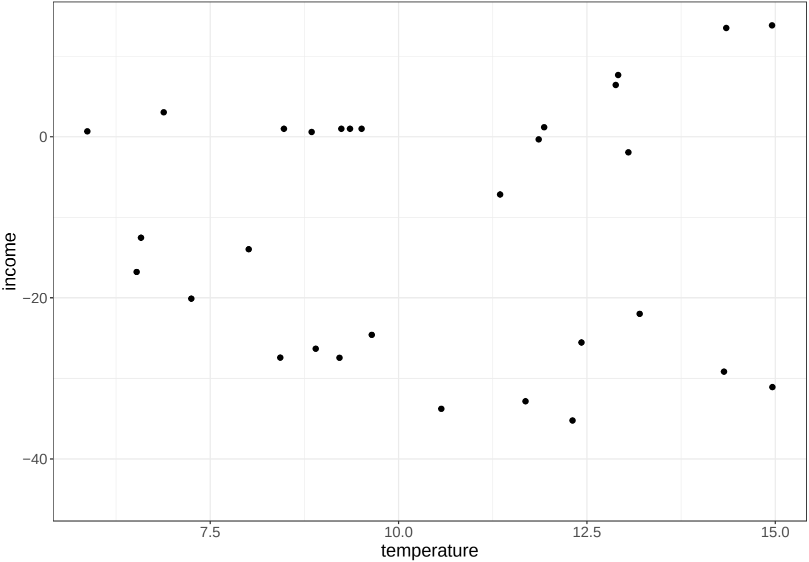
Predicting Without Prior Knowledge

- Suppose we are given temperature as feature (or explanatory variable) and income as target variable (or dependent variable).
- Then we could model income as follows

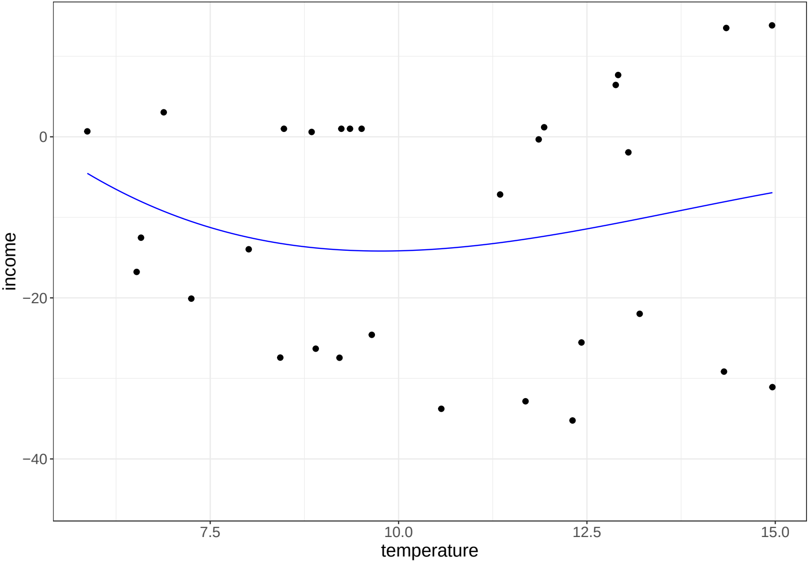
$$\begin{aligned}\text{income}_i = \text{intercept} + \beta_1 \cdot \text{temperature}_i + \\ \beta_2 \cdot \text{temperature}_i^2 + \beta_3 \cdot \text{temperature}_i^3 + \text{error}_i\end{aligned}$$

for $i = 1, \ldots, n$ and n observations.

Predicting Without Prior Knowledge



Predicting Without Prior Knowledge

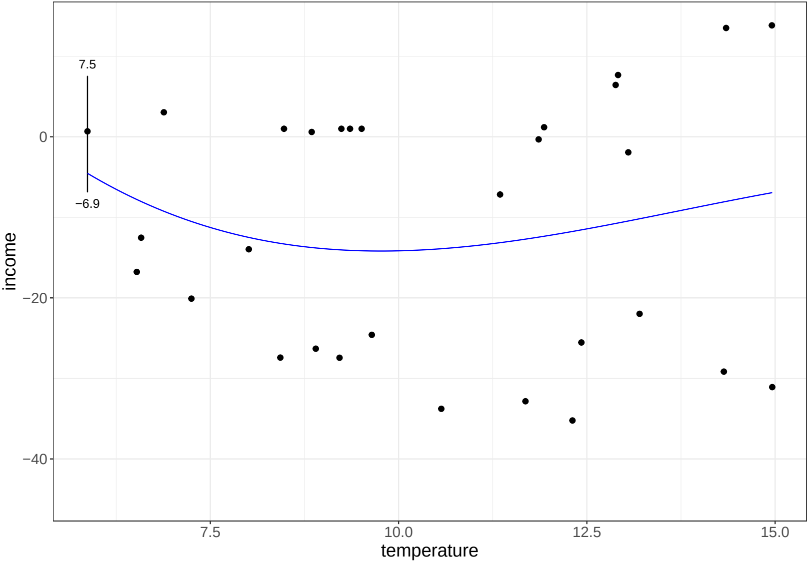


Incorporating The Prior Knowledge

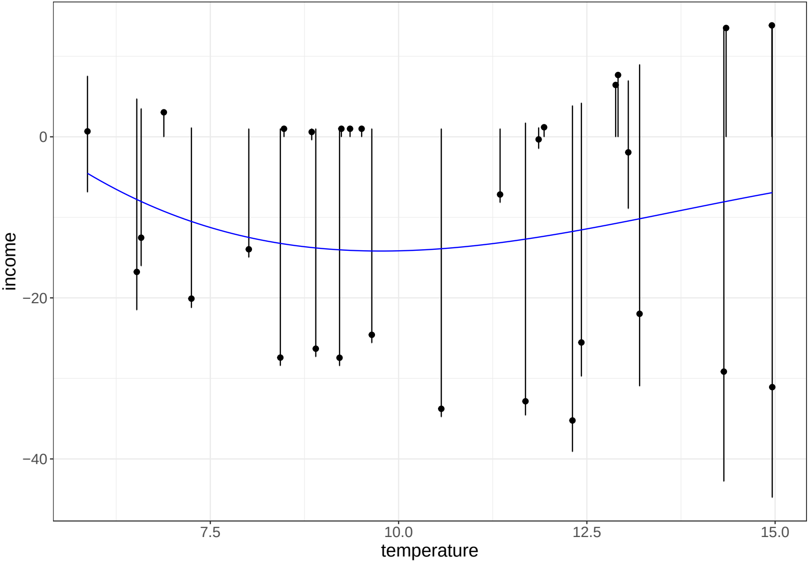
- We know for each observation i that
  - $\text{income}_i = \text{revenue}_i - \text{costs}_i$,
  - $\text{revenue}_i \geq 0$, and
  - $\text{costs}_i \geq 0$.
- Then also

$$-\text{costs}_i \leq \text{income}_i \quad \text{and} \quad \text{income}_i \leq \text{revenue}_i$$

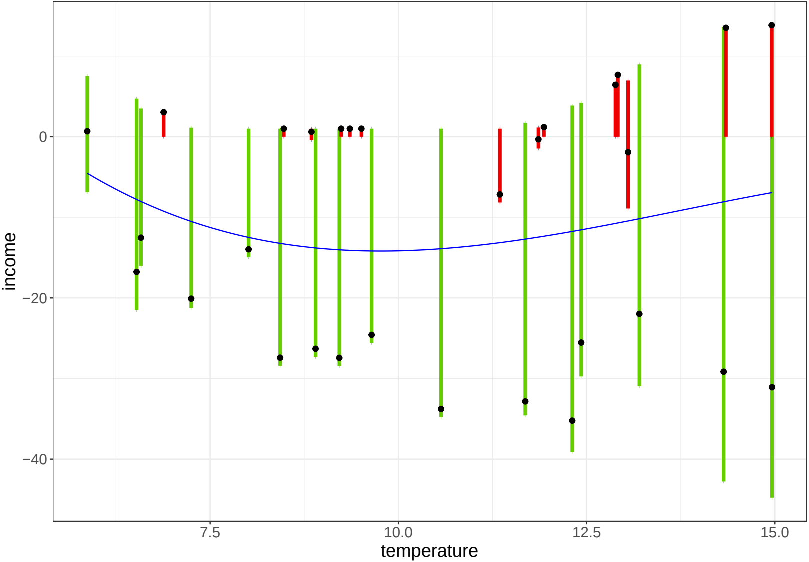
Prediction With Prior Knowledge



Predicting With Prior Knowledge



Predicting With Prior Knowledge



Utilise The Prior Knowledge

Predicting With Prior Knowledge

- Perfect prediction is usually hard to achieve.
- Without the prior knowledge we cannot be aware which of the predicted values are within their desired range.
- We can achieve better predictions when we utilise this prior knowledge!
- In addition, the predictions make more sense.

Predicting With Prior Knowledge

- Adjust the model coefficients β such that both goals are achieved: keep the training error as small as possible *and* let the predicted values stay in their corresponding intervals.
- We follow the notation of Abu-Mostafa [1993].
  - Every useful piece of information that can be processed into the loss function is a *hint*.
  - Friedman et al. [2001] (Section 13.3.3) also discusses hints.
  - 'A very powerful method for incorporating almost any type of prior knowledge into a neural network (or other non-linear statistical model) is training by hints.' (Lampinen et al. [1999])
- Using hints might seem outdated.
  - However, this concept is also known as prior probability distribution, imposing monotonicity, exploiting invariance, etc.

Technical Implementation

- As with ridge regression we can add a penalty to the Sum of Squared Residuals (SSR)

$$E_0 = (\boldsymbol{y} - \boldsymbol{X}\beta)'(\boldsymbol{y} - \boldsymbol{X}\beta)$$

- The penalty increases quadratically when a prediction is outside its desired range,

$$e_1(\hat{\boldsymbol{y}}_i) = \begin{cases} (-\boldsymbol{c}_i - \hat{\boldsymbol{y}}_i)^2 & \text{when } \hat{\boldsymbol{y}}_i < -\boldsymbol{c}_i, \\ 0 & \text{when } -\boldsymbol{c}_i \leq \hat{\boldsymbol{y}}_i \leq \boldsymbol{r}_i, \text{ and} \\ (\hat{\boldsymbol{y}}_i - \boldsymbol{r}_i)^2 & \text{when } \boldsymbol{r}_i < \hat{\boldsymbol{y}}_i. \end{cases}$$

with the predicted income $\hat{\boldsymbol{y}}_i = \boldsymbol{X}_i\beta$, the costs $\boldsymbol{c}_i$, and the revenue $\boldsymbol{r}_i$.

Technical Implementation

- The new loss function to be minimised with respect to β becomes

$$\alpha E_0 + (1 - \alpha) E_1$$

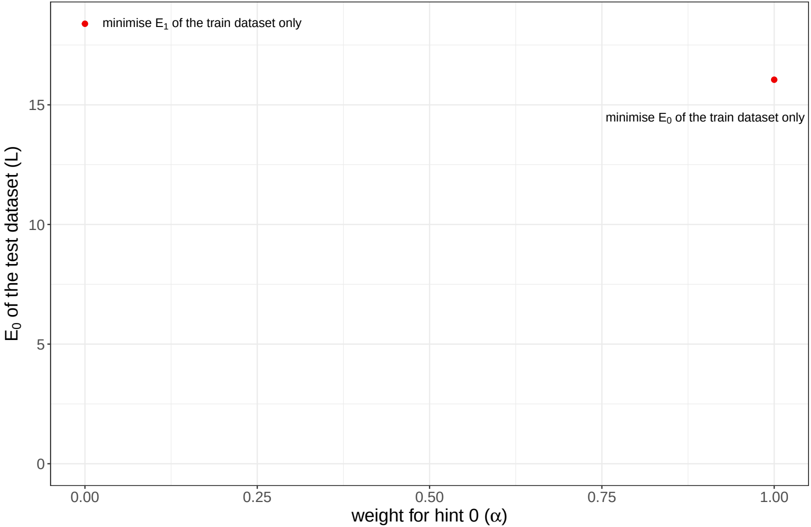
with E_0 the SSR of the train dataset, $\alpha \in [0, 1]$, and

$$E_1 = \sum_{i=1}^{n} e_1(\hat{\boldsymbol{y}}_i)$$

  - Note that with $\alpha = 1$ we return to E_0.
- As with ridge regression, optimise α with respect to the SSR of a *test* dataset (label that $L(\alpha)$), using, *e.g.*, cross-validation.
  - $L(\alpha) = \sum_{j=1}^{m} (\boldsymbol{y}_j - \boldsymbol{X}_j \beta)^2$

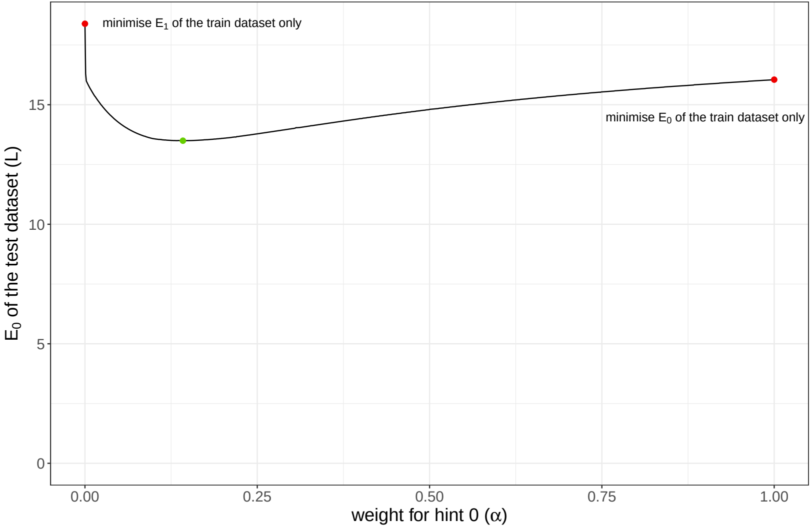
Results

E_0 of the test dataset (L) for different values of the weight for hint 0 (α).

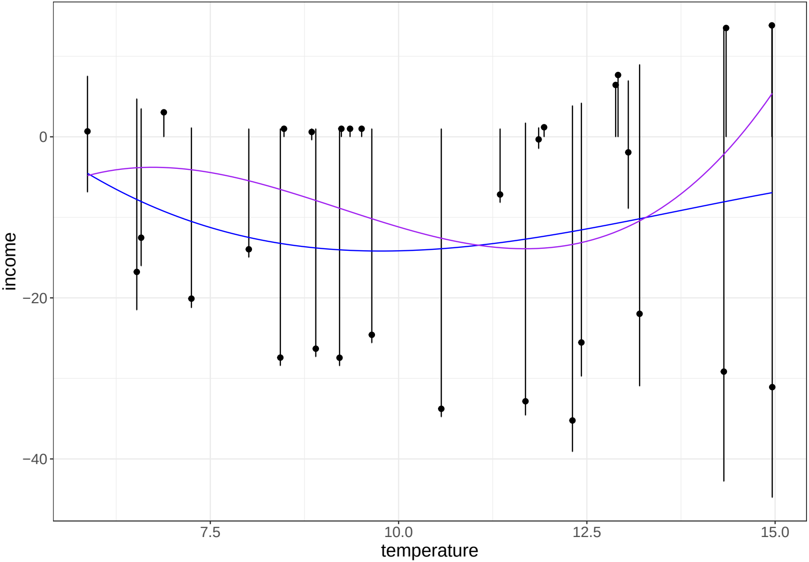


Results

E_0 of the test dataset (L) for different values of the weight for hint 0 (α).



Results



Results

- Not all simulations result in an improvement at all, but the majority does.
- And of the simulations that did result in an improvement (71.1%), the improvement was statistically significant (p-value $<$ 1e-05) and relevant (improvement is on average 4.4%).

Theory

Theoretical Analysis

- Has E_0 of the test dataset with respect to α always a unique minimum?
- Why is the weight relative close to hint 1 whereas you might expect closer to hint 0?
- Could we have achieved the same result by transforming the target variable $\boldsymbol{y}$ and explanatory variables $\boldsymbol{X}$?
- How do non-linear models such as eXtreme Gradient Boosting perform relative to the 'classical' linear model?
- What are other hints?

References

Yaser S. Abu-Mostafa. A method for learning from hints. pages 73–80, 1993.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Jouko Lampinen, Paula Litkey, and Harri Hakkarainen. Selection of training samples for learning with hints. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, volume 2, pages 1438–1441. IEEE, 1999.