Fakultät für Mathematik und Informatik
Ruprecht-Karls-Universität Heidelberg

Combining Probability Forecasts by Isotonic Recursive Partitioning

Diplomarbeit
von
Christian Rohrbeck

Betreuer:
Prof. Dr. Tilmann Gneiting
Dr. Thordis L. Thorarinsdottir

9. August 2012

Abstract

Accurate forecasts are very important for large parts of the economy for avoiding high costs and damages. In the last years, there is a growing belief that predictions have to be in form of probabilistic forecasts. In many situations, we have access to the forecasts of several organizations and institutes. Therefore, it seems favorable to aggregate these individual forecasts for generating calibrated and sharp forecasts. Unfortunately, the most intuitive method, the linear opinion pool, lacks calibration if the individual forecasts are calibrated. In this thesis, we investigate and compare two methods, isotonic recursive partitioning and beta-transformed linear opinion pool, in the context of calibration, sharpness and from a computational viewpoint, by implementing the approaches in `R`. The methods are illustrated by a simulation study and a case study. The case study uses statistical and National Weather Service temperature and probability of precipitation forecasts. We find that both methods can generate calibrated and sharp forecasts even if the individual forecasts are not calibrated. Additionally, we expand the theory of isotonic recursive partitioning.

Zusammenfassung

Um hohe Schäden und Kosten zu vermeiden, sind für einen großen Teil der Wirtschaft exakte Vorhersagen von großer Wichtigkeit. In den letzten Jahren setzte sich immer mehr die Überzeugung durch, dass Vorhersagen probabilistisch sein sollten. In vielen Situationen haben wir Zugriff auf Vorhersagen von verschiedenen Organisationen oder Instituten. Deshalb erscheint es vorteilhaft diese einzelnen Vorhersagen zu kombinieren, um kalibrierte und scharfe Vorhersagen zu erhalten. Unglücklicherweise erzeugt die naheliegendenste Methode, der linear opinion pool, unkalibrierte Vorhersagen für den Fall, dass die einzelnen Vorhersagen kalibriert sind. In dieser Diplomarbeit untersuchen und vergleichen wir zwei Methoden, isotonic recursive partitioning und beta-transformed linear opinion pool, in Hinsicht auf Kalibration, Schärfe und von einem algorithmischen Blickpunkt aus, in dem wir die Methoden in `R` implementieren. Die beiden Methoden werden mit Hilfe einer Simulations- und einer Fallstudie illustriert. Die Fallstudie basiert auf mit Hilfe von statistischen Methoden erzeugten Temperatur- und Regenwahrscheinlichkeitsvorhersagen und den entsprechenden Vorhersagen des National Weather Service. Wir zeigen, dass beide Methoden kalibrierte und scharfe Vorhersagen erzeugen können, unabhängig davon ob die Einzelvorhersagen kalibriert sind oder nicht. Außerdem erweitern wir die Theorie des isotonic recursive partitioning.

Contents

List of Figures

List of Tables

1 Introduction

Large parts of the economy are sensitive to random events like weather, economic shocks, etc. Therefore, there is a critical need for forecasts for such events. Within various scientific disciplines like meteorology, hydrology, business studies or demography, there exists a growing belief that forecasts should be probabilistic in nature and in form of predictive distributions (Gneiting and Raftery, 2007). Full predictive distributions allow for the assessment of forecast uncertainty and optimal decision making (Gneiting, 2008).

Probabilistic forecasting aims to provide calibrated and sharp predictive distributions for random events. "Calibration refers to the statistical consistency between the distributional forecasts and the observations and is thus a joint property of the predictions and the events that materialize. Sharpness refers to the concentration of the predictive distributions and is a property of the forecasts only" (Gneiting et al., 2007, page 243). Following Murphy and Winkler (1987) and Gneiting et al. (2007), probabilistic forecasts should be as sharp as possible, subject to calibration.

In many cases, several forecasts are available, where the forecasters have access to different information sources. For example, there may be forecasts from different experts or organizations for today's weather or the growth of the European economy. In these cases, there is a strong empirical evidence that combining information sets results in an improved predictive performance. However, because these experts are often in competition, we cannot ask them to combine their information sets for getting a better forecast. Thus, we search methods which generate calibrated and sharp predictive distributions from the individual forecasts.

The most intuitive method for aggregating individual forecasts into a combined forecast is the linear pool by Stone (1961) which has ubiquitous success in a large number of applications. Although other methods for combining exist, see for example, Genest and Zidek (1986) and Clemen and Winkler (1999), the linear pool is the favored one, as the works of Winkler (1968), Zarnowitz (1969) and Hall and Mitchell (2007) indicate (Gneiting and Ranjan, 2011). However, recent papers like Hora (2004) or Ranjan and Gneiting (2010) point at potential shortcomings and limitations. For example, following Hora (2004), any nontrivial convex combination of two calibrated density forecasts is uncalibrated (Gneiting and Ranjan, 2011). Consequently, there is a need for alternative methods.

Ranjan and Gneiting (2010) introduce the beta-transformed linear opinion pool

(BLP) for combining probability forecasts, where the outcomes are binary random variables. Gneiting and Ranjan (2011) expand the BLP approach to the class of cumulative distribution functions on $\mathbb{R}$. The main reference concerning our work is Luss et al. (2012). The authors consider the problem of predictive modeling and establish isotonic recursive partitioning (IRP) for solving the isotonic regression problem by Barlow and Brunk (1972), which is based on the squared error loss function. This approach may also be used in the framework of Ranjan and Gneiting (2010). Thus one question is, whether the IRP approach generates sharp and calibrated forecasts. Secondly, we are interested in a comparison of the BLP and IRP approaches. Hence, we implement the IRP approach in `R` and test it using the simulation example and the case study of Ranjan and Gneiting (2010). The case study uses the data set of Baars and Mass (2005).

Luss and Rosset (2011) formulate a generalization of IRP for a class of convex and differentiable scoring functions, called general isotonic recursive partitioning (GIRP). We further investigate the question, whether there exist scoring functions, for which the solution of the GIRP approach is equal to the one of the IRP approach.

In the context of probabilistic forecasting, there is a high interest in a generalization of the IRP approach to more general predictive distributions which do not only concern the combination of point forecasts or probability forecasts for binary random variables. General predictive distributions automatically lead to a different statistical framework. A next question is thus, whether we can extend the IRP approach to other settings, for example, the multinomial case.

The `R` implementation of the BLP approach of Ranjan and Gneiting (2010) was kindly provided by Roopesh Ranjan and Tilmann Gneiting and the dataset of Baars and Mass (2005) was kindly provided by Jeff Baars, Cliff Mass, Roopesh Ranjan and Tilamm Gneiting. One subfunction of the IRP implementation is based on the Matlab implementation by Luss et al. (2012) that the authors kindly sent to us.

The remainder of this work is organized as follows: The IRP and BLP approaches and their generalizations are introduced in Chapter 2. In Chapter 3, we describe our implementation of the IRP approach and its generalization, the GIRP approach, in `R` and discuss the arising difficulties and corresponding solutions. We investigate the equality of the GIRP and IRP approaches in Section 4.1 and give some connections to Barlow and Brunk (1972) in Section 4.2. In Section 4.3, we focus on an extension of the IRP approach for combining probability forecasts for categorical random variables. In Chapter 5, we apply our implementation of Chapter 3 to the simulation study and the case study of Ranjan and Gneiting (2010). As the IRP approach is not limited to combining probability forecasts, we also apply it on the temperature forecast data of Baars and Mass (2005). Finally, a summary is provided in Chapter 6.

2 BLP and GIRP Approaches

In this chapter, we introduce the beta-transformed linear opinion pool (BLP) of Ranjan and Gneiting (2010) and the isotonic recursive partitioning (IRP) of Luss et al. (2012). First, based on Ranjan and Gneiting (2010), we develop the statistical framework which we mainly consider in the following sections. Then, we introduce the BLP approach in Section 2.2 and the IRP and the GIRP approaches in Section 2.3. The beta-transformed linear opinion pool is, in contrast to general isotonic recursive partitioning (GIRP), non-parametric.

2.1. Statistical framework

We work within a probabilistic framework which considers the joint distribution of $(y, p_1, \ldots, p_m)$, where $y \in \{0, 1\}$ is a binary event and $p_1, \ldots, p_m \in [0, 1]$ are the probability forecasts of m forecasters for y. In this framework, a combined probability forecast is a random variable $\widehat{p}$ that is measurable with respect to the σ-algebra generated by $p_1, \ldots, p_m$. As already mentioned in the Introduction, probabilistic forecasting aims to provide calibrated forecasts.

Definition (Ranjan and Gneiting, 2010)
A probability forecast p is calibrated for the binary random variable y if the probability that y occurs conditional on p is equal to p. That is

$$\mathbb{P}(y = 1|p\,) = \mathbb{E}(y|p\,) = p$$

almost surely.

For example, the linear opinion pool

$$\widehat{p} = w_1 p_1 + \cdots + w_m p_m,$$

$$w_1, \ldots, w_m > 0, \qquad w_1 + \cdots + w_m = 1,$$

satisfies the condition of measurability. However, Ranjan and Gneiting (2010) prove, see Theorem 2.1, that the linear opinion pool lacks calibration in the sense that

$$\mathbb{P}(\mathbb{E}(y|\widehat{p}\,) \neq \widehat{p}\,) > 0,$$

if all forecasts $p_1, \ldots, p_m$ are calibrated. Thus, other methods for combining forecasts are needed.

Theorem 2.1 (Ranjan and Gneiting, 2010)
Suppose that $p_1, \ldots p_m$ are calibrated for the binary event y and such that $p_{i_1} \neq p_{i_2}$ with strictly positive probability for at least one pair $i_1 \neq i_2$. Consider the linear opinion pool

$$\widehat{p} = w_1 p_1 + \cdots + w_m p_m,$$

where $w_1, \ldots, w_k > 0$ and $w_1 + \cdots + w_m = 1$. Let

$$q = \mathbb{P}(y = 1|\widehat{p}) = \mathbb{E}(y|\widehat{p})$$

denote the recalibrated version of $\widehat{p}$, i.e. the conditional probability of y given $\widehat{p}$. Then the following results hold.

1. The linear opinion pool lacks calibration, in that $q \neq \widehat{p}$ with strictly positive probability.

2. The linear opinion pool $\widehat{p}$ lacks sharpness, in that

   $$\mathbb{E}(\widehat{p} - p_0)^2 < \mathbb{E}(q - p_0)^2,$$

   where $p_0 = \mathbb{E}(\widehat{p}) = \mathbb{E}(q) = \mathbb{E}(y)$. In words, both $\widehat{p}$ and q are marginally consistent, but on average $\widehat{p}$ is closer to its expectation, the naive climatological forecast p_0, than its recalibrated version q.

3. The recalibrated forecast q is calibrated, i.e. $\mathbb{P}(y = 1|q) = q$ almost surely, and it outperforms $\widehat{p}$, in that

   $$\mathbb{E}\left\{S(q, y)\right\} < \mathbb{E}\left\{S(\widehat{p}, y)\right\}$$

   for every strictly proper scoring rule [1].

The theoretically optimal combination of $p_1, \ldots, p_m$ is a probability forecast p^* such that

$$\mathbb{E}(p^* - y)^2 \leq \mathbb{E}(\widehat{p} - y)^2,$$

where $\widehat{p}$ is any measurable forecast with respect to the σ-algebra generated by $p_1, \ldots, p_m$. The forecast p^* is then the conditional expectation of y

$$p^* = \mathbb{E}(y|p_1, \ldots, p_m). \tag{2.1}$$

This forecast minimizes the expectation of any strictly proper scoring rule (Ranjan and Gneiting, 2010). By Theorem 2.1, p^* can only be a non-linear opinion pool of $p_1, \ldots, p_m$. However, the optimal combined probability forecast (2.1) is in general unknown and has to be estimated from a training data set.

[1] See Gneiting and Raftery (2007) for an introduction to the theory of strictly proper scoring rule S.

2.2. Beta-transformed linear opinion pool

The BLP approach applies a beta transform to the linear opinion pool, in order to recalibrate it. The combined forecast $\widehat{p}$ is of the form

$$\widehat{p} = B_{\alpha,\beta}\left(\sum_{k=1}^{m} w_k p_k\right), \tag{2.2}$$

where

$$B_{\alpha,\beta}(z) = B(\alpha,\beta)^{-1}\int_0^z t^{\alpha-1}(1-t)^{\beta-1}dt, \qquad z \in [0,1]$$

is the cumulative distribution function of the beta density and

$$w_1,\ldots,w_m \geq 0, \qquad w_1+\cdots+w_m = 1.$$

For $\alpha = \beta = 1$, we get the traditional linear opinion pool. Often it can be useful to set further constraints on the recalibration transform $B_{\alpha,\beta}$. For example, one might require that

$$B_{\alpha,\beta}(z)\begin{cases} \leq z & \text{for} \quad z \leq z_0 \\ \geq z & \text{for} \quad z \geq z_0 \end{cases}, \tag{2.3}$$

for some $z_0 \in (0,1)$. "For example, if the individual forecasts are calibrated, Theorem 2.1 suggests that the linear opinion pool is underconfident,in the sense that the calibration curve lies under the diagonal for small forecast probabilities, and above the diagonal for high probabilities, with a fixed point at some $z_0 \in (0,1)$" (Ranjan and Gneiting, 2010, page 78). Theorem 2 of Wallsten and Diederich (2001) supports the choice of $z_0 = 1/2$, under which condition (2.3) can be enforced by requiring that $\alpha = \beta \geq 1$ (Ranjan and Gneiting, 2010).

We now consider the estimation of the parameters $\alpha, \beta, w_1, \ldots, w_m$ of the BLP model. Let $(Y, P_1, \ldots, P_m)$ be the training data set, where $Y = (y_1, \ldots, y_n) \in \{0,1\}^n$ is the vector of available observations of binary random variables and $P_k = (P_{k1}, \ldots, P_{kn}) \in [0,1]^n$, $k \in \{1,..,m\}$, are the corresponding probability forecasts for Y issued by the k-th forecaster.

The aggregated BLP forecast has the form

$$\widehat{P_i} = B_{\alpha,\beta}\left(\sum_{k=1}^{m} w_k P_{ki}\right) \quad \text{for } i = 1,\ldots,n.$$

If we assume independence of the y_i , $i \in \{1,\ldots,n\}$, the log-likelihood function of the BLP model in (2.2) is equal to

$$
\begin{aligned}
&l(Y,P_1,\ldots,P_m;w_1,\ldots,w_m;\alpha,\beta) \\
&\quad = \sum_{i=1}^{n} \left[y_i \log\left(\widehat{P}_i\right) + (1-y_i)(1-\log\left(1-\widehat{P}_i\right)) \right] \qquad (2.4) \\
&\quad = \sum_{i=1}^{n} \left(y_i \log\left[B_{\alpha,\beta}\left(\sum_{k=1}^{m} w_k P_{ki} \right) \right] + (1-y_i) \log\left[1 - B_{\alpha,\beta}\left(\sum_{k=1}^{m} w_k P_{ki} \right) \right] \right)
\end{aligned}
$$

We get maximum likelihood estimates for the $k+2$ parameters $w_1,\ldots,w_k$, α, β under the constraints $w_1,\ldots,w_k \geq 0$, $w_1+\cdots+w_k=1$ and $\alpha>0$, $\beta>0$ by numerically optimizing the function $l(Y,P_1,..,P_m,w_1,..,w_k,\alpha,\beta)$. If wet set further constraints, such as (2.3), we get stricter constraints on the parameters.

The BLP is thus a parametric method for combining probability forecasts for binary events. The advantage of the parametric approach is that the number of parameters is linear in the number of forecasters and not exponential, as in the non-parametric case. This effect of an exponential growth in parameters is called the curse of high dimensions which we avoid with this method. Nonetheless, if we want to include a further forecast, we have to estimate all our parameters again and the previous results will not help us.

Gneiting and Ranjan (2011) formulate a generalization of the BLP approach for the full class of cumulative distribution functions $\mathcal{F}_{\mathbb{R}}$. Specifically, let $F_1,..,F_m \in \mathcal{F}_{\mathbb{R}}$ be cumulative distribution functions on $\mathbb{R}$. Then the combined cumulative distribution function $\widehat{F} \in \mathcal{F}_{\mathbb{R}}$ is of the form

$$
\widehat{F}(y) = B_{\alpha,\beta}\left(\sum_{k=1}^{m} w_k F_k(y) \right), \qquad z \in \mathbb{R}, \qquad (2.5)
$$

where $w_1,..,w_m$ are non-negative and the sum of them is equal to 1. Here, $B_{\alpha,\beta}$ denotes the cumulative distribution function of the beta density with parameters $\alpha,\beta>0$.

If each cumulative distribution function F_k has a Lebesgue density, f_k, $k\in\{1,..,m\}$, the combined predictive cumulative distribution function has Lebesgue density

$$
\widehat{f}(y) = \left(\sum_{k=1}^{m} w_k f_k(y) \right) b_{\alpha,\beta}\left(\sum_{k=1}^{m} w_k F_k(y) \right),
$$

where $b_{\alpha,\beta}$ denotes the beta density with parameters α, $\beta>0$. As in the case of binary outputs, we can estimate the parameters from a training data set $(Y,\underline{F}_1,..,\underline{F}_m)$, where $Y=(y_1,..,y_n)\in\mathbb{R}^n$ and $\underline{F}_k=(F_{k1},\ldots,F_{kn})$ are the corresponding predictive distributions of the k-th forecaster for Y. Let each predictive distribution function F_{ki} be absolutely continuous and have a Lebesgue density f_{ki}. Then, we can estimate the

parameters by maximizing the sum of the logarithmic scores:

$$\begin{aligned} & l(Y, \underline{F}_1, .., \underline{F}_m, w_1, .., w_m; \alpha, \beta) \\ & \qquad = \sum_{i=1}^{n} \log \widehat{f}(y_i) \\ & \qquad = \sum_{i=1}^{n} \left(\log \left[\sum_{k=1}^{m} w_k f_{ki}(y_i) \right] + \log \left[b_{\alpha,\beta} \left(\sum_{k=1}^{m} w_k F_{ki}(y_i) \right) \right] \right). \end{aligned} \tag{2.6}$$

The logarithmic score is a proper scoring rule, in the sense that forecasting the true density of the observation maximizes the expected score. It maps the density forecast and the realizing observation to the logarithm of the value that the density forecast attains at the observation. It is positively orientated, which means, the higher, the better. Nevertheless, if we assume independence between the n training cases, the corresponding estimated parameters can be seen as maximum likelihood estimates.

The generalization of the BLP is not of interest in our setting as the GIRP approach cannot combine predictive distributions. However, GIRP has the ability to combine forecasts that are not probabilistic.

2.3. Isotonic recursive partitioning

Concerning the problem of predictive modeling, we want to fit a model describing the dependence of our observation y on the forecasts $(p_1, \ldots, p_m)$, where $y, p_1, \ldots, p_m \in \mathbb{R}$. In the following, we work within a more general framework than that of Section 2.1, in the sense that the forecasts and the observations are in $\mathbb{R}$. Nevertheless, the results are equally applicable.

In the context of combining forecasts, it seems favorable to set isotonic constraints on the combined forecast $\widehat{p}$. For example, if each forecaster says that the probability of precipitation for today is lower than for tomorrow, $\widehat{p}$ should preserve this property. Thus, we achieve a lower combined probability of precipitation for today than for tomorrow. Therefore, we are searching for an isotonic function

$$\begin{aligned} g : \quad & \mathbb{R}^m \to \mathbb{R}, \\ & (p_1, \ldots, p_m) \mapsto g(p_1, \ldots, p_m) = \widehat{p}. \end{aligned} \tag{2.7}$$

that maps the individual forecasts to an aggregated forecast $\widehat{p}$.

Definition
A function $g : \mathbb{R}^m \to \mathbb{R}$ is called isotonic if $\forall\, p = (p_1, .., p_m)$, $\widetilde{p} = (\widetilde{p}_1, \ldots, \widetilde{p}_m) \in \mathbb{R}^m$:

$$p_k \leq \widetilde{p}_k, \;\; \forall k \in \{1, \ldots, m\} \;\; \Rightarrow \;\; g(p) \leq g(\widetilde{p}).$$

The isotonic function g has to be estimated from some training data. Let $(Y, P_1, .., P_m)$ be a given set of training data, where $Y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ is the vector of observations and $P_k = (P_{k1}, \ldots, P_{kn}) \in \mathbb{R}^n$, $k \in \{1, \ldots, m\}$, the corresponding forecast of the k-th forecaster for Y. In the case of probability forecasts for binary output, we get $Y \in \{0,1\}^n$ and $P_k \in [0,1]^n$, $k \in \{1, \ldots, m\}$.

Here, isotonic constraints are achieved by applying the considerations above to the training data. In general, the set of isotonic vectors is a closed convex cone in $\mathbb{R}^n$ (Barlow and Brunk, 1972). Define by I the set of isotonic constraints generated by the training data set. This set I can be characterized as follows

$$(i_1, i_2) \in I \Leftrightarrow P_{ki_1} \leq P_{ki_2} \quad \forall k \in \{1, \ldots, m\},$$

that is, in the case i_1 each forecaster has a lower prediction value than in the case i_2.

Additionally, the function g should minimize an objective value. If we define the objective value as the sum of the squared errors, we get the following optimization problem

$$g^* = \underset{g \text{ isotonic}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - g(P_{1i}, .., P_{mi}))^2. \tag{2.8}$$

For the probabilistic framework of Section 2.1, we use the negatively orientated Brier score, which is equal to the squared error.

Remark

The problem (2.8) corresponds to the isotonic regression problem by Barlow and Brunk (1972). Following their definition we have to solve

Minimize

$$\sum_{i=1}^{n} (y_i - g(P_{1i}, .., P_{mi}))^2 w_i \ ,$$

subject to

$$(P_{1i_1}, .., P_{mi_1}) \preceq (P_{1i_2}, .., P_{mi_2}) \Rightarrow g(P_{1i_1}, .., P_{mi_2}) \leq g(P_{1i_2}, .., P_{mi_2}),$$

where

$$(P_{1i_1}, .., P_{mi_1}) \preceq (P_{1i_2}, .., P_{mi_2}) \Leftrightarrow P_{ki_1} \leq P_{ki_2} \quad \forall\, k \in \{1, .., m\}, i_1, i_2 = 1, .., n$$

and

$$w_i \geq 0, \ \ \forall\, i \in \{1, .., n\}.$$

Any solution to this problem is called isotonic regression. As we weight all observations and corresponding forecasts equally, we set the weights w_i, $i \in \{1, .., n\}$, equal to 1. Barlow and Brunk (1972) give a characterization of the solution of the isotonic regression problem, see Barlow and Brunk (1972, Eq 2.8).

Some properties of g^* are well-known. "As many authors have noted, g^* comprises a partitioning of the space $\mathbb{R}^m$ into regions with no holes satisfying isotonicity properties defined below, with a constant fitted to g^* in each region." (Luss et al., 2012, page 1). In the context of probability forecasts, g^* comprises a partitioning of $[0,1]^m$. If the number of observations and the number of forecasters both grow large, two major concerns arise: statistical difficulty in form of overfitting as well as computational difficulty. In the worst case, it can happen that the number of forecasters is so high that the number of restrictions is very small. As a consequence, the fit for many points would be equal or relatively near to their corresponding observation. This might lead to overfitting.

A computationally attractive approach to solve the optimization problem in (2.8) can be found in the optimization and operational research literature. By repeatedly solving 'optimal cut problems', for which efficient algorithms exist, we split our data set into regions of decreasing size. This recursive approach generates isotonic models of increasing model complexity in each iteration step, ultimately leading to g^*. This method is called isotonic recursive partitioning (IRP). The approach is already mentioned in the paper by Maxwell and Muckstadt (1985). However, Maxwell and Muckstadt (1985) use a different objective value than the sum of the squared errors in (2.8)(Luss et al., 2012). In contrast to the BLP of Gneiting and Ranjan (2011), IRP is a non-parametric approach.

In the case of the isotonic regression problem, the solution is well-known: Observations are divided into subsets, where the fit in each set is equal to the set mean observation value. This can be seen by the conditions of Karush-Kuhn-Tucker (KKT), see Luss et al. (2012, Section 2.1) and Boyd and Vandenberghe (2004) for details.

If we want to generalize the optimization problem (2.8) to the class of scoring functions which are convex and differentiable, we have to solve

$$g^* = \underset{g \text{ isotonic}}{\operatorname{argmin}} \sum_{i=1}^{n} f_{y_i}(g(P_{1i}, .., P_{mi})), \tag{2.9}$$

where f_{y_i} is differentiable and convex and depends on the corresponding observation value y_i, but not on the other observations. Luss and Rosset (2011) expand the IRP approach to this framework and refer to it as generalized isotonic recursive partitioning (GIRP). As in (2.8), we can conclude properties of the solution (2.9) by KKT. The KKT conditions imply that as in the case of IRP, observations are divided into subsets with a constant solution in each subset. The solution depends on the scoring functions. Following Luss and Rosset (2011), with the KKT conditions we get a partitioning algorithm for solving (2.9) as follows:

Let

$$V \subseteq \{(y_i, P_{1i}, .., P_{mi}) : \ i = 1, \ldots, n\}$$

be a subset of the training data set. We define

$$\widetilde{V} = \{i \ : \ (y_i, P_{1i}, .., P_{mi}) \in V, \ i = 1, \ldots, n\}$$

as the set of corresponding indices and $I_V \subseteq I$ as the set of isotonic constraints generated by V. Following the KKT conditions, the value w_V that minimizes the sum of the scoring functions in (2.9) for the subset V is given by

$$w_V = \underset{z \in \mathbb{R}}{\operatorname{argmin}} \sum_{i \in \widetilde{V}} f_{y_i}(z). \tag{2.10}$$

This leads to the condition

$$\sum_{i \in \widetilde{V}} \frac{\partial f_{y_i}(z)}{\partial z}\bigg|_{w_V} = 0.$$

Suppose V is optimally split in the sense that a further partition is not beneficial or contradicts the isotonic constraints represented by I_V. According to the KKT conditions, it should be infeasible to find two subsets

$$(\widetilde{V}_1, \widetilde{V}_2) \in C_V = \left\{(A, B) | A, B \subseteq \widetilde{V}, A \cup B = \widetilde{V}, A \cap B = \emptyset, \nexists x \in A, y \in B : y \preceq x\right\},$$

which fulfill

$$\sum_{i \in \widetilde{V}_2} \frac{\partial f_{y_i}(z)}{\partial z}\bigg|_{w_V} - \sum_{i \in \widetilde{V}_1} \frac{\partial f_{y_i}(z)}{\partial z}\bigg|_{w_V} < 0. \tag{2.11}$$

The summation over $\widetilde{V}_2$ represents the change in the objective value of the optimization problem (2.9) due to an increase in the solution w_V of (2.11), whereas the summation over $\widetilde{V}_1$ represents the change in the objective value due to a decrease of w_V. Thus, an increase of the solution of the set V_2 and a decrease of the solution of V_1 will cause an overall decrease of the objective value. Following the KKT conditions, the optimal solutions are w_{V_1} and w_{V_2}. The GIRP approach is looking for two such subsets $\widetilde{V}_1^*$,$\widetilde{V}_2^*$ that minimize the left hand term in (2.11).

Thus, we get to the optimal cut problem

$$(\widetilde{V}_1^*, \widetilde{V}_2^*) = \underset{(\widetilde{V}_1, \widetilde{V}_2) \in C_V}{\operatorname{argmin}} \left\{ \sum_{i \in \widetilde{V}_2} \frac{\partial f_{y_i}(z)}{\partial z}\bigg|_{w_V} - \sum_{i \in \widetilde{V}_1} \frac{\partial f_{y_i}(z)}{\partial z}\bigg|_{w_V} \right\}. \tag{2.12}$$

This cut problem can be expressed as the binary program

$$x^* = \underset{x}{\operatorname{argmin}} \left\{ \sum_{i \in \widetilde{V}} x_i \ \frac{\partial f_{y_i}(z)}{\partial z}\bigg|_{w_V} \ : \ x_i \leq x_j \ \forall (i,j) \in I_V, \ x_i \in \{-1, 1\} \ \forall i \in \widetilde{V} \right\}. \tag{2.13}$$

As we have seen in (2.11), the sum of the scoring functions with $x_i = 1$ ($x_i = -1$) can be decreased by increasing (decreasing) the corresponding fits. Thus, we get

$$\widetilde{V}_1^* = \{i : x_i^* = -1\} \qquad \text{and} \qquad \widetilde{V}_2^* = \{i : x_i^* = 1\}$$

and the corresponding optimal subsets V_1^*, V_2^* of V as the corresponding subsets of the training data. If we set $f_{y_i}(z) = (z - y_i)^2$, we get the solution of the IRP approach.

Based on this theoretical approach, we can implement an algorithm. We start with the entire data set in one set. After each split, we save the resulting objective cut value

$$c^T x^* = \sum_{i \in \widetilde{V}} {x_i}^* \left. \frac{\partial f_{y_i}(z)}{\partial z} \right|_{w_V} \tag{2.14}$$

and perform the next cut with the smallest remaining cut value. The set V is optimally split if each entry of the solution x^* of (2.13) is equal to 1 or -1. The algorithm stops if there exists no further partition of the current sets.

Algorithm 1 Generalized Isotonic Recursive Partitioning by Luss and Rosset (2011)

Require: Observations $y_1, .., y_n$ and set of isotonic restrictions I
Require: $A = \{\{1, .., n\}\}$, $C = \{(0, \{1, .., n\}, \emptyset)\}$, $B = \emptyset$
Require: $k := 0$, $M_0 = (A, w_A)$
while $A \neq \emptyset$ **do**
  $k := k + 1$
  Let $(val, w^-, w^+) \in C$ be the potential partition with smallest val
  Update $A = (A \backslash (w^- \cup w^+)) \cup \{w^-, w^+\}$
  Update $C = C \backslash (val, w^-, w^+)$
  $M_k = (A \cup B, w_{A \cup B})$
  for all $v \in \{w^-, w^+\}$ **do**
    Set $c_i = \left. \frac{\partial f_i(x)}{\partial x} \right|_{w_V}$ $\forall i \in v$ and w_V is the weight of V defined in (2.10)
    Solve (2.13) with input c and the corresponding partial order I_V and get x^*
    if $x_1^* = \cdots = x_n^*$ (set is optimally split) **then**
      Update $A = A \backslash v$ and $B = B \cup v$
    else
      Let $v^- = \{i : x_i^* = -1\}$ and $v^+ = \{i : x_i = 1\}$
      Update $C = C \cup \left\{c^T x^*, v^-, v^+\right)\}$
    end if
  end for
end while
return B, indices of observations corresponding to the optimal sets

At each iteration step, the algorithm produces a model M_k which contains the partition of the data set after the k-th iteration step. Luss and Rosset (2011) prove that this algorithm is a no-regret partition algorithm of problem (2.13) in the sense that it does not cut through a set of the global optimal solution of (2.9), see Theorem 1 of Luss and Rosset (2011). Further, they demonstrate that the model M_k is in the class of isotonic models after each iteration step k, see Theorem 2 of Luss and Rosset (2011). Thus, we get a regularization path of isotonic models.

Consequently, we can stop the algorithm before it is near or equal to the global optimal solution. Further, the authors argue that the comparative improvement of the objective value is much greater in the early steps of the GIRP algorithm than in the later steps. Hence, it might be useful to stop the algorithm before reaching the global optimal solution, for reducing the problem of overfitting.

Remark
The generalized isotonic regression problem of Barlow and Brunk (1972) is to

Minimize

$$\sum_{i=1}^{n}[\Phi(g(P_{1i},..,P_{mi}))-y_i g(P_{1i},..,P_{mi})]w_i,$$

subject to

$$(P_{1i_1},..,P_{mi_1}) \preceq (P_{1i_2},..,P_{mi_2}) \Rightarrow g(P_{1i_1},..,P_{mi_2}) \leq g(P_{1i_2},..,P_{mi_2}),$$

where

$$(P_{1i_1},..,P_{mi_1}) \preceq (P_{1i_2},..,P_{mi_2}) \Leftrightarrow P_{ki_1} \leq P_{ki_2} \quad \forall\, k \in \{1,..,m\}$$

and

$$w_i \geq 0, \quad \forall\, i \in \{1,..,n\}\,.$$

Here, Φ is convex and proper, that is, it takes nowhere the value $-\infty$ and is not identical to $+\infty$. Luss and Rosset (2011) prove that under the assumption of differentiability on Φ, each generalized isotonic regression problem can be solved by GIRP.

3 Implementation in R

In this chapter, we implement the IRP and GIRP algorithms described in Chapter 2 in R and give a short overview over the differences between the two functions. Further, we describe computational difficulties and corresponding solutions. First, we introduce the standardization of the optimization problem (2.13) in Section 3.1 and then its implementation in Sections 3.2 and 3.3.

3.1. Standardization of the optimization problem

As already mentioned, the solution of the IRP approach is equal to the solution of the optimization problem (2.13) that is given by

$$x^* = \underset{x}{\operatorname{argmin}} \left\{ \sum_{i \in \widetilde{V}} x_i \left. \frac{\partial f_{y_i}(z)}{\partial z} \right|_{w_V} : \; x_i \leq x_j \; \forall (i,j) \in I_V, \; x_i \in \{-1, 1\} \; \forall i \in \widetilde{V} \right\}$$

and we aim to rewrite it as a standard linear optimization problem

$$\min_x \; c^T x, \;\; \text{subject to} \;\; Dx \leq d, \;\; x \geq 0. \tag{3.1}$$

If we change (2.13) such that $x_i \in \{0, 1\}$, $i = 1, \ldots, n$, the solution x^* does not change. Consequently, solving (2.13) is equal to solve

$$x^* = \underset{x}{\operatorname{argmin}} \left\{ \sum_{i \in \widetilde{V}} x_i \left. \frac{\partial f_{y_i}(z)}{\partial z} \right|_{w_V} : \; x_i \leq x_j \; \forall (i,j) \in I_V, \; x_i \in \{0, 1\} \; \forall i \in \widetilde{V} \right\}.$$

From this, we may deduce the matrix D and the vector d which represent the side-conditions in (3.1). A direct approach is the following.

Let $(i_1, i_2) \in I_V$. We want to exclude the case that $i_1 = 1$ and $i_2 = 0$. Thus, we set up an inequality

$$x_{i_1} - x_{i_2} \leq 0.$$

If we apply this to all members of I_V, we get $\#I_V$ inequalities. Finally, we have to set up n further inequalities, which bound the x_i to be at maximum 1. Consequently, we get the desired matrix D and the vector $d = 0$. Therefore, we can use existing R packages for solving (2.13).

3.2. Implementation of the IRP approach in R

As mentioned in Chapter 2, for both the IRP and GIRP approach, we need training data to estimate the isotonic function g. In the implementation, we represent the different forecasts in an $m \times n$ matrix, P, where m is the number of forecasts and n is the number of cases in the training data set. The second input are the observations Y of the training data set contained in the vector `y`. The forecasts of the test data set are represented in a $m \times \widetilde{n}$ matrix $\widetilde{P}$, where $\widetilde{n}$ is the number of cases in the test data set. This matrix `P_test` is our third and last input. For generating a forecast $\widetilde{y}$ for the test data set out of the training data, we implement the function `IRP`.

```
## Initialization of the parameters ##

P        <- matrix(0,n,m)
P_test   <- matrix(0,n_test,m)
y        <- matrix(0,n,1)

## Put in the data ##

P        <- ...
P_test   <- ...
y        <- ...

## Applies the IRP algorithm on the data and returns a vector
## with the forecasts for the training data set P_test

y_pred   <- IRP(y,P,P_test)
```

Listing 3.1: Setting and call of the function `IRP`

The function `IRP` itself combines three subfunctions which perform the IRP approach. The first and second subfunctions execute the approaches of Section 2.2 and Section 3.1. The third one computes the corresponding forecasts for the training data set. In the following subsections, we look at each of these functions and explain their functionality. Some of the three mentioned subfunctions use further functions which set up and solve the standardized optimization problem of Section 3.1. The code of each subfunction can be found in the Appendix B. For solving the linear program, we use the `R` libraries `lpSolve` and `linprog`. The function `IRP` has the following form:

```
IRP<-function(y,P,P_test){

library(lpSolve)
library(linprog)

## Get the number of forecasters and cases
m       <-  dim(P[1, ])
n       <-  dim(P[ , 1])

## Sort the data
p_temp  <-  order(P[ , 1])
y       <-  y[p_temp]

for(i in 2:m){
P[,i]   <-  P[ , i][p_temp]
}

## 1. subfunction: Sets up the isotonic constraints
A       <- createMatrixA(P)

## 2. subfunction: Executes the IRP approach
p_fits  <- IRP_cut(y, A)

## 3. subfunction: Computes the probability forecast
y_pred  <- yPrediction(P_test, p_fits, P, mean(y))

return(y_pred)
}
```

Listing 3.2: The `IRP` command

3.2.1. The `createMatrixA` - function

The `createMatrixA` function is implemented for generating the set of isotonic constraints I in Algorithm 1 of Chapter 2 from the training data forecasts P. Each isotonic constraint is a further side-condition of the optimization problem (2.13). In the end, the isotonic constraints are represented in the matrix A by

$$A[i_1, i_2] = 1 \text{ if } P[i_1, k] \leq P[i_2, k], \quad \forall k \in \{1, \ldots, m\}, \quad i_1, i_2 = 1, \ldots, n.$$

Otherwise, the entry is zero. Because we have already sorted the data set with respect to the first forecaster, $A[i,j] = 0$ if $i > j$. Here it is important to mention that the matrix A is not equal to the matrix D of Section 3.1. The matrix A just contains the information about the isotonic constraints, but is not directly used for solving (3.1).

For small data sets, this procedure is fine. But for a high number of training cases, the number of constraints grows large. Because each positive entry of A is one further constraint, it is a further side-condition. Consequently, we may get a high number of inequalities represented by the matrix D. For example, if we have a training data set with about 1,000 observations, we can at maximum get about 500,000 entries of the matrix which are equal to one and `R` cannot handle the resulting matrix D with 500,000 rows and 1,000 columns. Therefore, we have to reduce the number of entries in the matrix which are equal to one to a minimum. Our approach is to interpret the matrix A as the adjacency matrix of a graph.

Definition (Brouwer et al., 1989)
A graph is a pair $\Gamma = (V, E)$ consisting of a set V, referred to as the vertex set, and a set E of 2-subsets of V, referred to as the edge set.

If we think of our restrictions as a directed graph with n vertices, (i,j) is a edge if $A[i,j] = 1$. Thus, A is the adjacency matrix of a directed graph with n vertices and $\#I$ edges. Consequently, we have to reduce the set of edges E to a minimum $\widetilde{E}$ in the sense that each edge of E can be concluded by $\widetilde{E}$. Hence, we have to compute the transitive reduction of the directed graph.

We start in the first row of the matrix A and check whether there exists a case j, $1 \neq j$, with $A[1,j] = 1$. Second, we search for a third case k with the property that $A[1,k] = 1$ and $A[k,j] = 1$. If such a k exists, we can set $A[1,j]$ equal to 0 because the information about the isotonic constraint is already represented by the isotonic constraints given by $A[1,k]$ and $A[k,j]$. Because A is a upper triangular matrix, we just have to investigate the k's between 1 and j, because otherwise $A[k,j]$ is 0. In the context of the graphical model this means that the restriction generated by the edge $(1,j)$ is already represented by the edges $(1,k)$ and (k,j). After we have investigated all elements of the first row, we go on with the second, the third and so on until the matrix A represents the minimum number of restrictions.

In this way, we compute the matrix `A` which contains all necessary isotonic constraints for solving (2.13). All in all, with this procedure, we can reduce the number of side-conditions to a minimum and thus we need not to stop at a size of about 600 observations. Note that the `createMatrixA` function is the slowest of the three subfunctions because it has to make a lot of comparisons.

3.2.2. The `IRP_cut`-function

This main function generates the IRP cut by executing Algorithm 1 of Chapter 2. The `IRP_cut`-function is based on the Matlab implementation of Luss et al. (2012) wich was kindly provided by the authors. We start with the observation vector `y` and the matrix `A` which was generated by the `createMatrixA` function. Figure 3.1 illustrates the solution, `p_fits`, for combining two individual probability forecasts p_1, p_2 for 2,000 training data points.

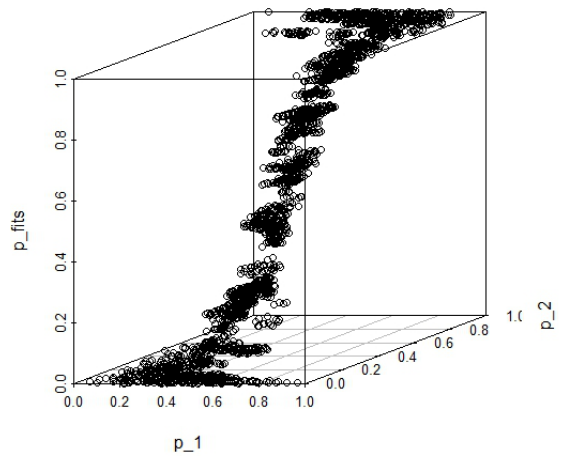

Figure 3.1.: Illustration of the fits in dependence on some training forecasts p_1 and p_2.

First, we check whether or not we can split the whole data set by calling the subfunction `IRP_wrapper` (the code is in the Appendix B). This subfunction puts up the matrix D and the vector d of Section 3.1 out of the matrix `A` and solves the optimization problem (2.13) with the help of the `linprog` and `lpSolve` libraries.

But this method has a limit of about 3,000 observations. If we want to apply this implementation on $5,000$ training data points, we have to implement the `IRP_wrapper` in `C++`. The reason is that the `lpSolve` generates a matrix for solving the linear programme of high dimension. Unfortunately, the dimension of this matrix is too high for `R` at a level greater than 3,000. But for our applications this limit was enough.

If we get a positive feedback that we can split the whole dataset, we update two help vectors, `cutgroups1` and `cutgroups2`, to save the split and a vector `cutvalues` for $c^T x$, defined in (2.14). The vector `cutgroups2` contains the knowledge about the subsets we already have investigated on the possibility to split and the vector `cutgroups1`, the optimal split of them. Consequently, after the first solution of the optimization problem, each entry in `cutgroups2` is equal to 1 and in `cutgroups1` each entry is 1 or 2 depending on the solution vector of the optimization problem. The variables

`cutcount1` and `cutcount2` are just the corresponding control variables for our help vectors. If we compare this approach to Algorithm 1 of Chapter 2, we get that our three help vectors span our set C, see line 3 and 14 of Algorithm 1 in Chapter 2.

After this first split, we start a while loop. At first, we update our variable `maxCutValue`. If all entries of cutvalues are equal to $10,000$, we stop the loop, otherwise we go on. From `cutgroups2`, we get the corresponding indices of the `maxCutValue`. Because we already know how to split set the by the vector `cutgroups1`, we check whether we can split the corresponding subsets further with the `IRP_wrapper`. If not, we set each corresponding entry of `cutvalues` equal to $10,000$. If yes, we update our three help vectors with the new results. In this part, we also update the corresponding entries of our solution vector `p_fits` by taking the mean other the appropriate subset.

This will go on as long as we can split the data set further. Because we know that we have a finite number of observations n and that two already separated sets cannot merge again, we can say with certainty that our algorithm will be finished in a finite period of time, independent of the data set. Finally, it returns the vector `p_fits` with the appropriate solutions of the training data set. This handling is a bit different from Algorithm 1, where we return the set of the indices B. However, we can deduce B by the output vector.

3.2.3. The `ypred`-function

The third subfunction calculates the forecasts for the test data set out of the `p_fits` which were calculated by the `IRP_cut` function. It has as further input the matrices `P_test`, `P` and the value `ImPred`; in this case the mean of the observations of the training data set. For each point of the test data set, we check which forecasts of the training data set are 'below' and which are 'above' the corresponding forecast. In this context, a training forecast is 'below' a test forecast if all single predictive values in the training forecast are smaller than the ones in the test forecast. We define 'above' analogously. Depending on the set of indices of points which are above or below, we calculate the corresponding combined forecast `y_pred`. Thus, we define a control variable `compare`.

If we just have either points below or above a test point, the parameter `compare` is equal to 1. We set the forecast of the test data point equal to the maximum of `p_fits` if there are only points below and equal to the minimum of `p_fits` if we just have points above. Of course, this may not be optimal because there probably exist two forecasts which are very close in the plane, but the difference of the corresponding combined forecasts is very high. But if we want to have isotonicity of the combined test forecasts independent of the training data set this is the only possibility.

If the value of `compare` is equal to 0, which means that we have neither forecasts of the training data set which are below nor above, we set the corresponding forecast value equal to the value `incomPred`, the mean of the observations of the training data

set in this case. This handling of the training data points which have not training data points below or above assures the isotonicity of our solution `y_pred`. Of course, we could set `ImPred` equal to any value in the interval from the smallest to the highest value of `p_fits`.

In most cases, we have training data points which are below and others which are above a corresponding test data point. In this case, the value `compare` is equal to 2. We set the forecast equal to the mean of the maximal solution value of the training forecasts which are below and the minimal solution value of the ones above. After we have calculated the combined forecast for each test data point, we return the vector `y_pred` with the combined forecast.

3.3. The `GIRP`-function

In this section, we give a description of the possibilities of the `GIRP`-function. Extending the `IRP`-function, it contains further parameters that can be useful for combining forecasts. The core of the `GIRP`-function is like the one of the `IRP`-function.

The parameters `y`, `P` and `P_test` are the same as in the previous case. As mentioned in Chapter 2, the GIRP is the generalization of the IRP in the way of generalizing the scoring function under which to minimize. Thus, the `GIRP`-function has a parameter called `loss_func`. The user can decide under which scoring function the solution of the GIRP approach should be calculated. This parameter has to fulfill two conditions:

1. It has to be differentiable in the sense that it is computable by the `R` function `D`.

2. The mode of the scoring function has to be `expression` and it must have the parameters `x` and `a`, where `x` is the forecast and `a` is the observation. For example, if we want to minimize a probabilistic forecast under the logarithmic score for a binary output, we set

```
loss_func=expression(-a*log(x)-(1-a)*log(1-x)).
```

The output is, as in the case of the `IRP`-function, a vector `y_pred_GIRP` with the forecasts for the test data set corresponding to the combined forecasts of the training data set. In the case of probabilistic forecasts, it may be helpful to change these forecasts a bit. If we have, for example, the logarithmic score and we have a combined forecast which is equal to 0 or 1, the probability for getting a penalty of ∞ is normally strictly positive. For avoiding this risk, we may decrease a forecast equal to 1 to the next lower fit and a forecasts equal to 0 to the next higher fit calculated by `IRP_cut`.

Our second extension to the `IRP`-function is that we can decide whether we want to know the number of subsets which were generated and the number of iteration steps

which were necessary for it. In the base setting, the number of subsets is printed, but not the number or iteration steps. By setting `out_iter=T`, the number of iteration steps will be printed, too.

As mentioned in Chapter 2, the problem of statistical overfitting can arise for a high number of training data points. For this reason, we defined the parameter `ch_p_fits`. After each update of the solution (2.10), the `GIRP_cut` function checks, if the comparative improvement of the objective value (2.9) of the previous solutions to the updated solutions is higher than `ch_p_fits`. If not, the corresponding entries of the vector `cutvalues` are set to 10,000 and there will not be a further partition of this set. Otherwise, we will go on as described in Algorithm 1 of Chapter 2. At standard, this parameter is 0 and the algorithm stops if no further splits are possible.

```
GIRP<-function(
y,P,P_test,loss_func,out_iter,out_num,imPred,ch_p_fits){

library(lpSolve)
library(linprog)

## Variable declaration and sorting
n       <- length(P[,1])
m       <- length(P[1,])

p_temp <- order(P[,1])
y       <- y[p_temp]

for (i in 1:m) P[,i] <- P[,i][p_temp]

## Generates a matrix representing the isotonic constraints
A_GIRP <- createMatrixA(P)

## Does the GIRP cut
p_fits_GIRP <- GIRP_cut(y,A_GIRP,loss_func,out_iter,
                                    out_num,ch_p_fits)

## Get a forecast for the test data set
y_pred_GIRP <- yPrediction(P_test,p_fits_GIRP,P,imPred)

return(y_pred_GIRP)
}
```

Listing 3.3: The `GIRP` command

4 Theoretical extensions of the GIRP approach

In Section 2.3, we introduced the IRP approach for solving

$$\underset{g \text{ isotonic}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - g(P_{1i}, .., P_{mi}))^2 \tag{4.1}$$

and the more general GIRP approach for solving

$$\underset{g \text{ isotonic}}{\operatorname{argmin}} \sum_{i=1}^{n} f_{y_i}(g(P_{1i}, .., P_{mi})), \tag{4.2}$$

where $f_{y_i}, i \in \{1, \ldots, n\}$ is a convex and differentiable scoring function.

Further in Chapter 3, we explained our implementation of the IRP approach in `R` in detail. Additionally, we gave a short overview of our general implementation of the GIRP approach in Section 3.3. The disadvantage of the general implementation is that we need further subfunctions for calculating the weights, see Equation 2.10, of the different subsets and consequently the solution of the GIRP approach. Thus, the computational cost is higher.

Hence, from a computational viewpoint, it is useful to find conditions on the scoring functions $f_{y_1}, \ldots, f_{y_n}$ such that the IRP implementation is applicable to solving the optimization problem (4.2). In Section 4.1, we formulate conditions on the scoring functions for binary and general output in $\mathbb{R}$ separately, for which the solution of the GIRP approach is equal to the one of the IRP approach. If there is equality between the solutions of the IRP and GIRP approaches, there may also exist equality within each iteration step. However, we mainly concentrate on the conditions for equality in the case that the algorithm signalizes that a further partition is not beneficial or contradicts the isotonicity of the solution. In the following, we define the 'GIRP-solution' as the solution of the GIRP approach of (4.2). In the same way, we define the 'IRP-solution' as the solution of the IRP approach of (4.1).

As already mentioned in Chapter 2, the GIRP approach refers to the general isotonic regression problem by Barlow and Brunk (1972). Luss and Rosset (2011) proved that under the assumption of differentiability, the generalized isotonic regression problem defined by Barlow and Brunk (1972) can be solved by GIRP. In Section 4.2, we compare the class of general isotonic regression problems by Barlow and Brunk (1972) with the class characterized by the conditions of Section 4.1.2.

For general output, like temperature, wind speed, etc., GIRP is applicable, but not for combining probability forecasts for categorical variables. This leads to another statistical framework with one additional constraint. In Section 4.3, we formulate the statistical framework and investigate whether we can extend the GIRP approach to this framework by characterizing the set of possible solutions.

4.1. Equivalence of the IRP and GIRP approach

As described in Chapter 2, we consider a joint distribution of a binary random variable y and m corresponding probability forecasts. In this section, we formulate and prove conditions on the scoring functions $f_{y_1}, .., f_{y_n}$ such that the GIRP-solution is equal to the IRP-solution in the cases of binary and general output. At first, we consider the binary case with output 0 and after that, in Section 4.1.2, the more general case with output in $\mathbb{R}$.

4.1.1. Equivalence for binary output

Theorem 4.1
Let $Y = (y_1, .., y_n) \in \{0,1\}^n$ be the vector of observations of binary random variables and $P_k = (P_{k1}, .., P_{kn}) \in [0,1]^n, \quad k \in \{1, .., m\}$, the corresponding vector of forecasts for Y issued by the k-th forecaster. Further let

$$f_0,\ f_1 \in C^1((0,1)) \quad \text{be convex and} \quad f_{y_i} = \begin{cases} f_1 & \text{if} \quad y_i = 1 \\ f_0 & \text{otherwise} \end{cases}$$

with:

1. $$f_0'(z) \geq 0, \quad \forall z \in (0,1)$$

2. $$f_1'(z) \leq 0 \quad \forall z \in (0,1)$$

3. $$(-\infty, 0) \subseteq \left\{ \frac{f_0'(z)}{f_1'(z)} \ : \ z \in (0,1),\ f_1'(z) \neq 0 \right\}.$$

Then the structure of the GIRP-solution of (4.2) is equal to the structure of the IRP-solution of(4.1).

A more general formulation of Theorem 1 is the following Lemma 1, which we prove.

Lemma 1
Let $Y = (y_1, .., y_n) \in \{0,1\}^n$ be the vector of observations of binary random variables and $P_k = (P_{k1}, .., P_{kn}) \in [0,1]^n, \quad k \in \{1, .., m\}$, the corresponding vector of forecasts for Y issued by the k-th forecaster. Define $s = \#\{i \,:\, y_i = 1, \;\; i \in \{1, .., n\}\}$ as the number of hits and $\widetilde{s} = n - s$ as the number of misses. Further let

$$f_0,\; f_1 \in C^1((0,1)) \quad \text{be convex and} \quad f_{y_i} = \begin{cases} f_1 & \text{if} \quad y_i = 1 \\ f_0 & \text{otherwise} \end{cases}$$

with:

1.
$$f_0'(z) \geq 0, \quad \forall z \in (0,1)$$

2.
$$f_1'(z) \leq 0 \quad \forall z \in (0,1)$$

3.
$$\exists z_1 \in (0,1) \quad \text{such that} \quad f_1'(z_1) + \widetilde{s}\, f_0'(z_1) < 0$$

4.
$$\exists z_2 \in (0,1) \quad \text{such that} \quad s\, f_1'(z_2) + f_0'(z_2) > 0.$$

Then the structure of the GIRP-solution is equal to the structure of the IRP-solution.

Remarks

1. In this context, the expression "the structure of the GIRP-solution is equal to the structure of the IRP-solution" means that two training cases have the same combined forecast value in the GIRP-solution if and only if they have it in the case of the IRP-solution. However, the corresponding combined forecast value under the IRP-solution may differ from the combined forecast value under the GIRP-solution, which depends on the scoring functions f_0 and f_1.

2. For example, the logarithmic score for a binary output fulfills the conditions of Theorem 1. Consequently, solving (4.1) is equal to solving (4.2) for getting the structure of the GIRP-solution. Further, the logarithmic score and the squared error, on which the IRP approach is based, are both mean-consistent. Therefore, the IRP-solution is equal to the GIRP-solution and in general under the conditions of Lemma 1, thee GIRP-solution with respect to a mean-consistent scoring function is equal to the IRP-solution.

Proof
Idea:

1. Condition (3) and (4) of Lemma 1 guarantee that for each subset

$$V \subseteq \{(y_i, P_{1i}, \ldots, P_{mi}) \ : \ i \in \{1, , .n\}\},$$

there exists a value

$$w_V = \underset{z\in[0,1]}{\operatorname{argmin}} \ \sum_{i\in\widetilde{V}} f_{y_i}(z)$$

for which

$$\sum_{i\in\widetilde{V}} f'_{y_i}(w_V) = 0,$$

where

$$\widetilde{V} = \{i \ : \ (y_i, P_{1i}, \ldots, P_{mi}) \in V\}.$$

2. We prove that the solution of the optimization problem (2.13) is independent of the derivatives of f_0 and f_1 for every subset V of the observations.

3. Theorem 2 of Luss and Rosset (2011) implies that the number of iteration steps is finite. Further, after (2), the full observation set is split equal. After Algorithm 1 of Chapter 2, the corresponding objective value of the two resulting subsets decides which subset of these two we split first.
Consequently, the structures has not to be equal after each iteration step because the GIRP-algorithm for (4.2) may split another subset than the IRP-algorithm splits. However, if the two algorithms of (4.1) and (4.2) signalize that a further partition is not beneficial, the structure of the GIRP-solution is identical to the structure of the IRP-solution.

Proof:
Let V be a subset of the training data set, as defined in Section 2.3, with

$$H = \left\{i \in \widetilde{V} \ : \ y_i = 1\right\} \quad \text{and} \quad M = \left\{i \in \widetilde{V} \ : \ y_i = 0\right\}.$$

We want to split V allowing for the isotonic constraints represented by I_V. The minimization problem to solve is:

$$x^* = \underset{x}{\operatorname{argmin}} \left\{ \sum_{i\in\widetilde{V}} x_i \frac{\partial f_{y_i}(\widehat{y}_i)}{\partial \widehat{y}_i}\bigg|_{w_V} \ : \ x_{i_1} \le x_{i_2}, \ \forall (i_1, i_2) \in I_V, \ \ x_i \in \{-1, +1\} \, \forall i \in \widetilde{V} \right\},$$

where

$$I_V = \left\{(i_1, i_2) : \ \ P_{ki_1} \le P_{ki_2}, \ \ \forall k \in \{1, .., m\}, \ \ i_1, i_2 \in \widetilde{V}\right\}$$

is the set of isotonic constraints of the subset V.

Because of the setting in Lemma 1, we can simplify the sum of the scoring functions to

$$\sum_{i \in \widetilde{V}} f'_{y_i}(z) = s_V f'_1(z) + \widetilde{s}_V f'_0(z),$$

where

$$s_V = \#H \quad \text{and} \quad \widetilde{s}_V = \#M.$$

Now as a consequence of the assumption that $f_0,\ f_1 \in C^1((0,1))$, we know that

$$h(z) = s_V f'_1(z) + \widetilde{s}_V f'_0(z)$$

is a continuous function on the interval (0,1).

With conditions (3) and (4) of Lemma 1 we get

$$\begin{aligned} h(z_1) &= s_V f'_1(z_1) + \widetilde{s}_V f'_0(z_1) \\ &< s_V\ f'_1(z_1) + \widetilde{s}\ f'_0(z_1) \\ &= \underbrace{(s_V - 1)\ f'_1(z_1)}_{\leq 0} + \underbrace{f'_1(z_1) + \widetilde{s}\ f'_0(z_1)}_{<0} \\ &< 0 \end{aligned} \tag{4.3}$$

and

$$\begin{aligned} h(z_2) &= s_V f'_1(z_2) + \widetilde{s}_V f'_0(z_2) \\ &> s\ f'_1(z_2) + \widetilde{s}_V f'_0(z_2) \\ &= \underbrace{s\ f'_1(z_2) + f'_0(z_2)}_{>0} + \underbrace{(\widetilde{s}_V - 1) f'_1(z_1)}_{\geq 0} \\ &> 0 \end{aligned} \tag{4.4}$$

With (4.3), (4.4) and the intermediate value theorem, there has to exist a $w_V \in [z_1, z_2] \subseteq [0,1]$ with $h(w_V) = 0$.

$$\begin{aligned} &\Rightarrow 0 = h(w_V) = s_V f'_1(w_V) + \widetilde{s}_V f'_0(w_V) \\ &\Leftrightarrow f'_0(w_V) = -f'_1(w_V)\frac{s_V}{\widetilde{s}_V} \end{aligned}$$

Applied to (2.13), this implies

$$
\begin{aligned}
x^* &= \underset{x}{\operatorname{argmin}} \left\{ \sum_{i \in \widetilde{V}} x_i f'_{y_i}(z)\big|_{w_V} : \; x_{i_1} \leq x_{i_2} \;\; \forall (i_1, i_2) \in I_V, \;\; x_i \in \{-1, +1\} \, \forall i \in \widetilde{V} \right\} \\
&= \underset{x}{\operatorname{argmin}} \left\{ \sum_{i \in H} x_i f'_1(w_V) - \sum_{i \in M} x_i f'_1(w_V) \frac{s_V}{\widetilde{s}_V} : x_{i_1} \leq x_{i_2} \forall (i_1, i_2) \in I_V, \; x_i \in \{-1, +1\} \right\} \\
&= \underset{x}{\operatorname{argmin}} \left\{ f'_1(w_v) \left(\sum_{i \in H} x_i - \sum_{i \in M} x_i \frac{s_V}{\widetilde{s}_V} \right) : x_{i_1} \leq x_{i_2} \forall (i_1, i_2) \in I_V, \; x_i \in \{-1, +1\} \right\}.
\end{aligned}
$$

Because we have required that $f'_1(z) \leq 0$ on $(0, 1)$ and $w_V \in (0, 1)$, the solution x^* is independent of the derivatives $f'_1(w_v)$, $f'_0(w_v)$ and therefore the structure of the GIRP-solution is equal to the one of the IRP-solution.

□

4.1.2. Equivalence for general output in $\mathbb{R}$

We consider the joint distribution of a random variable y in $\mathbb{R}$ and m corresponding point forecasts. In Theorem 2, we formulate stronger conditions on the scoring functions, as in the previous case, such that the GIRP-solution of (4.2) is equal to the IRP-solution of (4.1).

Theorem 4.2
Let $Y = (y_1, .., y_n) \in \mathbb{R}^n$ be the observations and $P_k = (P_{k1}, .., P_{kn}) \in \mathbb{R}^n, \;\; k \in \{1, .., m\}$ the corresponding forecast of the k-th forecaster for Y. Further, let $f_{y_i} \in C^1(\mathbb{R})$ be convex with the property that the solution w_V, defined by (2.10) is equal to the observation mean for all subsets of the training data set. That is,

$$
w_V = \underset{z \in \mathbb{R}}{\operatorname{argmin}} \sum_{i \in \widetilde{V}} f_{y_i}(z) = \sum_{i \in \widetilde{V}} \frac{y_i}{\#V}, \;\; \forall \, V \subseteq \{y_i \; : \; i \in \{1, .., n\}\}. \tag{4.5}
$$

Then the GIRP-solution is equal to the IRP-solution.

Proof
As in the previous proof, we prove that the solutions of (4.1) and (4.2) are equal for each subset V of the training data set. Theorem 2 then follows as before.

First, we deduce two properties of the scoring functions $f_{y_1}, \ldots, f_{y_n}$ from the conditions of Theorem 2:

1. Because f_{y_i} is convex, f'_{y_i} is monotone non-decreasing. Additionally, $f'_{y_i}(y_i) = 0$

because of the unique solution w_V defined by Theorem 2. Therefore, we achieve

$$f'_{y_i}(z) \begin{cases} < 0 & \text{if } z < y_i \\ > 0 & \text{if } z > y_i \end{cases}$$

2. The previous Property 1 implies that

$$sgn(f'_{y_i}(z)) = sgn(z - y_i).$$

That is, $f'_{y_i}(z)$ is negative for a value z if and only if $z - y_i$ is negative.

As already mentioned in Section 3.2.1., we can interpret the set of isotonic constraints I as a directed graph Γ with vertex set $\{1, .., n\}$ and edge set E defined by

$$(i_1, i_2) \in E \Leftrightarrow (i_1, i_2) \in I.$$

By applying the concept of transitive reduction, we achieve the minimal edge set $\widetilde{E}$ defining I. If we apply this approach on the set of isotonic constraints $I_V \subseteq I$ of V, we achieve a graph $\widetilde{\Gamma}_V$ with vertex set $\widetilde{V}$ and edge set $\widetilde{E}_V$, which represents the minimal set of isotonic constraints defining I_V. A part of such a possible graph is illustrated in Figure 4.1.

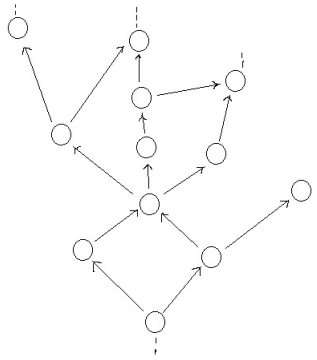

Figure 4.1.: Part of a graph Γ_V which represents the restriction set I_V.

For solving (4.1) and (4.2) by the GIRP approach, we have to solve the optimization problem (2.13). Obviously, the set of isotonic constraints I_V is equal for the corresponding optimization problems. Therefore, we achieve

$$x^* = \underset{x}{\operatorname{argmin}} \left\{ \sum_{i \in \widetilde{V}} x_i \, f'_{y_i} \underbrace{\left(\sum_{i \in \widetilde{V}} \frac{y_i}{\#V} \right)}_{=\overline{y}_V} : x_{i_1} \leq x_{i_2} \quad \forall (i_1, i_2) \in I_V, x_i \in \{-1, +1\}, \; i \in \widetilde{V} \right\} \tag{4.6}$$

as the corresponding optimization problem of the GIRP approach and

$$\widehat{x} = \underset{x}{\operatorname{argmin}} \left\{ \sum_{i \in V} x_i \ (\overline{y}_V - y_i) : \ \ x_{i_1} \leq x_{i_2} \ \ \forall (i_1, i_2) \in I_V, x_i \in \{-1, +1\}, \ i \in \widetilde{V} \right\} \tag{4.7}$$

as the optimization problem of the IRP approach.

We prove $x^* = \widehat{x}$ by contradiction, which implies directly that the subset V is split equally. Therefore, we define

$$G = \left\{ i \in \widetilde{V} \ x_i^* \neq \widetilde{x}_i \right\}$$

as the set of indices for which the corresponding components differ and assume $G \neq \emptyset$.

- $G = j, \ j \in \widetilde{V}$

  At first, we consider that G only contains one indice and thus the both solutions only differ in one component. Because I_V is equal for both optimization problems and because the isotonic constraints are not contradicted, the j-th components of $\widehat{x}$ and x^* can independently be chosen to be positive or negative.

  Assume $\widehat{x}$ and x^* are optimal. Therefore,

  $$\sum_{i \in \widetilde{V}} \widehat{x}_i (\overline{y}_V - y_i) \leq \sum_{i \in \widetilde{V} \setminus j} \widehat{x}_i (\overline{y}_V - y_i) - \widehat{x}_j (\overline{y}_V - y_i)$$

  and

  $$\sum_{i \in \widetilde{V}} x_i^* f'_{y_i}(\overline{y}) \leq \sum_{i \in \widetilde{V} \setminus j} x_i^* f'_{y_i}(\overline{y}) + \widehat{x}_j f'_{y_j}(\overline{y}).$$

  Because of Property 2, $\operatorname{sgn}(f'_{y_j}(\overline{y})) = \operatorname{sgn}(\overline{y} - y_j)$, we get either

  $$\sum_{i \in \widetilde{V}} x_i^* f'_{y_i}(\overline{y}) \geq \sum_{i \in \widetilde{V} \setminus j} x_i^* f'_{y_i}(\overline{y}) - \widehat{x}_j f'_{y_j}(\overline{y}) \qquad \lightning x^* \text{ optimal}$$

  or

  $$\sum_{i \in \widetilde{V}} \widehat{x}_i (\overline{y}_V - y_i) \geq \sum_{i \in \widetilde{V} \setminus j} \widehat{x}_i (\overline{y}_V - y_i) - \widehat{x}_j (\overline{y}_V - y_i) \qquad \lightning \widehat{x} \text{ optimal.}$$

  Thus, either x^* or $\widehat{x}$ cannot be optimal and therefore $x^* = \widehat{x}$.

- $\#G > 1$

  One property of the set G we can deduce, is the following:
  Intuitively, if we have two components $i_1, i_2 \in G$ of the solution vectors x^*, $\widehat{x}$ with the property $(i_1, i_2) \in I_V$, we achieve

1.

$$x^*_{i_1} = 1 \overset{(i_1,i_2)\in I_V}{\Rightarrow} x^*_{i_2} = 1 \overset{i_2\in G}{\Rightarrow} \widehat{x}_{i_2} = -1 \overset{(i_1,i_2)\in I_V}{\Rightarrow} \widehat{x}_{i_1} = -1$$

and

$$x^*_{i_1} = -1 \overset{i_1\in G}{\Rightarrow} \widehat{x}_{i_1} = 1 \overset{(i_1,i_2)\in I_V}{\Rightarrow} \widehat{x}_{i_2} = 1 \overset{i_2\in G}{\Rightarrow} x^*_{i_2} = -1.$$

2. Each further component with the property $(i_1, i_3) \in I_V$, $(i_3, i_2) \in I_V$ implies that $i_3 \in G$, because

$$(i)\ \ x^*_{i_1} \leq x^*_{i_3} \leq x^*_{i_2} \wedge x^*_{i_1} = x^*_{i_2} \Rightarrow x^*_{i_1} = x^*_{i_3}$$

and

$$(ii)\ \widehat{x}_{i_1} \leq \widehat{x}_{i_3} \leq \widehat{x}_{i_2} \wedge \widehat{x}_{i_1} = \widehat{x}_{i_2} \Rightarrow \widehat{x}_{i_1} = \widehat{x}_{i_2}$$

Formally, let i_1, $i_2 \in G$ with $(i_1, i_2) \in I$ then

$$i_3 \in \widetilde{V} \ : \ (i_1, i_3) \in I_V,\ (i_3, i_2) \in I_V \ \ \Rightarrow \ \ i_3 \in G.$$

Graphically this means, if there exists a path from i_1 to i_2, $i_1, i_2 \in G$, then all vertices on the path have to be in G too. Thus, we achieve a finite number of disjunct subgraphs with the property that there exists no path between any. Otherwise, we can combine them by the proved properties. Consequently, for example, we get subgraphs as illustrated in Figure 4.2. We investigate each one separately.

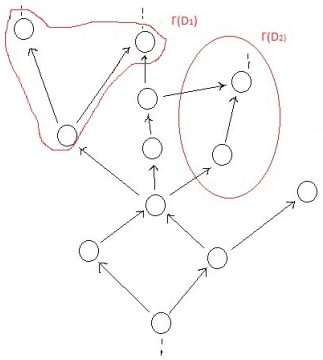

Figure 4.2.: Possible structure of the subgraphs representing G.

Let G_1 be the corresponding indices of a subset of G which characterize an arbitrary subgraph described by the properties above. Suppose that x^* and $\widehat{x}$

are optimal in the sense that they minimize the corresponding objective values. Because of the identical set of isotonic constraints for (4.6) and (4.7), we can choose for the corresponding indices G_1, if the corresponding x_i^* should be all negative or positive.

Because of the optimality of x^* and $\widehat{x}$, we deduce

$$\operatorname{sgn}\left(\sum_{i\in G_1}\widehat{x}_i\left(\overline{y}_V - y_i\right)\right) = -\operatorname{sgn}\left(\sum_{i\in G_1} x_i^* f'_{y_i}(\overline{y})\right).$$

Otherwise, we would change either x_i^* or $\widehat{x}$ for getting a smaller objective value. However, this leads to an contradiction to the Properties 1 and 2 at the beginning of this proof as follows:

Following the unique w_V defined by Theorem 2,

$$\sum_{i\in G_1} f'_{y_i}\left(\sum_{i\in G_1}\frac{y_i}{\#G_1}\right) = \sum_{i\in G_1}\left(\sum_{i\in G_1}\frac{y_i}{\#G_1} - y_i\right) = 0.$$

Further, $\sum_{i\in G_1} f_{y_i}$ is a finite sum of convex functions and therefore also convex. With Property 1 and 2 of the beginning of the proof, we conclude

$$\operatorname{sgn}\left(\sum_{i\in G_1}\widehat{x}_i\left(\overline{y}_V - y_i\right)\right) = \operatorname{sgn}\left(\sum_{i\in G_1} x_i^* f'_{y_i}(\overline{y})\right). \qquad \lightning \text{ Property 2}$$

Therefore, within the corresponding indices of G_1, the solution vectors of (4.6) and (4.7) do not differ. Because we investigated an arbitrary subgraph, the corresponding solution vectors x^* and $\widetilde{x}$ are identical. Consequently, the subset V is split equally. Because we investigated an arbitrary subset V, we have completely proved that the structure of the both solutions is equal.

The equivalence of the solutions follows from the assumption on the solution w_V. In the case of IRP, we calculated the fit of a given training data point by calculating the mean observation value of the corresponding subset V that contains the training data point. From the assumption on w_V we know that for the GIRP-solution, the corresponding fit for the training data point is also equal to the mean observation value of the corresponding subset in that it is contained. Because we have demonstrated that the corresponding subsets are equal, the fits are equal. Therefore, the IRP-solution and the GIRP-solution are equal and the proof is complete.

□

Remarks

1. A lot of commonly used scoring functions, fulfill the properties of Theorem 4.2. Therefore, it is often possible to perform the IRP-algorithm instead of the GIRP-algorithm.

2. There exists a more general version of Theorem 4.2 in the sense that we do not require differentiability on $\mathbb{R}$. We only use the differentiability in the optimization problem and we restrict the weight w_V to be equal to the mean of the observations of the corresponding subset V. Therefore, we only need differentiability of the scoring functions in the observation means of all possible subsets of the training data set.

3. For example, suppose we have output in $\mathbb{N}_0$, generated by Poisson distributed random variables and we use the negative Poisson log likelihood,

   $$f_{y_i}(z) = z - y_i \ln(z),$$

   for evaluating the forecasts. Because of the general version of Theorem 4.2, the GIRP-solution is equal to the IRP-solution.

4.2. Relations to the paper by Barlow and Brunk

In this section, we investigate the relation between the class of scoring functions of Section 4.1, for which the GIRP-solution is equal to the IRP-solution, and the general isotonic regression problem by Barlow and Brunk (1972). Like Luss and Rosset (2011), we just compare the members of Barlow and Brunk (1972) which are differentiable.

After Section 2.4 of Luss and Rosset (2011) and as mentioned in Section 2.3, each general isotonic regression problem of the form

$$\widehat{z}^* = \underset{z}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \Phi(\widehat{z}_i) - \widehat{z}_i y_i : \widehat{z}_{i_1} \leq \widehat{z}_{i_2} \ \forall (i_1, i_2) \in I \right\}$$

can be solved by applying the IRP-algorithm to the dataset and using the transformation

$$\widehat{z}_i^* = \phi^{-1}(\widehat{y}_i^*),$$

for getting the GIRP-solution. In this context, ϕ is the derivative of the convex function Φ and $\widehat{y}_i^*$ the solution of the IRP-algorithm.

At first we investigate whether we have any general isotonic regression problems for which the GIRP-solution is equal to the IRP-solution. As we can see, the intersection is the set of general isotonic regression problems with $\phi^{-1} = id$, but we want to prove

this formally.

Let $Y = (y_1, .., y_n) \in \mathbb{R}^n$ be an arbitrary vector. We look at the class of scoring functions with

$$f_{y_i}(z) = \Phi(z) - y_i z,$$

where $\Phi(z)$ is a differentiable and convex function and

$$\underset{z \in \mathbb{R}}{\operatorname{argmin}} \sum_{i \in \widetilde{V}} f_{y_i}(z) = \sum_{i \in \widetilde{V}} \frac{y_i}{\#V}, \forall \ V \subseteq \{y_1, \ldots, y_n\}.$$

We look which conditions on $\Phi(z)$ are necessary for the property

$$\underset{z \in \mathbb{R}}{\operatorname{argmin}} \sum_{i \in \widetilde{V}} f_{y_i}(z) = \sum_{i \in \widetilde{V}} \frac{y_i}{\#V}, \forall \ V.$$

It follows

$$\begin{aligned}
\underset{z \in \mathbb{R}}{\operatorname{argmin}} \sum_{i \in V} f_{y_i}(z) &= \underset{z \in \mathbb{R}}{\operatorname{argmin}} \sum_{i \in V} \Phi(z) - y_i z, \\
&\Leftrightarrow 0 = \sum_{i \in \widetilde{V}} \phi(\widehat{z}) - y_i, \\
&\Leftrightarrow \#V \phi(\widehat{z}) = \sum_{i \in \widetilde{V}} y_i, \\
&\Leftrightarrow \widehat{z} = \phi^{-1}\left(\sum_{i \in \widetilde{V}} \frac{y_i}{\#V} \right) \overset{!}{=} \sum_{i \in \widetilde{V}} \frac{y_i}{\#V}.
\end{aligned}$$

Because $Y = (y_1, .., y_n) \in \mathbb{R}^n$ is arbitrary this implies that

$$\phi^{-1}(\widehat{z}) = id(\widehat{z}).$$

Consequently, the intersection between the class of general isotonic regression problems by Barlow and Brunk (1972) and the class defined in Theorem 4.2 is a modified squared error.

Remark
Another interesting question in this context comes up if we investigate the negative logarithmic score for binary events:

$$\underset{\widehat{p}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} -y_i \log(\widehat{p}_i) - (1 - y_i) \log(1 - \widehat{p}_i) : \widehat{p}_{i_1} \leq \widehat{p}_{i_2} \quad \forall (i_1, i_2) \in I \right\}. \tag{4.8}$$

In the context of IRP, we set

$$\widehat{p}_i = g(P_{1i}, .., P_{mi}).$$

By substitution with

$$z_i = \log\left(\frac{g(P_{1i}, .., P_{mi})}{(1 - g(P_{1i}, .., P_{mi}))}\right),$$

the solution of (4.9) is equivalent to solve

$$\widehat{z} = \underset{z}{\operatorname{argmin}} \sum_{i=1}^{n} \log(1 + e^{z_i}) - z_i y_i, \;\; z_{i_1} \leq z_{i_2} \; \forall \, (i_1, i_2) \in I.$$

The function $\Phi(z_i) = \log(1 + e^{z_i})$ is strictly convex. Hence, the related general isotonic regression problem is a member of the class defined by Barlow and Brunk (1972). The negative logarithmic score in (4.8) is otherwise a member of the our class defined in Theorem 4.2.

Thus the question in this context is, if it is possible to convert each member of the class of scoring functions defined in Theorem 4.2 to a function of the class by Barlow and Brunk (1972) or conversely under the assumption that Φ is differentiable. Then we would have that the intersection between the functions which can be solved by GIRP and the isotonic regression problems solved by Barlow and Brunk (1972) could by substitution be represented by our class defined in Theorem 2.

So we check, whether for each member of the class of general isotonic regression problems, there exists a differentiable function γ with

$$z^* = \underset{z \in \mathbb{R}}{\operatorname{argmin}} \sum_{i \in \widetilde{V}} \Phi(\gamma(z)) - y_i \gamma(z) = \sum_{i \in \widetilde{V}} \frac{y_i}{n}, \;\; \forall \, V \subseteq \{y_i \; : \; i \in \{1, .., n\}\}. \tag{4.9}$$

We assume that γ is differentiable.

$$\begin{aligned}
(4.10) &\Leftrightarrow 0 = \sum_{i=1}^{n} \Phi'(\gamma(z^*))\gamma'(z^*) - \gamma'(z^*) y_i \\
&\Leftrightarrow 0 = \sum_{i=1}^{n} \Phi'(\gamma(z^*)) - y_i \\
&\Leftrightarrow \sum_{i=1}^{n} \frac{y_i}{n} = \Phi'(\gamma(z^*)) = \phi(\gamma(z^*))
\end{aligned}$$

Because we aim for

$$z^* = \sum_{i=1}^{n} \frac{y_i}{n},$$

we get that

$$\gamma = \phi^{-1}.$$

For this conclusion we have to assume that ϕ is differentiable. Because Φ is convex, ϕ is monotonically nondecreasing. Thus $\Phi(\phi^{-1}(z)) - \phi^{-1}(z) y_i$ is also a convex function. Consequently we can say, if Φ is two times differentiable we can convert by substitution each member of the class of general isotonic regression problems to a member of the class of scoring functions for which the IRP-solution is equal to the GIRP-solution.

4.3. Aspects of combining probability forecasts for categorical variables

In this section we investigate, whether the IRP approach explained in Section 2.3 can be used for combining probability forecasts for categorical events. Thus, we reformulate the framework of Section 2.1 and investigate resulting properties and consequences for an implementation. At first, we want to motivate this by an example.

Suppose, we want to know, whether tomorrow's temperature T will be below 5 degrees, between 5 and 15 degrees or above 15 degrees. Therefore, we ask m forecasters for their prediction and get $\underline{p_1}, .., \underline{p_m}$. Each forecaster gives us a probability forecast of the form

$$\underline{p_k} = (\underline{p_k}_{T<5}, \underline{p_k}_{T\in[5,15]}, \underline{p_k}_{T>15}), \;\; k \in \{1, .., m\}$$

and we want to combine these single forecasts for getting a better forecast. In contrast to the statistical framework in Section 2.1, we get the additional condition that the 1-norm of our combined probability forecast vector has to be equal to 1. Thus, the statistical framework of Section 2.1 changes in the following way.

We work within a probabilistic framework which considers a joint distribution of

$$(\underline{y}, \underline{p_1}, .., \underline{p_m}),$$

where $\underline{y} \in \{0,1\}^l$ is a random vector of length l which has entries zero except in one, where it has entry 1. Further, $\underline{p_1}, \ldots, \underline{p_m} \in [0,1]^l$ are probability forecast vectors of m forecasters for $\underline{y}$ with 1-norm 1. Transfered to the example, we have $l = 3$ and $\underline{y} = (y_{T<5}, y_{T\in[5,15]}, y_{T>15})$ and $\underline{p_1}, \ldots, \underline{p_m}$ are the corresponding forecasts of the m forecasters.

The theoretically optimal combined forecast $\underline{p}^*$ in the sense that

$$\mathbb{E}\left\|\underline{p}^* - \underline{y}\right\|_2^2 \leq \mathbb{E}\left\|\underline{p} - \underline{y}\right\|_2^2,$$

for any measurable $\underline{p}$ with respect to the σ-algebra generated by $(\underline{p_1}, \ldots, \underline{p_m})$, is the conditional expectation of the random vector $\underline{y}$ given the probability forecasts $\underline{p_1}, \ldots, \underline{p_m}$, that is,

$$\underline{p}^* = \mathbb{E}(\underline{y}|\underline{p_1}, \ldots, \underline{p_m}) \tag{4.10}$$

But as in the binary case, the optimal combined probability forecast is in general unknown and has to be estimated from a training data set.

Like Luss and Rosset (2011), we want to fit a model describing the dependence of the observation vectors $\underline{y_1}, .., \underline{y_n}$ concerning the corresponding forecast vectors. As in the binary case, it seems useful to assume that in case each forecaster gives us a higher

probability forecast value for the output j than in another case, the corresponding combined probability forecasts for this output should achieve this property. So we transform the isotonicity of the one-dimensional case, to an isotonicity in each component.

Hence, we are searching for a function

$$\begin{aligned} g: \quad & [0,1]^{l\times m} \to [0,1]^l \\ & \left(\underline{p_1},..,\underline{p_m}\right) \mapsto g\left(\underline{p_1},..,\underline{p_m}\right) \\ & \qquad = \left(\left[g\left(\underline{p_1},..,\underline{p_m}\right)\right]_1,..,\left[g_l\left(\underline{p_1},..,\underline{p_m}\right)\right]_m\right), \end{aligned} \tag{4.11}$$

with the properties

1. $$\underline{p_k}_j \leq \underline{\widehat{p}_k}_j \;\; \forall\, k \in \{1,..,m\} \Rightarrow [g(\underline{p_1},..,\underline{p_m})]_j \leq [g(\underline{\widehat{p}_1},..,\underline{\widehat{p}_m})]_j, \tag{4.12}$$
   where $\underline{p_k}_j$ is the probability forecast of the k-th forecaster for the output j. So with $[g^*(\underline{p_1},..,\underline{p_m})]_j$ we refer to the j-th component of the combined probability forecast vector and the function g should be isotonic in each component.

2. $$\sum_{j=1}^{l}[g(\underline{p_1},..,\underline{p_m})]_j = 1 \tag{4.13}$$
   This is the condition that the sum of the single component forecasts of one forecaster should be equal to one.

Out of these two properties, we can already conclude an important property of the function g, which we formulate in the following Lemma 2.

Lemma 2
If g is a function with the properties (4.12) and (4.13) which is non-constant in one component and the number of components l is greater than 2, then the following holds:

1. g is non-constant in each component
2. If for one case, each forecaster hands in a greater prediction value for the output j than in another case, the aggregated forecasts for the j-th component differ. That is, following the argument of isotonicity,
   $\underline{p_k}_j < \underline{\widehat{p}_k}_j, \forall k \in \{1,..,m\} \Rightarrow [g(\underline{p_1},..,\underline{p_m})]_{j_1} < g[(\underline{\widehat{p}_1},..,\underline{\widehat{p}_m})]_{j_1}$

Remarks

In Chapter 2, this strict relation is not true. There it could be that each forecaster gives us in the one case a higher forecast than in the other case, but the combined forecast values could be the same. Now we prove that this is not true in the framework described before and we have a strict relation.

Because of a lot of indices, we will prove Lemma 2 for the case that we have only two forecasters, $\check{\mathrm{Z}}m = 2$. However, for higher m the proof is analogous. In the following we refer to the forecast of the j-th component of the k-th forecaster in the i-th case by $\underline{p_k}_j^{\,i}$.

Proof

Without loss of generality, we may suppose that g is non-constant in the first component. Assume, we have two forecasts $\underline{p}^{(1)} = (\underline{p_1}^{(1)}, \underline{p_2}^{(1)})$, $\underline{p}^{(2)} = (\underline{p_1}^{(2)}, \underline{p_2}^{(2)}) \in [0,1]^{2\times l}$, where the 1-norm of each forecast vector is equal to 1. Further, these two probability forecast vectors have to satisfy the following properties:

1. Each forecaster in $\underline{p}^{(2)}$ has a higher forecast value in the first component than in $\underline{p}^{(1)}$, that is
$$\underline{p_k}_1^{(1)} < \underline{p_k}_1^{(2)}, \quad \forall k \in \{1,2\}\,.$$

2. The combined probability for the first component is equal for both forecasts, that is
$$[g(\underline{p}^{(1)})]_1 = [g(\underline{p}^{(2)})]_1.$$

In the second part of the proof, we will prove that this is a contradiction to the properties (4.12) and (4.13). However, at first we define a function

$$\begin{aligned}\vec{\delta}_{j_1,j_2} : \mathbb{R} &\mapsto \mathbb{R}^l \\ \delta &\mapsto \vec{\delta}_{j_1,j_2}(\delta) = (0,\cdots,0,\delta,0\cdots 0,-\delta,0,\cdots,0)^{\mathrm{T}},\end{aligned}$$

that is, we increase the j_1-th component of the zero vector by δ and decrease the j_2-th component of the zero vector by δ.

1. Suppose that an arbitrary component j^* of the function g, $[g(\cdots)]_{j^*}$, is constant fulfilling property (4.13). Because we assumed that the function g is non-constant in the first component, there exists a value $\delta^* = (\delta_1^*, \delta_2^*)$ with
$$[g(\underline{p_1}^{(1)} - \vec{\delta}_{1,j^*}(\delta_1^*), \underline{p_2}^{(1)} - \vec{\delta}_{1,j^*}(\delta_2^*))]_1 \neq [g(\underline{p_1}_1^{(1)}, \underline{p_2}^{(1)})]_1.$$
However,
$$[g(\underline{p_1}^{(1)} - \vec{\delta}_{1,j^*}(\delta_1^*), \underline{p_2}^{(1)} - \vec{\delta}_{1,j^*}(\delta_2^*))]_{j^*} = [g(\underline{p_1}_1^{(1)}, \underline{p_2}^{(1)})]_{j^*}.$$
This is a contradiction to property (4.13), because the other components do not change, but the sum of the first and j^*-th component changes. Consequently, we proved the first statement of Lemma 2.

2. Let $\underline{p}^{(3)} = (\underline{p_1}^{(3)}, \underline{p_2}^{(3)})$ be a third forecast with the property that each forecast for the first component is higher than in $\underline{p_1}^{(1)}$ but smaller than $\underline{p_1}^{(2)}$, that is

$$\underline{p_1}_1^{(1)} < \underline{p_1}_1^{(3)} < \underline{p_1}_1^{(3)}$$

and

$$\underline{p_2}_1^{(1)} < \underline{p_2}_1^{(3)} < \underline{p_2}_1^{(3)}.$$

Because of condition (4.13), we get

$$[g(\underline{p}^{(1)})]_1 = [g(\underline{p}^{(3)})]_1 = [g(\underline{p}^{(2)})]_1. \tag{4.14}$$

By conditions (4.12) and (4.13) and the property that g is non-constant in each component by part (1) of Lemma 3, we get that there exists a value $\delta^* = (\delta_1^*, \delta_2^*)$ such that

$$[g(\underline{p_1}^{(1)} - \vec{\delta}_{1,j^*}(\delta_1^*), \underline{p_2}^{(1)} - \vec{\delta}_{1,j^*}(\delta_2^*))]_1 \neq [g(\underline{p_1}_1^{(1)}, \underline{p_2}^{(1)})]_1, \tag{4.15}$$

but

$$[g(\underline{p_1}^{(1)} - \vec{\delta}_{1,j^*}(\delta_1^*/2), \underline{p_2}^{(1)} - \vec{\delta}_{1,j^*}(\delta_2^*/2))]_1 = [g(\underline{p_1}_1^{(1)}, \underline{p_2}^{(1)})]_1. \tag{4.16}$$

That means, in the neighborhood of $\underline{p}^{(1)}$ exists a further forecast which generates the same forecast, but in the same direction, we get to a further forecast, where the combined forecast differs. It may happen that $\underline{p_1}^{(1)}$ is a little bit special, in the sense that it lies on the border between two sets of different forecast values. Then, we search a point in the neighborhood of $\underline{p_1}^{(1)}$ and perform the following steps in the same way.

Further, Condition (4.14) implies that $\forall j^* \in \{2, .., l\}$

$$[g(\underline{p_1}^{(1)} - \vec{\delta}_{1,j^*}(\delta_1^*), \underline{p_2}^{(1)} - \vec{\delta}_{1,j^*}(\delta_2^*))]_{j^*} \neq [g(\underline{p_1}_1^{(1)}, \underline{p_2}^{(1)})]_{j^*},$$

but

$$[g(\underline{p_1}^{(1)} - \vec{\delta}_{1,j^*}(\delta_1^*/2), \underline{p_2}^{(1)} - \vec{\delta}_{1,j^*}(\delta_2^*/2))]_{j^*} = [g(\underline{p_1}^{(1)}, \underline{p_2}^{(1)})]_{j^*}.$$

Consequently, we achieve

$$\begin{aligned} &[g(\underline{p_1}^{(1)} - \vec{\delta}_{1,j^*}(\delta_1^*/2), \underline{p_2}_2^{(1)} - \vec{\delta}_{1,j^*}(\delta_2^*/2))]_1 - [g(\underline{p_1}_1^{(1)}, \underline{p_2}^{(1)})]_1 \\ &\quad = [g(\underline{p_1}_1^{(1)}, \underline{p_2}^{(1)})]_{j^*} - [g(\underline{p_1}^{(1)} - \vec{\delta}_{1,j^*}(\delta_1^*/2), \underline{p_2}^{(1)} - \vec{\delta}_{1,j^*}(\delta_2^*/2))]_{j^*}. \end{aligned} \tag{4.17}$$

and

$$[g(\underline{p_1}^{(1)} - \vec{\delta}_{1,j^*}(\delta_1^*/2), \underline{p_2}_2^{(1)} - \vec{\delta}_{1,j^*}(\delta_2^*/2))]_{j_1} - [g(\underline{p_1}_1^{(1)}, \underline{p_2}_2^{(1)})]_{j_1} = 0. \tag{4.18}$$

Therefore, we get a very strong condition on the function g. Because we have at least three components, we can also increase our forecast in the first and decrease in two other components. In this case, we increase the first forecast by δ^* and decrease the second and third component by $\delta^*/2$. This leads, together with property (4.12) and (4.13), to

$$\begin{aligned}
&[g(\underline{p_1}^{(1)} - \vec{\delta}_{1,j^*}(\delta_1^*/2), \underline{p_2}_2^{(1)} - \vec{\delta}_{1,j^*}(\delta_2^*/2))]_1 - [g(\underline{p_1}_1^{(1)}, \underline{p_2}^{(1)})]_1 \\
=&[g(\underline{p_1}^{(1)} - \vec{\delta}_{1,j^*}(\delta_1^*/2), \underline{p_2}_2^{(1)} - \vec{\delta}_{1,j^*}(\delta_2^*/2))]_2 - [g(\underline{p_1}^{(1)}, \underline{p_2}^{(1)})]_2 \\
&+ [g(\underline{p_1}^{(1)} - \vec{\delta}_{1,j^*}(\delta_1^*/2), \underline{p_2}_2^{(1)} - \vec{\delta}_{1,j^*}(\delta_2^*/2))]_3 - [g(\underline{p_1}^{(1)}, \underline{p_2}^{(1)})]_3 \\
=&0 \quad \lightning \quad (4.13)
\end{aligned}$$

Therefore, the assumption that the combined forecast values of $\underline{p}^{(1)}$ and $\underline{p}^{(2)}$ are equal is contradicted and consequently Lemma 2 is proved.

□

Based on Lemma 2, we can conclude even stronger properties of g, which we do in two steps in the following Conclusion. As in the proof before, we set $m = 2$ because of a lot of indices. Further, we set the number of categories l equal to three.

Conclusion

1. After Lemma 3, two forecasts $\underline{p}^{(1)}$ and $\underline{p}^{(2)}$ can only have the same combined forecast value in the first component if

$$\underline{p_1}_1^{(1)} \leq \underline{p_1}_1^{(2)} \wedge \underline{p_2}_2^{(1)} \geq \underline{p_2}_2^{(2)},$$

or

$$\underline{p_1}_1^{(1)} \geq \underline{p_1}_1^{(2)} \wedge \underline{p_2}_2^{(1)} \leq \underline{p_2}_2^{(2)}.$$

Consequently, the set of points which generate the same combined forecast for the first component are limited to a line in $[0,1]^2$ constrained by the property above.

One example is illustrated in Figure 4.3 for the first component. The x-axis represents the probability forecasts of the first and the y-axis the probability forecasts of the second forecaster for the first output.

2. We prove that our set of possible solutions for g under the given isotonic constraints and the assumption that g is non-constant in each component is limited to the case that g is linear in each component.

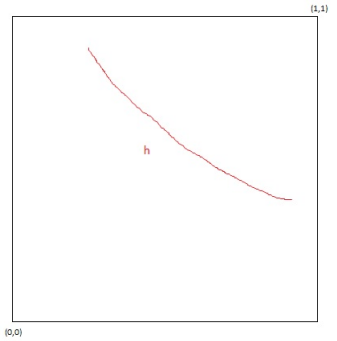

Figure 4.3.: Example of a set, for which the combined probability forecasts may be equal.

Consider the forecast $\underline{p}^{(1)} = (\underline{p_1}^{(1)}, \underline{p_2}^{(1)})$ as above. Further, let $\delta = (\delta_1, \delta_2)$ and $\vec{\delta}_{j_1,j_2}(\delta)$ be defined as in the proof before. For simplicity, we define

$$[\Delta G_{j_1,j_2}(\delta)]_{j_3} = [g(\underline{p}^{(1)})]_{j_3} - [g(\underline{p_1}^{(1)} - \vec{\delta}_{j_1,j_2}(\delta_1), \underline{p_2}^{(1)} - \vec{\delta}_{j_1,j_2}(\delta_2))]_{j_3}$$

as the difference in the j_3-th component if we change the probability forecasts for the j_1-th component by δ^* and the probability forecasts for the j_2-th component by $-\delta^*$. Of course, this value can only be non-zero if $j_3 \in j_1, j_2$. Otherwise, we would have a contradiction to condition (4.12).

Let δ, j_1, j_2 and j_3 be arbitrary but fixed. Then we get by the condition (4.13) that

$$[\Delta G_{j_1,j_2}(\delta)]_{j_1} = -[\Delta G_{j_1,j_2}(\delta)]_{j_2} = -[\Delta G_{j_1,j_3}(\delta)]_{j_3} = [\Delta G_{j_1,j_3}(\delta)]_{j_1}. \tag{4.19}$$

So if we change the forecasts in the j_1-th component by δ and the j_2-th component respectively the j_3-th component by $-\delta$ the change has to be equal for the j_2 and j_3-th component. This is the same property as (4.17). This of course holds for the case that we increase the j_2-th component by δ and decrease one of the other two components by δ. Thus

$$[\Delta G_{j_2,j_3}(\delta)]_{j_2} = -[\Delta G_{j_2,j_3}(\delta)]_{j_3} = -[\Delta G_{j_2,j_1}(\delta)]_{j_1} = [\Delta G_{j_1,j_2}(\delta)]_{j_1}.$$

The last equality holds, because a change in the j_3th component by $-\delta$, should be independent of the second component which is changed.

Therefore,

$$[\Delta G_{j_2,j_3}(\delta)]_{j_3} = [\Delta G_{j_1,j_3}(\delta)]_{j_3} \overset{(4.17)}{=} [\Delta G_{j_1,j_2}(\delta)]_{j_2}. \tag{4.20}$$

We change the forecast in the j_1-th and the j_2-th component by $\delta/2$ and the j_3-th component by $-\delta$ then we get by condition (4.14):

$$[\Delta G_{j_1,j_3}(\delta/2)]_{j_1} + [\Delta G_{j_2,j_3}(\delta/2)]_{j_2} = -[\Delta G_{j_2,j_3}(\delta)]_{j_3}.$$

$$\overset{(4.17),(4.18)}{\Leftrightarrow} [\Delta G_{j_1,j_3}(\delta/2)]_{j_1} + [\Delta G_{j_1,j_3}(\delta/2)]_{j_1} = [\Delta G_{j_2,j_3}(\delta)]_{j_2}$$
$$\Leftrightarrow 2[\Delta G_{j_1,j_3}(\delta/2)]_{j_1} = [\Delta G_{j_1,j_3}(\delta)]_{j_1}.$$
$$\Rightarrow 2[\Delta G_{j_1,j_3}(\delta/2)]_{j_3} = [\Delta G_{j_1,j_3}(\delta)]_{j_3}$$
$$\Rightarrow 2[\Delta G_{j_1,j_2}(\delta/2)]_{j_2} = [\Delta G_{j_1,j_2}(\delta)]_{j_2}.$$

If we go on in the same way with $\delta/4, \delta/8, \ldots$ we get that g is linear in the direction δ in the point $(\underline{p_1}, \underline{p_2})$ in an countable number of points. If we have

$$4[\Delta G_{j_j,j_3}(\delta/4)]_{j_1} = [\Delta G_{j_1,j_3}(\delta)]_{j_1}$$

we also get

$$3[\Delta G_{j_1,j_3}(\delta/4)]_{j_1} = [\Delta G_{j_1,j_3}(3\delta/4)]_{j_1}$$

because

$$[\Delta G_{j_1,j_3}(\delta/2)]_{j_1} + [\Delta G_{j_1,j_2}(\delta/4)]_{j_1} = [\Delta G_{j_1,j_3}(3\delta/4)]_{j_1} = 3[\Delta G_{j_1,j_3}(\delta/4)]_{j_1}.$$

Because δ was arbitrary, we get that g is linear in all directions in which the function g is non-constant. Further, we have said at the beginning that the forecast $\underline{p}^{(1)}$ is arbitrary but fixed and so we get that it is linear in each point. This implies that g has to be linear in each component, because j_1, j_2, j_3 were arbitrary but fixed too.

We just have to think about why it is enough that the g is linear in a countable set of points. This follows by Lemma 3 we proved before. Because there exists a direction in which we have the strict relation. We further know that we can get as near as we want to each point, because we can choose $\delta/2^n$, n great enough. Consequently, our solution set of isotonic functions, under the assumption that the solution is non-constant in each component, is strongly limited to the class of functions which are linear in each component.

Next, we consider our data set. Let

$$(\underline{Y}, \underline{P_1}, \ldots, \underline{P_m})$$

be a given training data set of size n, where

$$\underline{Y} = (\underline{y_1}, \ldots, \underline{y_n})^{\mathrm{T}} \in \{0,1\}^{l \times n}$$

are the observations and

$$\underline{P_k} = (\underline{P_k}_{\,1}, \ldots, \underline{P_k}_{\,n})^{\mathrm{T}} \in [0,1]^{l \times n}, \;\; k \in \{1, \ldots, m\}$$

$$\underline{P_k}_{\,i} = (\underline{P_k}_{\,i}^{\,1}, .., \underline{P_k}_{\,i}^{\,l}) \in [0,1]^l$$

are the corresponding forecasts of m forecasters for $\underline{Y}$. So we define $\underline{P_k}_{\,i}^{\,j}$ as the probability forecast of the k-th forecaster in the i-th case for the output j, $j \in \{1, \ldots, l\}$.

Additionally, our function g^* defined by (4.11), (4.12) and (4.13) should minimize an objective value. In this section, we define this objective value as the General Brier score. Thus we get the following minimization problem:

$$g^* = \underset{g \text{ with}(4.12),(4.13)}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| \underline{y_i} - g(\underline{P_1}_{\,i}, .., \underline{P_m}_{\,i}) \right\|_2^2 \tag{4.21}$$

After Lemma 2 and the following Conclusion, we have to calculate a isotonic function which is linear or constant in each component or linear in two components and in the remaining components constant. An approach for calculating such a isotonic function g which minimizes (4.21) is unknown to us.

5 Applications

In this chapter, we test our implementation, described in Chapter 3, on two examples. First, we use the setting of the simulation study in Ranjan and Gneiting (2010) and investigate the calibration and sharpness of the IRP forecast and the convergence of the IRP-solution. In the second application, we use the data set by Baars and Mass (2005) for combining model output statistics (MOS) and National Weather Service forecasts (NWS). At the end of each application, we compare to the BLP approach.

5.1. Simulation study of Ranjan and Gneiting (2010)

5.1.1. Setting

Based on Luss and Rosset (2011) and following the statistical framework of Section 2.1, we describe a model that gives rise to a joint distribution of

$$(y, p_1, p_2),$$

where y is a binary random variable and p_1, p_2 are probability forecasts of two forecasters, which have access to potentially distinct sources of information.

Specifically, let

$$p = \Phi(a_1 + a_2),$$

where

$$a_1 \sim \mathcal{N}(0, 1), \quad a_2 \sim \mathcal{N}(0, 2)$$

are independent variables. Here, Φ denotes the standard normal cumulative distribution function. Let y be a Bernoulli random variable with conditional success probability

$$\mathbb{P}(y = 1|p) = \mathbb{E}(y|p) = p.$$

The first forecaster has only notice of a_1 whereas the second forecaster has only notice of a_2. The corresponding forecasts are the conditional event probabilities:

$$p_1 = \mathbb{P}(y = 1|a_1) = \mathbb{E}(y|a_1) = \mathbb{E}(p|a_1) = \mathbb{E}(\Phi(a_1 + a_2)|a_1) = \Phi(a_1/\sqrt{3}) \tag{5.1}$$

and

$$p_2 = \mathbb{P}(y = 1|a_2) = \mathbb{E}(y|a_2) = \mathbb{E}(p|a_2) = \mathbb{E}(\Phi(a_1 + a_2)|a_2) = \Phi(a_2/\sqrt{2}). \tag{5.2}$$

Obviously, p, p_1 and p_2 are calibrated forecasts for the random variable y. In the following, we want to combine p_1 and p_2 by BLP and IRP. We generate a training data set consisting of 2,000 pairs of observations and corresponding forecasts. Similarly, we generate a test data set consisting of 1,500 such pairs.

5.1.2. Results

Table 5.1 shows the resulting Brier scores for four repetitions of generating data sets and combing them by IRP. The last column are the results of Ranjan and Gneiting (2010) together with the corresponding score for IRP. The forecast "CP" is the conditional event probability if someone has notice of a_1 and a_2. It is the optimum that we can reach.

Table 5.1.: Comparison of the IRP and BLP approaches for the simulation study by Ranjan and Gneiting (2010)

Forecast	Repetition 1	Repetition 2	Repetition 3	Repetition 4	RG (2010)
p_1	0.2006	0.2025	0.2054	0.2043	0.2094
p_2	0.1744	0.1616	0.1740	0.1646	0.1657
BLP	0.1160	0.1079	0.1203	0.1094	0.1137
IRP	0.1190	0.1083	0.1237	0.1147	0.1186
CP	0.1141	0.1056	0.1182	0.1081	0.1126

Obviously, combining the two single forecasts p_1, p_2 by IRP leads to an improvement. In all four cases, the Brier score of the combined forecast is strictly smaller than the Brier scores of the single forecasts. By comparing the Brier score of BLP and IRP, we can see that BLP is a little bit better than the IRP. However, they are both near to the optimal solution CP. Figure 5.1 shows the empirical distributions of the forecasts p_1, p_2 and the combined IRP forecast for a training data set of 1,500 forecasts.

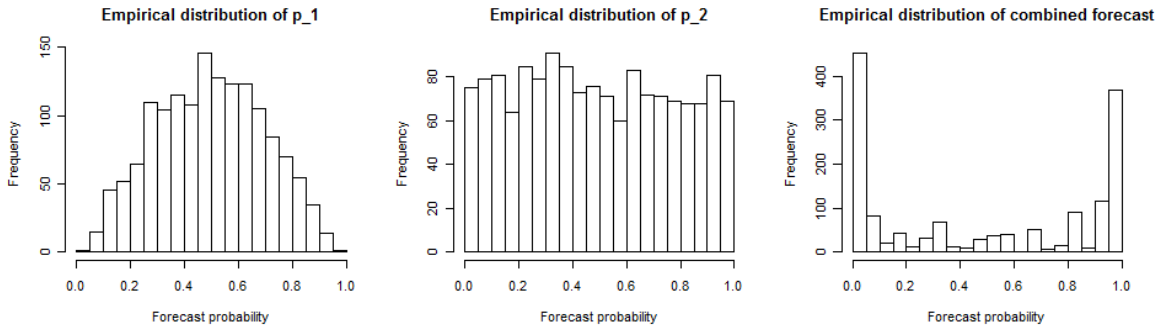

Figure 5.1.: Empirical distributions of the single forecasts and the combined IRP-forecast.

Further we investigate whether the combined IRP-forecast is calibrated or not. We

check this by the reliability diagram and the 95% bootstrap interval by Broecker and Smith (2007). A reliability diagramm is a graph of the observed frequency plotted against the forecast probability of an event. For our statistical framework this leads to:

Let $(\widetilde{P}, \widetilde{Y})$ with $\widetilde{Y} = (\widetilde{Y}_1, \ldots, \widetilde{Y}_{\widetilde{n}}) \in \{0,1\}^{\widetilde{n}}$ the observations and $\widetilde{P} = (\widetilde{P}_1, \ldots, \widetilde{P}_{\widetilde{n}}) \in [0,1]^{\widetilde{n}}$ the corresponding forecasts of the test data set, where $\widetilde{n}$ is the size of our test data set. We now split the unit interval with the help of a partition

$$0 = q_0 < q_1 < \cdots < q_J = 1$$

into subintervals $[q_{j-1}, q_j)$, $j \in \{1, \ldots, J\}$ and plot

$$\frac{\#\{p_i \in [q_{j-1}, q_j), Y_i = 1)\}}{\#\{p_i \in [q_{j-1}, q_j)\}} \quad \text{against} \quad \frac{q_{j-1} + q_j}{2}.$$

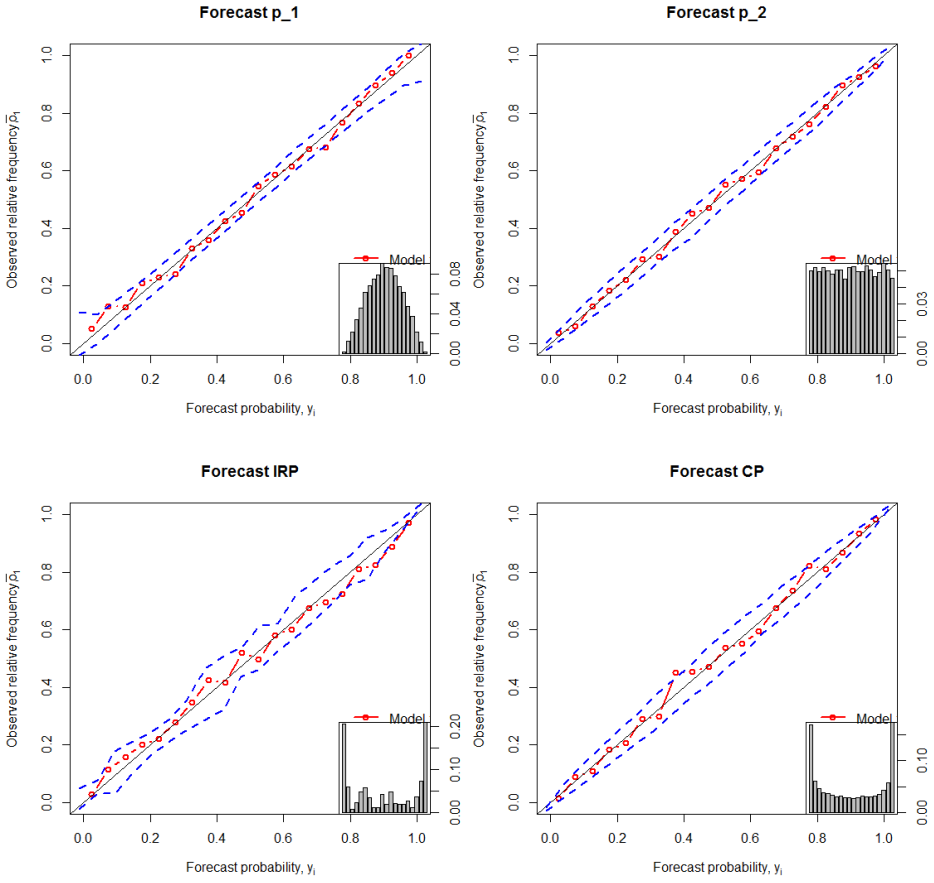

Figure 5.2.: Calibration curves and 95% bootstrap intervals under the null hypothesis of calibration for the two single forecasts p_1, p_2, our combined IRP forecast and the calibration curve of the optimal forecaster CP.

We have taken a training data set of 2,000 and a test data set of 10,000 observations and corresponding forecasts. The 95% bootstrap technique by Bröker and Smith

(2007), illustrated in the plots of Figure 5.2 by the broken lines, gives pointwise lower and upper critical values under the null hypothesis of calibration. If our forecast is calibrated, the plotted graph should be within the 95% bootstrap interval with probability 95%. As we can see in Figure 5.2, this property is fulfilled and we can say that our forecast is calibrated.

As already mentioned in the introduction, following Gneiting et al. (2007), probabilistic forecasts should be as sharp as possible, subject to calibration. Sharpness in the context of a binary random variable is the measure of the difference from a probability forecast to the climatological forecast, in our case $p = 1/2$. A forecaster who does not have any information would choose the climatological forecast. From the histograms in Figures 5.1 and 5.2, we conclude that the IRP is sharper than the single forecasts p_1 and p_2, because it differs stronger from the climatological forecast than p_1 or p_2. Hence, we get a calibrated forecast which is sharper than the single forecasts. Consequently, we are consistent with the statistical principle that we want to generate a sharp forecast under the condition of calibration.

Ranjan and Gneiting (2010) also investigates the performance of the BLP in the case of an uncalibrated forecast. We change the setting of Section (5.1) in the way that we set

$$p_2^* = \Phi\left(\frac{1}{5} + \frac{a_2}{2}\right).$$

In Figure 5.3, we can see that this new forecast p_2^* is uncalibrated. As in the case before, we generate a training data set of 2,000 and a test data set of 1,500 observations and corresponding forecasts.

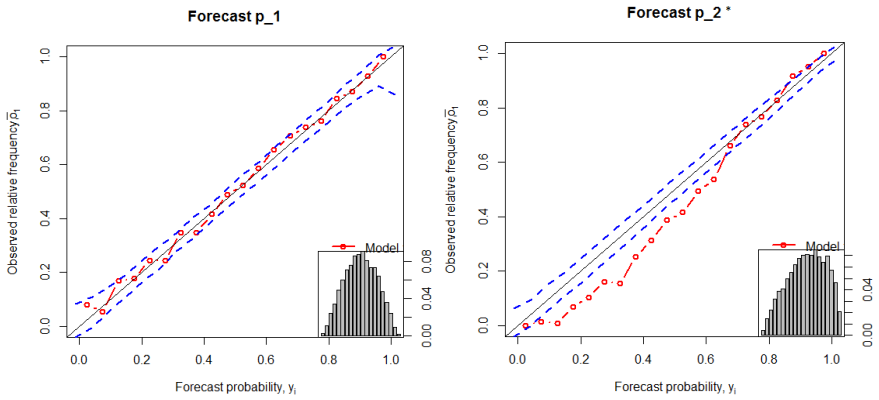

Figure 5.3.: Calibration curves and 95% bootstrap intervals under the null hypothesis of calibration for the two single forecasts p_1 and p_2^*.

Using the IRP approach we combine the two single forecasts p_1 and p_2^*. The results together with the results of Ranjan and Gneiting with the added result of the IRP

approach are shown in Table 5.2.

Table 5.2.: Comparison of the IRP and BLP approach for the simulation study by Ranjan and Gneiting (2010)

Forecast	Repetition 1	Repetition 2	Ranjan and Gneiting (2010)
p_1	0.2110	0.2102	0.2094
p_2	0.1735	0.1765	0.1740
BLP (symmetric)	0.1186	0.1272	0.1215
BLP (asymmetric)	0.1105	0.1197	0.1132
IRP	0.1112	0.1223	0.1204
CP	0.1088	0.1182	0.1126

The symmetric BLP, which means that $\alpha = \beta$ in the cumulative beta distribution, is worse than the IRP. But the IRP is worse than the general BLP. The solution of the IRP and the BLP is still very close to the optimal solution CP. If we investigate the calibration again, we get that the IRP is still calibrated, as we can see in Figure 5.4.

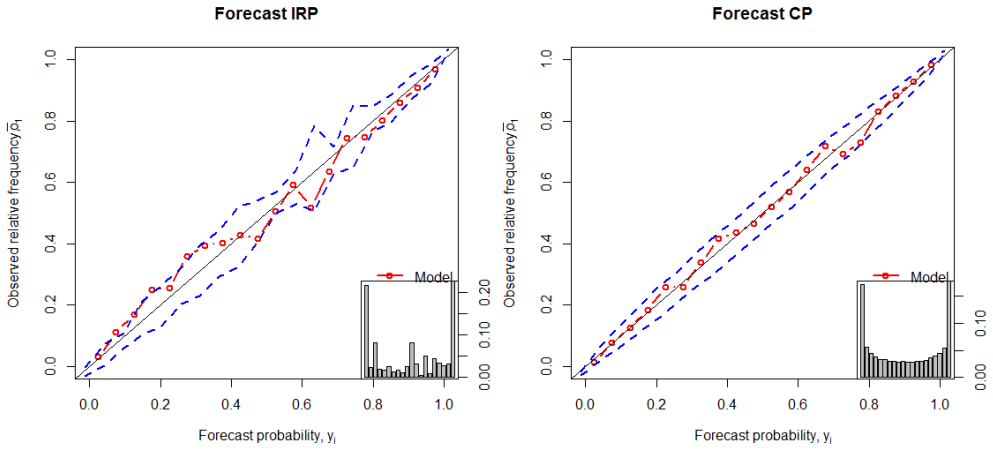

Figure 5.4.: Calibration curves and 95% bootstrap intervals under the null hypothesis of calibration for IRP and CP for combining the calibrated forecast p_1 and the uncalibrated forecast p_2^*.

At last, we investigate the convergence of the Brier score of the IRP solution under the condition of an increasing size of the training data set. For this reason, we have taken a test data set with 1,500 observations and started with 10 observations in the training data set. Until 100 our training data set grows in steps of size 10, afterwards in steps of size 50 until it reaches a limit of 400. We repeat this 15 times and plot the mean of the Brier score. For comparison we also plotted the results for training sets of size 500, 1,000, 1,500 and 2,000.

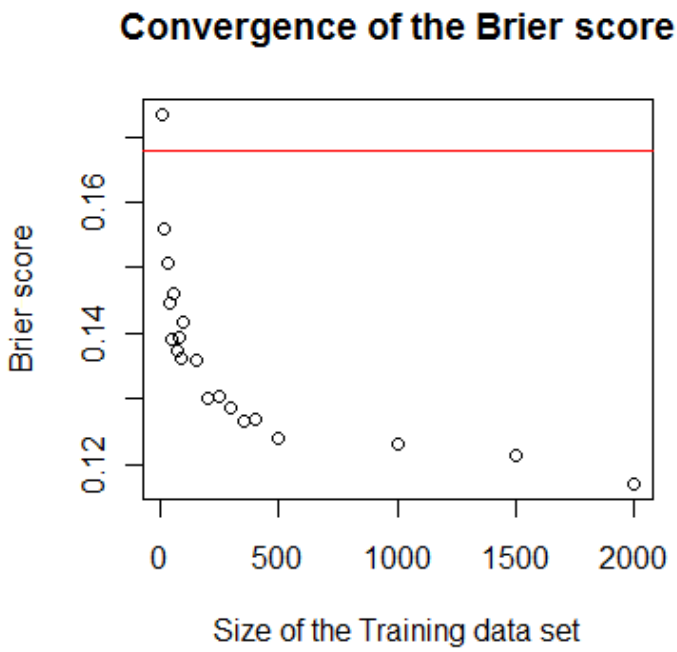

Figure 5.5.: Plot of the Brier score in dependence on the size of the data set. The Brier score of the forecast p_2 is illustrated by the constant line.

We can see in Figure 5.5 that just having a small training data set of about 20 data points is enough for getting a better result than just taking the forecast p_2. The reason for the non constant decreasing graph in the interval from 20 to 100 is the randomness of our procedure for generating the training data set. Because we just have made 15 repetitions, an IRP forecast generated from a training data set of size 70 can be surpassed by a IRP-forecast generated with 60 data points.

Further, we can see that the function is decreasing in the second part. This shows that we do not have the problem of overfitting in this situation. As mentioned in Chapter 3, the size of the training data set is restricted to 3,000 if we want to calculate it by `R`. Therefore, it seems that overfitting is not a problem in the context of the simulation study by Ranjan and Gneiting (2010).

5.2. Data set by Baars and Mass (2005)

5.2.1. Data and setting

Daily model output statistics (MOS) and National Weather Service (NWS) forecasts of maximum temperature (MAX-T), minimum temperature (MIN-T) and probability of precipitation (POP) were collected for 29 stations from July 1st, 2003 to March 3rd, 2008. All these stations were spread across the USA to represent a wide range of

geographical areas. The exact positions are illustrated in Figure 5.6. As we can see, there is a high concentration of stations to the northern East Coast and in the south of the Great Lakes. Further, each of them was near to a weather forecast office (WFO). Four forecasts were taken for each station; the local NWS forecast and three model output statistics forecasts, Global Forecast System (GMOS), Eta (EMOS) and NGM (NMOS).

Figure 5.6.: NWS locations used in the study.
Source: journals.ametsoc.org/doi/full/10.1175/WAF896.1

As in the work by Baars and Mass (2005), the reported times for all meteorological reports are given according to UTC (Universal Coordinated Time). In contrast to the MOS forecasts, which were taken from the 0000 UTC cycle model, for example 6 pm Central Standard time, the NWS forecasts were gathered from the early morning, at about 1000 UTC or 0400 Pacific standard time (Baars and Mass, 2005). This implies an advantage to the NWS forecasts, because they not only have access to the MOS forecasts, but also to consider the 6-9 h further development of the weather. Each cycle contains the forecasts and verifications for 48 hours, two maximum temperature (MAX-T), two minimum temperature (MIN-T) and four 12h probability of precipitation (12h-POP) forecasts.

The definitions of observed maximum temperatures follow the NWS definitions with MAX-T (MIN-T) equal to the maximum (minimum) temperature between 0700 and 1900 (1900 and 0700) local time. The data set considers two forecasts per day for POP, between 0000-1200 UTC and 1200-0000 UTC. Because the forecast period is 48 h, precipitation data for four periods were examined (day 1: 1200-0000, day 2: 0000-1200 , 1200-0000, day 3: 0000-1200). It was found that each station has all forecasts and corresponding observations for about 85%-90% of the days (Baars and Mass, 2005). In the case of missing forecasts or observations, the whole data of the station for the date was left out.

Baars and Mass (2005) compared the performance of the individual and linearly

combined MOS forecasts for POP. The authors concluded that a linear opinion pool of the MOS forecasts is equal or even better than the NWS forecast at almost all stations. In the following, we consider the three MOS forecasts and the NWS forecast for POP as individual forecasts which we aggregate by IRP and BLP.

At first, we combine the POP forecasts of the MOS and NWS forecasts by splitting the data set in a training set, from July 1st, 2003 to June 30th 2005, and a test data set, from July 1st, 2005 to March 3rd, 2008. Additionally, we investigate the performance in the case that we use the last two months for generating a forecast for the following two months. Finally, the same procedure is done with MIN-T and MAX-T forecasts in Section 5.3. In contrast to the paper by Ranjan and Gneiting (2010), we give the results of each station separately.

We consider two different sets of individual forecasts. At first, we only consider the three MOS forecasts as individual forecasts and aggregate them by IRP and BLP. In the second, we use the MOS forecasts and the NWS forecast to generate the IRP and BLP forecasts. With this approach, we investigate whether we get a better performance if we use a further forecast, the NWS forecast. On the other hand, we could deduce overfitting of the data set if the results are worse in the case that we use all individual forecasts.

5.2.2. Results for POP

As we have seen in Section 4.1, the GIRP-solution of the logarithmic score and the IRP-solution are equivalent. Consequently, we only evaluate the performance of the IRP-solution with the help of the Brier score. Tables 5.3 and 5.4 show the results for 9 of the 29 stations for the two cases we consider. The complete results of the data set can be found in the Appendix Table A.1 and Table A.2.

Tables 5.3 and 5.4 show that the BLP is often better than the IRP, but not always. A reason is that the single forecasts themselves are already quite good and combining them does not lead to an improvement as high as in Section 5.1, where we get a high reduction of the Brier score. The first station in Albuquerque (ABQ) is removed from the tables because the test data set only has a size of 19 observations. All of these observations recognize precipitation and thus this station is not good for evaluating the performance of the different forecasts. The nine stations above show that in some, but not in all cases, BLP and IRP lead to an improvement of the forecast in the sense that they reduce the Brier score.

By comparing the results in Table 5.3 and Table 5.4, we recognize that they often do not differ a lot. An exception is the station in Las Vegas ("LAS"), where we achieve an improvement of about 0.03 (compare Table Table 1 and Table 2 of the Appendix). Consequently, if we use all four individual forecasts, this may lead to a better performance. Further, we can say that the additional use of the NWS forecast does not incur overfitting. The reason is that the NWS forecast depends on the MOS forecasts, be-

cause we designed the setting in the way, that the National Weather Service knows the MOS forecasts. Therefore a high MOS forecast for precipitation will probably imply a high one of the NWS.

Table 5.3.: Brier scores for combining the three MOS forecasts for POP by IRP and BLP.

Station	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	0.077	0.084	0.079	0.076	0.084	0.077	0.113	0.064	0.068
EMOS	0.089	0.092	0.086	0.071	0.089	0.084	0.115	0.074	0.071
NMOS	0.125	0.095	0.092	0.088	0.097	0.082	0.110	0.077	0.074
NWS	0.079	0.081	0.075	0.069	0.116	0.082	0.109	0.064	0.074
BLP	0.077	0.087	0.076	0.069	0.082	0.071	0.105	0.065	0.063
IRP	0.084	0.083	0.080	0.068	0.079	0.071	0.107	0.070	0.062

Table 5.4.: Brier scores for combining the three MOS forecasts and the NWS forecast for POP by IRP and BLP.

Station	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	0.077	0.084	0.079	0.076	0.084	0.077	0.113	0.064	0.068
EMOS	0.089	0.092	0.086	0.071	0.089	0.084	0.115	0.074	0.071
NMOS	0.125	0.095	0.092	0.088	0.097	0.082	0.109	0.077	0.074
NWS	0.079	0.081	0.075	0.068	0.116	0.082	0.109	0.064	0.074
BLP	0.077	0.087	0.073	0.067	0.083	0.069	0.101	0.064	0.063
IRP	0.082	0.083	0.077	0.067	0.097	0.071	0.105	0.070	0.061

In the context of Section 5.1, Figure 5.5 suggests that we only need a small training data set for getting a good performance of IRP. However, the results in Table 5.3 and 5.4 suggest that just combining a small size of training dates by IRP leads to a bad performance. Therefore, we investigate this fact by taking the observations of the last two months, for forecasting the next two months. We start with July and August 2003 and generate the BLP and IRP forecasts for September and October. Then we take August and September for forecasting October and November and so on. We evaluate the performance of by taking the mean over the single Brier scores. As before, we consider two cases with just combining the MOS forecasts in the one and combining all individual forecasts in the other case. Tables 5.5 and 5.6 show the results for the same nine stations as before. The complete results of the data set can be found in the Appendix Table A.3 and Table A.4.

As we can see, the BLP and IRP forecast are sometimes even worse than each individual forecast and the IRP is usually the worst one. But what is the reason? Even if we just combine the MOS forecasts, the results do not change a lot and sometimes they are even a bit worse. There are three important differences to our simulation study in Section 5.1, which lead to these results.

Table 5.5.: Brier scores for forecasting the following two months by using the previous two months by combing the corresponding MOS forecasts by IRP and BLP.

Station	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	0.069	0.099	0.087	0.080	0.092	0.083	0.121	0.065	0.071
EMOS	0.070	0.103	0.091	0.079	0.091	0.083	0.121	0.071	0.074
NMOS	0.078	0.112	0.096	0.093	0.102	0.086	0.118	0.075	0.078
NWS	0.072	0.100	0.083	0.076	0.110	0.086	0.116	0.064	0.076
BLP	0.090	0.104	0.088	0.086	0.091	0.082	0.117	0.069	0.078
IRP	0.094	0.119	0.107	0.114	0.102	0.096	0.132	0.087	0.087

Table 5.6.: Brier scores for forecasting the following two months by using the previous two months by combing the corresponding MOS and NWS forecasts by BLP and IRP.

Station	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	0.078	0.099	0.087	0.080	0.092	0.083	0.121	0.065	0.071
EMOS	0.090	0.103	0.091	0.079	0.091	0.083	0.121	0.071	0.074
NMOS	0.114	0.112	0.096	0.093	0.102	0.086	0.118	0.075	0.078
NWS	0.084	0.100	0.083	0.076	0.110	0.086	0.116	0.064	0.076
BLP	0.094	0.110	0.093	0.095	0.100	0.087	0.116	0.068	0.079
IRP	0.116	0.119	0.106	0.114	0.109	0.094	0.132	0.086	0.089

1. As we have already seen, the MOS and NWS forecasts are very good forecasts, whereas p_1 and p_2 in the simulation study are not.

2. There exists a temporal dependence. It is more probably to get a day without precipitation, if the previous day has none too. The same property holds for the case of precipitation.

3. p_1 and p_2 are independent, but the MOS and NWS forecasts are not. Especially in this study, the National Weather Service has information on the MOS forecasts of the corresponding day. Even the MOS forecasts are not independent, because they all use the previous observation to get their forecast.

As already mentioned, point 3 implies that there are many members in the set I of isotonic constraints and that the combination of three and four single forecasts do not differ so much. Further this implies that the optimization problem of Chapter 2 has many side-conditions. Thus the number of partitions is lower. In combination with the small size of the training data set this leads to worse results than in Section 5.1.

Table 5.7.: MSE for combining the MAX-T MOS forecasts by IRP. The temperatures are in degrees Fahrenheit.

Station	ABQ	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	12.84	12.52	13.38	14.79	15.23	16.78	13.63	12.68	16.97	29.40
EMOS	15.82	14.81	17.84	17.67	15.37	17.53	16.70	14.12	19.67	25.63
NMOS	22.54	451.48	19.22	20.18	17.60	23.09	20.15	16.80	27.07	34.82
NWS	12.49	11.87	12.22	12.82	12.41	14.20	14.25	12.60	13.82	20.10
IRP	12.59	54.53	15.79	14.78	13.46	15.87	14.76	12.59	20.50	32.77

Table 5.8.: MSE for combining the MAX-T MOS and NWS forecasts by IRP. The temperatures are in degrees Fahrenheit.

station	ABQ	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	12.84	12.52	13.38	14.79	15.23	16.78	13.63	12.68	16.97	29.40
EMOS	15.82	14.81	17.84	17.67	15.37	17.53	16.70	14.12	19.67	25.63
NMOS	22.54	451.48	19.22	20.18	17.60	23.09	20.15	16.80	27.07	34.82
NWS	12.49	11.87	12.22	12.82	12.41	14.20	14.25	12.60	13.82	20.10
IRP	12.32	52.06	15.08	14.21	12.82	15.04	14.48	12.31	18.78	28.89

5.2.3. Results for Temperature

In this section, we test the performance of the IRP approach for combining temperature forecasts. Here, we cannot compare the IRP to the BLP because, as already mentioned in Chapter 2, the BLP cannot handle this. We use the MIN-T and MAX-T MOS and NWS forecasts by Baars and Mass (2005). Here, we can use the station in Albuquerque for evaluating IRP because we have, in contrast to the previous Section 5.2.2 where only 19 observations were available, many observations. In this context, we have to use the squared error to evaluate the forecasts. As in the case of combining predictions for precipitation, we set the training period from July, 1st 2003 to June, 30th 2005 and the rest as the test period.

Because the results for the maximum temperature and the minimum temperature imply the same conclusion, we only discuss the solutions of combining the maximum temperature in this section. All the temperatures are given in degrees Fahrenheit. The results for the minimum temperature as well as for the maximum temperature can be found in the Appendix. We get the following mean squared errors (MSE) illustrated in Tables 5.7 and 5.8 for combining MOS forecasts for the maximum temperature of the first ten stations. The complete results of the 20 stations be found in the Appendix Table A.5 and Table A.6.

Table 5.9.: MSE for combining the MOS and NWS forecasts for one season by using the data of the last season.

Station	ABQ	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	11.40	12.25	13.17	14.09	16.07	16.60	13.62	12.84	16.51	27.62
EMOS	14.11	15.58	17.36	17.66	15.57	17.08	16.61	14.23	17.50	23.09
NMOS	22.38	267.67	18.92	19.94	18.67	24.03	19.65	16.18	22.70	33.39
NWS	11.26	12.47	12.23	12.85	12.67	14.10	14.83	12.50	13.44	21.14
IRP	13.43	35.20	16.32	15.67	15.74	17.96	14.75	14.05	19.33	26.67

Table 5.10.: MSE for combining the MOS and NWS forecasts for one season by using the data of the last season.

Station	ABQ	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	11.40	12.25	13.17	14.09	16.07	16.60	13.62	12.84	16.51	27.62
EMOS	14.11	15.58	17.36	17.66	15.57	17.08	16.61	14.23	17.50	23.09
NMOS	22.38	267.67	18.92	19.94	18.67	24.03	19.65	16.18	22.70	33.39
NWS	11.26	12.47	12.23	12.85	12.67	14.10	14.83	12.50	13.44	21.14
IRP	13.24	35.12	16.33	15.42	14.90	17.00	14.76	14.24	18.17	25.92

We see that the IRP forecast in Atlanta ("ATL") is very bad. This is implied by the NMOS forecast. The other stations show that sometimes the IRP is better than each and sometimes there are one or two forecasts, which are better. But IRP is never the worst forecast. This conclusion is comparable to the one of section 5.2.2. The additional input of the NWS forecast improves the performance of the IRP in every of the stations above, as we can see in Table 5.7. and Table 5.8. Like in the previous section, we test the IRP approach on small training data sets. This is done as follows:

We want to improve the forecast of the temperatures for a season, by using the data of the same season of the previous year. For example, we use the data from the summer 2005 to predict the temperatures of the summer 2006. We split the months as follows:

- Winter: December, January, February, March
- Spring: March, April, May, June
- Summer: June, July, August,September
- Autumn: September, October, November, December

We perform the IRP implementation of Chapter 3 and get the results. In Table 5.9 and Table 5.10, we can see the results for the well-known 10 stations. The complete

results of the data set can be found in the Appendix Table A.7 and Table A.8. We take the mean over all seasons and do not look at each season separately.

The results show that a combination of the four forecasts does not lead to an improvement of the forecast. Further we don't see any evidence for that overfitting because the results don't differ in Table 5.9 and Table 5.10. This conclusion is equivalent to the one of the previous Section 5.2.2. The reasons for this are the same as in the case of combining the probabilities of precipitation.

However, we see that the question, if we should use IRP and BLP for combining probability forecasts cannot be answered directly. It depends on the quality of the forecasts as well as the dependence structure of the single forecasts. The simulation study in Section 5.1 shows that there is the possibility to achieve a high improvement of the MSE, the calibration and the sharpness. But as the case study shows, this is not guaranteed neither in the case of probability forecasts nor in the case of point forecasts.

6 Summary and Discussion

Based on the work by Ranjan and Gneiting (2010) and Luss et al. (2012), we have analyzed the two corresponding methods for aggregating individual forecasts in the framework of a binary random variable, formulated by Ranjan and Gneiting (2010). The aim was to generate a better forecast in in the sense that it is sharper, subject to calibration. The BLP and the IRP, discussed in Chapter 2, show that there exist parametric as well as non-parametric approaches for combining individual forecasts. The BLP uses the linear opinion pool by Stone (1961) and recalibrate it whereas the IRP approach provides a solution of the isotonic regression problem by Barlow and Brunk (1972). We further discussed the generalization of the BLP by Gneiting and Ranjan (2011) that provides the possibility for aggregating cumulative distribution functions. In contrast, the GIRP by Luss and Rosset (2011) generalizes the IRP approach to the class of convex and differentiable scoring functions on $\mathbb{R}$.

In Chapter 3, we standardized the linear optimization problem of the GIRP approach. In the following, we used existing `R`-packages for implementing the IRP and GIRP approaches separately. We have seen that our implementation is limited by the size of the training data set and the set of scoring functions whose derivative is computable by `R`. However, under this conditions, the implementation is quite fast and applicable for any output in $\mathbb{R}$.

Next, in Chapter 4, we investigated several theoretical aspects of the GIRP approach. First, in Section 4.1, we formulated conditions such that the IRP generates the same solution as the GIRP. In the framework of Ranjan and Gneiting (2010), we got conditions such that the structure of the solutions is equal, see Section 4.1.1. Next, we considered output in $\mathbb{R}$ and formulated conditions such that the two solutions are equal, see Section 4.1.2. This was motivated by the fact that the running time of the GIRP-implementation was significantly higher than that of the IRP-implementation. Consequently, using the IRP-algorithm instead of the GIRP-algorithm reduces the computational effort.

In Section 4.2 we gave further connections between the general isotonic regression problem by Barlow and Brunk (1972) and the class of scoring functions, for which the IRP-solution and the GIRP-solution are identical. Because Luss et al. (2012) do not investigate a generalization of the IRP approach to the class of categorical events, we focused on expanding the IRP approach to this class. At first, we deduced several strong properties from the statistical framework and got very strong restrictions on

the class of solution functions.

The simulation study conducted in Section 5.1, which is based on Luss and Rosset (2011), empirically confirms that the BLP as well as the IRP approach generates calibrated and sharp forecasts. We verified this by reliability diagrams and 95% bootstrap intervals established by Bröcker and Smith (2007). The property of calibration is fulfilled independently of the calibration of the individual forecasts. Further, we directly compared the IRP and BLP approach by using the Brier score. In the context of the simulation study, IRP is a bit worse than BLP but both are near to the theoretical optimum. IRP outperforms each individual forecast even if the training set is very small.

Finally in Section 5.2, we considered the data set by Baars and Mass (2005) and applied our implementation of Chapter 3 to the MOS and NWS forecasts for probability of precipitation as well as for the minimum and maximum temperature. We have got that the BLP often is better than the IRP(compare the results in the Appendix). Further, the aggregated forecasts do not outperform the individual forecasts in the simulation study by Ranjan and Gneiting (2010). In the case where we used a small training data set, the combined forecasts were often even worse than the individual forecasts.

The work discussed in this thesis expands the results of Luss et al. (2012) in several ways. The most important expansion is that we investigated the IRP in the framework of statistical forecasting. By applying the IRP to the simulation study by Ranjan and Gneiting (2010), we empirically confirm that the IRP approach generates calibrated and sharp forecasts. Further, we compared the IRP to another existing aggregating method, the BLP, and get that the BLP empirically generates better forecasts.

We also expanded the results of Luss et al. (2012) and Luss and Rosset (2011) by formulating conditions such that IRP and GIRP are equal which is desirable from a computational viewpoint. Further, we focused on a possible extension to the class of categorical events without getting a final result. However, the deduced properties are a good basis for further investigation. The third and last expansion is that we implemented the IRP and GIRP in `R` and investigated arising difficulties and explain corresponding solutions.

We now turn to connections to work from other scientists. Giang (2011) formulates an approach comparable to the IRP approach. The author verifies his aggregating method with the help of the simulation study by Ranjan and Gneiting (2010). However, we also investigate the GIRP approach and therefore achieve more general results.

The thesis at hand suggests several starting points for further research. At first, a final result for the problem in Section 4.3 seems desirable. Further, one may investigate whether IRP can be used for evaluating two forecasters. Think of the case in which we have one forecaster and we want to know if we should combine his forecasts with predictions of a second forecaster. Then it would be useful if the IRP approach signal-

Table 6.1.: Comparison of the Brier score for combining CP p_1 and just applying IRP on CP

forecast	Brier score
p_1	0.207
CP	0.116
p_1 and CP combined by IRP	0.122
CP and CP combined by IRP	0.118

izes that the information of one forecaster is already covered by the information of the second forecaster. In this context, we assume that the single forecasts are calibrated.

We tested this in the setting of Section 5.1 with

$$p_1 = \Phi(a_1/\sqrt{2})$$

and

$$p = \Phi(a_1 + a_2).$$

So we set the second forecaster equal to the CP forecast. We performed the implementation of Chapter 3 one time with the forecasts p_1, p and the other time just with p. The results illustrated in Table 6.1 show that just applying the IRP algorithm with the CP forecast is better than combining it with the forecast p_1. We have generated 1,000 training and 1,000 test data points. This setting is quite special because the second forecast is equal to the best theoretical forecast. Thus the question remains whether we can in general use IRP to check whether one information source is covered by the information source of the second forecaster, subject to the case that the corresponding forecasts are calibrated.

Bibliography

J. Baars and C. F. Mass. Performance of national weather service forecasts compared to operational, consensus and weighted model output statistics. *Weather and Forecasting*, 20(6):1034–1047, 2005.

R. Barlow and H. Brunk. The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67(337):658–672, 1972.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 7th edition, 2004. ISBN 978-0-521-83378-3.

J. Bröcker and A. L. Smith. Increasing the reliability of reliability diagrams. *Weather and Forecasting*, 22(3):651–661, June 2007.

A. Brouwer, A. M. Cohen, and A. Neumaier. *Distance-regular graphs*, volume 18 of *Ergebnisse der Mathematik und ihrer Grenzgebiete 3.Folge*. Springer-Verlag Berlin, Heidelberg, 1989. ISBN 3-540-50619-5.

R. T. Clemen and R. L. Winkler. Combining probability distributions from experts in risk analysis. *Risk Analysis*, 19:187–203, 1999.

C. Genest and J. Zidek. Combining probability distributions: A critique and an annotated bibliograpy. *Statistical Science*, 1:114–135, 1986.

P. Giang. Non-parametric synthesis of private probabilistic predictions. In *Twenty-Fifth Annual Conference on Neural Information Processing Systems*, 2011.

T. Gneiting. Editorial: Probabilistic forecasting. *Journal of the Royal Statistical Society Ser. A*, 171:319–321, 2008.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378, 2007.

T. Gneiting and R. Ranjan. Combining predictive distributions. *Preprint*, June 2011. arXiv:1106.1638.

T. Gneiting, F. Balabdaoui, and A. E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Association Ser. B*, 69:243–268, 2007.

S. G. Hall and J. Mitchell. Combining density forecasts. *International Journal of Forecasting*, 23:1–13, 2007.

S. C. Hora. Probability judgements for continuous quantities: Linear combinations and calibration. *Management Science*, 50:597–604, 2004.

R. Luss and S. Rosset. Generalized isotonic regression. *preprint*, April 2011. arXiv:1104.1779.

R. Luss, S. Rosset, and M. Shahar. Efficient regularized isotonic regression with applications to gene–gene interaction search. *Annals of Applied Statistics*, 6:253–283, 2012.

W. Maxwell and J. Muckstadt. Establishing consistent and realistic reorder intervals in production-distribution systems. *Operations Research*, 33(6):1316–1341, 1985.

A. H. Murphy and R. L. Winkler. A general framework for forecast verification. *Monthly Weather Review*, 115:1330–1338, 1987.

R. Ranjan. *Combining and Evaluating Probabilistic Forecasts*. PhD thesis, University of Washington, 2009.

R. Ranjan and T. Gneiting. Combining probability forecasts. *Journal of the Royal Statistical Society Ser. B,*, 72:71–91, June 2010.

M. Stone. The linear pool. *Annals of Mathematical Statistics*, 32:1339–1342, 1961.

R. D. C. Team. *R: A Language and Enviroment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, http://www.R-project.org, 2011. ISBN 3-900051-07-0.

T. Wallsten and A. Diederich. Understanding pooled subjective probability estimates. *Mathematical Social Science*, 41:1–18, 2001.

R. L. Winkler. The consensus of subjective probability distributions. *Management Science*, 15:B61–B75, 1968.

V. Zarnowitz. The new asa-nber survey of forecasts by economic statisticians. *American Statistician*, 23:12–16, 1969.

Appendix

A. Results

Table A.1.: Brier scores for the complete data set of Baars and Mass (2005) for combining the three MOS forecasts for POP by IRP and BLP. The first nine stations are also presented in Table 5.3.

station	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN	DTW
GMOS	0.077	0.084	0.079	0.076	0.084	0.077	0.113	0.064	0.068	0.083
EMOS	0.089	0.092	0.086	0.071	0.089	0.084	0.115	0.074	0.071	0.086
NMOS	0.125	0.095	0.092	0.088	0.097	0.082	0.110	0.077	0.074	0.103
NWS	0.079	0.081	0.075	0.069	0.116	0.082	0.109	0.064	0.074	0.099
BLP	0.077	0.087	0.076	0.069	0.082	0.071	0.105	0.065	0.063	0.073
IRP	0.084	0.083	0.080	0.068	0.079	0.071	0.107	0.070	0.062	0.077

station	IAH	IND	JAN	LAS	LGA	MIA	MSO	MSP	MSY
GMOS	0.081	0.088	0.079	0.148	0.076	0.125	0.103	0.081	0.079
EMOS	0.090	0.089	0.084	0.119	0.085	0.136	0.107	0.084	0.086
NMOS	0.093	0.102	0.085	0.180	0.082	0.134	0.115	0.102	0.084
NWS	0.085	0.084	0.077	0.163	0.079	0.127	0.105	0.086	0.076
BLP	0.082	0.082	0.075	0.133	0.072	0.125	0.099	0.079	0.080
IRP	0.087	0.083	0.079	0.120	0.075	0.130	0.104	0.079	0.081

station	OKC	ORD	PDX	PHL	PHX	SEA	SFO	SLC	STL
GMOS	0.062	0.097	0.089	0.078	0.037	0.104	0.038	0.128	0.076
EMOS	0.073	0.093	0.095	0.083	0.041	0.118	0.046	0.090	0.081
NMOS	0.081	0.109	0.098	0.084	0.042	0.112	0.055	0.124	0.090
NWS	0.065	0.096	0.091	0.080	0.040	0.101	0.045	0.080	0.079
BLP	0.063	0.088	0.084	0.071	0.035	0.109	0.036	0.078	0.072
IRP	0.066	0.095	0.088	0.074	0.040	0.101	0.036	0.078	0.077

Table A.2.: Brier scores for the complete data set of Baars and Mass (2005) for combining the three MOS forecasts and the NWS forecast for POP by IRP and BLP. The first nine stations are also presented in Table 5.4.

station	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN	DTW
GMOS	0.077	0.084	0.079	0.076	0.084	0.077	0.113	0.064	0.068	0.083
EMOS	0.089	0.092	0.086	0.071	0.089	0.084	0.115	0.074	0.071	0.086
NMOS	0.125	0.095	0.092	0.088	0.097	0.082	0.109	0.077	0.074	0.103
NWS	0.079	0.081	0.075	0.068	0.116	0.082	0.109	0.064	0.074	0.099
BLP	0.077	0.087	0.073	0.067	0.083	0.069	0.101	0.064	0.063	0.072
IRP	0.082	0.083	0.077	0.067	0.097	0.071	0.105	0.070	0.061	0.077

station	IAH	IND	JAN	LAS	LGA	MIA	MSO	MSP	MSY
GMOS	0.081	0.088	0.078	0.148	0.076	0.125	0.103	0.081	0.079
EMOS	0.089	0.089	0.084	0.119	0.085	0.136	0.107	0.084	0.086
NMOS	0.094	0.102	0.085	0.180	0.082	0.134	0.114	0.102	0.085
NWS	0.085	0.085	0.077	0.163	0.079	0.127	0.105	0.086	0.076
BLP	0.081	0.080	0.074	0.136	0.071	0.128	0.098	0.078	0.080
IRP	0.085	0.083	0.079	0.091	0.074	0.133	0.103	0.078	0.079

station	OKC	ORD	PDX	PHL	PHX	SEA	SFO	SLC	STL
GMOS	0.062	0.097	0.089	0.078	0.037	0.104	0.038	0.128	0.076
EMOS	0.073	0.093	0.095	0.082	0.041	0.118	0.046	0.090	0.081
NMOS	0.081	0.109	0.098	0.084	0.042	0.112	0.055	0.124	0.090
NWS	0.065	0.096	0.091	0.080	0.040	0.101	0.045	0.080	0.079
BLP	0.061	0.088	0.084	0.070	0.035	0.101	0.036	0.078	0.072
IRP	0.064	0.091	0.087	0.075	0.041	0.100	0.035	0.072	0.076

Table A.3.: Brier scores for the complete data set of Baars and Mass (2005) for combining the three MOS forecasts of two months for predicting POP by IRP and BLP. The first nine stations are also presented in Table 5.5.

station	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN	DTW
GMOS	0.078	0.099	0.087	0.080	0.092	0.083	0.121	0.065	0.071	0.091
EMOS	0.090	0.103	0.091	0.079	0.091	0.083	0.121	0.071	0.074	0.092
NMOS	0.114	0.112	0.096	0.093	0.102	0.086	0.118	0.075	0.078	0.109
NWS	0.084	0.100	0.083	0.076	0.110	0.086	0.116	0.064	0.076	0.103
BLP	0.091	0.104	0.088	0.086	0.091	0.082	0.117	0.069	0.078	0.095
IRP	0.119	0.119	0.107	0.114	0.102	0.096	0.132	0.087	0.087	0.104

station	IAH	IND	JAN	LAS	LGA	MIA	MSO	MSP	MSY
GMOS	0.084	0.092	0.077	0.036	0.078	0.125	0.108	0.084	0.083
EMOS	0.088	0.090	0.082	0.037	0.085	0.133	0.113	0.085	0.088
NMOS	0.094	0.104	0.086	0.040	0.084	0.132	0.120	0.098	0.093
NWS	0.087	0.090	0.076	0.037	0.078	0.126	0.113	0.087	0.083
BLP	0.089	0.092	0.082	0.063	0.076	0.133	0.113	0.086	0.085
IRP	0.119	0.105	0.099	0.052	0.094	0.171	0.139	0.102	0.121

station	OKC	ORD	PDX	PHL	PHX	SEA	SFO	SLC	STL
GMOS	0.074	0.098	0.095	0.082	0.038	0.120	0.040	0.109	0.083
EMOS	0.083	0.095	0.098	0.083	0.043	0.124	0.051	0.086	0.084
NMOS	0.089	0.108	0.105	0.088	0.045	0.121	0.059	0.106	0.093
NWS	0.076	0.098	0.100	0.083	0.041	0.117	0.048	0.086	0.088
BLP	0.080	0.103	0.097	0.082	0.054	0.122	0.053	0.090	0.085
IRP	0.095	0.115	0.123	0.086	0.055	0.134	0.067	0.102	0.098

Table A.4.: Brier scores for the complete data set of Baars and Mass (2005) for combining the three MOS forecasts and the NWS forecast of two months for predicting POP by IRP and BLP. The first nine stations are also presented in Table 5.6.

station	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN	DTW
GMOS	0.078	0.099	0.087	0.080	0.092	0.083	0.121	0.065	0.071	0.091
EMOS	0.090	0.103	0.091	0.079	0.091	0.083	0.121	0.071	0.074	0.092
NMOS	0.114	0.112	0.096	0.093	0.102	0.086	0.118	0.075	0.078	0.109
NWS	0.084	0.100	0.083	0.076	0.110	0.086	0.116	0.064	0.076	0.103
BLP	0.094	0.110	0.093	0.095	0.100	0.087	0.116	0.068	0.079	0.099
IRP	0.116	0.119	0.106	0.114	0.109	0.094	0.132	0.086	0.089	0.105

station	IAH	IND	JAN	LAS	LGA	MIA	MSO	MSP	MSY
GMOS	0.084	0.092	0.077	0.036	0.078	0.125	0.108	0.084	0.083
EMOS	0.088	0.090	0.082	0.037	0.085	0.133	0.113	0.085	0.088
NMOS	0.094	0.104	0.086	0.040	0.084	0.132	0.120	0.098	0.093
NWS	0.087	0.090	0.076	0.037	0.078	0.126	0.113	0.087	0.083
BLP	0.092	0.093	0.087	0.064	0.076	0.137	0.119	0.091	0.088
IRP	0.117	0.099	0.100	0.051	0.093	0.174	0.141	0.099	0.122

station	OKC	ORD	PDX	PHL	PHX	SEA	SFO	SLC	STL
GMOS	0.074	0.098	0.095	0.082	0.038	0.120	0.040	0.109	0.083
EMOS	0.083	0.095	0.098	0.083	0.043	0.124	0.051	0.086	0.084
NMOS	0.089	0.108	0.105	0.088	0.045	0.121	0.059	0.106	0.093
NWS	0.076	0.098	0.100	0.083	0.041	0.117	0.048	0.086	0.088
BLP	0.094	0.101	0.104	0.086	0.063	0.135	0.071	0.097	0.086
IRP	0.095	0.114	0.124	0.088	0.056	0.134	0.071	0.102	0.097

Table A.5.: MSE for the complete data set of Baars and Mass (2005) for combining the three MOS forecasts for predicting MAX-T by IRP. The first ten stations are also presented in Table 5.7. The scores are in degree Fahrenheit.

station	ABQ	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	12.84	12.52	13.38	14.79	15.23	16.78	13.63	12.68	16.97	29.40
EMOS	15.82	14.81	17.84	17.67	15.37	17.53	16.70	14.12	19.67	25.63
NMOS	22.54	451.48	19.22	20.18	17.60	23.09	20.15	16.80	27.07	34.82
NWS	12.49	11.87	12.22	12.82	12.41	14.20	14.25	12.60	13.82	20.10
IRP	12.59	54.53	15.79	14.78	13.46	15.87	14.76	12.59	20.50	32.77

station	DTW	IAH	IND	JAN	LAS	LGA	MIA	MSO	MSP	MSY
GMOS	23.07	9.793	16.74	12.08	9.56	13.34	4.25	25.43	18.62	7.90
EMOS	23.85	14.99	16.63	17.07	9.30	16.16	5.59	34.71	24.45	10.18
NMOS	28.13	15.19	22.41	17.94	12.83	17.75	6.46	26.59	23.05	12.18
NWS	22.32	10.53	13.55	11.33	8.23	12.25	4.21	20.46	16.99	8.76
IRP	22.03	12.88	14.76	15.18	9.17	13.69	5.04	33.84	17.99	8.15

station	OKC	ORD	PDX	PHL	PHX	SEA	SFO	SLC	STL
GMOS	21.66	16.33	13.60	14.03	10.21	12.49	15.09	17.49	18.70
EMOS	23.35	16.43	19.18	15.43	10.85	17.39	15.66	17.63	19.26
NMOS	31.28	22.21	16.68	18.73	13.82	14.64	16.79	23.14	25.64
NWS	15.03	15.79	13.72	11.81	8.28	11.53	14.07	14.69	17.29
IRP	28.46	14.76	13.17	12.74	8.69	12.53	14.08	21.37	17.76

Table A.6.: MSE for the complete data set of Baars and Mass (2005) for combining the three MOS forecasts and the NWS forecast for predicting MAX-T by IRP. The first ten stations are also presented in Table 5.8. The scores are in degree Fahrenheit.

station	ABQ	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	12.84	12.52	13.38	14.79	15.23	16.78	13.63	12.68	16.97	29.40
EMOS	15.82	14.81	17.84	17.67	15.37	17.53	16.70	14.12	19.67	25.63
NMOS	22.54	451.48	19.22	20.18	17.60	23.09	20.15	16.80	27.07	34.82
NWS	12.49	11.87	12.22	12.82	12.41	14.20	14.25	12.60	13.82	20.10
IRP	12.32	52.06	15.08	14.21	12.82	15.04	14.48	12.31	18.78	28.89

station	DTW	IAH	IND	JAN	LAS	LGA	MIA	MSO	MSP	MSY
GMOS	23.07	9.793	16.74	12.08	9.56	13.34	4.25	25.43	18.62	7.90
EMOS	23.85	14.99	16.63	17.07	9.30	16.16	5.59	34.71	24.45	10.18
NMOS	28.13	15.19	22.41	17.94	12.83	17.75	6.46	26.59	23.05	12.18
NWS	22.32	10.53	13.55	11.33	8.23	12.25	4.21	20.46	16.99	8.76
IRP	21.74	12.88	14.07	14.12	8.97	13.67	5.00	28.97	17.18	8.04

station	OKC	ORD	PDX	PHL	PHX	SEA	SFO	SLC	STL
GMOS	21.66	16.33	13.60	14.03	10.21	12.49	15.09	17.49	18.70
EMOS	23.35	16.43	19.18	15.43	10.85	17.39	15.66	17.63	19.26
NMOS	31.28	22.21	16.68	18.73	13.82	14.64	16.79	23.14	25.64
NWS	15.03	15.79	13.72	11.81	8.28	11.53	14.07	14.69	17.29
IRP	23.48	14.06	12.57	12.72	8.37	12.22	13.75	19.17	17.43

Table A.7.: Brier scores for the complete data set of Baars and Mass (2005) for combining the three MOS forecast of a season for predicting MAX-T by IRP. The first ten stations are also presented in Table 5.9. The scores are in degree Fahrenheit.

station	ABQ	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	11.40	12.25	13.17	14.09	16.07	16.60	13.62	12.84	16.51	27.62
EMOS	14.11	15.58	17.36	17.66	15.57	17.08	16.61	14.23	17.50	23.09
NMOS	22.38	267.67	18.92	19.94	18.67	24.03	19.65	16.18	22.70	33.39
NWS	11.26	12.47	12.23	12.85	12.67	14.10	14.83	12.50	13.44	21.14
IRP	13.43	35.20	16.32	15.67	15.74	17.96	14.75	14.05	19.33	26.67

station	DTW	IAH	IND	JAN	LAS	LGA	MIA	MSO	MSP	MSY
GMOS	20.69	10.08	16.45	13.15	8.69	13.20	4.35	24.90	18.49	7.78
EMOS	21.09	14.39	19.09	16.31	8.52	16.06	5.60	29.85	24.05	9.34
NMOS	25.72	14.68	21.95	16.61	12.29	17.77	6.11	26.73	23.65	11.17
NWS	20.14	10.16	14.55	12.12	7.49	12.42	4.32	20.47	17.78	8.05
IRP	22.67	13.46	18.43	14.16	10.46	14.06	5.63	28.98	22.32	10.22

station	OKC	ORD	PDX	PHL	PHX	SEA	SFO	SLC	STL
GMOS	21.72	16.22	13.96	13.95	10.20	12.87	14.84	18.09	17.73
EMOS	22.10	16.07	18.95	15.10	10.63	17.01	16.39	17.48	20.01
NMOS	27.03	21.35	17.33	19.69	14.19	15.26	16.19	23.43	24.00
NWS	15.56	15.25	14.04	12.31	8.14	11.87	13.99	14.34	16.61
IRP	21.62	16.78	15.32	14.71	11.69	15.40	16.24	19.65	21.11

Table A.8.: Brier scores for the complete data set of Baars and Mass (2005) for combining the three MOS forecast and the NWS forecast of a season for predicting MAX-T by IRP. The first ten stations are also presented in Table 5.10. The scores are in degree Fahrenheit.

station	ABQ	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	11.40	12.25	13.17	14.09	16.07	16.60	13.62	12.84	16.51	27.62
EMOS	14.11	15.58	17.36	17.66	15.57	17.08	16.61	14.23	17.50	23.09
NMOS	22.38	267.67	18.92	19.94	18.67	24.03	19.65	16.18	22.70	33.39
NWS	11.26	12.47	12.23	12.85	12.67	14.10	14.83	12.50	13.44	21.14
IRP	13.24	35.12	16.33	15.42	14.90	17.00	14.76	14.24	18.17	25.92

station	DTW	IAH	IND	JAN	LAS	LGA	MIA	MSO	MSP	MSY
GMOS	20.69	10.08	16.45	13.15	8.69	13.20	4.35	24.90	18.49	7.78
EMOS	21.09	14.39	19.09	16.31	8.52	16.06	5.60	29.85	24.05	9.34
NMOS	25.72	14.68	21.95	16.61	12.29	17.77	6.11	26.73	23.65	11.17
NWS	20.14	10.16	14.55	12.12	7.49	12.42	4.32	20.47	17.78	8.05
IRP	22.18	13.46	17.70	14.13	10.45	13.88	5.46	24.87	22.62	10.12

station	OKC	ORD	PDX	PHL	PHX	SEA	SFO	SLC	STL
GMOS	21.72	16.22	13.96	13.95	10.20	12.87	14.84	18.09	17.73
EMOS	22.10	16.07	18.95	15.10	10.63	17.01	16.39	17.48	20.01
NMOS	27.03	21.35	17.33	19.69	14.19	15.26	16.19	23.43	24.00
NWS	15.56	15.25	14.04	12.31	8.14	11.87	13.99	14.34	16.61
IRP	19.65	16.48	14.75	14.44	11.56	14.93	17.64	17.88	20.34

Table A.9.: MSE for the complete data set of Baars and Mass (2005) for combining the three MOS forecasts for predicting MIN-T by IRP. The scores are in degree Fahrenheit.

station	ABQ	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	11.68	8.05	11.77	13.20	11.53	6.81	12.40	11.76	11.78	17.65
EMOS	12.97	7.93	11.07	12.34	11.85	8.55	11.49	12.72	11.57	15.97
NMOS	11.67	328.11	15.84	15.94	16.22	10.14	14.26	14.24	16.66	22.75
NWS	10.54	7.08	9.73	10.99	10.05	7.61	12.42	11.64	9.20	17.18
IRP	11.85	48.53	10.27	11.50	14.93	7.21	10.99	11.20	11.06	21.94

station	DTW	IAH	IND	JAN	LAS	LGA	MIA	MSO	MSP	MSY
GMOS	9.74	10.30	11.96	10.34	9.24	7.18	4.62	18.55	14.01	7.86
EMOS	10.71	12.56	12.62	10.98	11.75	7.55	5.80	17.70	14.82	9.70
NMOS	13.26	14.45	16.83	13.69	18.09	11.21	7.82	19.16	19.01	15.75
NWS	9.62	10.31	10.97	9.11	9.20	8.44	5.05	17.80	13.22	7.56
IRP	9.28	10.90	11.59	9.96	9.50	7.56	5.51	19.52	13.30	8.11

station	OKC	ORD	PDX	PHL	PHX	SEA	SFO	SLC	STL
GMOS	14.74	14.96	8.58	7.40	9.92	6.47	6.33	18.59	9.93
EMOS	14.94	14.54	9.26	9.24	9.38	6.61	7.97	19.45	11.61
NMOS	20.06	19.96	9.83	10.18	9.71	6.99	7.57	20.79	13.75
NWS	12.93	13.67	9.13	7.60	7.76	6.28	7.43	15.65	8.79
IRP	15.71	14.93	8.29	7.70	9.42	6.06	6.83	21.87	10.88

Table A.10.: MSE for the complete data set of Baars and Mass (2005) for combining the three MOS forecasts and the NWS forecast for predicting MIN-T by IRP. The scores are in degree Fahrenheit.

station	ABQ	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	11.68	8.05	11.77	13.20	11.53	6.81	12.40	11.76	11.78	17.65
EMOS	12.97	7.93	11.07	12.34	11.85	8.55	11.49	12.72	11.57	15.97
NMOS	11.67	328.11	15.84	15.94	16.22	10.14	14.26	14.24	16.66	22.75
NWS	10.54	7.08	9.73	10.99	10.05	7.61	12.42	11.64	9.20	17.18
IRP	11.80	47.94	10.18	11.25	14.73	7.43	11.42	11.19	10.13	22.33

station	DTW	IAH	IND	JAN	LAS	LGA	MIA	MSO	MSP	MSY
GMOS	9.74	10.30	11.96	10.34	9.24	7.18	4.62	18.55	14.01	7.86
EMOS	10.71	12.56	12.62	10.98	11.75	7.55	5.80	17.70	14.82	9.70
NMOS	13.26	14.45	16.83	13.69	18.09	11.21	7.82	19.16	19.01	15.75
NWS	9.62	10.31	10.97	9.11	9.20	8.44	5.05	17.80	13.22	7.56
IRP	8.87	10.35,	11.15	9.58	9.27	7.35	5.45	19.19	13.22	7.82

station	OKC	ORD	PDX	PHL	PHX	SEA	SFO	SLC	STL
GMOS	14.74	14.96	8.58	7.40	9.92	6.47	6.33	18.59	9.93
EMOS	14.94	14.54	9.26	9.24	9.38	6.61	7.97	19.45	11.61
NMOS	20.06	19.96	9.83	10.18	9.71	6.99	7.57	20.79	13.75
NWS	12.93	13.67	9.13	7.60	7.76	6.28	7.43	15.65	8.79
IRP	15.09	14.10	8.08	7.44	9.09	6.04	6.99	20.17	9.90

Table A.11.: Brier scores for the complete data set of Baars and Mass (2005) for combining the three MOS forecast of a season for predicting MIN-T by IRP. The scores are in degree Fahrenheit.

station	ABQ	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	10.39	7.42	10.75	12.57	11.19	7.26	11.99	11.54	9.80	16.58
EMOS	11.44	7.73	10.44	11.43	11.57	8.80	11.70	12.33	10.91	16.14
NMOS	11.27	200.76	14.08	14.61	15.68	10.63	14.48	13.51	13.88	22.14
NWS	9.75	7.21	9.13	10.28	10.25	7.40	12.27	11.66	9.51	16.49
IRP	12.19	29.70	10.75	12.39	12.78	8.85	12.10	12.11	12.13	19.30

station	DTW	IAH	IND	JAN	LAS	LGA	MIA	MSO	MSP	MSY
GMOS	9.72	10.02	11.50	10.12	8.81	7.21	4.79	17.80	13.34	7.74
EMOS	10.22	12.03	11.97	10.15	10.96	8.04	5.69	18.08	13.80	9.00
NMOS	12.21	14.02	15.03	12.59	17.18	11.89	7.34	19.18	17.52	15.57
NWS	9.38	9.76	10.73	9.36	8.98	8.49	5.23	17.40	12.77	7.77
IRP	10.23	11.26	13.00	10.08	9.48	8.35	5.68	20.16	14.29	8.70

station	OKC	ORD	PDX	PHL	PHX	SEA	SFO	SLC	STL
GMOS	13.36	14.19	8.60	7.57	8.86	6.83	6.49	16.87	9.26
EMOS	13.65	13.69	9.74	8.70	8.71	6.90	7.99	17.18	10.50
NMOS	18.91	18.48	9.87	10.20	9.01	6.85	7.57	18.79	12.41
NWS	12.68	12.72	8.97	7.69	7.11	6.50	7.76	14.72	8.82
IRP	16.24	14.47	9.69	8.27	9.92	7.26	7.95	20.22	10.87

Table A.12.: Brier scores for the complete data set of Baars and Mass (2005) for combining the three MOS forecast and the NWS forecast of a season for predicting MIN-T by IRP. The scores are in degree Fahrenheit.

station	ABQ	ATL	BHM	BNA	BOI	BOS	BWI	CLE	DAL	DEN
GMOS	10.39	7.42	10.75	12.57	11.19	7.26	11.99	11.54	9.80	16.58
EMOS	11.44	7.73	10.44	11.43	11.57	8.80	11.70	12.33	10.91	16.14
NMOS	11.27	200.76	14.08	14.61	15.68	10.63	14.48	13.51	13.88	22.14
NWS	9.75	7.21	9.13	10.28	10.25	7.40	12.27	11.66	9.51	16.49
IRP	12.20	29.64	10.38	12.21	12.69	8.93	12.56	12.09	11.66	19.20

station	DTW	IAH	IND	JAN	LAS	LGA	MIA	MSO	MSP	MSY
GMOS	9.72	10.02	11.50	10.12	8.81	7.21	4.79	17.80	13.34	7.74
EMOS	10.22	12.03	11.97	10.15	10.96	8.04	5.69	18.08	13.80	9.00
NMOS	12.21	14.02	15.03	12.59	17.18	11.89	7.34	19.18	17.52	15.57
NWS	9.38	9.76	10.73	9.36	8.98	8.49	5.23	17.40	12.77	7.77
IRP	10.14	11.07	12.74	10.04	9.55	8.32	5.65	20.04	13.61	8.73

station	OKC	ORD	PDX	PHL	PHX	SEA	SFO	SLC	STL
GMOS	13.36	14.19	8.60	7.57	8.86	6.83	6.49	16.87	9.26
EMOS	13.65	13.69	9.74	8.70	8.71	6.90	7.99	17.18	10.50
NMOS	18.91	18.48	9.87	10.20	9.01	6.85	7.57	18.79	12.41
NWS	12.68	12.72	8.97	7.69	7.11	6.50	7.76	14.72	8.82
IRP	15.78	14.33	9.47	8.37	9.34	7.28	8.10	18.81	10.40

B. Code

```
#################################################################
## Function IRP which finds a forecast for training datas y
## with the help of the training datas y,P
#################################################################
#Input:
# P_test   : n_test x m matrix of the test points
# P        : n      x m matrix of the training points
# y        : n      x 1 matrix of the observations of the training data set
#
#Output:
# y_pred   : n_test x 1 matrix with predictions for P_test
#################################################################

IRP<-function(y,P,P_test){

library(lpSolve)
library(linprog)

## Parameter ssetting and sorting
n<-length(P[,1])
m<-length(P[1,])

p_temp <- order(P[,1])
y      <- y[p_temp]

for ( i in 1:m){
P[,i]<-P[,i][p_temp]
}

## Sets up matrix A for the optimization problem
A<-createMatrixA(P)

## Performs the IRP algorithm of Chapter 2
p_fits <- IRP_cut(y,A)

## Calculate predictions out of the p_fits
y_pred <- yPrediction(P_test,p_fits,P,mean(y))

return(y_pred)
}
```

```
#########################################################################
## Creates a Matrix A with all the necessary isotonicity constraints
#########################################################################
#Input:
# P: Matrix of dimension m x n with the n forecasts of the m forecasters
#
#Output:
# Matrix A of dimension n x n
# A: A[i,j] == 1 if p_1[i]< p_2[j] and p_2[i]< p_2[j]
# but: if A[i,j]==1 and A[j,k]==1, A[i,k]==0 because of numerical reasons
#########################################################################

createMatrixA<-function(P){

n<-length(P[,1])
m<-length(P[1,])
A<-matrix(0,nrow=n,ncol=n)

## Set A[i,j]==1 if P[i,k] < P[j,k] for each component k in 1:m

 for (i in 1:(n-1)){
  for (j in (i+1):n){
   if (sum(P[i,] <= P[j,])== m){
    A[i,j]<-1
   }
  }
}

## Set A[i,j]==0 if A[i,k]==1 and A[k,j]==1

for (i in 1:(n-2)){
  for (j in (i+2):n){
   k<-i+1
   while(A[i,j]==1 && k<(j-1)){
    k<-k+1
    if(sum(P[i,]<P[k,])==m && sum(P[k,]<P[j,])==m) A[i,j]<-0
   }
  }
}

return(A)
}
```

```
#################################################################
## Cuts the dataset in smaller data sets and returns
## the optimal fits for the training data
#################################################################
#Input:
# y: n x 1 vector of the observations
# A: n xn matrix which represents the isotonicity constraints
#
#Output:
#  p_fits: n x 1 vector of optimal fits for the training data
#################################################################

IRP_cut<-function(y,A){

#### Parameters for the algorithm ####
cutvalues       <- matrix(0,nrow=1,ncol=n)
cutgroups1      <- matrix(0,nrow=1,ncol=n)
cutgroups2      <- matrix(0,nrow=1,ncol=n)
cutcount1       <- 0
cutcount2       <- 0
p_fits          <- matrix(mean(y),ncol=1,nrow=n)

## first iteration step ###
x   <- IRP_cutproblem_wrapper(y,A)
x_1 <- which(x<0.5)
x_2 <- which(x>0.5)

if (length(x_2)<n && (n-length(x_2))<n){

 cutvalues_help <- sum(y[x_2]-mean(y))-sum(y[x_1]-mean(y))
 for (j in 1:n)cutvalues [j] <- cutvalues_help
 for (j in x_2)cutgroups1[j] <- cutcount1+1
 for (j in x_1)cutgroups1[j] <- cutcount1+2
 for (j in 1:n)cutgroups2[j] <- cutcount2+1

 cutcount1   <- cutcount1 + 2
 cutcount2   <- cutcount2 + 1
 maxCutValue <- 0
}

if (length(x_1)==n || length(x_1)==0) maxCutValue = 10000

## the following iteration steps ##

while (maxCutValue !=10000){

maxCutValue <- min(cutvalues)
```

```
indCutValue <- which.min(cutvalues)
indsToCut   <- which(cutgroups2==cutgroups2[indCutValue])
temp          <- sort(unique(cutgroups1[indsToCut]))

if (10000!=maxCutValue){
 for (k in 1: length(temp)){

  indsToCut           <- which(cutgroups1==temp[k])
  p_fits[indsToCut]   <- mean(y[indsToCut])

  x   <- IRP_cutproblem_wrapper(y[indsToCut],A[indsToCut,indsToCut])
  x_1 <- which(x<0.5)
  x_2 <- which(x>0.5)
  L   <- length(indsToCut)

 if (length(x_1)<L && length(x_2)<L){
  cutvalues_help  <- sum(y[indsToCut[x_2]]-mean(y[indsToCut]))
                     -sum(y[indsToCut[x_1]]-mean(y[indsToCut]))
  for (j in indsToCut)cutvalues[j]        <- cutvalues_help
  for (j in indsToCut[x_1])cutgroups1[j] <- cutcount1+1
  for (j in indsToCut[x_2])cutgroups1[j] <- cutcount1+2
  for (j in indsToCut)cutgroups2[j]       <- cutcount2+1

  cutcount1   <- cutcount1 + 2
  cutcount2   <- cutcount2 + 1
  maxCutValue <- 0
  }
 if (length(x_1)==L || length(x_1)==0){
  cutvalues[indsToCut] = 10000
  }
 }
}
}

return(p_fits)
}
```

```
#####################################################################
## Solves the linear optimization problem for the optimal cut
#####################################################################
#Input:
# y_temp : n_temp x 1 vector of the observations
# A_temp : n_temp x n_temp matrix with the isotonicty constraints
#
#Output:
# x: n x 1 vector with x[i] is 0 or 1 representing the two sub-groups
#################################################################### ###########

IRP_cutproblem_wrapper <- function(y_temp, A_temp)
{
z<-0
row<-1
n_temp<-length(y_temp)

## count the number of isotonicity constraints

for(i in 1:(n_temp-1)){
 for(j in (i+1):n_temp){
  if(A_temp[i,j]==1) z<-z+1
 }
}

A_input<-matrix(0,z+n_temp,n_temp)
b<-matrix(0,z+n_temp,1)

## generate a z x n_temp Matrix A_input with x_i - x_j <= 0 if A[i,j]==1
## and a z x 1 vector b with zeros

for (i in 1:(n_temp-1)){
 for (j in (i+1):n_temp){
  if (A_temp[i,j]==1){
    A_input[row,i]  <- 1
    A_input[row,j]  <- -1
    b[row]          <- 0
    row             <- row + 1
    }
  }
 }

## add the constraints that x_i<=1 for all i in 1 to n to A_input and set
## b[(z+1): (z+n)]=1

for(i in 1:n_temp){
   A_input[row,i] <- 1
```

```
    b[row]          <- 1
    row             <- row + 1
}

## solve the problem  max((p-mean(p))*x) with Ax<=b with linprog library

c<- mean(y_temp)-y_temp

solution<-solveLP(c,b,A_input,maximum=FALSE,,
                  const.dir = rep( "<=", length( b) ),lpSolve=TRUE)\$solution

return(solution)
}
```

```
#################################################################
## Finds the best forecast for a test data set y
#################################################################################
#Input:
# P_test: n_test x m  matrix of the forecasts of the test data set
# p_fits: forecasts of the training data set generated by IRP
# P     : n x m       matrix of the forecasts of the training data set
# imPred: prediction value for a point if there aren't any below or above
#
#Output:
# y_pred: n_test x 1 vector with forecasts for the test data set
#################################################################

yPrediction<-function(P_test,p_fits,P,imPred){

y_pred           <- matrix(0,length(P_test[,1]),1)
number_training  <- length(p_fits)
m                <- length(P[1,])

## Calculates the forcasts for each test point seperated

for (i in 1:length(P_test[,1])){

indsL   <- 0
indsU   <- 0
compare <- 1

## Finds all the points of the training data set
## which are below the test point

for( j in 1:number_training){
    if(sum(P[j,] < P_test[i,])==m) indsL<-c(indsL,j)
    }

## Finds all the points of the training data set
## which are above the test point

for( j in 1:number_training){
    if(sum(P[j,]>P_test[i,])==m) indsU<-c(indsU,j)
    }

## if there aren't any points above or below the
## forecast value is incompPred

if( length(indsL)+length(indsL)==0){
  compare   <- 0
  y_pred[i] <- imPred
```

```
  }

## if there are only points above chooses the minimum value of p_fits

if ( length(indsL)==1){
  compare   <- 2
  y_pred[i] <- min(p_fits)
}

## if there are only points below chooses the maximum value of p_fits

if ( length(indsU)==1){
  compare   <- 2
  y_pred[i] <- max(p_fits)
}

## if there are points below or above chooses the mean
## of the maximum value of the points below the point
## and the minimum value of the points above the point

if (compare==1){
   indsL   <-  indsL[2:length(indsL)]
   indsU   <-  indsU[2:length(indsU)]

   ##find_comparables(y[i,],p_1,p_2)
   p_L_fits  <- max(p_fits[indsL])
   p_U_fits  <- min(p_fits[indsU])
   y_pred[i] <- (p_L_fits+p_U_fits)/2
   }

}
return(y_pred)
}
```

Erklärung

Hiermit versichere ich, dass ich meine Arbeit selbstständig unter Anleitung verfasst habe, dass ich keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, und dass ich alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entlehnt sind, durch die Angabe der Quellen als Entlehnung kenntlich gemacht habe.

Datum Unterschrift