International Institute for
Applied Systems Analysis
Schlossplatz 1
A-2361 Laxenburg, Austria

Tel: +43 2236 807 342
Fax: +43 2236 71313
E-mail: publications@iiasa.ac.at
Web: www.iiasa.ac.at

Interim Report **IR-07-034**

Qualitative models of climate variations impact on crop yields

Wojciech Kotłowski (wojciech.kotlowski@cs.put.poznan.pl)

Approved by

Marek Makowski (marek@iiasa.ac.at)
Leader, Integrated Modeling Environment Project

November 2007

Foreword

This report describes the research the author advanced during his participation in the 2005 Young Scientists Summer Program (YSSP), and continued after the YSSP, albeit – due to the time constraints – with a much lower intensity than during the YSSP.

The reported research deals with application of rough sets theory to the analysis of complex data thus contributes to the IME research agenda, which includes development of methodologies and software tools for analysis and support of complex decision problems. The reported work is interdisciplinary in nature. It not only requires combining thorough comprehension of mathematical modeling, optimization techniques, and relational databases with the understanding of complex analysis of agriculture and climatic data in the context of modeling crop yields. It also required cooperation with colleagues of the IIASA Land Use Change and Agriculture (LUC) Program. Moreover, the author was a member of a three-person YSSP team working on researching different approaches to the same problem.[1] Such a team work was very fruitful. We mention here only the main aspects of this successful collaboration:

- Different approaches are complementary, i.e., they provide various insights to the same problem.
- Team-members having different skills and experiences have learned from each other.
- Time-consuming jobs of data collection, verification, and processing (by organizing a dedicated database and the supporting software) were shared by team members.

The Summer-time of the YSSP is only three months short, and this type of research requires substantial amount of time for the initial stage of the research, which includes a correct specification of the problem, and then data collection, verification, and processing. The problem is challenging, therefore no standard ready-to-use methods were available. The author therefore had to first advance his research on machine learning and rough sets theory, develop or adapt software to support the new approaches to data analysis, and then to verify various approaches with the data before discussing the results with the colleagues from the LUC Program. Therefore only preliminary results were achieved during the Summer'2005, and documented in a draft report.

The constructive criticism by the LUC colleagues had led to the development of better methods and the corresponding software tools. Therefore the results presented in this report are substantially better than the results achieved during the Summer'05. One of the main advantages of the described approach is that it does not require an advanced expert knowledge nor assume anything about the data distribution: all conclusions are derived from the data. Moreover, although the approach is relatively (in comparison to most commonly used data analysis approaches) simple it is characterized by high accuracy and efficiency.

[1] The results of another approach are described by Bartosz Kozłowski in IR-07-038.

Abstract

This report presents application of machine learning methodology in modeling process. The objective is to identify and explain impact of weather variations on crop yields, and to test the approach on the agricultural and climatic data from the USA.

First, separation of weather and non-weather factors is performed by trend identification — two methods of trend identification are considered. Then the importance of the attributes is assessed using information gain measure and attributes are also aggregated into seasons. Finally, four types of classification methods (support vector machines, nearest-neighbors classifier and two variants of decision rules induction) are applied to the data and the results are compared and analyzed.

The proposed approach differs from standard approaches to the crop yields modeling. It does not require a lot of expert knowledge, nor assume anything about the data distributions. All conclusions are drawn from data and the final model is build only from data. This approach is much simpler, however it maintains high accuracy and performance.

Keywords: machine learning, rule induction, rough sets theory, attribute selection, dominance-based rough sets approach (DRSA), decision support, multicriteria decision analysis.

Acknowledgments

This report and all the research described was done during my participation in the Young Scientists Summer Program (YSSP) 2005 at the International Institute of Applied Systems Analysis in Laxenburg, Austria.

I would like to thank to Dr Marek Makowski for his supervision and lot of priceless help he gave me during the program. I also give many thanks to Günther Fisher and Harrij Van Velthuizen for valuable advices and answers, Bartosz Kozłowski and Gleb Peterson for their cooperation.

Finally, I would like to thank prof. Roman Słowiński from Poznań University of Technology for scientific help, and the Polish National Member Organization of IIASA for the financial support which made his participation in the YSSP possible.

About the Author

Wojciech Kotłowski received his M.Sc. in Computer Science from Poznań University of Technology in 2004. The title of his thesis was "System of gradual rules induction with monotonic relationship between premise and conclusion". He is a Ph.D. student in the Laboratory of Intelligent Decision Support Systems. He also received his M.Sc. in Physics of the Physics Department at Adam Mickiewicz University, specializing in theoretical physics (quantum optics and quantum information). His main scientific interests include: machine learning, rough sets theory, multicriteria decision analysis and decision support systems.

Contents

List of Tables

List of Figures

Qualitative models of climate variations impact on crop yields

Wojciech Kotłowski[*] *(wojciech.kotlowski@cs.put.poznan.pl)*

1 Introduction

It is obvious that even nowadays, despite the high technological level, weather conditions have a crucial influence on crop yields. However, by using common sense and experience of those involved in agriculture only a general conclusion can be drawn with lack of precision, since the whole process seems to be complex. Moreover, there are dozens of non-weather factors (fertilization, mechanization, soil quality), which made an even stronger influence and also undergo temporal changes. The most common way for crop yields analysis is to build a sophisticated model, basing it on expert knowledge, agriculture, climatology, hydrology, etc., taking into account many different variables, dependent from a big amount of parameters. This approach may give good results, however it is very complex and usually depends on many assumptions, which in reality does not need to be valid.

There is another possibility to cope with modeling. On the contrary to the above described *model-driven* approach, one can neglect most of the expert knowledge and try to build a model to explore knowledge from data — which is known to be *data-driven* approach. Such an approach has of course both advantages and disadvantages. The main disadvantage is that it neglects certain facts well known by experts, it strongly simplify the whole system and sometimes wrong conclusions can be drawn from data. The advantage is that the whole modeling process is usually much simpler and it does not need a deep domain knowledge and uses less assumptions.

In computer science there is a data-driven methodology called *machine learning*. Originally it is part of artificial intelligence and is related to systems, which learn how to predict or react using only information found in data. However, this is a collection of statistical tools, which can be used for data analysis and modeling of complex systems. The modeling process starts with data and tries to get as much information from it as possible, with the smallest possible number of assumptions. Unlike parametric statistics, there are usually no assumptions about a priori distribution. The theory behind machine learning introduces bounds on real error, which are *distribution free*, i.e. the bounds hold whatever the distribution generating the examples (observations). The only thing assumed is that there exists some probability distribution from which data are drawn, so data are assumed to be *independently drawn and identically distributed* (“iid”). Therefore

[*]Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland.

this methodology can give better results if no knowledge about data set (e.g. probability distributions) is given.

The main goal of the analysis is to explain the impact of weather on crop yields. Having the weather data (precipitation, average temperature, minimal temperature, maximal temperature) collected monthly and yield data collected yearly we may try to build a model describing the impact of the weather variability on crop yields.

The idea is to build a qualitative model using methodology which comes from machine learning, namely decision rules. From a statistical point of view, this is a kind of nonparametric regression. The qualitative approach is considered, since it is much easier to understand by decision maker and since we believe the very exact (numerical) values of yields obtained using the model are meaningless because of high and unknown uncertainties and imprecision of analysis. However, this qualitative approach may still be used for future prediction and build model is exact and precise.

In our study, we use data from the United States and China. The reason is that in these two regions the data were easily accessible for us. We focus only on the most important crops, like rice and soya in China or corn and wheat in USA.

2 Methodology

Before going into details, let us introduce some symbols, which will be used through the paper.

The data consists of time series from different points (regions) — we have both temporal and spatial distribution and different approaches must be used to deal with them. By y_i we mean a time series of yields from region i (where i might be an index from the set of counties, but also a set of provinces — we do not specify it here). If we want to mention the exact value of yields in certain year t, we will write $y_i(t)$ (t is a discrete time variable measured in years). Moreover, by $p_i(\tau)$, $t_i^{\text{avg}}(\tau)$, $t_i^{\text{min}}(\tau)$, $t_i^{\text{max}}(\tau)$ we denote precipitation, average, minimal and maximal temperature in point i and time τ (τ measured in months).

2.1 Upscaling

We need to deal with the different resolutions of both spatial and temporal distribution of yield and weather data. Since we are able to cope with the temporal differences while model constructing, the spatial difference in resolutions of yield and weather data seems to be more inappropriate and should be eliminated in data preprocessing phase. We will describe briefly which assumptions were adopted for upscaling both in USA and China.

2.1.1 U.S. Weather Data

Weather data in USA are collected by 4280 weather stations spread across the whole country and the yield data are collected at the level of counties. There are 3255 counties, which gives us in most cases one-to-one correspondence between stations and counties. In order to map weather stations (described by longitude and latitude coordinates) to counties, administrative map has been used. However, in some counties there might be

more than one station, or there might be no station. Therefore, an upscaling procedure has to be used. The rules of upscaling are as follows:

- If there are more then one station in a given county, we take the average of each weather attribute collected from these stations.
- If there is no station in a given county, we do not consider this county during the analysis because without having any additional knowledge this is the most prudent approach.

2.1.2 China Weather Data

Weather data in China are collected in square grids of size $30' \times 30'$ (about 56km $\times$ 56km on the equator, smaller width above and below). Yields data are collected in much smaller resolutions - by provinces. Thus an efficient method of upscaling was needed. The administrative map, which assigns a province name to each point (and therefore to each grid) is available. By using the map, we aggregated the weather in each province and do the analysis on the province level. We introduce weight w_i to each grid, which is the area of cultivated land in each grid. Using the weight w_i and having (for instance) precipitation $p_i(\tau)$ in some time τ, where i is the index of the grid, we may write:

$$p_r(\tau) = \frac{\sum_{i \in \text{PROV}_r} w_i p_i(\tau)}{\sum_{i \in r} w_i} \tag{1}$$

where r is a province and PROV_r is the set of all grids in province r.

2.2 Trend Identification

It is not obvious how to distinguish between the influence of climate and the influence of other factors, such as fertilization, mechanization, soil type, etc. Since we want to investigate only the climate impact, we need to identify the non-climate factors in the preprocessing phase.

We assume no additional knowledge about these factors and their temporal changes. One thing, which should be taken into account is the fact, that the amount of yields keeps growing (with variations caused by weather) along the time, and the growth seems to be linear. This is due to general agricultural development, improving the methods of crops cultivation, inventing better types of fertilizers.

The identification of non-climate (non-weather) factors comes with assumption that all of them corresponds only to long-term variations as opposite to short-term variations of weather. Therefore, we smoothened the signal (time series) so as to obtain the general trend and neglect variations. We proposed and applied two methods of obtaining the trend:

- Assume that the trend is linear; an advantage of this approach is that the trend is simple (it has only two degrees of freedom), so a linear function will only explain the trend, not the variations. One can easily verify, that as the amount of yields grow, also the amplitude of variations grow and we may guess that impact of climate depends on value of the trend. So we think that instead of least squares regression

it is better to use weighted least squares. So, if $y(t)$ is a real value of yields, $\hat{y}(t)$ is an estimator, instead of minimizing:

$$\sum_i (\hat{y}(t_i) - y(t_i))^2 \tag{2}$$

we minimize:

$$\sum_i \frac{1}{y(t_i)} (\hat{y}(t_i) - y(t_i))^2 \tag{3}$$

After identifying the trend $\hat{y}(t)$ we use the value $y(t) - \hat{y}(t)$, which corresponds to weather impact. However, if we want the further analysis to be consistent with the assumption that weather impact depends on value of trend (in other words, we consider relative variations), we define the weather impact on yields as:

$$c(t) = \frac{y(t) - \hat{y}(t)}{\hat{y}(t)} \tag{4}$$

so we take into account only the percentage value of variations.

- Assume that trend has a polynomial form. We have arbitrarily chosen two types of polynomial, with a high degree (15) and a low degree (5). The polynomial approximation is more flexible than the linear one and may be applied when there no constant development pace. Both models will be considered in parallel, and the one with better accuracy will be chosen.

Two examples of trend identification were shown on Figure 1. On first chart there are real data (yields) and linear trend. Notice that the linear function captures the growth of the yields quite well (because the growth is roughly constant). On the second chart one can see pure weather impact on yields from (4), i.e. percentage value of variations; the variations mildly increase with time and become less "stable". The third picture shows polynomial trend identification. Polynomial function fits the data very well (linear trend may fail here, because the growth does not look linear), but it may also capture the variations (see the right end of the line) which should be avoided.

2.3 Classification Task

2.3.1 Machine Learning

Machine learning is a domain based on statistics and data analysis. It deals with problems where the relation between input and output variables is not known or hard to get. The idea is to train a system how to obtain the proper function only by giving the system examples (observations). In other words, the system must *generalize* the knowledge taken from a set of examples. This is why the learning process such machines is often called *induction* or *inductive learning*.

The goal of the analysis is, as it was mentioned before, to explain the amount of yields in terms of weather variability. It would be possible to explain yields by using some statistical tools, e.g. linear regression. However, we always assume a priori a certain kind of function and a certain kind of distribution on data set.

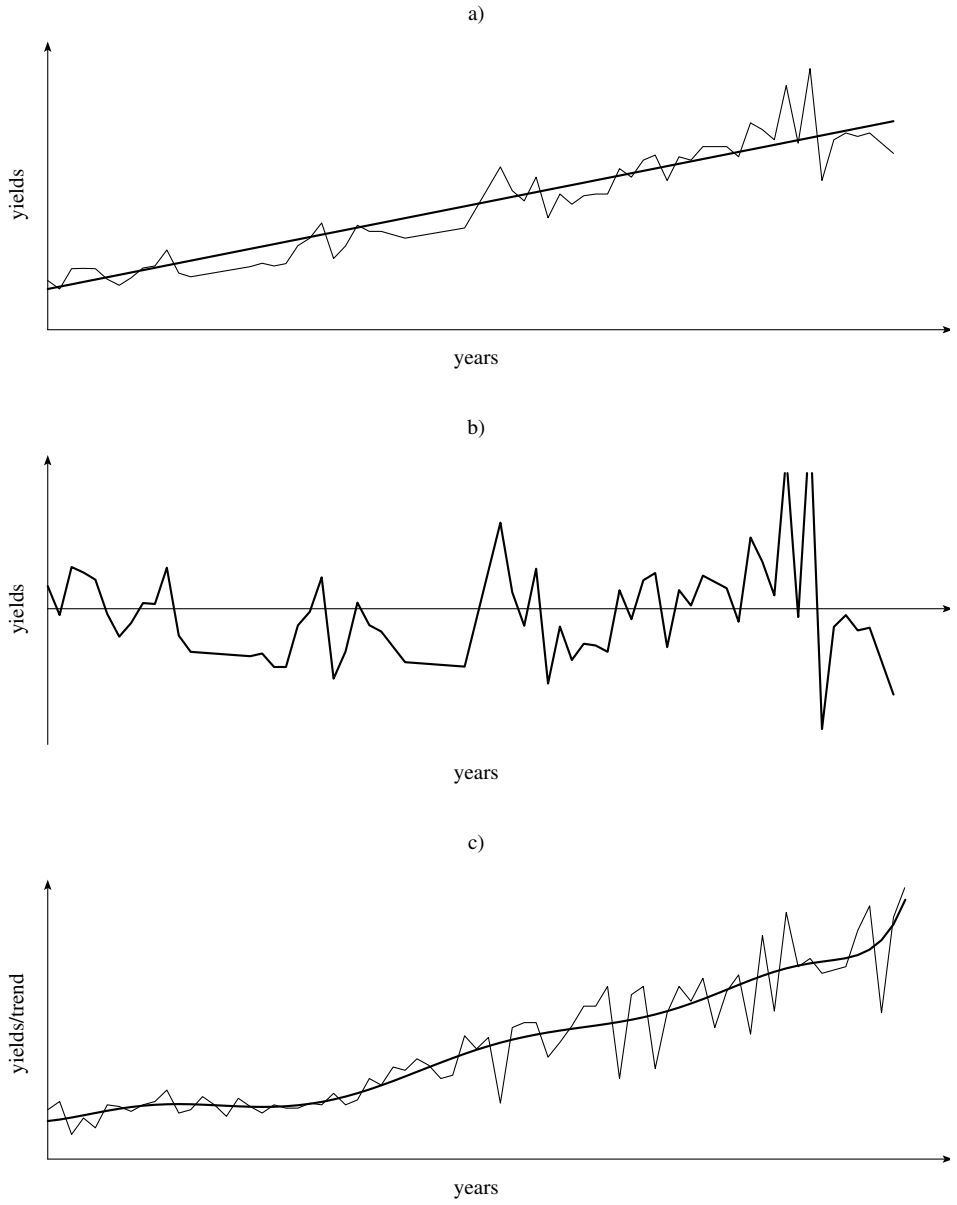


Figure 1: Trend identification in data — winter wheat yields from Accomack (Virginia): a) yields and linear trend b) climate impact $c(t)$ (note that different scale for yields has been used) c) corn yields from Alexander (Illinois) — polynomial trend

Machine learning methodology usually does not need such assumptions or the assumptions are much weaker. In particular, decision rules induction may give any kind of function and, as the number of observations increases, any kind of function can be approximated with any given accuracy.

General introduction to machine learning can be found in [9].

2.3.2 Classification vs. Regression

To build a model we need a set of observations (objects) with their characteristics, which is called a *training set*. The variables describing objects are called *attributes* or *features* (in this paper, first term will be used). Every object is described by n *conditional* attributes and one *decision attribute* — the attribute we want to explain in terms of the condition attributes. In other words, we want to find a function $f\colon A_1 \times \ldots \times A_n \to D$, where A_i is a domain of i-th conditional attribute D is a domain of decision attribute. If D is a finite set, one call such analysis *classification* and each of elements of D defines a *class*. If D is infinite and continuous, one calls the analysis *regression*.

In the approach described in this paper we use classification for explanation of weather impact on yields. This is due to following facts:

- It is much easier for a decision maker to understand and interpret the model, when there is a small possible number of output values; e.g. it is easier to speak about yield in terms of "low" or "high" than giving the exact value of percentage of variation around the trend.
- There is too much uncertainty in the model phenomenon to even try to predict exact values of yield. Uncertainties comes from the fact, that we narrow the analysis only to climate factors and we try to estimate any other factor by trend identification. Therefore results are rough, so it is better to do the analysis on a smaller resolution (a smaller number of possible output values).

To perform classification, we need to convert continuous decision attribute into discrete attribute with finite number of values (classes). This problem will be considered later.

2.3.3 Decision Table

To gather information about a training set in a comprehensive way, one can introduce a *decision table*. Simplifying the notation, we have a set of ℓ objects (observations) described by n *conditional* attributes $a_1, \ldots, a_n$ and one *decision attribute* d (the attribute we want to explain). Let $a_i(x)$ be a value of object x on attribute a_i and $d(x)$ — a value of object x on decision attribute. We recall, that by A_i and D we denote the domain of attributes, i.e. the set of all possible values of this attribute. Such set of objects with a description by the set of attributes is called *decision table*, since it may be shown as a table where rows are objects and columns are attributes (see Table 1).

2.4 Attribute Assessment and Transformation

There is often a need for assessing general importance of condition attributes. By importance of a given attribute we mean the influence of this attribute on the value of the decision attribute, or in other words, the amount of information about the decision attribute that a given attribute brings to the problem.

There are many different measures of importance used in machine learning algorithms [9]. Here, we focus on the most popular one, based on concept of *entropy* introduced in the information theory.

$t^{\mathbf{min}}$(**march**)	$t^{\mathbf{max}}$(**may**)	$t^{\mathbf{max}}$(**june**)	p(**june**)	**yields**
-5.2	24.7	29.5	92.4	low
-3.8	24.1	27.4	197.4	high
...	...	...	...	...
-4.2	25.6	25.7	99.8	low
-2.8	20.2	31.0	117.6	high

Table 1: Decision table. Attribute *yields* is a decision attribute d, rest are conditional attributes $\{a_1, \ldots, a_4\}$

Given a collection of objects S, containing subset S_1 of objects from class 1, subset S_2 of objects from class 2, ..., subset S_m of objects from class m, the entropy of S is:

$$H(S) = \sum_{i=1}^{m} -p_{S_i} \log p_{S_i} \tag{5}$$

where

$$p_{S_i} = \frac{|S_i|}{|S|} \tag{6}$$

is estimator for probability of object from class i. We will focus on two-class problem, where we have only positive class $\oplus$ and negative class $\ominus$, so the definition (5) can be written as:

$$H(S) = -p_{\oplus} \log p_{\oplus} - p_{\ominus} \log p_{\ominus} \tag{7}$$

The measure we use to grade attribute a_i is called *information gain*. It is the expected reduction of entropy caused by partitioning the set of objects according to this attribute:

$$IG(S, a_i) = H(S) - \sum_{v \in A_i} \frac{|S_v|}{|S|} H(S_v) \tag{8}$$

where the sum is over all possible values from domain A_i of attribute a_i and S_v is subset of objects for which $a_i(x) = v$, $S_v = \{x \in S : a_i(x) = v\}$.

Since we deal with continuous attributes (i.e. each has an infinite number of values), first we *discretize* the attribute so it has only two values. It is done, by taking a threshold c_i. If we denote new, discretized attribute by a'_i then for each object, if $a_i(x) < c_i$ then $a'_i(x) = 0$ and if $a_i(x) \geqslant c_i$ then $a'_i(x) = 1$. The threshold c_i is chosen to maximize $IG(S, a_i)$.

The attributes can be ranked by value of $IG(S, a_i)$. Attribute a_i with high value of information gain are important for classification, since introducing such attribute increases the information about the decision attribute.

The thing, which can possibly increase the prediction accuracy of learning algorithms is the transformation of the attributes. Having a set of attributes $\mathcal{A}$, we transform them to a new set $\mathcal{A}'$, which might be more appropriate for our purposes. In the presented analysis, we join monthly attributes to seasonal attributes (e.g. instead of precipitation in June, July and August, we use precipitation in summer defined as the average of the monthly values) and try to improve accuracy of the model. By transformation we can also aim at decrease the number of attributes. In our study, two sets of attributes has been used: the original set of month-based attributes and the transformed set of season-based attributes.

2.5 Support Vector Machines

The following chapter is based mostly on introductory texts [4, 10].

For simplicity of further equations we assume that decision of each object has values from the set $D = \{-1, 1\}$, where we code the "positive" class (e.g. high yields) by 1 and "negative class" (low yields) by -1. We also denote the vector of conditional attributes for object x, i.e. $(a_1(x), \ldots, a_n(x))$ by $\mathbf{x}$. Assume that there exist an unknown probability distribution $P(\mathbf{x}, d(x))$ (we remind, that $d(x)$ is a decision of the object x), from which the observations are drawn. We would like to find a function $f(\mathbf{x}, \alpha)$, where α is some parameter's vector, which minimizes the expected test error (*risk*) [12]:

$$R(\alpha) = \int \frac{1}{2}|d(x) - f(\mathbf{x}, \alpha)|dP(\mathbf{x}, y) \tag{9}$$

It is not possible to perform this task directly (since the underlying probability distribution is unknown). Therefore one usually minimize the approximation of (9), *empirical risk*:

$$R_{emp}(\alpha) = \frac{1}{2\ell}\sum_{i=1}^{\ell}|d(x_i) - f(\mathbf{x}_i, \alpha)| \tag{10}$$

However, we need to define apriori the class of function $\{f(\mathbf{x}, \alpha)\}$. Too narrow class may lead to situation, that function, which minimize (9) fall outside the class (high bias). On the other hand, too rich class compared with ℓ may lead to overfitting to given data (high variance). Moreover, minimization process should be computationally efficient. Functions for which the last criterion is surely satisfied are linear functions of the form:

$$f(\mathbf{x}, \mathbf{w}, b) = \mathbf{w} \cdot \mathbf{x} + b \tag{11}$$

For simplicity we later omit the parameters $\mathbf{w}, b$ and denote $f(\mathbf{x})$. Using linear functions greatly simplifies the optimization process. However, for some real-life problems, the dependencies between output and input are strongly nonlinear and it is not possible to obtain adequate linear model. Then, to still the advantages of linear functions and extend the function class, one introduce a nonlinear mapping:

$$\begin{array}{rcl} \mathbf{\Phi}\colon \mathbb{R}^n & \to & \mathcal{F} \\ \mathbf{x} & \mapsto & \mathbf{\Phi}(\mathbf{x}) \end{array} \tag{12}$$

Now one can consider the same function class, but defined on $\mathcal{F}$ instead of $\mathbb{R}^n$. The class of linear function would be now:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{\Phi}(\mathbf{x}) \rangle + b \tag{13}$$

In some cases, input data enters the optimization problem only in form of the dot products $\mathbf{x}_i \cdot \mathbf{x}_j$. Then, mapping the objects to $\mathcal{F}$ we obtain:

$$\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_i) =: k(\mathbf{x}_i, \mathbf{x}_j) \tag{14}$$

where $k(\mathbf{x}_i, \mathbf{x}_j)$ is a *kernel function*. This trick simplifies greatly the complexity of the problem, since the only thing we need is $n \times n$ kernel matrix, even if $\mathcal{F}$ has a very

high dimension. Conversely, introducing a function $k(\mathbf{x}_i, \mathbf{x}_j)$ corresponds to some input space transformation $\mathbf{\Phi}(\mathbf{x})$. The necessary and sufficient conditions that symmetric kernel functions must satisfy, are Mercer's conditions [12]:

$$\int k(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) f(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geqslant 0 \tag{15}$$

for all $f \in L_2(\mathbb{R}^n)$. The equivalent finite dimensional conditions is the requirement of positive semi-definiteness of matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

The most popular algorithms using the kernel trick are support vector machines (SVM). One uses the linear class of function and minimize simultaneously norm of the weight vector $\mathbf{w}$ and number of misclassified objects in training set (so called "soft margin" case). The corresponding mathematical programming problem is following:

$$\begin{aligned} \text{min:} \quad & \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{\ell} \xi_i & (16) \\ \text{subject to:} \quad & d(x_i)(\mathbf{w} \cdot \mathbf{x}_i + b) \geqslant 1 - \xi_i \qquad i = 1 \ldots \ell \end{aligned}$$

where C is a complexity constant (corresponding to the capacity of a class of functions). Thus one shall minimize the norm subject to constraint to avoid objects in unit margin around the decision boundary $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = 0$. Since the goal function is convex and the constraints are linear, there is no "duality gap" so that the optimal solution of problem (16) and the optimal solution of the dual problem form the saddle point, therefore one can obtain the optimal solution to the primal problem by solving the dual. The dual problem is the following:

$$\begin{aligned} \text{max:} \quad & \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j d(x_i) d(x_j) \mathbf{x}_i \cdot \mathbf{x}_j & (17) \\ \text{subject to:} \quad & \alpha_i \geqslant 0 \qquad i = 1 \ldots \ell \\ & \sum_{i=1}^{\ell} \alpha_i d(x_i) = 0 & (18) \end{aligned}$$

Notice, that input values appear in the problem only by the values of dot products. Thus, it is possible to apply the kernel trick here. The only change in the corresponding nonlinear extension of (17) is the kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ instead of the dot product.

SVM approach has an advantage of controlling the complexity of function class by minimizing the norm in the function space. The space itself can be highly dimensional, but the appropriate complexity parameter C should avoid overfitting to data.

2.6 K Nearest Neighbours

K Nearest neighbours [8] algorithm is a very simple yet efficient classification and regression procedure (here we consider the classification case only). It is a member of a family of methods called "lazy learning" algorithms, since in fact there is no learning process beforehand – all we need for classification is the training set.

The classification process proceeds as follows. Assume we have some distance measure (metric) $\delta: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$, which assigns a non-negative real value to each pair of objects. Suppose we classify an object x described by the corresponding vector of condition attributes $\mathbf{x}$ (notation as in Section 2.5). Consider k closest (in terms of distance measure δ) objects from the training set, denoted $x_1, \ldots, x_k$. The object x is classified by assigning the majority vote over nearest neighbours $x_1, \ldots, x_k$, i.e. assigning the most frequent decision value of nearest neighbours. If there is a tie on the frequency of two or more decision values, another procedure must be used to resolve it. Usually the class with the lowest label is chosen or random choice between tied classes is made. In case of binary classification, ties can be avoided by simply choosing the odd value k.

Nearest Neighbours are proved to be (under mild assumptions about probability distribution and value of k) asymptotically optimal for any distance measure. However, in case of finite training set, a choice of metric can be crucial to the performance of the method. A metric which is known to work well in many cases is simple Euclidean distance:

$$d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}|| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{19}$$

However, all the attributes must be normalized (scaled to the $[0, 1]$ range) or standardized (zero mean and variation equals to unity) before using the Euclidean distance, otherwise attributes with larger values would dominate those with smaller values during the distance calculation. The Euclidean metric will be used in our study. Its main drawback is that it treats each attribute equally (without taking into account its importance) and thus is sensitive to the number of irrelevant attributes (which do not have influence on the decision attribute). On the other hand, it is simple and works well in most cases.

2.7 Decision Rules

2.7.1 Introduction

Decision rule induction is a well-known methodology in machine learning domain, used for inducing the knowledge from a data set. The knowledge is represented in a very simple form of implications: "if *conditions* then *conclusion*". Since it may be treated as a part of data analysis and statistics, one can think about this methodology as a general method of nonparametric regression. A nice overview of this methodology can be found in [9].

Decision rules considered here has a general form:

$$\text{if } condition_1 \wedge condition_2 \wedge \ldots \wedge condition_p \text{ then } conclusion \tag{20}$$

In other words, there is one conclusion and some conditions connected with AND operator. It is easy to show that if we have a rule, in which some of the conditions are connected with OR operator, it is possible to resolve it to a set of rules of the form (20).

We consider only conditions of the form "$a_i(x) \geqslant c_i$" and "$a_i(x) \leqslant c_i$", since all of the condition attributes used in our study are continuous. Then a conclusion has the form "$d(x) = \beta$".

The strength and generality of the model is easier to see if we write the decision rule in the following way. Assuming (without loss of generality) that smallest possible decision value is 0, decision rule R_k can be treated as a function $d_k(x)$ of the form:

$$d_k(x) = \beta\theta(a_{i_1}(x) - c_{i_1}) \cdot \ldots \cdot \theta(a_{i_k}(x) - c_{i_k})\theta(c_{i_{k+1}} - a_{i_{k+1}}(x)) \cdot \ldots \cdot \theta(c_{i_p} - a_{i_p}(x)) \quad (21)$$

where $\theta(x)$ is step function:

$$\theta(x) = \begin{cases} 1 & \text{if } x \geqslant 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (22)$$

The function explaining the decision attribute, $d(x)$, is composed from local functions $d_k(x)$ in the two different ways. In case of the classical rule induction the composition has the following form:

$$d(x) = \max_{R_k \in \mathcal{R}} \{d_k(x)\} \quad (23)$$

where $\mathcal{R}$ is a set of decision rules. In case of ensembles of decision rules (explained below), the function takes the additive form:

$$d(x) = \sum_{k=1}^{M} d_k(x) \quad (24)$$

One can easily check, that any "reasonable" (integrable) function $f(x)$ can be approximated (with norm) by $d(x)$. This shows the power of the decision rule model as a universal approximation method.

The generality of the model is also its drawback. It can be shown (in computational learning theory [1]), that the wider the function space is, the bigger number of objects we need to build accurate model and avoid overfitting to data. Therefore it is not enough to find the set of rules which is consistent with training set, one usually try to find the simplest set of rules, following a paradigm called *Ockham razor* [2], which says that the simplest possible description seems to be most the appropriate.

We say, decision rule R_i is satisfied by object x (or we say that x *covers* the rule) if the object satisfy all of the conditions (left hand side part of the rule). Then, the right hand side value of the rule (conclusion) is denoted by $R_i(x)$.

2.7.2 Classical Rule Induction

There are many approaches, which leads to obtain a set of decision rules from a decision table. In this subsection we focus on "classical" way of rule induction. Roughly speaking, in concordance with the already mentioned Ockham razor paradigm, the goal is to induce the smallest number of strongest possible decision rules which properly cover (i.e. cover indicating proper decision value) all of the objects.

Decision rule R_1 is *stronger* than R_2 if **both** conditions hold:

1. R_1 and R_2 has the same right hand side (they assign objects to the same class)
2. For each object x, if R_2 covers x, then R_1 also covers x; we remind, that rule covers the object, if the object satisfies all the condition on the left hand side of the rule; In other words R_1 is *more general* than R_2.

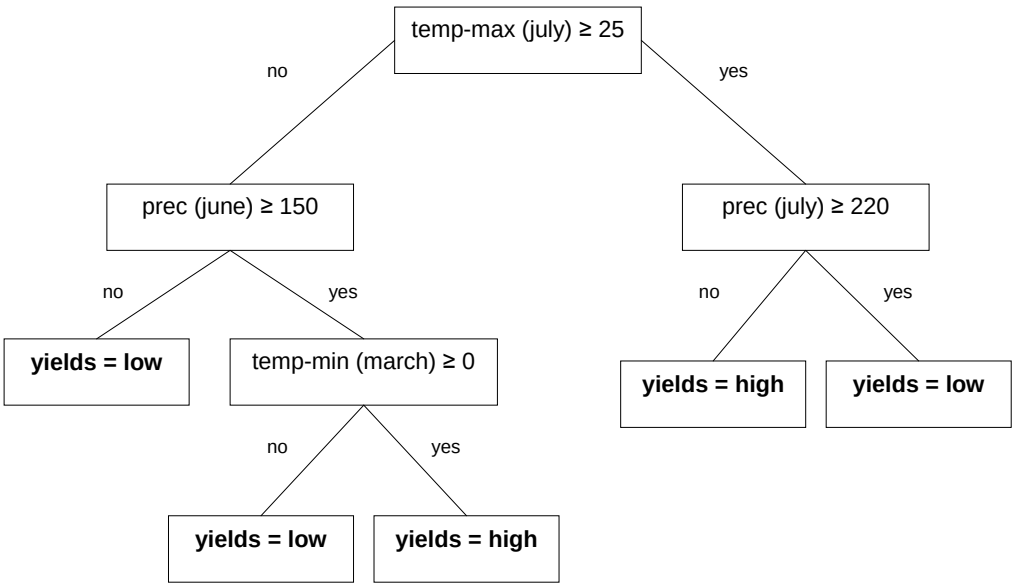


Figure 2: An example of decision tree

Decision rule R_1 is *minimal* if there exists no other decision rule R_2 such that R_2 is stronger than R_1.

Strictly speaking the objective of decision rule induction algorithm is to obtain the smallest set of minimal rules. Unfortunately this problem was proven to be NP-complete, hence all the induction algorithms are heuristic.

There are many algorithms for performing rule induction. We decided to use PART algorithm [6]. The algorithm of rule induction is based on very famous algorithm for generating decision trees — C4.5 [11], and the latter one must be explained first.

A decision tree is a tree graph structure, where in each node there is a test. A test on attribute a_i (we restrict ourselves again to continuous attributes) has following form: "$a_i \geqslant c_i$". Depending on the answer (true or false) we move to one of the two possible children of the node and perform another test. Testing is continued until a leaf of the tree is found, where the indicator of the class is located (final assignment to a class). An example of the decision tree is shown of a Figure 2.

C4.5 builds a tree by ranking the attributes due to information gain measure (IG). A test is performed on attribute with the highest IG and the process is repeated recursively for each child, until the exact assignment to the class is found. However, too complex tree may easily lead to overfitting, thus *pruning* procedure is performed to reduce some subtrees into nodes.

PART deals with such pruned partial trees and extract from a path from root to the leaf a decision rule. It continue the process until all the examples are covered. Details of the algorithm may be found in [6].

2.7.3 Rule Induction by Boosting

In the recent years, a new methodology based on combining the family of simple classifiers was introduced under the name *boosting* [7, 8]. Suppose we have already trained $m-1$ classifiers on the training data. Roughly speaking, the mth classifier is learned by putting more weight to the objects which were incorrectly classified by previous classifiers and less weight to the objects which did not cause any trouble before. Then, weights are updated according to the results of the mth classifier. The procedure is repeated.

The connection with decision rule induction follows from the observation that a single decision rule can be regarded as a simple classifier and that a decision rules induction is nothing else than construction of a family of such simple classifiers. The algorithm used here has been proposed in [3] and is based on the so called *forward stagewise additive modeling* [8], a general scheme of algorithm that creates an ensemble.

Assume that $F(\mathbf{x})$ is a classification function, which assigns to each object a real value. If $F(\mathbf{x})$ is positive, the object is classified to the "positive" class, if $F(\mathbf{x})$ is negative – to the "negative" class, respectively. Assuming $D = \{-1, 1\}$ ("negative" class is denoted by -1, "positive" by 1), object x is classified correctly if $d(x)F(\mathbf{x}) > 0$. We can model this by introducing the so called *loss function* $L(d(x), F(\mathbf{x}))$ which is a penalty for a wrong classification of the object x. The simplest loss function is so called *zero-one loss*:

$$L_{0-1}(d(x), F(\mathbf{x})) = \begin{cases} 1 & \text{if } d(x)F(\mathbf{x}) \leqslant 0 \\ 0 & \text{if } d(x)F(\mathbf{x}) > 0 \end{cases} \tag{25}$$

However, this function is neither continuous nor differentiable. Those properties are desired in induction algorithm, where each step is done using the values of gradient vector. Therefore the *exponential loss function* is used:

$$L_{exp}(d(x), F(\mathbf{x})) = \exp(-d(x)F(\mathbf{x})) \tag{26}$$

The objective of the induction algorithm is now to minimize the loss function on the whole training set:

$$\min L = \sum_{i=1}^{\ell} L_{exp}(d(x_i), F(\mathbf{x_i})) \tag{27}$$

This is done in a greedy way. We start with $F(\mathbf{x}) \equiv 0$ and induce a rule $R_1(\mathbf{x})$ which minimizes the total loss (27). Then we update $F(\mathbf{x}) := F(\mathbf{x}) + R_1(\mathbf{x})$. We proceed in the following way until we obtain some fixed number of rules M. Then, the classification function has the form:

$$F(\mathbf{x}) = \sum_{i=1}^{M} R_i(\mathbf{x}) \tag{28}$$

It can be shown that such strategy of learning base classifiers is equivalent to above-mentioned boosting technique [8].

2.7.4 Measure of the Accuracy

As soon as the model is built, it is important to check, how reliable the model is. In case of binary classification, which is the case of our study there is only one general measure of the classifier's accuracy. Assume each object $(\mathbf{x}, d(\mathbf{x}))$ (where by $\mathbf{x}$ we mean values of condition attributes and by $d(\mathbf{x})$ – value of the decision attribute) is drawn independently from a probability distribution $P(\mathbf{x}, d(\mathbf{x}))$. Then the measure of the accuracy of the classifier $c(\mathbf{x})$ is the probability of error also called the *classifier's risk*:

$$Acc(c(\mathbf{x})) = \Pr(c(\mathbf{x}) \neq d(\mathbf{x})) \tag{29}$$

where the probability is taken according to the distribution P. Since we do not know P, we cannot directly calculate the accuracy. We estimate it from the data instead. It is done during the *testing phase*. It would not be reasonable to estimate the probability of error with the same data it was build on, because knowing the values of the decision attribute in training set, one can build a model as accurate as we want to. Therefore one divides the set of data into two parts — training and testing set. One builds model on training set and tests it (estimates the accuracy measure) on the testing set.

The method of accuracy estimation used in most cases is simply the frequency of errors in the testing set:

$$\hat{Acc} = \frac{\text{number of errors}}{|T|} \tag{30}$$

where T is a testing set and by error we mean the situation, when the classifier assigns object to wrong class.

We use a standard way of choosing a testing set — *k-fold cross validation*. The idea is to divide the whole set of data into k subsets of (almost) the same size. Then make one of those subsets a testing set and the rest is a training set. We repeat this process k times on each subset and take the average accuracy as a reliable estimation of real prediction error. In case when k equals to number of objects, this procedure is called *leave-one-out*.

In the study we use $k = 10$. Moreover, we repeat 10-fold cross validation 10 times. This is due to the fact that the accuracy measure obtained from a single cross-validation procedure is a random variable due to the random splits of dataset into training and testing parts. If one repeats the procedure many times, the variation of the measure decreases and the measure stabilizes.

3 Results

We present partial results for China and complete results for USA, using both the weather and crops data sets.

All the results were generated using programs written by the Author in Java language. Most of used algorithms come from the open-source Java package for data mining, WEKA [5]. For more information, see http://www.cs.waikato.ac.nz/ml/weka/

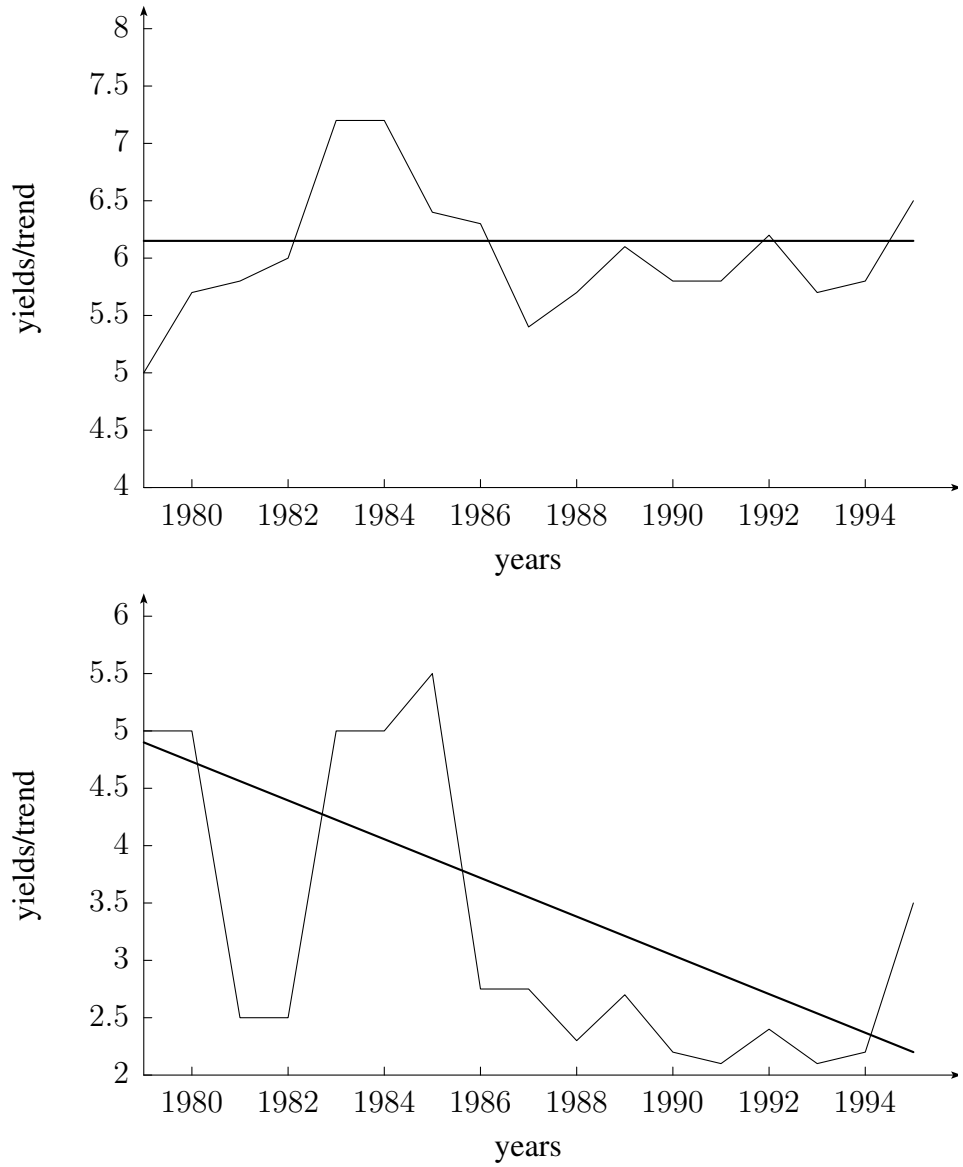


Figure 3: Examples of yields time series in China violating assumption of non-weather factors identification

3.1 Results for China

3.1.1 Data

We were provided with the following data sets for China:

- **Weather data.** A data set containing information about weather from 1900-1995 in the whole globe. Earth was divided into 30' $\times$ 30' grids (so there are 720 grids horizontally and 360 vertically). In each grid, for every month in each year for the above written time period three parameters were given:
  - average monthly temperature
  - monthly temperature difference

  - monthly precipitation

  Additionally, for each grid the area of cultivated land were given (in percents)

- **Yield data.** The yield data set with five kinds of crops. Values of yields were given for 30 provinces in China (one value for each province for each crop), for time period 1979-2000 (one value in each year one value).

- **Administrative map.** Data set with mapping from longitude and latitude grids into provinces, used during the upscaling phase.

3.1.2 Summary

After analyzing the results of trend identification, we decided to not do analysis of impact of climate on yields for China. There are two reasons justifying the decision:

- **Invalid assumptions for trend identification.** The assumption we used to justify the identification of non-weather factors by detrending data was that impact of weather corresponds to short-term variations, as opposite to long-term impact of non-weather factors. However, this assumption was violated in the case of China. One can see two examples of yield time series on Figure 3 and linear trends fitted to data. There are not systematic temporal changes in the amount of yields and no trend in data is visible. Moreover, the time series are too short to obtain any reliable estimation for trend. This is caused by a very strong impact of non-climate factors, which undergo sudden temporal changes and make the separation of weather factors not possible.

- **Small amount of data.** For each province there are only 16 years, in which we were provided with both weather and crop data. Such time series are too short to draw any conclusion about impact of weather variability.

3.2 Results for USA

3.2.1 Data

We were provided with the following data sets for United States:

- **Weather data.** A data set containing information about weather stations in USA and the measurements done by the stations. Observations were given for a period of time about 1930-2004 (there were observations from some stations for pre 1920). For each year, for each month four parameters were given:

  - average monthly temperature
  - minimal monthly temperature
  - maximal monthly temperature
  - monthly precipitation

Some stations provide only precipitation measurements and some provide only temperature. Moreover, there are some "missing" values, i.e. in some months are no values available. Stations usually have more than one thermometer and may perform more than one measurement at the same time. All of those problems were solved during the preprocessing phase, by averaging values over thermometers for each station and later upscaling weather observations from station level into grid level. All observations with missing values were removed.

- **Station data.** A data set containing information about the positions of each station. This information was needed during the upscaling phase.
- **Yield data.** A data set with an amount of yield for different kinds of crops. Values of yields were given for each county usually for the time period about 1930-2004 (but sometimes much shorter periods). We concentrate our analysis on two crops:
  - maize — yields were considered in the following states: Iowa, Illinois, Indiana
  - winter wheat — yields were considered in the following states: North Carolina, South Carolina, Virginia

  We chose one crop to be a winter crop and another to be a summer crop. We concentrate only on three states, close to each other to preserve the same weather conditions.

Finally, after preprocessing phase there were 9630 observations for maize and 3390 observations for wheat.

3.2.2 Identifying the Trend

The first part of the analysis (after preprocessing — cleaning data and upscaling) was related to removing impact of non-weather factor by trend identification. Both methods of trend analysis — linear and polynomial — were performed. For each county and for each crop linear regression was performed independently and different coefficients were calculated.

However, there was a change in the way the linear regression was done. This is due to the fact that about 1950 there was a significant change in USA in agricultural development. The pace of improvement increased because of the change in fertilization, mechanization and generally growing efficiency in this area. Thus, the slope of linear trend has changed. That is why using two different regression lines instead of a global one gave a better accuracy and fit to data.

Finally we selected a piecewise linear trend. The node (brake point) was chosen to be located in 1950 and two linear regressions were applied independently — one for the period before 1950, another from 1950 till present. In case of polynomial trend, such partition was unnecessary, since polynomial curve had enough degrees of freedom to adapt to this change. Both kinds of trends were shown in Figure 4.

Having the signal $y(t)$, value of the trend $\hat{t}(t)$, we identified weather impact $c(t)$ as a percentage variation around the trend:

$$c(t) = \frac{y(t) - \hat{y}(t)}{\hat{y}(t)} \tag{31}$$

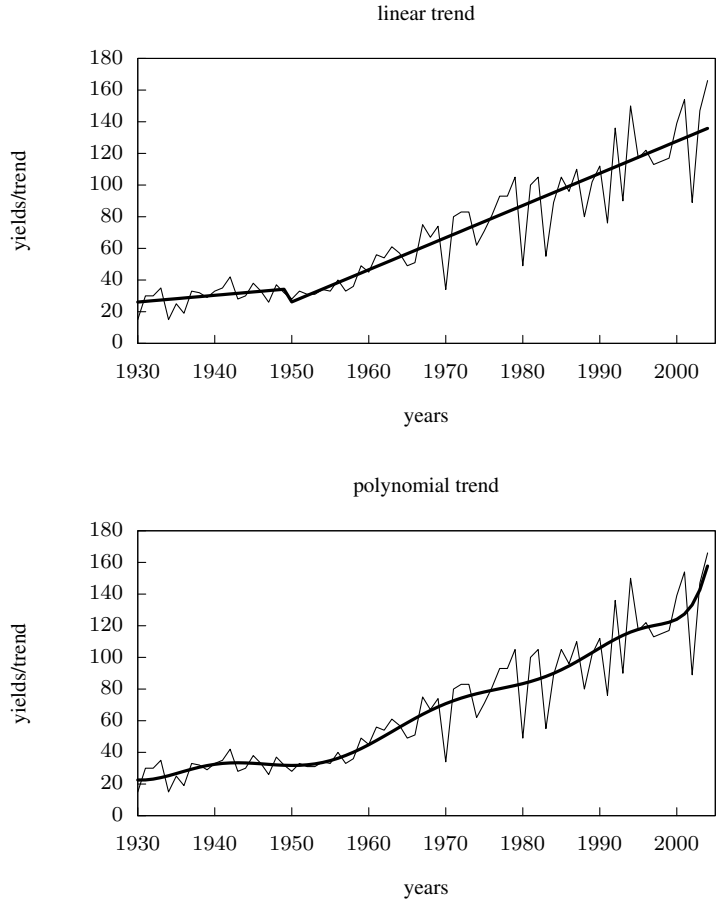


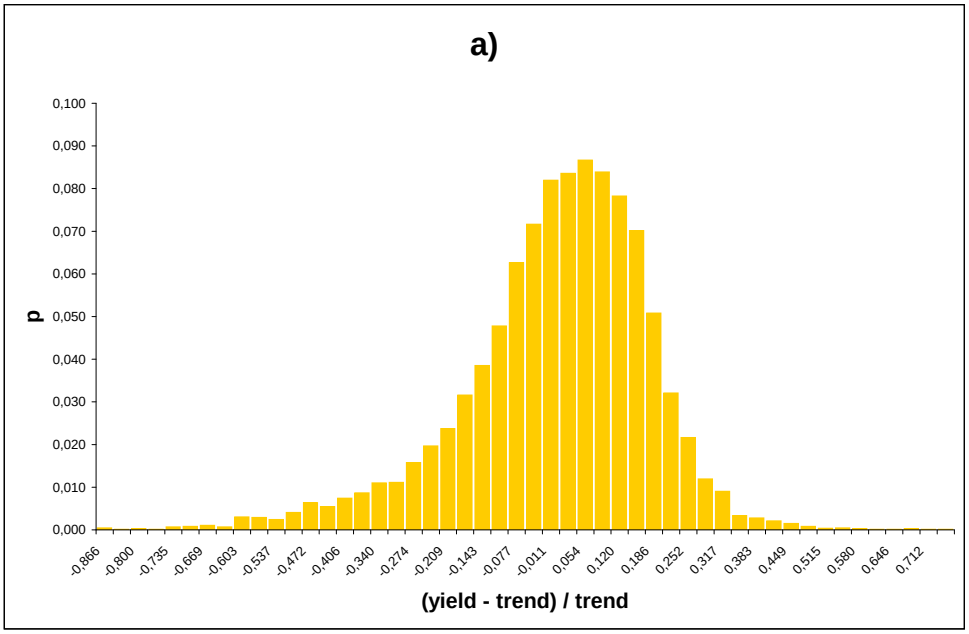
Figure 4: Typical yield time series and trends. In 1950 there is a change in trend for linear case — small "jump" in this year is only the effect of interpolation

The interpretation is as follows: if $c(t) = 0$ then impact of weather is neither positive nor negative for yields. Positive value of $c(t)$ means positive impact of weather, precisely: the ratio of yields to normal (average) value (e.g. $c(t) = 0.2$ is 20% growth of yields caused by the weather). If $c(t)$ was negative, the interpretation is analogous; particularly, if $c(t) = -1$, there were no yields at all.

In Figures 5 and 6 one can see the frequency distribution of values of climate impact $c(t)$ for both types of trend and both crops. Most of values are around average value — 0 (what was expected). Outliers are very rare and there are no impacts stronger than 85%.

Linear regression coefficients have also such interpretation:

- the slope may be regarded as a pace of development in analyzed agricultural area (county), related to improvement in fertilization and mechanization
- the intercept is a constant value and may be interpreted as a constant condition of the considered agricultural area, related to e.g. soil quality



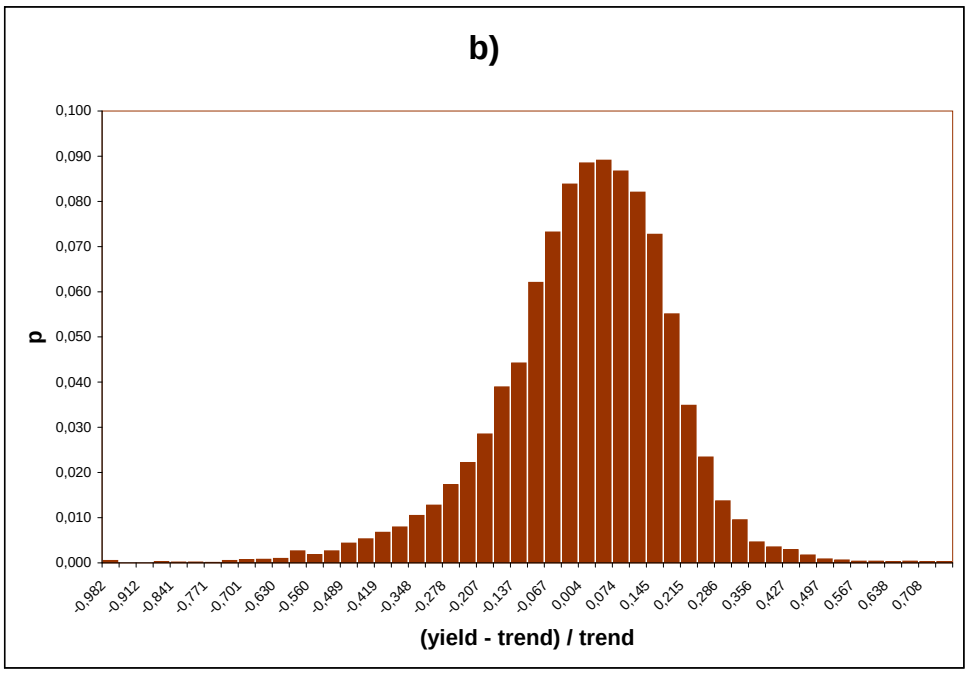


Figure 5: Frequency distribution for (a) maize (b) wheat using piecewise linear trend

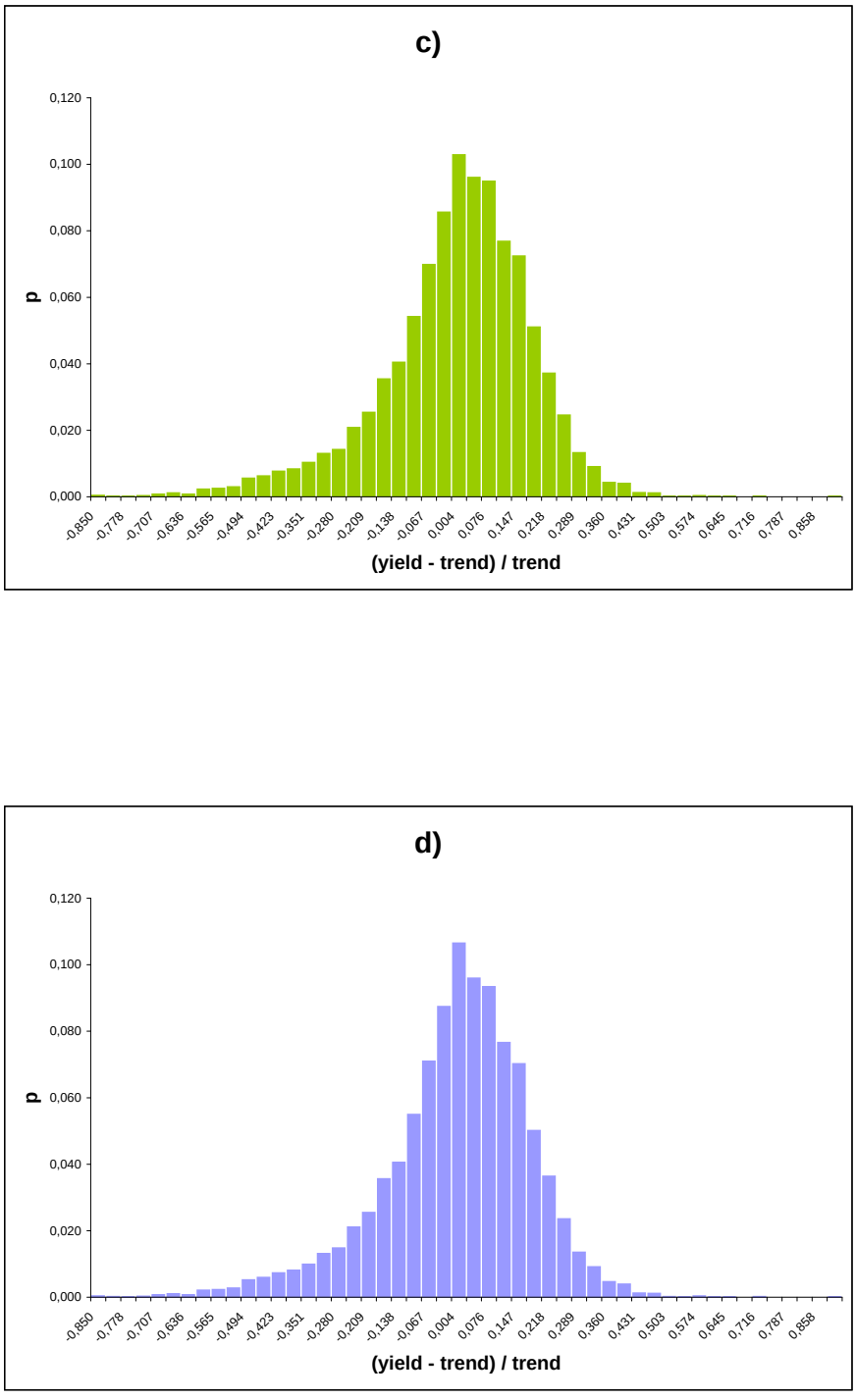


Figure 6: Frequency distribution for (c) maize (d) wheat using polynomial trend

3.2.3 Discretization of Decision Attribute

To perform a classification, a decision attribute must have a finite number of possible values. Since the yields data are continuous, we had to discretize the decision attribute into some intervals. We decided to use only two classes: "low" and "high", since they are easy to interpret, and decision rules are simple and understandable. The most obvious division is to treat negative yield values (under the trend) as low, and positive (over trend) as high. However, there should be a more clear border between classes, especially that the non-weather factor and trend identification were not exact (linear or polynomial trend may only approximate real values of non-weather impact). Upon the advice of the LUC Program researchers we removed values close to the trend (close to 0) to obtain a gap between classes. Since there is no good reason to choose a certain threshold value, below which observations will observations will be removed, we decided to perform the analysis for different threshold values, ranging from 0 (no observations removed) to 0.1. The results for each threshold will be presented later.

3.2.4 Preparing and Assessing the Attributes

linear		**poly-5**		**poly-15**	
monthly					
attr.	IG	attr.	IG	attr.	IG
p(jul)	0.176	p(jul)	0.117	p(jul)	0.137
$t^{\max}$(aug)	0.123	t^{avg}(aug)	0.087	$t^{\max}$(aug)	0.096
$t^{\max}$(jul)	0.119	$t^{\max}$(aug)	0.082	t^{avg}(aug)	0.094
t^{avg}(aug)	0.114	$t^{\max}$(jul)	0.058	$t^{\max}$(jul)	0.076
t^{avg}(jul)	0.095	$t^{\min}$(aug)	0.057	t^{avg}(jul)	0.066
$t^{\min}$(aug)	0.067	t^{avg}(jul)	0.056	$t^{\min}$(aug)	0.062
$t^{\max}$(jun)	0.046	$t^{\min}$(nov)	0.026	p(jun)	0.026
p(jun)	0.418	p(jun)	0.024	$t^{\min}$(jul)	0.025
$t^{\min}$(jul)	0.033	$t^{\min}$(jul)	0.024	$t^{\min}$(nov)	0.024
$t^{\max}$(sep)	0.029	t^{avg}(mar)	0.019	$t^{\max}$(jun)	0.024
seasonal					
attr.	IG	attr.	IG	attr.	IG
p(summer)	0.147	p(summer)	0.092	p(summer)	0.104
$t^{\max}$(summer)	0.104	$t^{\max}$(summer)	0.065	$t^{\max}$(summer)	0.078
t^{avg}(summer)	0.087	t^{avg}(summer)	0.057	t^{avg}(summer)	0.068
p(spring)	0.039	$t^{\min}$(summer)	0.029	$t^{\min}$(summer)	0.034
$t^{\min}$(summer)	0.037	p(autumn)	0.016	p(spring)	0.018

Table 2: Maize. Information gain for top 10 monthly attributes and top 5 seasonal attribute, separately for three types of trend (linear, 5-degree polynomial and 15-degree polynomial.

In order to predict the yields, weather attributes must be selected. The selection vary between different kinds of crops:

linear		**poly-5**		**poly-15**	
monthly					
attr.	IG	attr.	IG	attr.	IG
p(apr)	0.042	p(apr)	0.040	p(apr)	0.040
$t^{\min}$(oct)	0.034	$t^{\min}$(oct)	0.033	$t^{\min}$(oct)	0.033
t^{avg}(oct)	0.033	$t^{\max}$(sep)	0.025	t^{avg}(oct)	0.025
t^{avg}(dec)	0.026	$t^{\min}$(dec)	0.024	$t^{\min}$(dec)	0.025
$t^{\min}$(may)	0.025	t^{avg}(oct)	0.024	$t^{\max}$(sep)	0.024
$t^{\max}$(dec)	0.023	t^{avg}(dec)	0.023	t^{avg}(dec)	0.024
$t^{\max}$(sep)	0.021	$t^{\min}$(may)	0.022	$t^{\min}$(may)	0.021
p(nov)	0.019	t^{avg}(may)	0.020	t^{avg}(may)	0.019
p(oct)	0.019	$t^{\max}$(dec)	0.019	$t^{\max}$(dec)	0.019
$t^{\max}$(jul)	0.017	p(nov)	0.019	p(nov)	0.017
seasonal					
attr.	IG	attr.	IG	attr.	IG
p(spring)	0.029	p(spring)	0.026	p(spring)	0.025
$t^{\min}$(winter)	0.019	$t^{\min}$(winter)	0.011	$t^{\min}$(winter)	0.011
t^{avg}(winter)	0.017	t^{avg}(winter)	0.010	t^{avg}(winter)	0.010
t^{avg}(summer)	0.012	$t^{\max}$(winter)	0.007	$t^{\max}$(winter)	0.007
p(autumn)	0.010	p(autumn)	0.005	p(autumn)	0.005

Table 3: Wheat. Information gain for top 10 monthly attributes and top 5 seasonal attribute, separately for three types of trend (linear, 5-degree polynomial and 15-degree polynomial.

- for maize we use weather data for 12 months, beginning with December of the previous (comparing to yields) year.
- for winter wheat the starting month of data is September in the previous year till August in yields year

We also transform attributes by merging monthly weather parameters into seasonal parameter – for each parameter instead of 12 monthly values we get 4 seasonal values. Values were merged in the following way: December, January, February into Winter, March, April, May into Spring, June, July, August into Summer and September, October, November into Autumn. Since we wanted to take into account both resolutions (monthly and seasonal), we end up with two different sets of attributes (two crops, two trend, two sets of attributes) and in fact 12 different datasets (two crops $\times$ three kind of trend $\times$ two sets of attributes).

Next step of analysis was to assess the importance of the attributes according to the information gain (IG) measure. We fixed some threshold value (value 0.075 was chosen). For each type of crop (maize, wheat), for each set of attributes (monthly, seasonal) and for each type of trend (linear, polynomial with degree 5, polynomial with degree 15) IG was calculated for every attribute using all the observations (without removing those below the threshold). The results are shown in tables 2 and 3. The higher IG is, the more information the attribute brings.

threshold	corn			wheat		
	linear	**poly-5**	**poly-15**	**linear**	**poly-5**	**poly-15**
0	100%	100%	100%	100%	100%	100%
0.01	95%	95%	95%	95%	95%	96%
0.02	90%	91%	91%	90%	91%	91%
0.03	84%	86%	86%	85%	86%	86%
0.04	79%	82%	82%	80%	81%	81%
0.05	74%	78%	78%	75%	77%	77%
0.06	69%	73%	73%	70%	72%	72%
0.07	64%	68%	68%	65%	68%	68%
0.08	59%	63%	63%	61%	63%	62%
0.09	54%	59%	59%	56%	58%	58%
0.1	50%	55%	55%	51%	53%	54%

Table 4: Percent of observations for chosen thresholds and trend

Analyzing of the results, we draw the following conclusions:

- Information gain for maize is much higher than for wheat, which means that the weather attributes are less informative for wheat. One reason for this phenomenon is the average length of time series for wheat which is much shorter (due to smaller number of historical observations) than for maize. On the other hand, wheat yield may depend less on the weather conditions.
- Information gain is highest in case of the linear trend for both crops. It mean that a simple trend works best with our data. Probably polynomial trends have too many degrees of freedom, therefore they catch not only the trend but also a part of the weather-caused variations.
- Aggregation of monthly weather conditions into seasons decreased the information gain. Thus, aggregation leads to the loss of information.
- In case of maize, the most important attributes are those related to Summer months: June, July and August. The analysis shows that the crop growth strongly depend on the amount of precipitation in July.
- In case of wheat, precipitation in April has the greatest influence. Then the temperatures in late Autumn and early Winter follows.

3.2.5 Classification Results

The final stage was to evaluate the performance of four different learning algorithms:

- Support Vector Machines (SVM) – with default complexity parameter $C = 1$ and linear kernel function.
- k nearest neighbours – with $k = 1$ (one nearest neighbor).

- classical decision rules induction – PART algorithm based on C4.5 tree induction. All the parameters were chosen to be the default (in WEKA software).
- Ensemble of decision rules – rule induction based on boosting strategy, with the default parameters (50 rules, specificity parameter $\alpha = 0.25$).

Each of the algorithm has been evaluated on the 132 datasets:

- for each crop (maize, wheat)
- for each kind of trend (linear, 5-degree polynomial, 15-degree polynomial)
- for each set of attributes (monthly, seasonal)
- for each threshold value (between 0 and 0.1 with 0.01 step) – see Section 3.2.3.

Each evaluation was based on repeating 10 times 10-fold cross validation to obtain low variation of the results. Therefore for each evaluation, the classifier was learned 100 times. Summarizing, in our test we performed 52800 learning procedures, each on the dataset with thousands of objects. The computations lasted 4 days on two-processor machine with $4GB$ RAM.

In Table 4 the percentage of observations taken into account for each dataset is shown. For threshold 0 none observation is removed (100%), while for the highest considered threshold (0.1) about 50% of observations is removed from the dataset. The results of the study (prediction accuracies of each classifier) are summarized in Tables 5-8.

maize						
threshold	**monthly**			**seasonal**		
	linear	**poly-5**	**poly-15**	**linear**	**poly-5**	**poly-15**
0	71.81	70.31	71.38	68.41	67.38	68.05
0.01	73.08	71.19	72.24	69.50	68.12	68.90
0.02	74.12	72.12	72.60	70.24	68.77	69.49
0.03	75.20	73.22	73.65	71.03	69.76	70.33
0.04	76.15	73.92	74.50	71.90	70.73	70.98
0.05	77.01	74.48	75.20	72.73	71.36	71.75
0.06	77.84	75.04	75.76	73.51	71.56	72.05
0.07	78.58	75.77	76.27	73.97	72.20	72.7
0.08	79.57	76.55	76.89	74.36	72.71	73.21
0.09	80.38	77.01	77.62	75.40	73.32	73.53
0.1	80.83	77.75	78.16	76.24	73.96	73.74
wheat						
threshold	**monthly**			**seasonal**		
	linear	**poly-5**	**poly-15**	**linear**	**poly-5**	**poly-15**
0	64.40	65.34	64.91	55.58	55.35	55.36
0.01	64.66	66.31	65.92	55.48	56.45	56.22
0.02	65.19	66.71	66.79	56.17	56.85	55.84
0.03	66.78	67.23	67.57	56.24	56.97	56.45
0.04	67.73	68.35	68.43	56.63	56.82	57.28
0.05	68.72	69.41	68.92	57.29	57.45	57.56
0.06	69.90	70.75	69.79	57.01	57.34	56.70
0.07	71.32	71.72	70.97	57.52	57.51	56.79
0.08	71.96	72.30	71.81	57.98	58.09	57.64
0.09	73.17	73.75	73.48	58.81	58.76	58.01
0.1	73.61	73.57	73.24	58.69	59.42	58.26

Table 5: Prediction accuracy (in %) for SVM

As expected, for each of the classifiers, the prediction accuracy increases as the threshold increases. This is due to the fact, that the classes become more and more separated

maize						
threshold	**monthly**			**seasonal**		
	linear	**poly-5**	**poly-15**	**linear**	**poly-5**	**poly-15**
0.	81.10	82.59	82.31	70.66	70.64	70.96
0.01	83.29	84.70	84.03	72.12	72.30	72.36
0.02	85.82	86.61	86.07	73.85	73.77	73.98
0.03	87.23	88.34	87.83	74.77	75.29	75.42
0.04	88.72	90.36	89.78	76.53	76.37	76.21
0.05	89.68	91.67	91.00	77.38	77.41	77.43
0.06	90.86	92.40	92.14	78.39	78.27	78.38
0.07	91.88	93.49	93.32	79.73	78.91	78.99
0.08	92.99	94.23	94.36	80.06	80.26	80.02
0.09	93.86	95.28	94.92	81.33	80.99	80.63
0.1	94.16	95.66	95.33	82.25	81.75	81.40
wheat						
threshold	**monthly**			**seasonal**		
	linear	**poly-5**	**poly-15**	**linear**	**poly-5**	**poly-15**
0	71.74	69.07	69.32	63.19	62.59	62.36
0.01	72.57	70.79	71.23	63.80	63.80	63.69
0.02	74.36	72.52	72.85	64.31	64.69	64.91
0.03	75.76	73.59	73.01	65.05	64.56	64.66
0.04	77.36	74.71	74.52	66.37	65.19	64.97
0.05	78.96	75.88	75.44	67.41	66.16	66.09
0.06	80.55	77.78	76.53	68.15	67.64	66.78
0.07	81.90	78.85	78.08	69.28	67.57	67.23
0.08	82.14	79.56	79.36	69.97	68.26	68.61
0.09	83.91	80.89	80.51	70.93	69.08	69.39
0.1	83.43	81.06	80.05	70.81	69.45	69.07

Table 6: Prediction accuracy (in %) for kNN.

(we remove the observations around the trend). Investigation of the results shows that if a given classifier is better than other classifier on a single threshold, it is better for each threshold. Therefore it is enough to compare the classifiers on a single threshold value. Here, for better view, we compare the classifiers on two extreme threshold values (0 and 0.1).

The best results were obtained using k nearest neighbours algorithm. It outperforms all the other classifiers, reaching the accuracy for maize of more than 95% for threshold 0.1 and 82% for threshold 0 (i.e. for all the observations), and for wheat 83% and 72%, respectively. This shows that the yield prediction using weather attributes gives very accurate results, or in other words, the weather conditions explains the yields in a strong degree. Especially for high threshold, when we predict only the extreme observations (yield values far away from trend), the predictions are almost certain. We stress that those results were obtained without any additional domain knowledge, only with use of statistical tools for trend identification and classification.

The algorithm which is second with respect to the accuracy, is the ensemble of decision rules, based on boosting strategy of rule induction. It outperforms the classical rule induction algorithm (PART) in each case, reaching the accuracy for maize of 89% for threshold 0.1 (while PART has only 83%), 77% for threshold 0 (while PART – 73%) and for wheat 78% and 67%, respectively (in case of PART – 70% and 62%, respectively). Ensemble, although working slightly worse than kNN, has advantage of being much easier to interpret (since it is a family of decision rules).

Finally, SVM works similar to PART. It achieves worse results for maize (81% for threshold 0.1 and 72% for threshold 0) and better results for wheat (74% and 65%, respectively). This might be caused by the nonlinear nature of the weather impact on crop

maize						
threshold	**monthly**			**seasonal**		
	linear	**poly-5**	**poly-15**	**linear**	**poly-5**	**poly-15**
0	72.60	72.88	72.94	67.65	66.19	67.19
0.01	74.25	74.26	74.42	68.68	67.11	67.92
0.02	75.86	76.07	75.72	69.24	67.66	68.53
0.03	76.72	77.39	77.61	70.52	68.67	69.55
0.04	78.52	78.92	79.02	71.99	69.62	70.06
0.05	79.59	79.86	79.87	72.06	70.28	71.15
0.06	80.83	80.91	80.49	73.40	70.92	71.84
0.07	81.48	81.62	82.12	73.97	71.87	72.66
0.08	82.45	82.51	82.71	74.43	72.51	73.22
0.09	83.12	83.19	83.23	75.56	72.92	73.08
0.1	83.10	83.72	83.44	76.12	73.07	73.89
wheat						
threshold	**monthly**			**seasonal**		
	linear	**poly-5**	**poly-15**	**linear**	**poly-5**	**poly-15**
0	61.47	61.88	61.40	55.71	55.02	54.29
0.01	61.98	62.40	62.64	55.66	55.66	54.76
0.02	62.64	62.85	63.27	55.99	55.31	55.45
0.03	64.36	63.81	63.31	55.72	55.61	54.60
0.04	65.55	64.76	64.65	55.07	55.86	54.90
0.05	65.99	65.56	65.67	57.99	55.46	55.06
0.06	67.10	66.71	66.40	58.38	56.91	56.85
0.07	68.94	66.90	66.64	58.88	57.00	57.08
0.08	69.23	68.68	68.23	60.04	57.06	57.16
0.09	69.63	69.29	68.61	60.39	57.30	57.11
0.1	70.40	68.77	69.25	59.83	57.48	57.09

Table 7: Prediction accuracy (in %) for PART.

yields. SVM with linear kernel is a linear function of the condition attribute values, while both decision rules and kNN will approximate even strongly nonlinear functions. Probably the results of the SVM might be improved using different kernel functions (especially so called Radial Basis Function kernel for which the function shape resembles kNN function).

Another thing apparent in Tables 5-8 is the dominance of the linear trend over the polynomial trends. In almost all the cases, linear trend identification works best, then the 15-degree polynomial and the 5-degree polynomial at the end. This shows that a simple trend identification procedure should be used to remove the long-term factors. Those more complicated procedures (polynomial) have more degrees of freedom, therefore they overfit to the data, hiding the variations caused by climate. The only exception, when polynomial trend works better than a simple linear one, is the case of kNN and maize.

Finally, the study shows that using seasonal attributes leads to decrease of the accuracy. It was already apparent when the information gains were calculated and IG values for seasonal attributes were much lower than those for monthly attributes. The reason of this deterioration is the loss of information during the aggregation of attributes. Probably three-months time period are too coarse for considering weather conditions.

3.2.6 Distribution of Errors

Each classifier was characterized by one single value – estimated prediction accuracy by repeated 10-fold cross validation. However, it might be of interest to know what is the distribution of prediction accuracy for different years. The distribution was calculated as follows. For each classifier, we chose a certain year. Then we trained the classifier on the

maize						
threshold	**monthly**			**seasonal**		
	linear	**poly-5**	**poly-15**	**linear**	**poly-5**	**poly-15**
0	75.76	76.65	76.88	69.29	67.52	68.57
0.01	78.14	78.08	78.34	70.48	69.02	69.76
0.02	79.95	79.62	79.71	71.69	69.91	70.80
0.03	81.25	81.46	81.41	73.11	71.58	72.44
0.04	83.06	83.34	83.04	74.33	72.66	73.48
0.05	84.04	84.27	84.42	75.49	73.68	74.84
0.06	85.45	85.41	85.55	76.83	74.05	75.17
0.07	86.44	86.11	86.59	77.54	75.45	76.22
0.08	87.43	87.19	87.53	78.48	75.99	76.95
0.09	88.49	88.17	88.15	79.63	77.06	77.51
0.1	89.14	89.11	88.95	80.83	77.45	78.32
wheat						
threshold	**monthly**			**seasonal**		
	linear	**poly-5**	**poly-15**	**linear**	**poly-5**	**poly-15**
0	66.61	66.08	65.49	57.32	57.32	56.43
0.01	67.89	66.96	66.71	57.54	57.80	57.32
0.02	68.32	68.03	68.05	59.18	58.19	57.29
0.03	70.07	68.64	68.90	59.08	58.25	58.30
0.04	71.48	70.91	70.24	59.96	58.84	58.77
0.05	72.59	71.46	70.99	60.50	59.77	59.16
0.06	73.92	72.59	71.59	61.30	60.48	60.17
0.07	75.55	73.77	72.97	62.13	61.43	61.36
0.08	75.94	75.06	74.68	62.83	62.09	61.92
0.09	76.27	75.41	75.08	64.22	63.01	62.43
0.1	77.69	76.01	75.06	65.34	63.88	63.11

Table 8: Prediction accuracy (in %) for ensemble of decision rules.

dataset excluding the observation from the chosen year. Then we tested the classifier on the data from the chosen year. We restricted the analysis only to the case of linear trend, monthly attributes and chosen threshold 0.075. The accuracy distributions are shown on Figure 7.

The distribution is varying strongly, without any visible trend and regularity. However, one can notice, that there are several years for which prediction accuracy is very high and some years for which it is unusually low and this phenomenon does not depend on the type of classifier (all of them show increase/decrease of the accuracy in the same year). For a given crop, all the charts, although having different average height and oscillation amplitudes, have a similar shape. The explanation for this fact could be the following:

- Years with a high prediction accuracy might be "typical", i.e. years for which no extraordinary phenomena appeared. Then, the weather factors explain most of the crop growth.

- Years with low prediction accuracy might be "untypical", i.e. those years for which some phenomena appeared not taken into account in our analysis. The phenomena might have origin in environmental conditions (flood, disaster), but might also have different origin (crop disease, economical crisis).

Figure 7: Distributions of accuracy for each year. The left column shows results for maize, the right column – for wheat.

4 Summary

4.1 Conclusions

In this section we will summarize the most important conclusions drawn in the previous Sections.

- Piecewise linear trend identification seems to be generally better than polynomial ones. This is due to the low number of degrees of freedom in case of linear trend which ensures catching only long-term changes in the trend.

- Aggregation of weather conditions into seasons deteriorated the results, probably due to too coarse time resolution of weather factors.

- Information gain values differs strongly between attributes, which means that there is a small number of attributes with great influence on crop yields and many attributes with hardly any influence on yields. Therefore, in some months weather conditions seems to be crucial for further yields, while in the other months the conditions does not affect the crop much.

- Estimating non-weather factors by using the trend was successful, since accuracy obtained by the best model is about 95% which is very high (misclassification occurs only once per 20 observations). Even without removing any observations around the trend, we still obtain accuracy 82% (misclassification one per 5 observations), which is much better than a random guessing.

The results clearly show the advantages of the presented methodology for studying impacts of weather on yields. Without any domain knowledge and without any assumptions about the analyzed processes the rules model achieved high accuracy and consistency with expert knowledge. However, there are several possibilities to improve the presented results and obtain new insights into the analyzed problem. Namely:

- A more precise preprocessing should be performed on the data sets being used. At first, incorrect data should be detected and removed. Moreover, outlier analysis may be done (outliers need different approach since by definition they are unusual; thus no additional rule is induced to cover the outliers, because for any induction algorithm, rules with very small support are usually not preferred).

- A more sophisticated approach to trend identification may be applied to the weather data to detect and account for a long-term weather changes. Also, a different method of estimating the trend coefficients may be applied: instead of least squares used in this study, minimizing L_1 norm (least absolute values) can be performed which has advantage of being less sensitive to outliers.

- We believe, that the most important phase of the analysis is a separation of non-weather factors, which cannot be done precisely using a simple trend. The high accuracy of the obtained model shows that in general the approach was proper and gave satisfactory results. However, we think further research might be done in either improving the method of detrending data or introducing non-weather attributes to the analysis. This might be done with cooperation of the researchers from LUC Program. Both improvements should lead to more reliable results.

- Finally, we want to stress that after the suggested extensions, it will be much easier to apply this methodology, and the corresponding software to other sets of weather and crop data.

References

[1] Angluin, D. *Computational learning theory: Survey and selected bibliography*. In Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing (May 1992), pp. 351–369.

[2] Blumer, A.; Ehrenfeucht, A.; Haussler, D.; Warmuth, M. K. *Occam's razor* Inf.Proc.Lett. 24, 377-380, 1987. Proceedings of the Twelfth International Conference on Machine Learning. pg. 194–202, 1995.

[3] Błaszczyński, J., Dembczyński, K., Kotłowski, W., Słowiński, R., and Szelag, M.: Ensembles of Decision Rules. *Foundations of Computing and Decision Sciences*, **31** (2006) 21–232

[4] Burges, C. J. C., A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, **2(2)** (1998) 121-167

[5] Eibe Frank and Ian H. Witten (2005) *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005

[6] Eibe Frank and Ian H. Witten, *Generating Accurate Rule Sets Without Global Optimization*. In Shavlik, J., ed., Machine Learning: *Proceedings of the Fifteenth International Conference*, Morgan Kaufmann Publishers, 1998, San Francisco, CA

[7] Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. *J. of Computer and System Sciences*, 55 **1** (1997) 119–139

[8] Hastie, T., Tibshirani, R., Friedman, J. H., The Elements of Statistical Learning, Springe (2003)

[9] Mitchell, Tom M., *Machine Learning*, McGraw-Hill Science, 1997.

[10] Müller, K., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B., An Introduction to Kernel-Based Learning Algorithms. IEEE Transactions on Neural Networks, **12** (2001).

[11] Quinlan, J. R., *C4.5: Programs for Machine Learning*, San Mateo 1993, CA: Morgan Kaufman

[12] Vapnik, V.: The Nature of Statistical Learning Theory, New York, Springer-Verlag (1995)