AN INTRODUCTION TO CHEMOMETRICS FOR FOOD SCIENCE

***Jerzy Tyszkiewicz*[1], *Stanisław Tyszkiewicz*[2]**

[1]*Institute of Informatics, Warsaw University, Warszawa;* [2]*Institute of Meat and Fat Research, Warszawa, Poland*

Key words: chemometrics, foods, kinetics, algorithm, machine learning

In this paper we present an introduction to the computer science methods of chemometrics. Chemometrics is understood here very broadly, as the area covering the methods used to analyze vast amounts of numerical data obtained in the course of chemical, physical and sensoric experiments in the area of food science and technology.

INTRODUCTION

The International Chemometrics Society (ICS) offers the following definition of chemometrics: *Chemometrics is the science of relating measurements made on a chemical system or process to the state of the system via application of mathematical or statistical methods. Chemometric research spans a wide area of different methods, which can be applied in chemistry. There are techniques for collecting good data (optimization of experimental parameters, design of experiments, calibration, signal processing) and for getting information from these data* (*statistics, pattern recognition, modeling, structure-property-relationship estimations*).

Bearing in mind that most of the readers of this paper have chemical or biological background, we present chemometrics as it is seen from the computer science point of view, concentrating on the typology of methods used to get information from the data collected in experiments.

WHERE THE DATA COMES FROM

Nowadays food science evolves in the same direction as all the applied experimental sciences: partially or even completely automated experiments yield vast amounts of raw data as the output. The main scientific task is to analyze it and uncover the real processes hidden behind the columns of numbers. One should keep in mind that working with the natural, biologically diverse material, all the data exhibits, apart from the processes to be found, also statistical differences following the normal distribution. Even more, the materials investigated in food science change their properties with time, and the state in which they should be investigated lasts for a short period of time, only.

The properties of food products which are measured are mainly those measurable in the sense of "fundamental sciences": chemistry, physics, biology, physiology, each of which offers its own, specific methodologies, measures, and terminology to be applied.

The situation of the researcher can be quite different: one extreme is that he possesses the complete mathematical model of the process, and the whole task reduces to fitting the data with the theoretical curves, determining the constants and taking care of the measurement errors. On the other extreme one may have data recording many parameters of the experiment and not even a slightest clue what kind of statistical regularities the data exhibit, let alone to know which parameter determines which and how.

FUNDAMENTALS

We deal with *samples* or *objects*, typically denoted by s, which can be thought of as experimental samples in a laboratory, or items on a production line in a factory. *Attributes* or *values* are functions v which assign to every object a number $v(s)$, representing a certain property of s. Weight, length, fat content of a meat unit, age, length, fat content, sex of an animal, fraction of unsaturated fatty acids, volume and weight of an olive oil probe, *etc*. can be the attributes. Note that sex of an animal, even though not a number by itself, can be easily and conveniently represented by a number (say: 0 for male, 1 for female). Most of the natural attributes are nominated values, and are expressed as multiples of an arbitrarily chosen measurement unit.

Throughout the text, most of the time we assume that the problem we face is the following: given is an attribute v, which is either difficult or expensive to evaluate exactly for each object individually. The examples might be: the fat amount in a carcass (expensive to evaluate because it would require a large amount of labor); the optimal "best before" date (impossible to evaluate because this time becomes known precisely when the item does not qualify for sale any longer). Our wish is to find a method to estimate the value

Author's address for correspondence: Jerzy Tyszkiewicz, Institute of Informatics, Warsaw University, ul. Banacha 2, 02-097 Warszawa, Poland; e-mail: jty@mimuw.edu.pl

$v(s)$, as accurately as possible, by the value $f(v_1(s),..., v_n(s))$ of a function f, which can be computed given the values of the attributes $v_1(s),..., v_n(s)$ which in turn can be evaluated easily and cheaply. So we seek a methodology to find such functions f, given the list of easy to evaluate attributes of s.

Summing up, our data are vectors of numbers, of fixed length. In our terminology, each vector is the sequence $\mathbf{v}(s)=(v_1(s),..., v_n(s))$ of the values of certain attributes of the object s. Besides that, we have a number of vectors: $\mathbf{v}(s_1),...,\mathbf{v}(s_N)$, for which the values of the target attribute v are known. These are the reference values.

The progress of the chemometrics is due to the combined impact of two fundamental factors: (i) the progress of the instrumental methods, which enlarges the list of easily evaluable attributes; and (ii) the progress in the methodology of determining approximations of an attribute by functions of other attributes.

The first of them is driven by the advances of the "fundamental sciences" we have already mentioned: chemistry, physics, biology and physiology.

This text deals almost exclusively with the second component above, describing some of the methods offered by mathematics, statistics and computer science for the purpose of analyzing data gathered using the methods of the respective "fundamental sciences".

DISTRIBUTIONS AND STATISTICS

As a result of an experiment we get datasets of various formats. From both scientific and practical standpoint the data should represent important qualitative and quantitative characteristics of the objects which have undergone experimental examination. The quantitative characteristics are generally expressed in the widely accepted units of the SI system.

For the processing purpose, the quantitative data is expressed as floating point numbers of certain precision (number of decimal places). Further processing is done using the same precision. This seems obvious, but has some non-obvious consequences.

One of them is that *numerical errors* may occur. They are caused by the fact that, *e.g.* a product of two very large numbers, which are representable in the given precision, may well be too large to be representable itself. In this case, the result of the multiplication can be quite unpredictable, and depends on the software and hardware used to process the data. A similar phenomenon can happen on the opposite end of the precision scale: a multiple of two numbers which are positive may be calculated as 0 because the real positive result of the multiplication is too close to 0 to be representable in the given precision as a nonzero number. And, of course, even if these extreme errors do not occur, still significant digits of the data can get lost in the processing, if they are located close to the extremes.

These problems can be easily avoided by choosing a reasonable multiple of the basic unit to represent the data in numerical form. The available choices of the multiples range from tera (prefix T), which is 10^{12} to atto (prefix a), which is 10^{-18}; therefore there is always a choice for which the numbers are within a reasonable range.

Besides the errors introduced by the computer processing itself, there are of course measurement errors, unavoidable in experimental sciences. Certainly, the accuracy of the calculated values cannot exceed the accuracy of the input data. In case the real measurement error Δv of the value v is difficult to estimate as an absolute value, it can be estimated in terms of a relative error δv according to the formula:

$$\delta v = \frac{10}{d \cdot 10^{n-1}} = \frac{1}{d} 10^{2-d}$$

where d is the first significant digit of the approximation, and n is the number of the significant digits.

In most of the cases, the final results of the computations depend on many parameters v_i of the initial, raw data, each of which contributes its own error to the final error of the computed value $v=f(v_1,v_2,...)$. If the errors of $v_1, v_2,...$ are $\Delta v_1, \Delta v_2,...$, then the error Δv of the result can be estimated as:

$$\Delta v = \left|\frac{\partial f}{\partial v_1}\right| \Delta v_1 + \left|\frac{\partial f}{\partial v_2}\right| \Delta v_2 + ...$$

The above formula is particularly useful for estimating errors of parameters represented as dimensionless numbers. Such numbers are quantities which describe a certain physical system and which are a pure numbers without any physical units. They are typically defined as products or ratios of quantities which do have units, in such a way that all units cancel. Their importance is based on the fact that two systems, which have the same dimensionless number, are quite similar to each other.

In practice, the estimation of the measurement errors is done experimentally. One performs a large number of identical experiments in identical conditions. The distribution of the measured values of the quantity under investigation often gives sufficient information about the measurement errors. Typically this distribution follows the normal distribution with a certain expected value μ and standard deviation σ. They are called the estimators of that distribution. It is usually the case when the error is a consequence of many independent factors, none of which dominates the others. The well-known formula for the normal distribution is:

$$f(v) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{(v-\mu)^2}{2\sigma^2}\right).$$

For a sufficiently large set of n measurements mean (average, expected value) μ (no matter whether for a normal distribution or any other one) is estimated as the average of the measurements:

$$\mu \cong \bar{v} = \frac{1}{n}\sum_{i=1}^{n} v_i .$$

The expected value $\bar{v}$ itself is also distributed according to the normal distribution with standard deviation $\sigma_{\bar{v}} = \frac{\sigma}{\sqrt{n}}$. The latter value is assumed to be the average error of the average, while the standard deviation is estimated as:

$$\sigma \cong Sv = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(v_i - \bar{v})^2} .$$

Standard deviation is a square root of the variance var(v), whose formula can be seen under the root sign.

Variance has a generalization, which is the covariance. Its formula is as follows:

$$\text{cov}(v,w)=\frac{1}{n-1}\sum_{i-1}^{n}(v_i-\overline{v})(w_i-\overline{w}) .$$

Covariance is important when one wants to use principal component analysis PCA. It is an important statistical procedure, which allows one to reduce the number of attributes in the data and to discover the most significant relations. Let us therefore assume that we have a number of vectors $\boldsymbol{v}(s_1),\ldots,\boldsymbol{v}(s_N)$ representing the data, each of length n.

Here is the principle of PCA.

- In the first step, for each attribute v_i, we replace v_i(s) by the value $v_i(s)-\overline{v}_\mathrm{i}$ in the whole dataset.
- The second step is to create the covariance matrix, whose entries are covariances among all the attributes in the dataset. In column i and row j this matrix has value cov(v_i,v_j).
- The third step is to compute the eigenvalues and the corresponding eigenvectors of the covariance matrix.
- Transform the dataset so that the new coordinates are computed along the eigenvectors. A mathematical principle guarantees that the vectors are pairwise perpendicular and can be used to provide an alternative coordinate system for the dataset.
- The larger the modulus of an eigenvalue, the more significant is the corresponding eigenvector for the differences observed in the data. One can often decide to ignore dimensions corresponding to the eigenvectors for which the eigenvalues have a very small absolute value, since they contribute very little to diversity of the data. This is the method to eliminate, *e.g.*, redundant attributes, which are highly correlated with other ones.

Most major statistical software systems have PCA among their standard tools.

An important element of the initial analysis of the data is the elimination of non-systematic errors. In the case of a parameter distributed according to the normal distribution with standard deviation σ, the data items suspected to carry such an error should be those that have extreme values, outside of the range for that parameter. They should be eliminated from the dataset. If the dataset consists of a large number of items, the elimination of such elements can also be based on a normality test, like the test χ^2. If the number of data items is small, Dixon's Q test can be used. In this test for an extremal element x_n, suspected of carrying a non-systematic error, the quantity Q is computed according to the formula

$$Q=\frac{v_{n-1}-v_n}{R} ,$$

where v_{n-1} is the data item closest to v_n, R is the range in the complete dataset (with all the elements included). The resulting value of Q is compared to the critical values Q_0 computed for various significance levels α and cardinalities n of the dataset. Table 1 summarizes these values for the first few cardinalities and two often used values of α.

In the datasets characterizing real life objects or phenomena not all of the items carrying a non-systematic error manifest themselves with extreme values of certain attributes, and hence the statistical analysis of each of the parameters alone does not allow one to detect such objects. If two or more parameters are correlated, an unusual combination of values of those parameters can indicate that at least one of them carries such an error. A simple example of this situation is the following: if in a mixture of three substances the relative content of each of them is determined in an independent measurement, the sum of the obtained values should give 100%, up to the measurement error. However, if this sum is, let's say, 50%, at least one of the values definitely carries a non--systematic error, even if each of the measured individual contents does not stand out among the other measurements.

In general, analyzing concentrations poses one more difficulty. If the dominating content can occur in concentrations close to 100%, the error cannot have the normal distribution any more, as the limit for all the results of the measurements is 100%. In such situations, it is recommended to transform the results according to the formula

$$y=\arcsin(\sqrt{x}) .$$

On the other extreme of concentrations, they are limited by the *limit of determination L_D*. It is based on the probability α of getting a false positive detection (the substance does not occur, but is detected) and the probability β of a false negative detection (the substance does occur, but is not detected). These values are usually assumed at the level of 0.05.

The impact of the concentration of the detected substance on the *randomized standard deviation RSD* of the results of measurements is given by the Horwitz formula [Horwitz, 1982]:

$$RSD=2^{(1-0.5\log_{10}C)}=2C^{-0.1505} ,$$

where C is the relative concentration. Consequently, for the concentration of 100% RSD is 2%, for 1% RSD is 4%, for 0.1% RSD is 5.6 %, for 1 ppm it is 16% and for 1 ppb it is 45%. The empirical formula of Horwitz has been determined based on interlaboratory experiments described in over 6000 reports. The experiments covered a very diverse spectrum of products, ranging from food, through pharmaceuticals, cosmetics, paints, pesticides, to drinking water. It can be used as a universal method to validate the methods and analytical procedures used in experiments.

WHEN THE THEORY IS KNOWN

The most comfortable situation is encountered when the theory governing an experiment is known in advance and the task reduces to the determination of the involved constants and performing error analysis.

Still the problem is far from being trivial. There are of course simple theoretical models, for which there is no real problem in the data analysis. On the other hand, there are quite difficult ones, like, *e.g.* the heat transportation model,

TABLE 1. Critical values for the Dixon's Q test depending on the cardinality n of the dataset and significance levels α.

Q_0 n	3	4	5	6	7	8	9	10	11	12
α=0.05	.941	.764	.642	.560	.507	.468	.437	.412	.392	.376
α=0.01	.988	.889	.780	.698	.637	.590	.555	.527	.502	.482

which is governed by a system of non-linear partial differential equations. It can be easily met in the food technology, *e.g.* when a food item should be heated to reach a prescribed temperature in all of its volume. For such models, there are serious problems with determining the boundary conditions, taking the anisotropy and/or inhomogenity of the item into account, *etc*. In such circumstances, it may be well the case that it is easier to adapt and use methods which do not assume the knowledge of the theory governing the process, than to apply the theory itself.

If all those difficulties have been overcome, after deriving the equation, it should be solved, and the solutions verified experimentally. While the second task is the more important one, the first can also be automated, at least to a certain degree. There are software tools, called *computer algebra systems*, designed exactly to help in symbolical mathematical calculations, like solving differential equations. Among the major systems of this kind one should name *Maple*, *Mathematica*, and *MuPAD*. All of them have similar capabilities, and the differences typically reduce to the user interface organization. Their functionalities can differ significantly only in the support of some very advanced mathematical theories. Basic calculus, statistics and graphical capabilities of all of them are very similar. In the considered example the equations can be solved exactly, and the solutions can be expressed by closed formulas. We would like to stress out that modern computer algebra systems are not difficult in use at all. For a person who knows what a differential equation is, solving it using *Maple*

Smoke mass differential equation for an open smoke chamber. Smoke of density C0 and production yield mu per time unit t enters the chamber; V is the chamber volume and C(t) the density of smoke in the chamber at time t. Smoke disappears at the rate a.

```
> EQ1:=diff(C(t),t)=(mu/V)*C0-(mu/V)*C(t)-(a/V)*C(t);
```

$$EQ1 := \frac{\partial}{\partial t}C(t) = \frac{\mu\, C0}{V} - \frac{\mu\, C(t)}{V} - \frac{\alpha\, C(t)}{V}$$

The symbolic solution of this equation:

```
> S1:=dsolve({EQ1,C(0)=0});
```

$$S1 := C(t) = \frac{\mu\, C0}{\mu+\alpha} - \frac{e^{\left(-\frac{(\mu+\alpha)t}{V}\right)}\mu\, C0}{\mu+\alpha}$$

Smoke mass differential equation for a closed smoke chamber. The symbols have the same meaning as above.

```
> EQ2:=diff(C(t),t)=(mu/V)*C0-(a/V)*C(t);
```

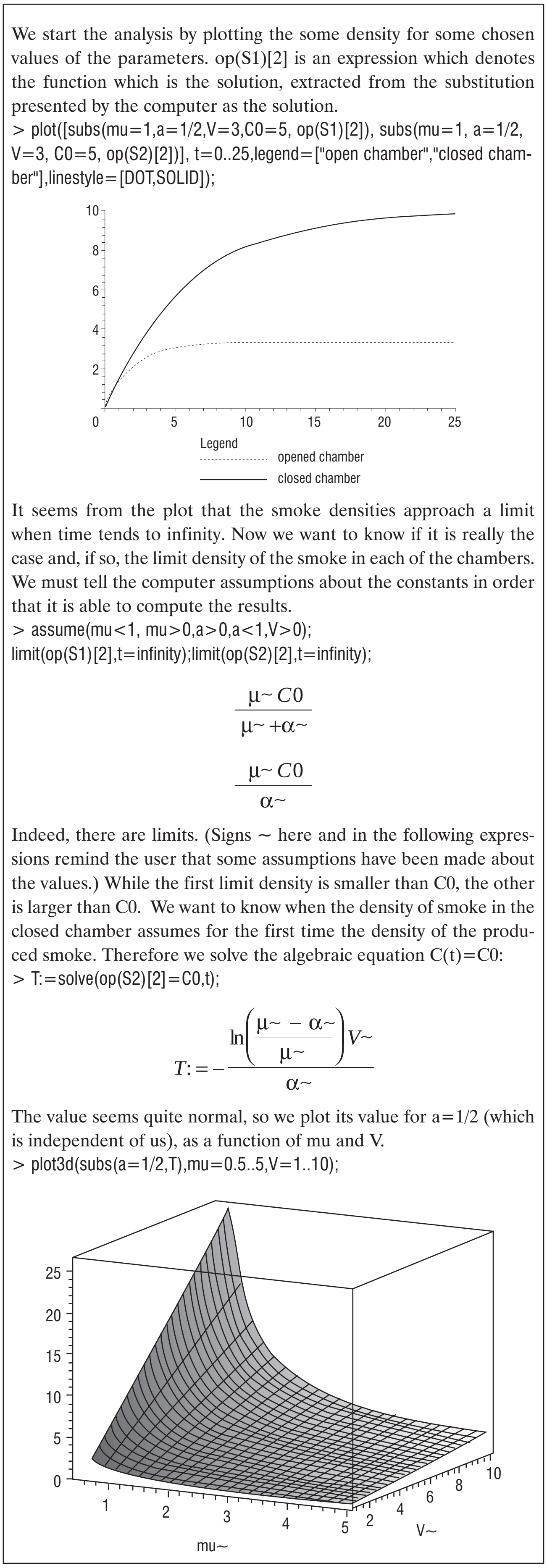
$$EQ2 := \frac{\partial}{\partial t}C(t) = \frac{\mu\, C0}{V} - \frac{\alpha\, C(t)}{V}$$

Again the symbolic solution:

```
> S2:=dsolve({EQ2,C(0)=0});
```

$$S2 := C(t) = \frac{\mu\, C0}{\alpha} - \frac{e^{\left(-\frac{\alpha t}{V}\right)}\mu\, C0}{\alpha}$$

FIGURE 1. Symbolic solutions of differential equations describing smoke density in an open and a closed smoke chamber, using *Maple*.



We start the analysis by plotting the some density for some chosen values of the parameters. op(S1)[2] is an expression which denotes the function which is the solution, extracted from the substitution presented by the computer as the solution.

```
> plot([subs(mu=1,a=1/2,V=3,C0=5, op(S1)[2]), subs(mu=1, a=1/2, V=3, C0=5, op(S2)[2])], t=0..25,legend=["open chamber","closed chamber"],linestyle=[DOT,SOLID]);
```

It seems from the plot that the smoke densities approach a limit when time tends to infinity. Now we want to know if it is really the case and, if so, the limit density of the smoke in each of the chambers. We must tell the computer assumptions about the constants in order that it is able to compute the results.

```
> assume(mu<1, mu>0,a>0,a<1,V>0);
limit(op(S1)[2],t=infinity);limit(op(S2)[2],t=infinity);
```

$$\frac{\mu\text{~}\, C0}{\mu\text{~} + \alpha\text{~}}$$

$$\frac{\mu\text{~}\, C0}{\alpha\text{~}}$$

Indeed, there are limits. (Signs ~ here and in the following expressions remind the user that some assumptions have been made about the values.) While the first limit density is smaller than C0, the other is larger than C0. We want to know when the density of smoke in the closed chamber assumes for the first time the density of the produced smoke. Therefore we solve the algebraic equation C(t)=C0:

```
> T:=solve(op(S2)[2]=C0,t);
```

$$T := -\frac{\ln\left(\frac{\mu\text{~} - \alpha\text{~}}{\mu\text{~}}\right)V\text{~}}{\alpha\text{~}}$$

The value seems quite normal, so we plot its value for a=1/2 (which is independent of us), as a function of mu and V.

```
> plot3d(subs(a=1/2,T),mu=0.5..5,V=1..10);
```

FIGURE 2. Analysis of the solutions of differential equations, found in Figure 1.

or *Mathematica* should not be harder than using a calculator to do simple statistics of a few numbers.

After making the general comments, let us consider an example, describing the methods and tools, which may ease the analysis of the results. Figures 1 and 2 present an example session of *Maple*, in which the differential equations of Tyszkiewicz [1999] are solved. In that paper, a theoretical analysis of the process of filling a smoking chamber with smoke is investigated in two cases: the first is an open cycle chamber, through which the smoke flows, while the other is a closed smoke cycle chamber, to which the smoke is recycled after re-heating.

Maple and its relatives are general-purpose programs. Of course, there is a whole pletora of very specialized ones, which can be used for calculations within a specific field. An example might be the *HyperChem* [1992] program for quantum theory calculations of intra- and intermolecular interactions. It can be used for analyzing molecular level interactions between various food contents, as demonstrated by Mazurkiewicz [1997]. Of course, number of similar examples from different areas of food science, using tools from different areas of chemistry, biology, genetics, physics, *etc*., could be given.

MACHINE LEARNING

In this and the following section we do not assume that the theory behind the experiment is known.

A standard method to proceed is then the use of *machine learning* methods. This area of computer science is concerned with the creation of algorithms, whose purpose is to *learn* to compute a specific function, based on a number of training examples, for which the values of the function are provided. The algorithms then make an internal representation of the training examples and the results of the function. After this learning phase, the algorithm is used to compute the function for new arguments, which were not in the training set. Generally, the value of the function is computed by comparing the argument to the arguments found in the training set and interpolating, in one form or another, the values of the function known for those elements, with the present, previously unknown argument. Reference works for the whole section are Witten and Frank [1999], Cichosz [2000] Osowski [2000], and Rutkowska *et al.* [1999].

There exists a large number of subfields in the machine learning, each devoted to a specific methodology of creating learning algorithms.

In this paper we use a very simple methodological distinction between two subfields of machine learning, which seems to distinguish quite well those methods, which are routinely used in the chemometry of food science, and those, which are not.

The first subfield is the group of methodologies in which the information gathered by the algorithm after the learning phase is in a human-readable format. In particular, it allows the user to inspect the data, and possibly modify it to achieve a better performance of the algorithm. This group of methods is well represented by applications in food science and technology.

The second subfield is the group of methodologies in which the information gathered by the algorithm after the learning phase is in general unreadable for a human, making the user completely dependent on the machine. This group of algorithms seems to have only very few applications in food science.

In the following two sections we discuss these two classes of algorithms.

Learning human-readable information

In most of the cases the task of data analysis can be reduced to the problem of distinguishing, based on the result of the measurement, whether the examined sample has certain property or not. In many cases there is no theory that tells us how that property depends on the measurement results, or at least no such theory is known to us. However, there is a whole area of computer science, called *machine learning*, devoted to creation of computer algorithms capable of learning on provided examples, how that property can be decided, given the measurements.

For the purpose of illustration, we describe two such algorithms here.

The first of them is kNN, which is shorthand for "*k nearest neighbors*". k is a natural number and is a parameter of the algorithm. It can be set to any odd natural number. In practical applications, the typical values of k are 5 or 7. It can be used to approximate the value of an attribute a whose possible values are 0 and 1, only. Its data are vectors of numbers, of fixed length. In our terminology, each vector is the sequence $\boldsymbol{v}(s)=(v_1(s),...,v_n(s))$ of the values of certain attributes of the object s. In kNN, each such vector is considered as an element of the Euclidean space $\mathbf{R}^n$.

During the first phase the algorithm "learns" the attribute a in the following way: it is given a large number of vectors: $\boldsymbol{v}(s_1),...,\boldsymbol{v}(s_N)$, for which the values of the attribute v are known. These vectors form the knowledge of the algorithm.

In the second phase, the algorithm is supposed to approximate the value of the attribute v, *i.e.* for each new object s the algorithm should output either 0 or 1, given the vector $\boldsymbol{v}(s)$ as the information about s. The following procedure is used for that:

For each $0 \leq i \leq N$ the Euclidean distance

$d(\boldsymbol{v}(s_i),\boldsymbol{v}(s)) = \sqrt{\sum_{0\leq k\leq n}(v_k(s_i)-v_k(s))^2}$ in the space $\mathbf{R}^n$ is computed.

k among the objects $s_1,...,s_N$ are selected: those, which give vectors with the smallest distance to the vector $\boldsymbol{v}(s)$.

If the majority of the selected k objects have value of the attribute v equal to 0, the algorithm gives the result 0, otherwise (*i.e.* the majority of the selected k objects has value of the attribute v equal to 1), the algorithm gives the result 1.

As an example, consider the (oversimplified) case when $n=2$, so the vectors can be drawn on the plane as in Figure 3.

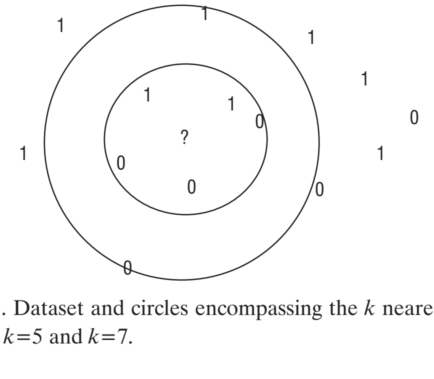


FIGURE 3. Dataset and circles encompassing the k nearest neighbours of ? for $k=5$ and $k=7$.

Each 1 in the picture is a vector $\boldsymbol{v}(s)$ for an object satisfying $\boldsymbol{v}(s)=1$, and similarly for 0. The question mark ? is the vector $\boldsymbol{v}(s)$ of an object s whose value of the attribute v is unknown. If k is 5, then the algorithm gives result 0, as indicated by the inner circle, encompassing the 5 closest symbols, among which there are three 0s and two 1s. With $k=7$ the algorithm still gives the same answer 0 – inside the outer circle there are the 7 closest symbols, four of which are 0s and only three 1s.

Observe an important feature of this algorithm: the result is not only the answer that $v(?)$ is probably 0, but also an *argument* why this should be so. In the example, the argument could be that among the objects s which are most similar to ?, the majority has value $v(s)=1$.

The method of (lazy) decision trees is used, exactly as kNN, to classify an unknown sample, given a set of training samples, Witten and Frank [1999], Cichosz [2000].

Input: An unknown sample and a set of training samples.

1. Choose a sensor and a threshold value, split the set of training samples into two subsets according to whether the value of that sensor exceeds the threshold or not.

2. Apply the same procedure to the subset in which the unknown sample falls, according to the value of the chosen sensor and threshold.

3. Stop splitting when the set of training samples contains samples of one type, only.

4. Classify the unknown sample to be of the type of the samples in the set.

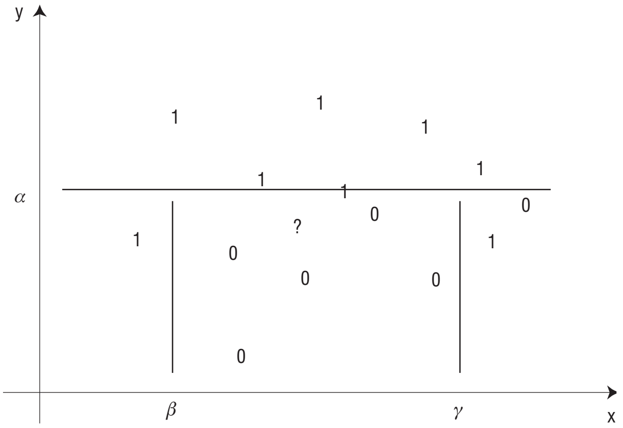


FIGURE 4. Dataset and splits introduced by a lazy decision tree algorithm while attempting to classify ?.

Figure 4 presents an example of a set of values and the splitting lines.

The crux of the method is the principle how to choose the sensor and how to set the threshold in point 2 of the algorithm above. We used the number of pairs with different answers that are differentiated by that split to choose the sensor and threshold. At each stage of the computation, the optimal choice of sensor and threshold is made, *i.e.* the one who guarantees the maximal number of pairs correctly differentiated.

Assuming that the precise location of the samples is represented by the top pixel of the number, the picture represents the splits of the lazy decision tree algorithm makes in the previous example. The threshold values of the parameters are denoted (in the order the splits are made) α, β, and γ. As one can see, the result of this algorithm gives is 0, exactly as in the kNN case.

Observe an important feature of this algorithm: the result is not only the answer that $v(?)$ is probably 0, but also an *argument*, in a human-readable form, why $v(?)$ should be 0. In the example, the argument could be rephrased in natural language as follows:

- For most of the objects s with $v(s)=1$ we have $y(s)>\alpha$ while $y(?)<\alpha$.
- For all of the (very few) objects s with $v(s)=1$ and $y(s)<\alpha$ we have either $x(s)<\beta$ or $x(s)>\gamma$, while $\beta<x(?)<\gamma$.
- Therefore $v(?)$ should not be 1, *i.e.*, $v(?)=0$.

Moreover, the decision process can be conveniently represented in a form of a tree, as in Figure 5 below.

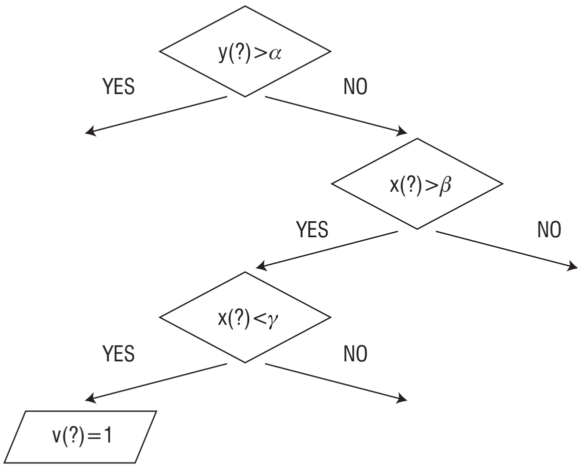


FIGURE 5. An example of a lazy decision tree, corresponding to the dataset on Figure 4.

Fragments of the tree, which are not visited in order to determine the answer for the present object ? are left unevaluated. A simple observation leads to the conclusion that after answering YES to the very first test (or NO to the second one), we may immediately conclude that $v(?)=1$. However, we do not recognize (and neither record) it until we attempt to classify another object for which $y(?)>\alpha$. For each object, only the fragments of the tree, which are necessary for its classification, are computed. This is the difference between the lazy decision trees (which we consider in this example) and the classical ones, where all the arcs and branches of the tree are constructed before any object is classified. The lazy version is more efficient because it is often the case that large portions of the full tree correspond to quite unusual objects of the training set (often due to "fat errors"). Similar objects are never encountered again, and hence the lazy tree is quite sufficient for handling them.

The latter method has been used by Maciejak *et al.* [2003] for detecting the smell of ammonia contaminating food samples. The data has been provided by an electronic nose *Cyranose* 320. The vapor of a sample of food is examined using an electronic nose. The result of measurement is a vector of 32 numbers – the resistances of the sensors of the electronic nose after saturation. The decision should be made whether or not the sample is polluted with ammonia.

There is no theory of the electronic nose. Its sensors are tiny polymer blocks, which absorb the vapors and increase in size. The enlargement is measured by the increase of their electric resistance. Each of the resistors changes its characteristics during its lifetime, as well as senses uncontrollable changes in the environment (like, *e.g.* the smell of blossoming flowers in the spring or increased concentration of ozone after a thunderstorm). Most likely the characteristics of particular sensors can differ in two otherwise identical electronic noses. Consequently, all we can hope for is to *learn* how our particular electronic nose reacts on ammonia. Of course it would be extremely difficult for a human.

Yet another example of a learning algorithm, which yields human-readable information, is *calibration of instruments*. We describe it on an example: calibrating a NIR machine NIRLab N-200 from Büchi Labortechnik AG [2003]. The main task is to make measurements with the machine on a number of samples of known characteristics. Say, if we want to measure the fat content of the samples, we obtain the infrared spectra of a specially prepared set of samples by the NIR, and subsequently determine their fat contents by other laboratory methods, to get the reference data. Next, a special software system is used for calibrating the NIR to measure the fat content in the probes.

1. First a sequence of mathematical operations is applied to the spectra. These may include applications of the Fast Fourier Transform FFT (this is done always), and then several other from a rich palette of possibilities, like computing derivatives of the spectra according to several available numerical algorithms, applying mathematical transformations like log(x), $1/x$, x^2, *etc*., selecting the most significant fragments of the spectra, *etc*.
2. The set of the transformed spectra is then split into two parts. The first one is the reference set.
3. The fat contents in the samples from the second, test set are estimated by comparing their transformed spectra to the transformed spectra from the reference set by a proprietary algorithm.
4. A quality measure of the estimations, the so-called Q-value, is computed based on the estimations and the known fat contents of the test samples.
5. The procedure is repeated starting from point 1. again, for a new sequence of mathematical operations.
6. Altogether a large number of sequences of operations is tested. The one, which gives the highest Q-value, becomes the calibration, together with the set of reference samples chosen.

Subsequently, the fat content of new, unknown samples is calculated by transforming their spectra by that chosen set of mathematical operations and comparing them to the analogously transformed spectra from the reference set – exactly as in point 3. of the above procedure. According to the information from Büchi, if the Q-value is at least 0.75, the calibration already qualifies for practical use, and the values of 0.85 or above promise excellent performance of the fat estimation.

The question arises, why we are so much interested in learning information in a form, which is usable for the human?

The answer is that this gives us several opportunities:

- We may be able to detect and correct non-systematic errors.
- We may be able to optimize manually our algorithms, based on the observation of methods they have learned. This is so, *e.g.* with the NIR calibration. According to the representative of Büchi Labortechnik AG, the company's experts can in many cases calibrate the system beyond what the software does automatically. Needless to say, they start from the best calibration obtained by the machine and, tuning the details of the calibration process, improve upon that.
- Finally, by analyzing the data gathered by a learning system one, at least theoretically, may discover the theory governing the process.

Learning for the machine

In contrary to the two algorithms described above, there are learning algorithms, in which, after the training samples are processed, the knowledge collected by the algorithm is represented in a form incomprehensible for humans. There are numerous such methods, only one of which has, to the best of our knowledge, been used in food technology applications. We describe two of them in a few words here, with the hope to point out their existence to the food science and technology community. Our choice of the presented methods is based on our personal, subjective feeling which of them are suitable for applications in this application area. We think they deserve experimental testing. The special feature of both technologies is that they come with a solid mathematical background. It has been namely mathematically demonstrated, that in simple cases and under mild additional assumptions, they are *guaranteed* to discover, in an automatic way, the optimal solutions to the problems they are applied to.

Neural networks

This method is the artificial neural networks. An artificial neural network is a (computer model of a) network of many simple processors (called neurons or units). These units are connected by communication channels, which usually transmit numeric data. The units work only on their local data (often a single number) and on the inputs they receive *via* the connections from their neighbors.

The initial phase is called "training", when the weights of connections are adjusted on the basis of data. In other words, neural networks learn from examples. The knowledge gathered this way is represented in the form of the channel weights, which is generally useless for humans. However, if trained carefully, neural networks may exhibit some ability to generalize their knowledge beyond the training data, that is, to give approximately correct answers for new cases that were not used for training.

Genetic algorithms

Genetic algorithms are based on a biological metaphor: They view learning as a process of competition among a population of evolving candidate problem solutions. Such candidates might be sequences of parameters in a linear formula $f(v_1(s),...v_n(s))$ for approximating an attribute $v(s)$ by the combination of the values of other attributes on the same objects together with abstract descriptions of strategies to choose the optimal measurement points to gather these data. However, equally well such solutions might be simple computer programs intended to perform a specified task. A "fitness" function evaluates each solution on the training data. The value of "fitness" describes the quality of the solution, as well as describes its chance to contribute to the next generation of solutions. Then, through operations analogous to mutation and gene transfer in sexual reproduction, the algorithm creates a new generation of solutions. This process continues until the solutions achieve the prescribed level of quality. A computer program simulates this artificial evolution scenario. After the solutions with high enough fitness are found in this "artificial evolution", they are subsequently used to solve the normal cases. Again, like in the case of neural networks, the fittest solutions, which form the knowledge of the algorithm, are often effective but the algorithm does not provide any evidence why they behave so well. Therefore this knowledge remains often incomprehensible to humans.

THREE EXAMPLES

In this section we describe three interesting examples of food science applications of computer science methods, or where such methods could have been used.

Growth of bacteria

The aim of the study of Tyszkiewicz *et al.* [2003] was to examine the growth of pattern strains: *Weisella viridescens* ATCC 12706 and *Escherichia coli* NCTC 8196, placed on the surface of slices of luncheon meat in the form of the inoculum of a determined concentration, during storage in various temperatures.

The formula below presents the dependence of the logarithm of the number of bacteria in one gram (CFU/g) $N(t)$ at the time t (counted in days) of sample storing.

$$N(t) = N_{\max}(1-e^{-K(t+t_0)}),$$

where $N_{\max}$ is the logarithm of the highest experienced number of bacteria (CFU/g), K is the constant speed of growth of bacteria, t is the time counted from the moment of sample contamination, and t_0 is the "initiation time", *i.e.* time necessary for the development of the tested microorganism to the initial level of contamination $N(t)=N_0$ for $t=0$.

The above equation has been derived theoretically at the assumption that the kinetic process of the growth of microorganisms is a first order chemical reaction. Under this assumption, $N(\tau)$ changes according to the differential equation

$$\frac{dN(\tau)}{d\tau} = K(N_{\max} - N(\tau)),$$

where τ is the time counted from the beginning of the growth of bacteria.

The equation after integration shall read as follows

$$\ln(N_{\max} - N(\tau)) = -K\tau + \ln C,$$

where $\ln C$ is an integration constant. If one determines the constant $\ln C$ at the boundary condition that for $N(0)=0$, one receives the next equation as follows

$$N(\tau) = N_{\max}(1-e^{-K\tau}),$$

which, after substituting $\tau=t+t_0$ gives the initial formula, which we wanted to derive.

The theoretical findings have been positively verified in experiments, whose descriptions can be found in the op. cit. Tyszkiewicz *et al.* [2003]. The portions of the sliced luncheon meat, manufactured in the form of sterile preserves were the subject of the research. The product, after taking out from cans, was sliced and contaminated on their surfaces with the inoculum of *Weisella viridescens* and *Escherichia coli* bacteria. The initial numbers N_0 of bacteria have been measured as a logarithm of the number of bacteria per one gram (CFU/g). The slices of luncheon meat, wrapped in the vacuum plastic bags were stored at a temperature of 2°C, 9 or 11°C and 20°C in the period of 1 to 4 weeks. In the determined intervals, *i.e.* after 1, 4, 7–8, 11–12 and 21–25 days the samples were taken for microbiological tests. The tests have been repeated twice using luncheon meat of a similar salt contents 2% NaCl, but of different humidity 68% and 60% and different fat contents 10.5% and 20.0% and thus of a different brain concentration ranging from 2.9% (1st repetition) to 3.2% (2nd repetition). The data analysis has led to the determination of the relevant constants for each of the experiments conducted.

The finding is that the growth of bacteria depends, first of all, on temperature. In general, the impact of temperature on the growth of bacteria in the range between the minimum growth temperature and thermal death temperature may be described by the Arrhenius equation, but at the assumption of the variable energy activation.

The value of the energy of activation for the experimental data in Joules per kilomol (JkM^{-1}) has been calculated with the application of the equation given by Loncin [1976] as follows

$$\ln\frac{K}{K_w} = \frac{A}{R}\left(\frac{T_w - T}{T_w \times T}\right),$$

where K_w is a pattern constant for the growth speed determined for the pattern temperature T_w, A is the activation energy, R – the gas constant.

As one can see in this example, we have an interesting two-layer theory governing the process. First is the growth of the bacteria itself, described by a simple differential equation. The second layer is the theory governing the behaviour of the constants in those equations as a function of temperature. The theoretical results have been derived manually, but a substantial reduction of the effort could have been achieved by using a computer algebra system (which has been indeed used only for visualization purposes in that paper). Needless to say, the theoretical derivations have been verified experimentally.

Identification of honeys

Identification of stain and origin of honeys is an interesting example of application of chemometric methods in food science. A classical method to identify the strain of a honey is to perform a so-called polynologic analysis, *i.e.* microscope observation of the insoluble fraction of honey. This fraction is to a large extent a mixture of pollens, fungal spores and algae. Its composition, estimated by manual counting under a microscope, allows one to determine the strain of the honey, which can be monofloral or multifloral, and identify the plants from which the honey has been collected. In order to mechanize and standarize this process, the method of near infrared spectrometry has been proposed by Piekut *et al.* [2000], which allows for the identification of several main components of honeys, like sugars investigated by Mates and Bosch-Reig [1997]. The geographical origin of the honeys can be determined by the analysis of the elements found in honey. It appears that the heavy metals present in the soil show up in the plants, and consequently also in the honeys made from nectars collected in that area, see Latorre *et al.* [2000], Latorre *et al.* [1999] and Leita *et al.* [1996].

A successful attempt to use the fluorescence spectra of honeys to determine their strain is due to Gębala [2003]. He has exposed honey samples to monochromatic light, changing its wavelength from 210 to 650 nm every 10 nm. At the same time the spectra of the induced fluorescence have been recorded in the range from 240 to 650 nm. The results of the experiment were complete spectra, illustrating how the intensity I_i of the fluorescence depends on the wavelengths of the source λ_w and the fluorescence λ_F. An example diagram for the buckwheat (*Fagopyrum esculatum*) honey is presented on Figure 6.

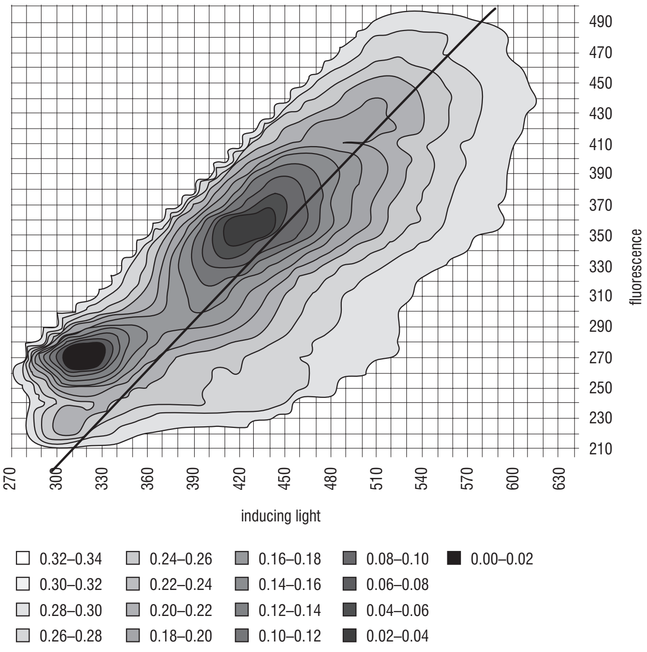


FIGURE 6. Intensity of fluorescence as a function of the inducing light for buckwheat (*Fagopyrum esculatum*) honey [after Gębala, 2003].

The maxima of the fluorescence intensity are typically of wavelength close to the wavelength of the inducing light. It is therefore possible to move from the two-dimensional spectrum to a one-dimensional one, which records the value of I_i at the wavelength $\lambda_e=\lambda_w+\Delta\lambda$ as a function of λ_w. $\Delta\lambda$ is the offset and can be in principle chosen arbitrarily, except that the line defined by $\lambda_e=\lambda_w+\Delta\lambda$ should not omit the extrema of I_i. Next, Gębala op cit. [2003] has collected a large number of spectra of honeys of different strains and origins. Considering the difference of each spectrum from the average of all those spectra, the author has been able to determine characteristic numerical attributes of these difference spectra. These attributes, after performing PCA, have been proven to suffice determine the composition and quality parameters of the honey.

It seems to the authors of the present paper, that an algorithm based on neural networks might be at least equally successful. First, it might easily analyze the complete two-dimensional spectra, as opposed to the reduced, one-dimensional ones created by Gębala. Next, it would be able to deal with them as a whole, rather than with a few chosen parameters only. Of course, the more data is passed to the identification algorithm, the better the accuracy of the method. This approach might therefore lead to a more accurate identification procedure. On the other hand, in one of the cases Gębala has been able to identify a (fake) linden honey as a mixture of two other strains. With the neural networks algorithm one probably could not make such an inference. This is the price of using learning algorithms, which gather knowledge in a format unreadable for humans.

Meatiness of pigs

Estimating the meatiness of fatteners is an important practical issue. It is also an example of a very effective application of modern data analysis methods. The price for pork carcasses depends on their meat content. Therefore one needs an objective method to evaluate it. The basic one is dissection – a destructive one, since after dissection a carcass is not a carcass any more. Therefore for a long time a simple yardstick has been used for measuring the lard thickness in a few specific locations along the backbone. In order to classify the carcass the results of measurements and the slaughter weight of the carcass were taken into account.

Only recently modern methods have been introduced to this field. Based on extensive zoometric data, the spatial distribution and quantitative dependencies among various tissues of the fattener have been determined. At the same time, modern instrumental technologies have allowed for non-destructive measurements of many of those parameters. In particular, the introduction of ultrasound detectors has had a great impact. They enable fast and reliable multipoint measurements of the thickness of the skin, fat, connective, bone, and muscle tissues. At present, the instrumental methods are already so accurate that the European Union has introduced a compulsory scheme called EUROP, which classifies carcasses into meatiness classes, described by Borzuta [1998]. It is an interesting fact, that this scheme must have been extended since its introduction. Improved breeding resulted in fatteners with meatiness over 60%, unforeseen by the authors of the original scheme. Part of the scheme is the requirement that all member states introduce approved instrumental methods to classify the carcasses. Apart from the instrument itself, the algorithm (or a mathematical formula), which determines the meatiness, is also approved by the EU. The accuracy presently required by the law is that the RSD does not exceed 2.5%. The reference value is, of course, determined using dissection. To date, according to the Institute of Meat and Fat Industry in Warsaw, three instruments have been approved for use in Poland: SGM, UltraFom 300 and AutoFom. The first two of them calculate the meatiness using fixed, empirically determined algebraic formulas involving measurement results. The last one uses a very interesting methodology. The carcass is moved continuously along a U-shape row of ultrasonic detectors, which measure the tissue thickness along the whole carcass on its spinal part and both sides (Figure 7).

The collected data is processed using a neural network algorithm (which must have been formally approved, too). This method has demonstrated accuracy substantially better than the required RSD≤2.5%. In fact, this seems close to the even theoretical accuracy limit. One must keep in mind that the reference data is meatiness estimated by the dissection method (the law requires that the simplified method of Markus and Walstra is used), which introduces its own error. In order to improve the instrumental methods even

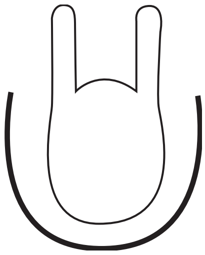


FIGURE 7. The detector placement of the AutoFom meatiness measurement method.

beyond that limit, a project has been recently discussed to use computer tomography results as the reference data in calibrating meatiness estimation procedures, see Dobrowolski *et al.* [2003].

SUMMARY

In this paper we have described a few statistical principles and computer science methods use in chemometry for data analysis. Some of them are well know and widely used, some other ones have only isolated applications, and some have not been used in that field to date. Our hope is that this description will prompt the researchers and practitioners to look at the methods developed in and provided by statistics and computer science. We have separated those tools into a few classes:

- Statistical methods, which help in non-systematic error removal, and control of the systematic errors. Statistics is also a tool to discover the simplest patterns in the data, which sometimes suffice already for successful deployment in practice.
- Computer algebra systems, which are the right aid when developing or applying an existing mathematical theory to interpret experimental data. Such systems can play a similar role for differential equations governing many natural processes, as calculators for adding rows of numbers. Computer algebra systems seem to remain relatively unknown in the food science and technology, while they clearly deserve much greater popularity.
- Learning algorithms, which present the learned knowledge in a form suitable for manual analysis. We described here the k nearest neighbors and lazy decision trees techniques. We have also pointed out that automatic calibration of an analytical instrument can be seen as an application of a learning algorithm.
- Learning algorithms, which present the learned knowledge in a form unsuitable for manual analysis. Our examples have been neural networks and genetic algorithms, whose popularity in the field is also far below what one could expect.
- We have concluded the paper with three examples from the literature, described in more detail to expose the applications of automated data analysis techniques, or the possibilities to do so.

ACKNOWLEDGEMENT

This research was supported by the State Committee for Scientific Research KBN, grant No. PBZ KBN(020) P06/1999.

REFERENCES

1. Borzuta K., Studies on usefulness of different methods of meatiness evaluation for the classification of porcine carcasses in the EUROP system. Roczn. Inst. Przem. Mięsn. Tłuszcz., 1998, 35/2, 5–84 (in Polish).
2. Cichosz, P., Systemy uczące się, 2000,WNT (in Polish).
3. Dobrowolski A., Romvari R., Allen P., Branscheid W., Horn P., X-Ray computed tomography as an objective method of measuring the lean content of a pig carcass. A study in the framework of the European Europigclass Project. 2003, *in*: Proceedings of 49th ICoMST Campinas, Brasil, 2003, pp. 371–372.
4. Gębala S., Studia nad wykorzystaniem widm fluorescencyjnych do identyfikacji odmian miodu, 2003, PhD Thesis, Akademia Morska w Gdyni (in Polish).
5. Horwitz W., Evaluation of analytical methods for regulation of foods and drugs. Anal. Chem., 1982, 54, 67A–76A.
6. HyperChem Computational Chemistry 1992. Autodesk Inc.
7. Latorre M., Pena R., Garcia S., Herrero C., Authentication of Galician (N.W. Spain) honeys by multivariate techniques based on metal content data. The Analyst, 2000, 125, 307–312.
8. Latorre M., Pena R., Pita C., Botana A., Garcia S., Herrero C., Chemometric classification of honeys according to their type II. Metal content data. Food Chem., 1999, 66, 263–268.
9. Leita L., Muhlbahova G., Ceso S., Barbattini R., Mondini C., Investigation of use of honey bees and honey bee products to asses heavy metals contamination. Environmental Monitoring and Assessment, 1996, 43, 1–9.
10. Loncin M., Génie Industriel Alimentaire. Aspects Fondamentaux. 1976, Masson.
11. Maciejak T.R., Kukawska-Tarnawska B., Tyszkiewicz J., Tyszkiewicz S., Multisensor odour detection and measurement of polluted food. Pol. J. Food Nutr. Sci., 2003, 12/53, SI1, 45–48.
12. Mates R., Bosch-Reig F., Sugar profiles of Spanish multifloral honeys. Food Chem., 1997, 60, 33.
13. Mazurkiewicz J., Calculations of intermolecular interactions of D-fructose with the HyperChem available programs. Pol. J. Food Nutr. Sci., 1997, 6/47, 31–39.
14. Osowski, S., Sieci neuronowe do przetwarzania informacji, 2000, Oficyna Wydawnicza Politechniki Warszawskiej (in Polish).
15. Piekut J., Witkowska A., Borawska M., Hejft R., An attempt to use spectrophotometric analysis in near infrared to distinguish honey species. Bromat. Chem. Toksykol., 2000, 32, 73–78 (in Polish).
16. Rutkowska D., Piliński M., Rutkowski L., Sieci neuronowe, algorytmy genetyczne i systemy rozmyte, 1999, PWN (in Polish).
17. Tyszkiewicz S., Kinetics of smoke filling of the smoke blow chambers with open and closed smoke cycle. Theoretical reflections on basis of mass balance. Roczn. Inst. Przem. Mięsn. Tłuszcz., 1999, 36, 191–196.
18. Tyszkiewicz S., Murzynowska M., Kitzman P., Moch P., Borys A., Kinetics of secondary micribial growth during storage of sliced luncheon meat. 2003, *in*: Proceedings of 49th ICoMST Campinas, Brasil, 2003, pp. 305–306.

19. Unique NIR solutions based on the NIRCAL Version 3.0 chemometric software, Büchi Labortechnik, 2003.
20. Witten I.H., Frank E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation, 1999, Morgan Kaufmann.

FINAL REPORT

Title of the research ordered project:
THE METHODOLOGICAL BASES OF THE EVALUATION OF THE QUALITY AND SAFETY OF THE NEW GENERATION FOOD (PBZ-KBN-020/P06/1999).

Title of the individual project:
Sensoric atmosphere analysis (electronic nose) in application to early detection of critical stages of selected food products during production and storage.

Institution:
Meat and Fat Research Institute, Jubilerska 4, 01-190 Warsaw, Poland.

Leader:
Prof. dr hab. Stanisław Tyszkiewicz

Co-workers:
dr inż. A. Borys, dr hab. J. Tyszkiewicz prof. UW, dr P. Kitzman, dr B. Kukawska-Tarnawska, dr inż. H. Makała, mgr M. Murzynowska, inż. S. Grześkiewicz, dr inż. M. Olkiewicz, dr inż. T. Płatek.

Key words:
Sensorous analysis of atmosphere, electronic nose, chemical activity, food, production process control, storage, food safety.

SYNTHESIS OF RESULTS

The aim of the project was to create methodological bases for process manufacturing control or storage of selected foodstuffs under the application of sensorous systems comprising "an electronic nose" designed for meat and edible fat processing. The basic problem is that the set of sensors must be "educated" relying on programming and qualification of analyzed impulse or impulses. Special mathematical techniques of integration and discrimination of experimental data exist. The authors of the project possessed an electronic nose (version Cyranose 320) made by Cyrano Sciences Inc., USA, designed with the use of polymer sensors reacting in the investigated atmosphere content *via* electric resistance changes. This instrument has 32 sensors with various characteristics and this feature allows obtaining considerably various spectra, peculiar to examined atmosphere. The principal feature of this apparatus is its portability, facility in the application under terrain conditions. The drawback of this apparatus is a big sensitivity to water moisture content in the examined atmosphere and in the air used for calibration and rinsing after subsequent measurement exposures. Another defect of an electronic nose is the instability of sensors, which enforces the need of frequent teaching activity (recalibration).

The instrument was checked on various objects, representing specified food products or characteristics conditions of examined objects and this supported usefulness of instrument to quality identification or to quantitative characterization of non-complex and reproducible objects (for instance water solutions of ammonium). Apparatus completely failed in the case of complex objects with non-reproducible matrix (for instance meat products contaminated with ammonium).

Considerable reproducible improvement was achieved after the application of alternative calculation method, not included in informatic (chemometric) instrumentation, for instance decsision trees, Rough-Sets or neural networks. This special technique brought excellent results in quantitative differentiation of two bacterial strains: *Escherichia coli* ATCC 700599 and *Weisella viridescens* ATCC 12706 cultivated in microbiological culture beds, then mixed together in various proportions. Equally good mathematical models have not been elaborated for more complex matrix (meat) so far. Reproducibility was also improved by supplementation of electronic nose with a system to standardization uptake of air from environment for instrument calibration and sensors refreshing. The essential element in this system is a washer with hygrostatic solution or solid state desicator. The application of the above-mentioned system eliminates the possibility of instrument transportation and limits the scope of the application only to laboratory use. Practical conclusions from the performed experiments pinpoint to quality investigations of accordance or lack of accordance with simple reproducible standard. Instrument could be recommended for detection of contamination with characteristic compounds, for instance with ammonium, with chlorine-emitting compounds, flavour active disinfectants, *etc*., in products or objects, on the whole. It could be possible to detect undesirable flavours deteriorating the quality of products, rancid lipids, buming stink or sulphuric compounds (mercaptanes). Attempts to apply Cyranose 320 for non-destructive examinations of complex flavours in products failed, in contrary to SPMS technique applied simultaneously, where the height of individual pics of GC spectrum depended on the concentration of a given compound.

The future prospects are as follows:

1. Elaboration of techniques and mathematical models suitable for quantitative differentiation of mixed cultures growing in meat matrix.
2. Model studies of technological processes dealing with hydrophobic and hydrophilic properties on example of investigative system based on triacylglycerols.