Knowledge mining sensory evaluation data: genetic programming, statistical techniques, and swarm optimization

Genet Program Evolvable Mach (2012) 13:103–133
DOI 10.1007/s10710-011-9153-2

Knowledge mining sensory evaluation data: genetic programming, statistical techniques, and swarm optimization

Kalyan Veeramachaneni · Ekaterina Vladislavleva · Una-May O'Reilly

Received: 4 January 2011 / Revised: 17 November 2011 / Published online: 20 January 2012

Abstract Knowledge mining sensory evaluation data is a challenging process due to extreme sparsity of the data, and a large variation in responses from different members (called assessors) of the panel. The main goals of knowledge mining in sensory sciences are understanding the dependency of the perceived liking score on the concentration levels of flavors' ingredients, identifying ingredients that drive liking, segmenting the panel into groups with similar liking preferences and optimizing flavors to maximize liking per group. Our approach employs (1) Genetic programming (symbolic regression) and ensemble methods to generate multiple diverse explanations of assessor liking preferences with confidence information; (2) statistical techniques to extrapolate using the produced ensembles to unobserved regions of the flavor space, and segment the assessors into groups which either have the same propensity to like flavors, or are driven by the same ingredients; and (3) two-objective swarm optimization to identify flavors which are well and consistently liked by a selected segment of assessors.

Keywords Symbolic regression · Sensory science · Ensembles · Non-linear optimization · Variable selection · Pareto genetic programming · Hedonic evaluation · Complexity control

K. Veeramachaneni (✉)
CSAIL, MIT, 32 Vassar Street, D-540, Cambridge, MA, USA
e-mail: kalyan@csail.mit.edu

E. Vladislavleva
Evolved Analytics Europe BVBA, Wijnegem, Belgium
e-mail: katya@evolved-analytics.com

U.-M. O'Reilly
CSAIL, MIT, 32 Vassar Street, D-534, Cambridge, MA, USA
e-mail: unamay@csail.mit.edu

1 Introduction

In this contribution, we knowledge mine sensory evaluation data collected for the purposes of flavor research and design. Extracting flavor information from sensory evaluation data has scientific and commercial value. Our particular experimental dataset is collected for hedonic flavor evaluation. See Fig. 1. Each sample (or query) supplied to the assessor is controlled by varying its flavoring ingredients. A hedonic response is obtained by asking an assessor to sense (e.g. smell, taste, view) something (e.g. a food product) and report how much he or she likes its flavor. The assessor reports how much he or she likes the flavor using 9 distinct language categories which range from “like extremely” to “dislike extremely” with “neutral” in the middle. These categoric responses are mapped to numeric scores per Fig. 2a.

In this contribution, we deal with a hedonic experiment dataset provided by Givaudan Flavors Inc. Research scientists at Givaudan prepared the experimental protocol, pre-screened the assessors, and conducted the experiment prior to our involvement in the project. Using a designed experiments approach, they prepared 36 different flavors. Each flavor is a composition of the same seven ingredients $k_1, k_2, \ldots, k_7$, called keys, at different concentrations. The ranges of concentrations of different ingredients is significant: some range between [0, 200], others between [0, 20]. Levels chosen for each ingredient in the experiment are either extrema or centers of each range. Each assessor, s, evaluates flavor $\mathbf{k}^j \in \mathbb{R}^7$ using 9-value language hedonic category scale and by assigning a liking score (*LS*) to it (as in the top of Fig. 1). Each of the 36 different flavors was evaluated by 69 assessors.

The intent of such a hedonic experiment is to facilitate understanding of which ingredients drive liking of a target population (assumed to be represented by the assessor panel), the consistency of liking preferences and to deliver insight into how to design or identify flavors that *most* consumers would *consistently* like. To extract information that provides this understanding is demanding because any dataset, ours being just one example, poses numerous technical challenges:

1. *The same flavors have been repeatedly measured and responses to them vary* In sensory evaluation studies, the same flavors are assessed by different assessors.

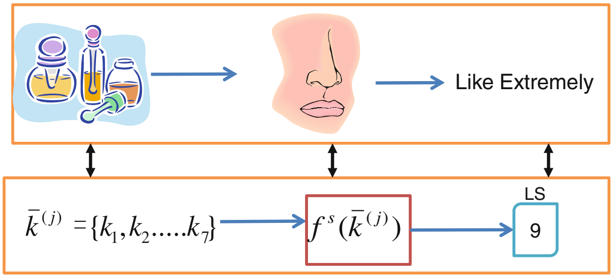


Fig. 1 Hedonic flavor evaluation data collection. *Top* the assessor reports a hedonic response to a presented item using a language category. *Bottom* flavor ingredients are varied to present different samples. Responses are converted to discrete, numeric, fixed range liking scores (LS)

(a)

LS	Rating
9	Extremely Like
8	Like Very Much
7	Like Moderately
6	Like Slightly
5	Neither Like Nor Dislike
4	Dislike Slightly
3	Dislike Moderately
2	Dislike Very Much
1	Dislike Extremely

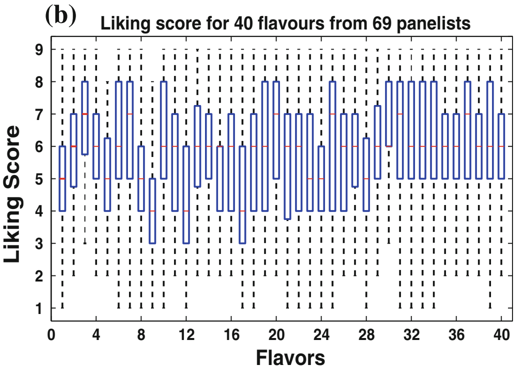


Fig. 2 **a** Category anchoring of the 9 point hedonic scale, **b** variation in the liking scores assigned by all assessors to a given flavor over all 40 flavors. There are 36 unique flavors in these 40 flavors. Box boundaries correspond to the interquartile range of 69 liking scores per flavor

The resulting data is best visualized to recognize the phenomenon we have to address. Figure 2b shows the responses for the 40 flavors for the 69 assessors. For 19 flavors, the responses vary across the complete range from 1 to 9. For most of the remaining flavors, the range is 7 or 8. This implies that assessors do not agree in their hedonic judgement. More technically, it implies that there can be no one single model that accurately describes the entire dataset. Instead, it is more desirable to construct a model for each assessor.

2. *The data is extremely sparse* The risk of rapid sensory exhaustion drives hedonic experiments to present an extremely small number of samples to an assessor. Our dataset is, in fact, only somewhat exemplary with respect to its sparsity[1]. It contains *more* flavor samples than usual but this amounts to only 36 flavors per assessor! Given the 7 dimensional space and range of each key, this is a very, very small fraction of samples of the flavor space. *How can such a small fraction be used to predict out of sample behavior if we model an assessor? How might analytic methods be validated for accuracy?* Whatever the challenge such a small sample size presents, it must be managed because modeling the entire panel is impossible due to the variation on the repeated measures.
3. *Ingredients interact hedonically in a non-linear manner* An additional complexity of sensory evaluation knowledge mining is the non-linear nature of the relationship between ingredients and a hedonic response. If the volumes of all ingredients in a flavor are doubled, the hedonic response will not necessarily stay the same, it might drastically change. This raises a question: *Could complex models be potentially necessary to accurately model an assessor's hedonic function?*
4. *Assessors may use the hedonic scale differently* An inquiring observer might wonder "What does the scale mean for each assessor?" Intuitively, people may

[1] The greater than normal number of samples were enabled by a proprietary method for delivering the flavor to the assessor which delays sensory fatigue.

not use the same scale to describe liking. When considering combining evaluations from many assessors, “how is it possible to ensure that the final outcome is scale invariant?”

5. *Designing well-liked flavors* Moskowitz has stated that “taste and smell, the chemical senses, are prime examples of inter-panelist differences, especially in terms of the hedonic tone (liking/disliking)” [19]. Continuing, Moscowitz remarks that “panelist to panelist differences are simply too great to ignore as just an inconvenience of the scientific quest.”
   In view of a business goal of designing flavors that many people like a lot, this phenomenon in the dataset logically raises questions such as: *“Can one flavor satisfy everybody? What is a realistic liking score for a flavor that satisfies everybody?”*

1.1 Project goals and a high level framework for solving them

Our knowledge mining framework is broadly depicted in Fig. 3. In general, to achieve our goals and address the challenges, we combine a variety of techniques including symbolic regression *via* genetic programming, ensemble methods, statistical techniques like Monte-Carlo sampling, Pearson correlation analysis, clustering and density estimation plus particle swarm optimization. The goals of the framework and high level methods of achieving them are:

Foundational Goal—Assessor Modeling Our first and foremost goal is to model each assessor. In other words, we seek to emulate individual assessors by building

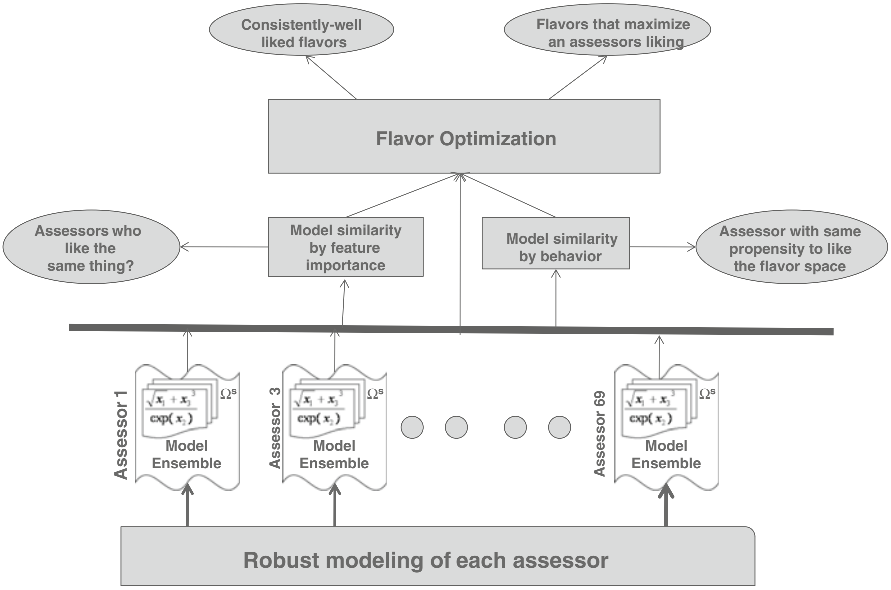

Consistently-well liked flavors
Flavors that maximize an assessors liking
Flavor Optimization
Assessors who like the same thing?
Model similarity by feature importance
Model similarity by behavior
Assessor with same propensity to like the flavor space
Assessor 1
Assessor 3
Assessor 69
$\sqrt{x_1 + x_3^3}$
exp(x_2)
Ω^s
Model Ensemble
Robust modeling of each assessor

Fig. 3 Knowledge mining framework for sparse sensory data with a focus on robust modeling for individual assessors, panel segmentation and finally optimal design of flavors

liking score functions (see Fig. 1). These functions relate the key ingredients to the hedonic liking score. The best way to address the sparse nature of the data is to generate as many explanations for it as possible. Therefore, we will derive a model ensemble for each assessor separately. Assessor modeling is a corner stone in our knowledge mining framework. It enables us to accomplish more extensive data analysis.

Analytic Goal #1—Segment the panel by driving ingredients and enable sensitivity analysis To accomplish this goal, our knowledge mining process will use the model ensemble developed for each assessor as a source of robust variable importance (i.e. driving ingredient) information. The frequency of variable occurrences in the models of the ensemble will be interpreted as information about the ingredients that drive the liking of an assessor. Model ensembles with the same dominance of variable occurrences and which demonstrate similar hedonic response directionality when the important variables are varied, will be grouped together. This method identifies assessors who are driven by the same ingredient set, in the same direction. Varying the input values of the important variables, while using the model ensembles of these panel segments, provides flavor design scientists with a means of conducting focused hedonic sensitivity analysis.

Analytic Goal #2—Segment the panel by propensity to like the flavor space Our goal is to to analyze an assessor's general propensity to respond either positively, negatively or neutrally to any flavor that is a combination of the 7 ingredients. After each assessor is identified into one of the positive, neutral or negative categories, we will consider those of each category to be a distinct segment. To accomplish this, we will take advantage of the assessor's model ensemble and generate the predicted behavior of the assessor over a very large set of unobserved flavors. We will then probabilistically describe the predicted responses. The three categories will be defined as ranges on the resulting distribution. This kind of segmentation enables flavor design scientists to take this sort of response variation into account when generating flavors that are maximally liked by many people.

Analytic Goal #3—Flavor optimization Our goal is to generate flavors that most people will maximally like. First, we will focus on achieving this for a single assessor. We will stochastically search for previously unobserved flavors that maximize the assessor liking by using the assessor's model ensemble to predict the hedonic response (i.e. liking score of) to each when our optimization requires an objective evaluation. In our second step, to find flavors that segments which are driven by the same ingredient like, we will combine predictions of multiple assessors and integrate their variation using a pair of objectives. The objectives will be related to maximizing the liking scores and minimizing the variance of the segment's responses.

We now proceed as follows: Sect. 2 addresses our foundational goal of assessor modeling. It answers the question:*"How to knowledge mine where there are only the responses of 69 assessors, each for only 36 samples?"* It summarizes a conventional approach to this kind of dataset then describes our proposed ensemble based symbolic regression (EBSR) approach. We present a variety of ensemble generation techniques, techniques to fuse multiple predictions from ensembles, and a technique to generate confidence measures. It presents the results of using our approach on the 69 assessors. Section 3 describes the steps to achieve our analytic goal #1. We provide two alternate methods to extract variable importance information from our ensembles for each assessor. We then show different segments of assessors derived using our technique. Section 4 provides the methodology to segment the assessors based on their propensity to like the flavor space to achieve analytic goal #2. We provide a Monte Carlo simulation technique to derive a probability density function for each assessors liking score. Three different clusters are formed within our panel of 69 assessors. Section 5 addresses analytic goal #3 by presenting a multi objective particle swarm optimization algorithm to optimize the flavors for a single assessor, a segment of assessors, and the entire panel. Section revists our challenges with this data mining problem and summarizes the solutions presented in this paper. Section 6 presents our conclusions and future work.

2 Foundational goal: assessor modeling

Consider a set of explanatory variables $\overline{x} = \{x_1 . . . x_n\}$, a response variable y and an unknown function $\mathcal{G}$ that relates $\overline{x}$ to y. Sparsity implies that very few data samples that explain $\mathcal{G}$ are available relative to the number of the explanatory variables, i.e. n.

In our problem the explanatory variables are the seven ingredients called *keys*, k_i. A flavor in the flavor space is a mixture by volume of these seven ingredients and the jth flavor is denoted by $\mathbf{k}^{(j)}$. 69 assessors evaluate 36 different flavors and select a rating for each that is translated to its *liking score*, *LS* per Fig. 2a.

Explanatory variables in the dataset were sampled using a D-optimal design for a full second-order model. The samples were chosen so that the standard errors (or variances) of the regression coefficients of a full second-order model are kept to a minimum. Keeping the standard errors of the regression coefficients at a minimum is important for analysis since these coefficients are the indications of which ingredients (and their interactions) drive liking or disliking.

Table 1 gives the notation for different variables used in this paper. We scale all key data to the same range in this study.

2.1 Conventional approach

A conventional approach to sensory evaluation data analysis is to explain the dependence between the key levels and the *average* liking scores of the entire panel by an empirical model. The model is constructed to approximate the average assigned liking score per flavor, and is usually a low-order polynomial obtained by linear regression. This model describes how much the panel, as an aggregate, likes any flavor, on average, in the space. Variable importance information is obtained from the analysis of model parameters. Variable sensitivity is studied based on predictions of the model.

Table 1 Problem specific variable description

Variable	Notation	Details
Flavor space	F	The design space of ingredient mixtures
Keys	k_i	$i \in \{1\ldots7\}$
Flavor	$\mathbf{k}$	A mixture of 7 keys, $\mathbf{k} = \{k_1, \ldots k_7\}$
A specific flavor	$\mathbf{k}^{(b)}$	A specific flavor denoted by superscript b
Assessor	s_n	$n \in \{1\ldots69\}$
Set of assessors	S	$S = \{s_1, s_2, \ldots s_{69}\}$
Observed flavors	F_o	$F_o = \{\mathbf{k}^{(1)}\ldots\mathbf{k}^{(40)}\}$
Unobserved flavors	F_B	$F_b = \{\mathbf{k}^{(41)}\ldots\mathbf{k}^{(10041)}\}$
Liking score function	$f^s(\mathbf{k}^{(j)}) = LS$	Relationship between a $\mathbf{k}^{(j)}$ and LS
lsd	$p(LS \mid s)$	Liking score density model for assessor s
Cumulative density	$P_x(LS \geq x \mid s)$	Probability of liking score $\geq x$
Assessor cluster	S_c	A subset of S, $c \in \{E, N, H\}$
Model	m	One model m for assessor s
Model ensemble	$\mathcal{M}^s$	Ensemble for assessor s, $\mathcal{M}^s = \{m_1^s, \ldots, m_j^s\}$
Ensemble response	$\mathcal{M}^s(\mathbf{k})$	All responses $\mathcal{M}^s : \{m_1^s(\mathbf{k}), \ldots, m_j^s(\mathbf{k})\} \in \mathbb{R}^{j(s)}$
Ensemble prediction	$Y(\mathcal{M}^s, \mathbf{k})$	$\mathrm{MedianAverage}(m_1^s(\mathbf{k}), \ldots, m_j^s(\mathbf{k})) \in \mathbb{R}$
Ensemble confidence	$C(\mathcal{M}^s, \mathbf{k})$	see Sect. 2.2.1

The single model that inevitably predicts the average liking score completely ignores assessor differences in the liking. Common sense favors the approach that builds models per assessor, extract as much information about the assessor from these models and then combines this information for multiple assessors when necessary. To mitigate this problem for flavor optimization and include some information about different liking patterns, in some cases a polynomial model is constructed for each assessor. Each of these models is then aggregated additively and maximized using a non-linear constrained optimization method.

The conventional approach to liking score modeling, analysis and optimization has three key problems:

- *Model error* The main problem with developing a single model for the panel is the true error (versus the lack of fit error). True error describes the difference in panelists predicted and actual responses. The single model only has average liking score predictive capability. Given the variance in responses across the panel for many flavors, this average is a poor match to each assessor's responses.
- *Lack of extrapolative capabilities* Models build using parametric linear regression (quadratic models on all ingredients) do not possess any reliable extrapolation capabilities (unless the true relationship between the liking scores and key levels is indeed quadratic). With a single model per assessor, due to sparsity of data, it is hard to build a model that is reliable (has good predicting

capabilities on unobserved points) and robust (less error prone on observed points), i.e. models of high accuracy and no over-fitting.

- *No confidence* A model constructed for the entire panel or for an individual assessor provides liking score predictions, but totally lacks information about the confidence in its predictions. The risk of producing untrustable predictions through modeling cannot be mitigated unless modeling methodology provides some measure of confidence in its prediction. This confidence measure does not accompany a conventional parametric linear regression model.
- *No optimization trade-offs* Flavor optimization is inherently multi-objective, given that assessors' liking on the same flavor vary. Maximizing the average liking score over the entire panel without taking into account variation among scores is insufficient, since it does not provide insights on flavors that *many* people like *a lot*. (A flavor with higher average liking score with a very high variance is by design less profitable than a flavor with a slightly lower, but very consistent liking score.)

The dimensionality of the design space is too high for parametric methods like spatial correlation (kriging) to perform well. It calls for semi-parametric and non-parametric methods, like neural networks and GP symbolic regression. This, in addition to the lack of confidence information in predictions, motivates the use of ensemble-based modeling methods. In principle, models obtained in both cases can be grouped into ensembles, but the extrapolative capabilities of GP symbolic regression are stronger and therefore we focus on GP symbolic regression in this study.

2.2 ParetoGP symbolic regression

Learning from sensory data is a perfect example of an application where *the* model does not exist. To gain prediction robustness on this sparse data, we adopt an ensemble based symbolic regression approach to provide multiple unbiased explanations of the input-output relationships in the data. These multiple models, also known as model set, $\mathcal{M}$ should contain diverse but high-quality models, which are constrained to approximate *all* training data samples well (high-quality) and are also constrained to diverge in predictions on unobserved data samples (diverse). When a sufficient number of models are generated, all of them can be used to determine both the prediction (by unifying their predictions) and the disagreement at an arbitrary point of the original variable space.

The *multiple model generating capability* of GP enables us to achieve these model sets on sparse data sets. To our surprise it is often ignored (or taken for granted) and a GP with single-objective fitness driven selection, and a single best-of-the run final solution (see [10, 11, 13, 25] among others) is used. In this paper, we attempt to fully exploit the *multiple model generating* capability of the GP. We employ a robust approach using ParetoGP which is symbolic regression *via* tree-based GP implemented with archiving (elite-based selection with elite preservation), two-objective selection and other defining features [28]. ParetoGP gives the aggregated final archive of multiple independent runs, called replicates. In a single run the algorithm performs the following operations:

1. *Initialize models* The following primitives are used for tree-based individuals: $\{+, -, *, /$, *inverse*, *power*$(x, const)$, *square*, *ln*, *exp*$\}$. They range in arity 1 to 2. The list of variables, which in our case are seven keys and real constants from $[-5, 5]$ are used as terminals. We rescaled our input variables to $\{0, 2\}$ range.
2. *Perform multi-objective evaluation* The models are evaluated under two objectives. The first one, model error, is defined as $1 - R^2$, where R is a correlation coefficient between scaled predicted and scaled observed response. The second objective, model complexity, is defined as the sum of all subtrees of the tree-based genome of the GP individual. The goal is to minimize both error and complexity.
3. *Archive the best models and update* An archive of individuals is created separately from the population and an elite-preservation strategy is employed. At generation $t + 1$, the archive, which is the elite set of best individuals discovered so far gets updated. Its size is limited to *ArchiveSize* by selecting the least-dominated individuals from the union of *Archive*(t) and *Population*$(t + 1)$ in the objective space of model error and model complexity.
4. *Vary the models* During each iteration, a new population is created using archive mutations and crossovers. In crossovers, parents are either both sampled from the archive, or one parent sampled from archive and one from the population (in both cases using Pareto tournament selection). This archive-based selection preserves genotypic diversity of GP individuals. The new individual is generated by using a sub-tree crossover with rate 0.9 and sub-tree mutation with rate 0.1. Every 10 generations, the population gets re-initialized to provide diversity and avoid inbreeding.

Other parameters for the ParetoGP are given in Table 2. A run is executed for a time interval, using all the observations, because using complexity as a second objective and collecting multiple solutions in accuracy-complexity trade-off space eliminates any requirement for an arbitrary maximum generation or cross-validation that would make the training data even more sparse. Some evolved models will "over-fit" but they can rationally be pruned *post-hoc* when the model set is finalized to be used for prediction. The time interval we chose is equivalent to 280 generations. Interval arithmetic is used to prune individuals with numerical inconsistencies and linear scaling is used to enhance effectiveness of evolution [9].

At the end of an experiment, the models in the archives of all runs are aggregated into a super archive. The non-dominated solutions in this archive form the super Pareto front (SPF). This is illustrated in Fig. 4. We call this a model set, $\mathcal{M}$ and generates these model sets for each subset of data samples corresponding to an assessor $\mathcal{F}_s(\overline{x})$. When repeated for all the assessors, symbolic regression creates rich sets of models. We generated model sets for all 69 assessors using our approach. Figure 5 shows the super Pareto front plots achieved for a subset of assessors (27 of them). We examined the results achieved for these 27 assessors. Table 3 summarizes the details of the model sets achieved for the 25 assessors. An average of 1,071 models are generated for each assessor. An average of 41 of lie on the SPF.

Table 2 ParetoGP experimental parameters

Parameters	Comments
# Replicates	5 unless stated otherwise
# Generations	280
Population size	1,000
Archive size	100
Fitness	$1 - R^2$
Complexity	expressional complexity
Crossover rate	0.9
Subtree mutation rate	0.1
Population tournament	5
Archive tournament	5

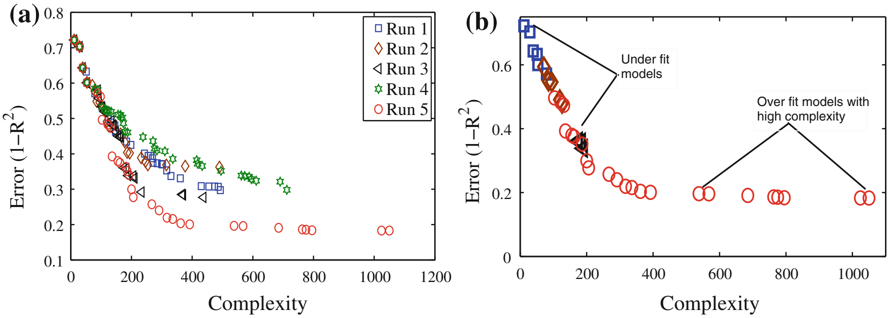


Fig. 4 An exemplar ParetoGP simulation on the sparse data **a** results from multiple runs of ParetoGP. Pareto fronts from each run show the trade-offs between model error ($1 - R^2$) and model complexity. **b** A super Pareto front is generated by aggregating the Pareto fronts from multiple runs. The super Pareto front has 37 models

2.2.1 *Ensemble selection*

The idea of using ensembles for improved generalization of the response prediction is by far not new in regression. It has been extensively used in neural networks (e.g., [7, 14, 17, 18, 35]) and even more extensively in boosting and machine learning in general (albeit, mostly for classification). See [2, 4, 5, 8, 22, 24, 29] for examples. Krogh and Vedelsby [14] presented the idea of using disagreement of ensemble models for quantifying the ambiguity of ensemble prediction for neural networks, but the approach has not been adapted to symbolic regression.

In [12], the authors describe an approach to selecting the models which form an ensemble: collect models that differ according to complexity and prediction accuracy. Complexity can be measured by examining some quantity associated with the GP expression tree or by considering how non-linear the expression is. Accuracy is the conventional error measure between actual and predicted observations.

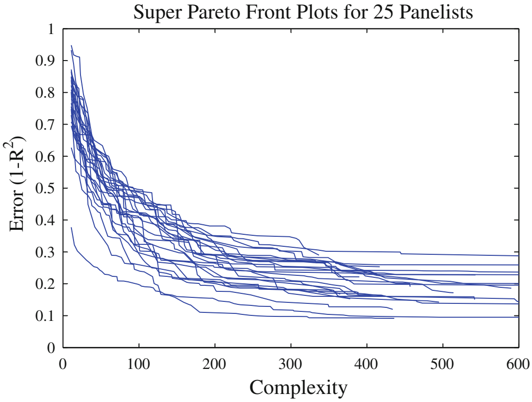


Fig. 5 Super Pareto fronts generated for multiple assessors using ParetoGP based symbolic regression

Table 3 Model ensemble results for multiple assessors

Parameters	Value
# of models	1,071
# of models in SPF	41

Specific predictions are considered to assess correlations and eliminate correlated models. Generally, each ensemble combines:

- A “box” of non-dominated and dominated models in the dual objective space of model prediction error and model complexity.
- A set of models with uncorrelated prediction errors on a designated test set of inputs. Here a model is selected based on a metric which expresses how its error vector correlates with other models’ error vector. The correlation must not exceed a value of ρ. The input samples used to compute prediction errors can belong to the test set (if available), or be arbitrarily sampled from the observed region.

Our intent is to use symbolic regression ensembles developed for each assessor as a surrogate for the assessor. When we want to “know” how much the assessor would like a (unobserved) flavor, we feed the flavor’s ingredient levels into each model of the ensemble and receive a prediction. The issue then is how to fuse the predictions from the multiple models. In fact, “fusing” (or unifying) the multiple predictions from an ensemble is an open area of research. In this contribution we choose a median-average method for its robustness (see [32]). We additionally exploit the ensemble to derive another piece of valuable information: confidence, expressed locally, for a specific prediction. We experiment with a confidence measure based on entropy. These are explained next in detail in Sect. 2.2.2.

2.2.2 *Predictions and confidence measures*

We use median-average for fusing multiple predictions from multiple members of ensembles and use entropy as a measure for confidence. We have previously used inter-decile range for the predictions as a measure for confidence.

Prediction via median-average The median prediction and its two neighbors in the prediction space are identified and averaged.

Confidence via entropy In this method the discrete probability mass function is formed for the liking score predictions, in $Y(\mathcal{M}^s, \mathbf{k})$, that are approximated to nearest integer value and the entropy [1] is evaluated using:

$$H = -\sum_{i=1}^{9} p(a_i) log(p(a_i)) \tag{1}$$

where $p(a_i)$ is the probability mass generated from the predictions for the liking score values $a = \{1, 2, \ldots 9\}$. Higher entropy implies higher random behavior. Figure 6 illustrates the properties of the entropy based confidence placement. It shows two different scenarios of the predictions. In the first scenario, the predictions in the $\overline{Y}^s$ are equally divided among the different values on the rating scale, giving each a probability of 1/9. In the second scenario, only three values are chosen by the predictions, i.e. 7 (1/9), 8 (1/9) and 9(7/9). The entropy is higher for the first scenario indicating random nature of the predictions. In fact the entropy is maximally bounded by this number and we denote this value as H_{max}. Higher entropy implies lower confidence. In the second scenario, due to the spiky characteristic of the discrete probability mass function, the entropy is lower and implies high confidence because many predictions converge to the same rating. Thus we define confidence as:

$$C_E(Y(\mathcal{M}^s, \mathbf{k})) = 1 - H_{norm}, \tag{2}$$

where $H_{norm} = \frac{H}{H_{max}}$. C_E's value lies between [0,1].

3 Analytic goal #1: segment the panel by driving ingredients and enable sensitivity analysis

Figure 7 graphically describes how goal #1 is achieved. The approach of this section, in the sensory science context, is:

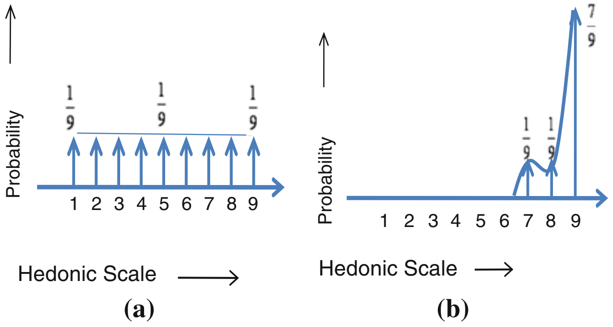


Fig. 6 Discrete probability mass of predictions for cases of **a** maximum entropy—low confidence and **b** low entropy—high confidence

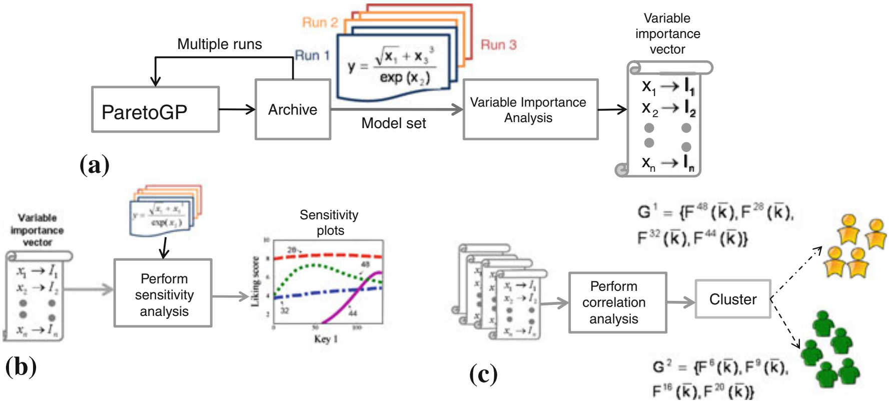

Multiple runs
Run 2
Run 1
Run 3
ParetoGP
Archive
Model set
Variable Importance Analysis
Variable importance vector
(a)
Variable importance vector
Perform sensitivity analysis
Sensitivity plots
Liking score
Key 1
(b)
Perform correlation analysis
Cluster
(c)
G¹ = {F⁴⁸(k̄), F²⁸(k̄), F³²(k̄), F⁴⁴(k̄)}
G² = {F⁶(k̄), F⁹(k̄), F¹⁶(k̄), F²⁰(k̄)}

Fig. 7 Panel segmentation and sensitivity analysis **a** the ensemble of models is analyzed for variable importances. **b** The variable importance information along with the archive of models is used for performing sensitivity analysis. **c** The variable importance for multiple assessors is used to perform correlation analysis and segment them

- to identify the variables that drive liking scores of the assessors,
- to segment the panel into groups that are driven by the same ingredient.
- and to understand the direction of the driving, i.e. the analysis of the changes in the liking scores caused by changes in the concentration of the keys (sensitivity analysis).

We make use of the model ensembles generated for each assessor to derive variable importance vectors and analyze their similarity and dissimilarity based on these vectors. In Sect. 3.1, we present two techniques to derive variable importance vectors.

3.1 Variable importance

Most non-evolutionary modeling methods are vulnerable to producing solutions that contain insignificant inputs. This results in a fast deterioration of prediction performance of final solutions as more irrelevant variables in the data are considered in the model.

A conventional approach to identify the true dimensionality of the problem is to perform a principal component analysis or a factor analysis. The former reduces the problem dimensionality to a smaller number of meta-variables which are linear combinations of the original variables. An alternative, factor analysis, extracts the latent dimensionality of the problem by determining the number of factors that contain the same information as the matrix of mutual correlations of data variables. The potential problem of these approaches (in analysis of non-linear systems) is that they only take into account mutual correlations between variables and hence miss the relevance of non-linear combinations of inputs to the response. As well, they do not select important variables from the original set, but create new variables in the new reduced set. They forfeit multicollinearity (which is most often present in real measurements).They are also sensitive to outliers.

One of the unique capabilities of genetic programming is its built-in power to select significant variables and gradually omit the variables that are not relevant while evolving models. Variable selection based on genetic programming has been exploited in various applications where the significant inputs are generally unknown (for examples see [36, 15, 21, 23, 26, 36]).

We use an effective method for variable presence analysis for a set of models generated using ParetoGP. Note that the method could also be used on the population of solutions at the end of a standard GP run.

We define a *variable importance vector* as a vector $V = \{\mathcal{I}_1, \mathcal{I}_2, \ldots \mathcal{I}_d\}$ of the importance of all explanatory variables $\{k_1, k_2, \ldots k_d\}$, in percents, arranged in the same order as the explanatory variables. The importances are relative if $\sum_{k=1}^{n} I_k = 100$.

In [33] the following methods were introduced:

1. *Presence-weighted variable importance* This method analyzes variable presence rates in a subset of models $\tilde{\mathcal{M}}$ from the ensemble archive and considers variables relevant if they have a high presence rate. The *aggregated importance* of the variable $k_i, i = 1\ldots, d$ computed on the basis of best models $\tilde{\mathcal{M}} = \{m_j\}, j = 1, \ldots, l$ is

$$\mathcal{I}_i^{(PW)}(k_i, \tilde{\mathcal{M}}) = \sum_{j=1}^{l} \frac{\delta(k_i, m_j)}{l}, \qquad (3)$$

where $\delta(k_i, m_j)$ is zero if k_i is not present in model m_j and one otherwise. This aggregated variable importance provides a robust estimation of relevance *if* $\tilde{\mathcal{M}}$ is hand selected for high-quality (i.e., fitness and complexity) from an experiment-archive derived from many independent runs.
The second variable importance metric resolves the problem of hand selecting $\mathcal{M}$ by eliminating the need for it.

2. *Fitness-weighted variable importance:* Fitness-weighted variable importance is calculated using all models (in the archive or in both archive and population) (see [27]). It first uniformly distributes the fitness of each model over all variables present in it, thus assigning a variable a score per each model it is present in. Then, it sums up the scores over all models, $\mathcal{M} = \{m_j, j = 1, \ldots, l\}$.

$$\mathcal{I}_i^{(FW)}(k_i, \mathcal{M}^s) = \sum_{j=1}^{l} \frac{\mathit{fitness}(m_j)}{\sum_{i=1}^{d} \delta(k_i, m_j)} \delta(k_i, m_j), \qquad (4)$$

Since the fitness of a model is uniformly distributed over all its variables, this creates an explicit bias towards variables occurring in lower dimensional solutions. Thus, the overall aggregated scores of irrelevant variables (only present in overfitting solutions) is much smaller than the overall score of relevant variables.

We use normalized fitness-weighted variable importances defined as:

$$\mathcal{I}_i^{(NFW)}(k_i,\mathcal{M}^s)=\frac{\mathcal{I}_i^{(FW)}(k_i,\mathcal{M}^s)}{\sum_i \mathcal{I}_i^{(FW)}(k_i,\mathcal{M}^s)}\cdot 100\%. \tag{5}$$

3.2 Segmentation by sensitivity to flavor ingredients

Having generated variable importance vectors for all the assessors, we are next interested in how similar one assessor is to another. This is equivalent to identifying people driven by the same key and, in sensory evaluation, is called segmentation. Segmentation enables design strategies for multiple, similar people and is highly useful.

We use the variable importance vector as the basis for deriving this similarity. Consider a model set denoted by $\mathcal{M}^s$ for an assessor s and the corresponding variable importance vector as V_s. Computing this for each assessor (s)'s model set, $\mathcal{M}_s$, we aggregate the 69 vectors of normalized fitness-weighted importances into a table of 69 rows, with their elements reflecting the relative importance of seven keys for predicting the liking scores of each panelist.

We then compute pairwise correlation between each pair of vectors and construct a correlation matrix defined by β. The entry β_{ij} in this matrix is the Pearson correlation coefficient between variable importance vectors of model set, i and j given by:

$$\beta_{ij}=\frac{\sum_{k=1}^{n}(V_i^k-\bar{V}_i)(\bar{V}_j^k-\bar{V}_j)}{(n-1)\sigma_{V_i}\sigma_{V_j}}, \tag{6}$$

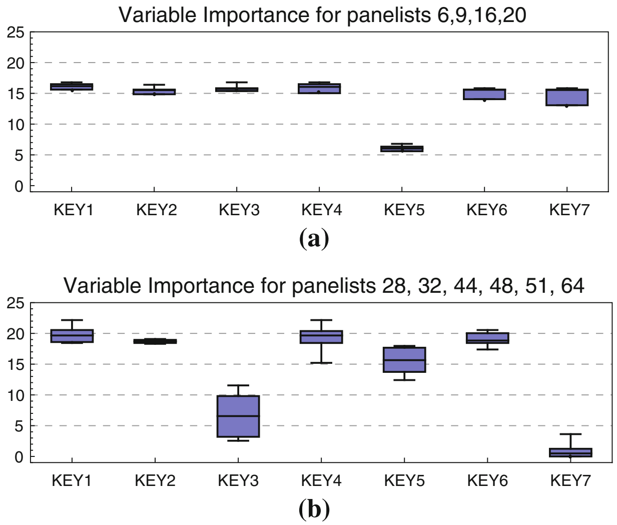
where σ_{V_i}, σ_{V_j} are sample standard deviations for V_i and V_j. We then group panelists with similar variable importance vectors using a correlation threshold θ to the matrix entries. A cluster is identified when all the pair-wise correlations exceed θ. By selecting pairs with correlations exceeding a threshold $\theta = 0.90$, we identify groups of assessors with similar variable importance vectors.[2]

There are five groups of assessors with high pair-wise correlations of importances between all members in a group. Fitness-weighted variable importances of two groups are illustrated in Fig. 8. Notice the high consistency in variable importance vectors per group, and the clear differences among the two groups. All the variables except k_5 are required to predict the individual liking scores of Group 1 consisting of panelists 6, 9, 16 and 20. All variables except k_3 are required for Group 2 consisting of panelists 28, 32, 44, 48, 51 and 64.

3.3 Variable sensitivity analysis

The variable importance vector enables a means of sensitivity analysis which supports efficient exploration of the design space to observe the response variable under selected conditions of the explanatory variables. Consider an explanatory variable set consisting of n variables where each variable can be explored in

[2] The choice of the θ threshold is highly influential in the subsequent conclusions.



(a)

(b)

Fig. 8 Distribution of variable importances for two distinct segments of panelists

r discrete step sizes. The total number of design exploration samples is n^r which is generally intractable.

To alleviate this, the variable importance vector can be used. The distribution of the percentages in the variable importance vector informs the choice of downsizing the sampling. The effects of q influential variables, where $q \ll n$ can be explored while the *non-influential* $n - q$ variables are clamped to a finite set of combinations, $c \ll (n - q)^r$. The q *influential* variables can be exhaustively sampled over q^g. For each sample, the predicted response of the predictive model ensemble is calculated using a median-average method [31]. The predictive model ensemble is derived by boxing the ensemble-archive. See [31] for more details. These predictions for the q most relevant variables can subsequently be visualized to observe the assesors's sensitivity under the clamped conditions. The values of $q,\ c$ and g are selected based on the needs of the modeling application. It is sensible to also reduce g as the importance ranking of an influential variable decreases. This supports coarser grained sampling in dimensions where variable importance is less and higher grained sampling where it is higher.

We now perform sensitivity analysis to understand whether a key has direct or inverse relation with the liking score. By doing this we can identify assessors, for whom both the ith key is the most important variable, but one might hate it as its concentration increases and another who might like it. We use Group 1 and Group 2 as our exemplars. In Fig. 9 we plot the individual ensemble predictions (median of predictions of ensemble members) of all assessors in the group for varying volume levels of k_1, while all other levels are clamped to their maximum. The step size in varying k_1 is domain related. The spread in ensemble predictions in Fig. 9 justifies once again the differences in the assessors. In Group 1 one segment of assessors $\{6, 16\}$ have monotonically increasing predicted liking scores for increasing levels

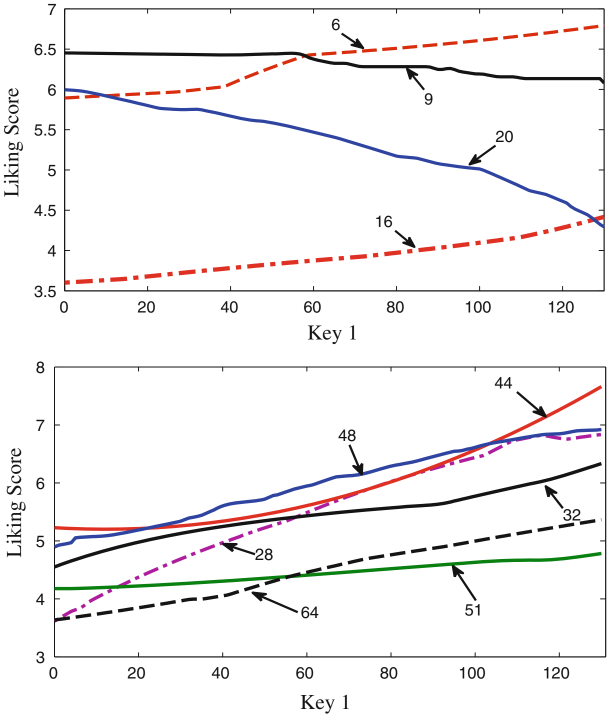


Fig. 9 Comparison of sensitivity of liking of panelists within Group 1 (*top*) and Group 2 (*bottom*) based upon varying the levels of key k_1 while k_3, $k_5 - k_7$ clamped to their maximum, and k_2 and k_4 clamped to zero. *Numbers* indicate assessor ids

of k_1, another segment, {9, 20} shows decreasing liking scores as k_1 is varied in the interval [0,130]. In Group 2 the liking score increases for all the assessors as k_1 is increased.

4 Analytic goal #2 segment the panel by propensity to like the flavor space

In our knowledge mining framework, we next make use of the ensemble, $\mathcal{M}^s$ designed for an assessor s to answer the question: "*How likely is an assessor to answer with a liking score/rating higher than X?*" The answer to this question allows us to categorize assessors, w.r.t. the flavor space, as: (1) Easy to Please, (2) Hard to Please, (3) Neutral. We accomplish this by modeling the probability density function given by $p(LS|s)$ for an assessor s [34].

Density estimation poses a critical challenge in machine learning, especially with sparse data. Even if we assume that we have finite support for the density function and it is discrete, i.e. $LS = \{1, 2, \ldots 8, 9\}$, we need sample sizes of the order of "supra-polynomial" in the cardinality of support [30]. In addition, if the decision variables are inter-dependent, as they are here, estimating a conditional distribution increases the computational complexity. Most of the research in density estimation has focused on identifying non-parametric methods to estimate distribution of the data. Research on estimation of density from very small sample sizes is limited[20, 30].

Figure 10 presents the steps taken to form this liking score probability density model. We first generate 10,000 untested flavors. We use the model ensemble $\mathcal{M}^s$, which gives us a set of predictions $\mathcal{M}^s(\mathbf{k})$. For each untested flavor we get a set of predictions (*not just one*), which plausibly represents all possible liking scores the assessor would give. We use these to construct the liking score density model, for an individual assessor. The process has five steps:

1. Generate flavor samples for the seven dimensional ingredients using a uniform random number generator.
2. Thoughtfully select an *ensemble* of models meeting accuracy and complexity limits to admit generalization and avoid overfitting and a correlation threshold to avoid redundancy.
3. Use *all* models of the ensemble to generate multiple predictions for the unseen flavor samples generated in 1.
4. With minor trimming of the extremes and attention to the discrete nature of liking scores, fit the predictions to a Weibull distribution.
5. Segment based on the Weibull distribution's probability mass.

It is significant to note that these steps respect the importance of avoiding premature elimination of any plausible information because the data is sparse. The ensemble provides all valid values of the random variable (in this case is the liking score prediction) when it is presented with new inputs. This extracts maximum possible information about the random variable, which supports more robust density estimation.

4.1 Deriving predictions by Monte-Carlo simulation

To generate the bootstrapped data of liking scores for the $F_b = \{\mathbf{k}^{(41)}\ldots\mathbf{k}^{(10041)}\}$ we follow the steps described in Algorithm 1.

4.2 Parametric estimation of the liking score density function

We use a parametric Weibull distribution to estimate $p(LS|s)$. The two parameters for the Weibull distribution, λ and r are called scale and shape respectively. A Weibull distribution is an adaptive distribution that can be made equivalent to an

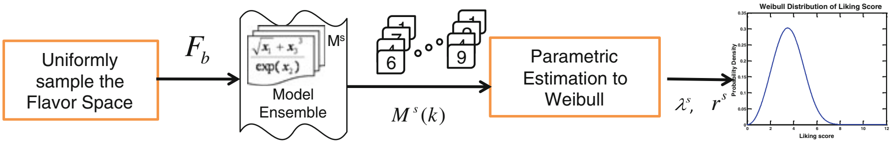

Uniformly sample the Flavor Space
F_b
M^s
$\frac{\sqrt{x_1 + x_3^3}}{\exp(x_2)}$
Model Ensemble
6
9
$M^s(k)$
Parametric Estimation to Weibull
λ^s, r^s
Weibull Distribution of Liking Score
Probability Density
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
2
4
6
8
10
12
Liking score

Fig. 10 Monte Carlo simulation to derive the liking score probability density model

Algorithm 1 Monte Carlo simulation to derive the *LS* data for an assessor *s*

```
Generate 10,000 flavors randomly, i.e., $F_b = \{\mathbf{k}^{41}\ldots\mathbf{k}^{10{,}041}\}$ (we use a fixed uniform lattice in the
  experiments, same for all assessors)
for ($\overline{k}^b \in F_b \forall b$) do
    (i) Collect all the predictions from Model ensemble, $\mathcal{M}^s$: $\mathcal{M}^s(\mathbf{k})$
    (ii) Sort the vector $\mathcal{M}^s(\mathbf{k})$
    (iii) Remove the bottom and top 10% of $\mathcal{M}^s(\mathbf{k})$ and call this vector $\overline{R}^{s,j}$
    (iv) Append $\overline{R}^{s,j}$ to $\overline{Y}^s$
end for
Fit the $\overline{Y}^s$ to a Weibull distribution. See Sect. 4.2
```

Exponential, Gaussian or Rayleigh distributions as its shape and scale parameters are varied. For our problem this is a helpful capability as an assessor's liking score follows any one of the three distributions. The derived Weibull distribution is:

$$p(LS;\lambda, r|s) = \begin{cases} \frac{r}{\lambda}\left(\frac{LS}{\lambda}\right)^{r-1} e^{-\left(\frac{LS}{\lambda}\right)^r} & \text{if } LS \geq 0 \\ 0 & \text{if } LS < 0. \end{cases} \tag{7}$$

In addition to steps taken in Sect. 4.1, we map the bootstrapped data to a range of the support of Weibull and the hedonic rating scale i.e., [1, 9]. There are some predictions in the $\overline{Y}^s$ which are below 1 or are above 9. We remove 80% of these predictions as outliers. We assign a liking score of 1 for the remaining 20% of predictions that are less than '1' in the prediction set. We similarly assign the liking score of '9' for the ones that are above 9. We use these 20% in $\overline{Y}^s$ to capture the scores corresponding to the "extremely dislike" and "extremely like" condition. Each plot line of Fig. 12b–d is a liking score density model.

4.3 Segmenting the assessors by propensity to like

Having estimated the data generated from the models for 10,000 flavors in $F_b = \{\mathbf{k}^{(41)}\ldots\mathbf{k}^{(10041)}\}$ using the methods described in the previous section, we can classify the assessors into three different categories (see Fig. 11). We divide the liking score range [1, 9] into three regions as shown in Fig. 12. The assessors are then classified by identifying the region in which the majority (more than 50%) of their probability mass lies (see Algorithm 2). This is accomplished by evaluating the cumulative distribution in each of these regions using:

$$P_{(l_1,l_2]}(LS;\lambda, r|s) = e^{-\left(\frac{l_1}{\lambda}\right)^r} - e^{-\left(\frac{l_2}{\lambda}\right)^r}. \tag{8}$$

We applied our methodology to the dataset of 66 assessors who can be individually modeled with adequate accuracy. 3 assessors were left out due to lack of adequate accuracy in their models. The first segment is the "hard to please" panelists. We have 23 assessors in this segment which is approximately 34.8% of the panel. These assessors have most of their liking scores concentrated between 1 and 3.5 range. We call these “hard-to-please” since low liking scores might imply that they are very choosy in their liking.

The second segment is the segment of “neutral assessors.” These assesors rarely choose the liking scores which are *extremely like* or *extremely dislike*. For most of the sampled flavours they choose somewhere in between and hence the name *neutral*. There are 31 assessors in this segment which is 47% of the total panel.

The final segment of assessors is the “easy to please” assessors. This segment of assesors reports a high liking for most of the flavors presented to them or may report moderate dislike of some. They rarely report “extremely dislike.” There are 12 assesors in this segment which is close to 18% of the total panel.

5 Analytic goal #3 flavor optimization

Using our models derived with ParetoGP, we now use a multi-objective particle swarm optimization detailed in Algorithm 3 and [16] to identify flavors that maximize the liking score function for (a) a single assessor, (b) a segment of assessors. In a particle swarm optimization algorithm, a particle is a candidate solution to the problem which in our case is the concentration levels of the 7 keys (a.k.a. ingredients), given by $\{k_{i1}, k_{i2} \ldots k_{i7}\}$, that compose a flavor. Solutions are randomly initialized. Each particle also stores in its memory the best position in the 7-dimensional space it has visited so far and is denoted by P_i. Initially, both P_i and k_i are the same. Particles are perturbed using a set of equations called velocity and position update equations.

Once a new set of positions are created for each particle, the particles are evaluated for different objective functions. The non-dominated solutions among the candidate solutions are identified using non-dominated comparisons. Two solutions are considered non-dominated if either of them are not better than the other along all the objectives. A particle’s memory is updated with the non dominated solutions

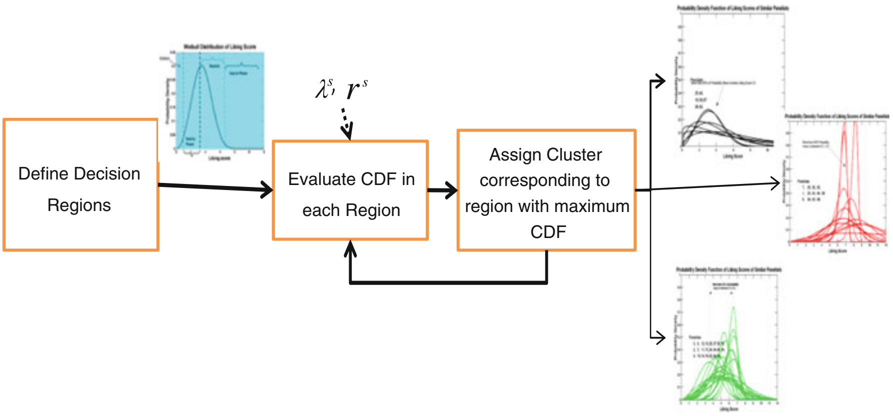

Define Decision Regions
λ^s, r^s
Evaluate CDF in each Region
Assign Cluster corresponding to region with maximum CDF

Fig. 11 Segmenting the assessors

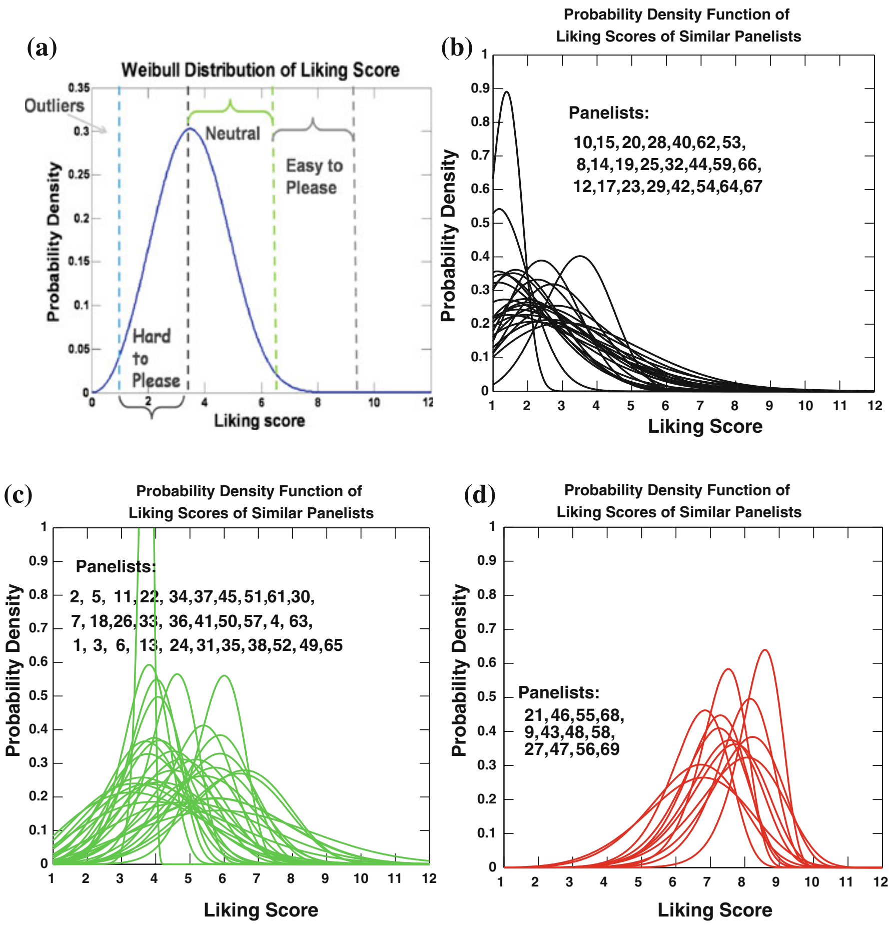


Fig. 12 Liking score density models: **a** decision regions for evaluating cumulative distribution, **b** hard to please assessors, **c** neutral assessors, **d** easy to please assessors

Algorithm 2 Segmenting the assessors

```
for $\forall s \in \{S\}$ do
    1. Calculate $P_{l_1,l_2}$ using estimated $(\lambda^s, r^s)$ for $(l_1, l_2] \rightarrow (1, 3.5]$, (3.5, 6.5] and (6.5, 9.5]
    2. Assign the assessor $s$, to the segment corresponding to the region where he/she has maximum cumulative density
    $s \leftarrow s + 1$
end for
```

and a global best solution is carefully selected. Thus, to resolve multiple objectives the algorithm evolves solutions towards their Pareto front. We chose particle swarm optimization algorithm since it is simple to implement and execute.

5.1 Objectives for single assessor optimization

To find optimal flavors for a single assessor, our objectives are (1) maximize liking score prediction (2) maximize the confidence. We use the median average method for liking score prediction, and use the entropy based measure of confidence described in Eqs. 1 and 2 of Sect. 2.2.2. An instantiation of our method for assessor 39 is shown in Fig. 13.

We experimented with 6 different assessors (each of whom rated the same 36 flavors) named A, B, C, D, E, F. In Fig. 14 the y-axis is, by definition, bounded at 1.0. A liking score with confidence of 1 (entropy of 0) implies that the model predictions for this flavor (when rounded to nearest integer) all agreed. We can observe that this liking score is generally between 7 and 7.5 for assessors E, A, F, at 8 for B and D, and at 6 for C.

5.2 Objective design for a segment of assessors

To identify flavors that are well-liked across the assessors, the first objective is to maximize the mean of the predicted liking scores of different assessors where each liking score prediction is weighted by its confidence:

$$LS^c = \frac{\sum_{s=1}^{N} C_s LS_s}{\sum_{s=1}^{N} C_s} \tag{9}$$

C_s is the confidence of the sth assessor's prediction derived from $\overline{Y}^s$, and LS_s is the liking score derived from $\overline{Y}^s$. The confidence and liking score can be estimated using any of the two methods described in 2.2.2.

We define a second measure called *consistency* that helps in two scenarios to, (a) counter the case in which the mean is driven by a few assessors that like a particular $\mathbf{k}^{(j)}$, and (b) push the flavors to the design space where they are mostly liked. The measure is derived using the LS_s derived from the $\overline{Y}^s$. The measure is given by:

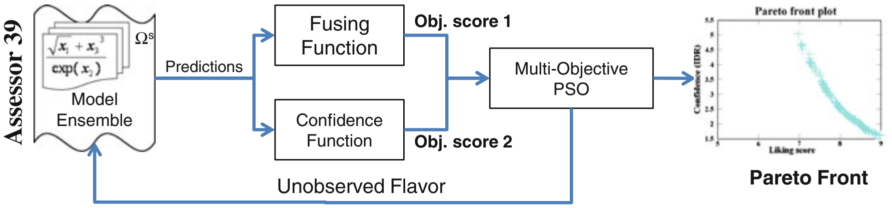


Fig. 13 Multi objective particle swarm optimization for identifying flavors that maximize the liking score as well as the confidence of a single assessor. The ensemble is used to generate multiple liking score predictions and a fused liking score prediction and confidence in this prediction is derived using these multiple predictions. These two form the objectives for the Particle swarm optimization algorithm. It attempts to maximize these two values and generates new unobserved flavors to query the ensembles for predictions. The output is flavors which approximate the true Pareto front in the 2-dimensional trade-off space

Algorithm 3 Multi-objective particle swarm optimizer for flavor optimization

```
$(k_{i_d})$: $d \Leftarrow$ key index, $i \Leftarrow$ particle index, $k \Leftarrow$ Key;
$\overline{k_i} \Leftarrow$ particle, $\overline{P_i} \Leftarrow$ $pbest$, $N \Leftarrow$ number of particles
1. Initialize the particles, $\bar{k}$ randomly in the search space with in the range for each key, $[l_i, u_i]$
2. Initialize $\overline{P_i}$ to be the same, $\forall\ i$
3. Initialize particle velocities denoted by $V_i$ randomly, $\forall\ i$.
4. Initialize parameters of PSO, interia: $\omega = 0.8$, cognitive learning rate: $\psi_1 = 1$, social learning rate:
   $\psi_2 = 1$
5. Evaluate objective function ($o_1$) and objective function ($o_2$) for the given $\overline{\overline{k}}$
6. Randomly initialize the $'gbest'$, $g$
for $t = 1$ to maxiter do
    for $i = 1$ to N do
        for $r = 1$ to d do
            $V_{ir}^{(t+1)} = \omega\ V_{ir}^{(t)} + \psi_1\ (P_{ir}^{(t)} - k_{ir}^{(t)})U[0,1] + \psi_1\ (P_{gr}^{(t)} - k_{ir}^{(t)})U[0,1]$
            $k_{ir}^{(t+1)} = k_{ir}^{(t)} + V_{ir}^{(t+1)}$
            where $U$[0,1] is a uniform random number between [0, 1], and
            $P_{ir}$ is $i^{th}$ particles best position along dimension $r$, $P_{gr}$ is the position of the globaly best
   performing particle along dimension $r$.
        end for
    end for
    for $i = 1$ to number of N do
        Evaluate $o_1$ and $o_2$ for $\overline{k_i}$
    end for
    Update the $\overline{P}$ with the non-dominated solutions
    Identify the $'gbest'$, $g = 5*U[0,1]$
    Store the $pbest$ vector for iteration $t$
    $t \leftarrow t + 1$
end for
7. Output $\overline{P}^{(t)}$
```

$$V = \sqrt{\frac{1}{N}\sum_{s=1}^{N}(LS_s^c - \overline{LS^c})^2}, \tag{10}$$

where $\overline{LS^c}$ is the mean of the liking score prediction of multiple assessors. Equation 10 simply evaluates the standard deviation of the liking scores generated by the multiple assessors. The second objective is to minimize the variability among the liking scores defined by (10). Figure 15 shows the interactions of different components of this design. The ensembles of different assessors are queried with an unobserved flavor. The predictions from the ensemble are then used to evaluate the confidence in the predictions using Eq. 2 in Sect. 2.2.2 and the median average is calculated. These values from multiple assessors are supplied to the "confidence weighted fusion" box where Eqs. 9 and 10 are evaluated. These are objective scores for the optimization.

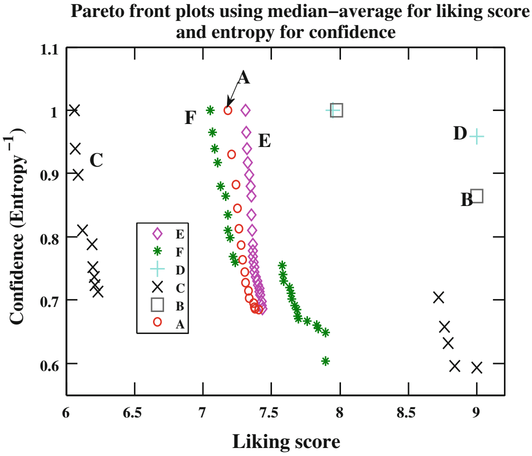


Fig. 14 Flavor optimization achieved by multi objective particle swarm optimization. Evolved Pareto front plots for 6 assessors. Each assessor is optimized independently using his/her own model ensemble

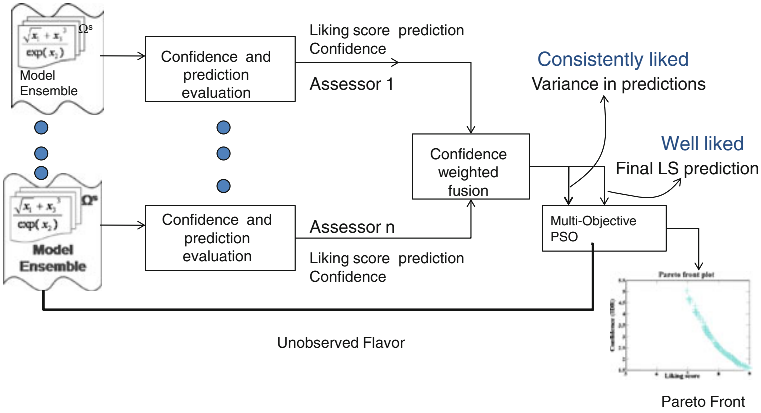


Fig. 15 Multi-objective particle swarm optimization for identifying flavors that are well liked by multiple assessors assembled into a segment by similar ingredient drivers. The ensemble of each assessor is used. For a given flavor composition, the liking score predictions from multiple assessors are weighted using the confidence and combined. The goal of the PSO is to maximize this combined liking score and minimize the variance in the liking score predictions from multiple panelists

5.2.1 Results for the entire panel

We experimented to generate flavors for the entire panel. Figure 16 shows the results achieved when we optimized to generate flavors for the entire panel. We were able to achieve a maximum liking score of 5.9 which is closer to *like slightly* on the hedonic scale. Most of the flavors we found for the entire panel had a liking score of 5 and above. Note that 5 is *neither like nor dislike*. Furthermore, we could not reduce the variance to less than a value of 1. In Fig. 16b, we show the variance in the ingredient concentration levels. It is interesting to see that to achieve consensus among the large number of assessors, we could only vary the concentration levels in a very narrow range. In the next section, we attempt to optimize flavors for a segments of assessors to see if we can improve the overall liking score.

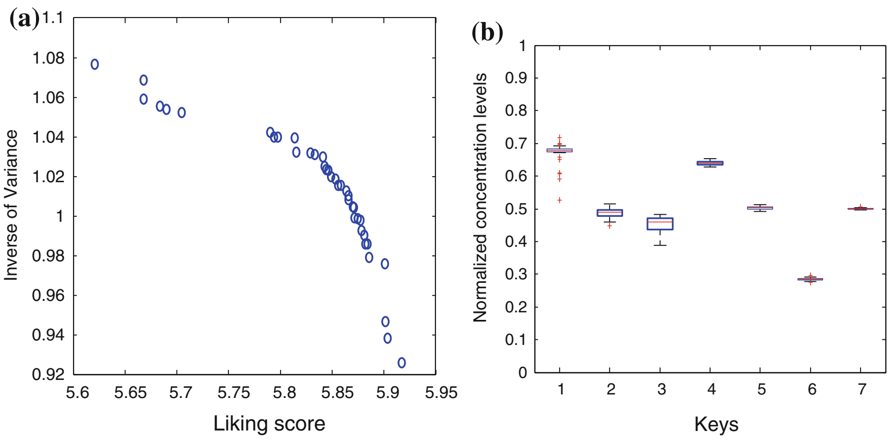


Fig. 16 **a** Pareto front plot for the entire panel. Each point on the plot is a flavor and is evaluated for how *well* it is liked (x-axis) and how consistently it is liked (y-axis). **b** Variation in the normalized concentration levels for different ingredients (aka keys) among the flavors on the Pareto front for different clusters. Box boundaries correspond to the interquartile range

5.2.2 *Results on segments of assessors driven by same ingredients*

We experimented with 6 segments of assessors selected for similar flavor preferences. These 6 segments were identified in Sect. 3. In this experiment each ensemble computes confidence with the entropy measure. Fig. 17 shows the iterative progress of the optimization algorithm for one of the segments. In the first few iterations there are only a few solutions on the evolved Pareto front. With additional iterations the front moves up and rightward and the algorithm identifies flavors with higher liking scores, [6, 6.5], that are lower in variance before finally identifying flavors with confidence-weighted liking scores ranging from [5.5, 7]

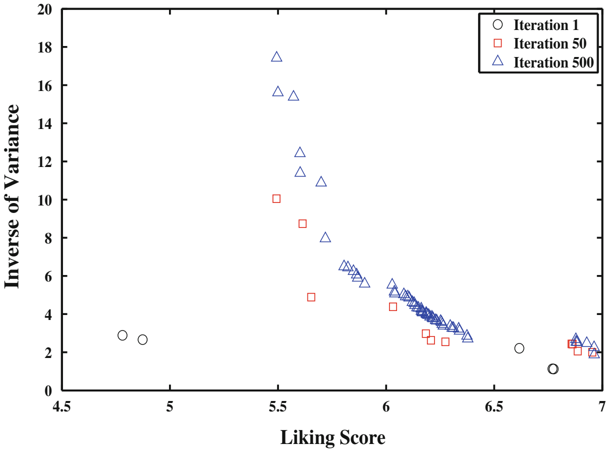


Fig. 17 Progress of multi-objective particle swarm optimization for a group of assessors. The final evolved Pareto front of flavors achieved is also shown

which have low variance (higher inverse of variance). Figure 18 shows the final Pareto fronts evolved for different segments. Each point on the front is a different flavor. For most clusters we are able to find flavors that have a fused liking score of higher than 7, which is "like moderately" on the hedonic scale. However, the variance at this high liking score is approximately 1. For two segments we are able to achieve flavors with liking score of 6 which is equivalent to "like slightly". While a liking score of 6 and above is only in the "slightly-like" and "moderately-like" hedonic range, it is understandable that more highly liked flavors cannot be identified because multiple assessors remain uniquely "individual" even when they are clustered by similar flavor preference. This is still consistent with the expectation that similar flavors will have a range of liking scores because of the assessors preferential similarities. Note that when we attempted to optimize flavors for individual assessors we were able to achieve liking scores in the range of 8–9 (see Fig. 14).

Figure 19 shows the box plots for the flavors on the evolved Pareto front for these 6 segments. The plots demonstrate the variance in key values among these flavors is very low for all the keys, although the liking score varies from [5.8, 7.0]. It appears that limiting the concentration to a certain range for all the keys is necessary to arrive at a reasonable consensus. Overall the results represent the challenges in working with hedonic-based panel aggregates.

6 Conclusions

In this paper, we presented an approach to model a liking score function for assessors employed in sensory evaluation. A GP based symbolic regression methodology was used to generate all plausible models that explain the given data. From these plausible models, a subset was selected that are diverse and that explain

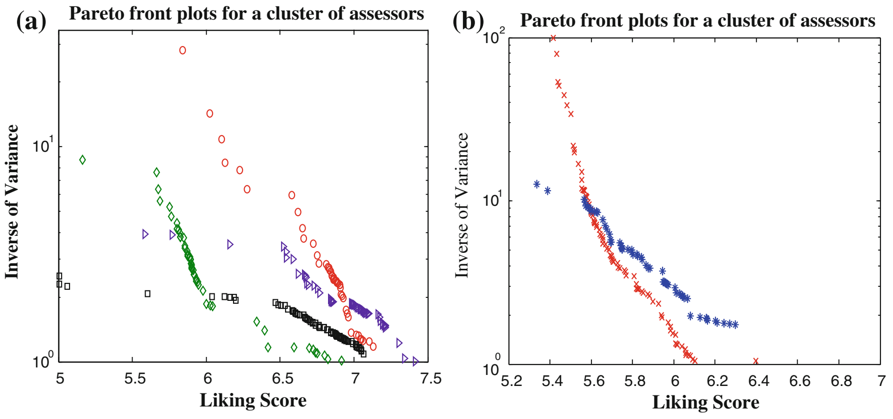


Fig. 18 Pareto front plots achieved for different segments of assessors. Each plot is for a different segment of assessors

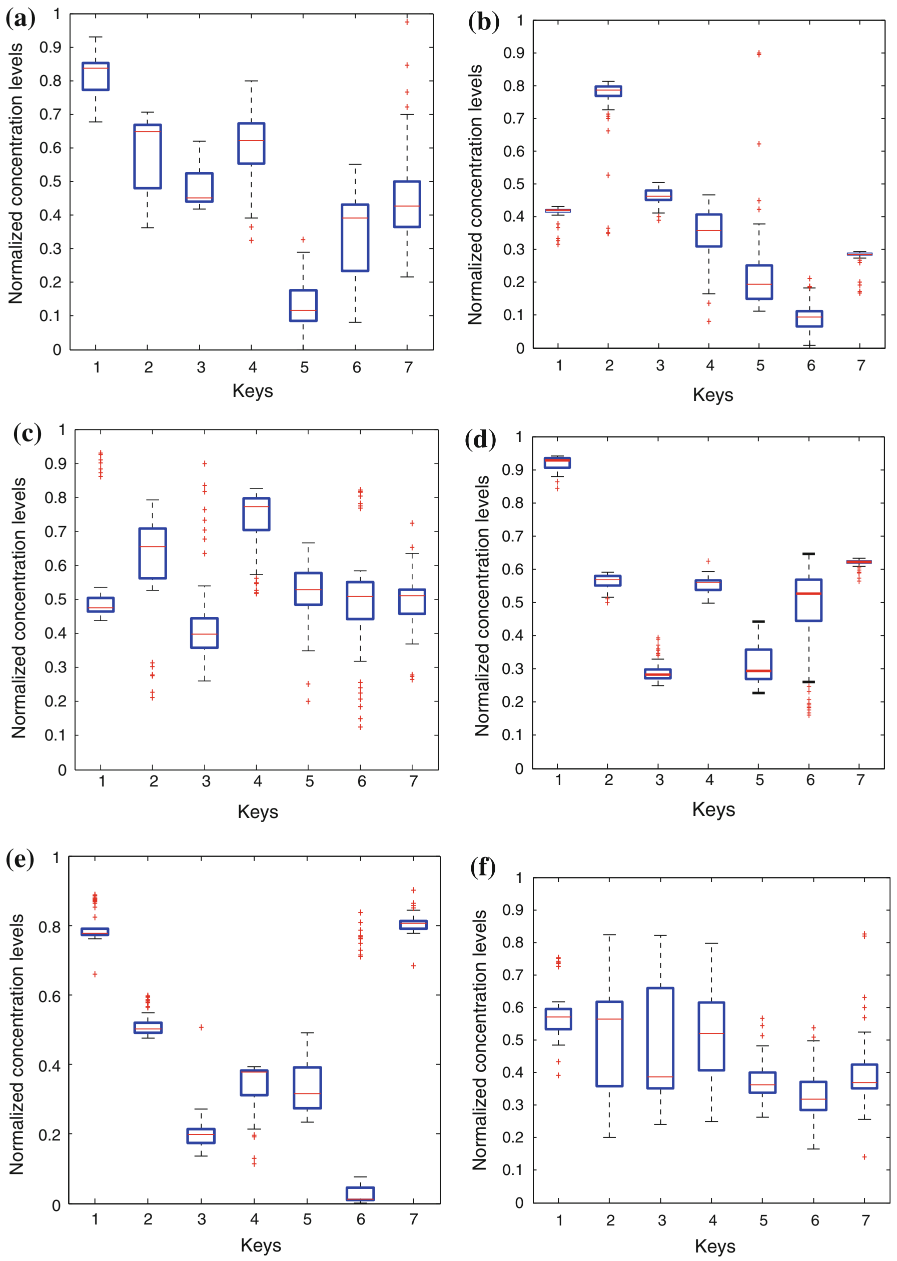


Fig. 19 Variation in the normalized concentration levels for different ingredients (aka keys) among the flavors on the Pareto front for different clusters. Box boundaries correspond to the interquartile range

as many as possible behaviors that the human expert can exhibit when a previously unseen flavor is presented. This formed an ensemble of liking score functions. We demonstrated a new means of knowledge mining with GP methods by conducting data analysis using model ensembles. In this new space, we respect the evidence that the response and explanatory variable relationship differs among assessors and exploit rather than inaccurately average the differences.

The information of variable importance facilitates model similarity clustering on this basis and efficient sensitivity analysis. Ensemble archives of Pareto genetic programming experiments furnish the model sets for variable important analysis that can be driven by presence or fitness weighting. The standard GP approach for variable selection, which analyses variable presence in just one successful solution, does not work in this context because the variable importance statistics of one model are not reliable. We identified 6 distinct groups in our panel using multiple models per assessors. The flavor liking of members of one group are driven by the same ingredients.

Our methodology postpones decision making regarding a *model*, a *prediction* and a *decision boundary* until the very end. For an unseen flavor, *all* the models are consulted and with minor trimming, their predictions are fit to a probability density function. Finally, as the macro-level behavior emerges and more is known about the assessors, decision boundaries can be rationally imposed on this probability space to allow their segmentation. Our approach allowed us to robustly identify segments in the panel based on their propensity to like the flavor space. We conjecture from our results that there are similar potential benefits across any sparse, repeated measure dataset.

We then integrated the ensemble of liking score functions per assessor into two multi-objective algorithms that identify the flavors that maximize (a) an assessor's liking score, or (b) a group of assessors' liking scores. We used a particle swarm optimization algorithm for this. We observed that for an individual assessor, the method identified flavors that were extremely liked according to the hedonic scale. However, when the flavors are optimized for a group of assessors, the maximum liking score that we achieved was around '7' which corresponds to "like moderately" on the hedonic scale.

There are many choices in this knowledge mining process: e.g. what data to aggregate and thresholds such as θ. They should, in general, be made by an expert on the data being modeled. A choice could depend on exogenous goals like market targeting. For example, one could decide to use average ratings of the panel. This would allow them to design flavors that maximize the liking scores according to this information. In this example, strong inter-assessor differences contra-indicate this approach. We observe that all variables are important for modeling the liking score of the entire panel and that there exist fundamental differences in the driving variables among individual panelists. This implies that an approach of designing flavors for the entire panel is likely to generate designs that will be suited to a broad population with a lesser degree of liking. Alternatively, if it is affordable to segment the panel into multiple segments and design flavors that satisfy these smaller segments, each resulting design would have higher likability inside a segment but less suitability across the broad population. The advantage of our approach is that all analysis decisions are postponed until the moment when the decision trade-offs become clear to the domain expert. To understand the trade-offs, the domain experts have access to efficient sensitivity analysis methods which will allow them to finally identify the directions in which the liking scores of panelists are steered by important keys.

We are enthusiastic about the results, primarily because they confirm that genetic programming symbolic regression methodology has evolved into a mature field capable of routinely solving real-world problems. In this case study, genetic programming allowed us to decompose a seemingly unsolvable problem (few samples with multiple responses of high variation) into a sequence of solvable problems generating insights at each step. The most exciting feature of the study is its efficiency. The complete analysis when automated takes a night (or a lot less if multiple cores are available). This, in combination with flavor optimization opens up opportunities for new on-line protocols of flavor design, generating new insights in days instead of months. Additionally, panel segmentation, derived on the basis of liking being influenced by the same ingredients in the same direction, will allow a clearer understanding of the hedonic responses to a product suite. When affordable, it may enable the development of products for particular segments leading to higher consumer satisfaction.

References

1. A. Antos, I. Kontoyiannis, in *Information Theory, 2001. Proceedings. 2001 IEEE International Symposium*. Estimating the entropy of discrete distributions. (IEEE Press, New York, 2001), pp. 45–45. doi:10.1109/ISIT.2001.935908
2. G. Folino, C. Pizzuti, G. Spezzano, GP ensembles for large-scale data classification. IEEE Trans. Evol. Comput. **10**(5), 604–616 (2006)
3. F.D. Francone, L.M. Deschaine, T. Battenhouse, J.J. Warren, in *Late Breaking Papers at the 2004 Genetic and Evolutionary Computation Conference*, ed. by M. Keijzer. Discrimination of unexploded ordnance from clutter using linear genetic programming (Seattle, Washington, USA, 2004). http://www.cs.bham.ac.uk/wbl/biblio/gecco2004/LBP022.pdf
4. Y. Freund, Boosting a weak learning algorithm by majority. Inf. Comput. **121**(2), 256–285 (1995)
5. Y. Freund, H.S. Seung, E. Shamir, N. Tishby, in *Advances in Neural Information Processing Systems*, 5th edn. Information, prediction, and query by committee (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993), pp. 483–490 [NIPS Conference]
6. R.J. Gilbert, R. Goodacre, B. Shann, D.B. Kell, J. Taylor, J.J. Rowland, in *Genetic Programming 1998: Proceedings of the Third Annual Conference*, ed by J.R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D.B. Fogel, M.H. Garzon, D.E. Goldberg, H. Iba, R. Riolo. Genetic programming-based variable selection for high-dimensional data (Morgan Kaufmann, University of Wisconsin, Madison, Wisconsin, USA, 1998), pp. 109–115
7. L.K. Hansen, P. Salamon. Neural network ensembles. IEEE Trans. Pattern Anal. Mach. Intell. **12**(10), 993–1001 (1990). doi:http://dx.doi.org/10.1109/34.58871
8. H. Iba, in *Proceedings of the Genetic and Evolutionary Computation Conference, vol 2*, ed. by W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, R.E. Smith. Bagging, boosting, and bloating in genetic programming (Morgan Kaufmann, Orlando Florida USA, 1999) pp. 1053–1060.
9. M. Keijzer, in *Proceedings of the 6th European Conference on Genetic programming, EuroGP'03*. Improving symbolic regression with interval arithmetic and linear scaling (Springer, Berlin, Heidelberg, 2003), pp. 70–82. http://portal.acm.org/citation.cfm?id=1762668.1762676
10. M. Keijzer, Scaled symbolic regression. Genetic Program. Evol. Mach. **5**(3), 259–269 (2004). doi:10.1023/B:GENP.0000030195.77571.f9
11. M.F. Korns, in *Genetic Programming Theory and Practice IV, Genetic and Evolutionary Computation*, vol. 5, chap. 16, ed. by R.L. Riolo, T. Soule, B. Worzel. Large-scale, time-constrained symbolic regression (Springer, Ann Arbor, 2006) pp. 299–314.

12. M. Kotanchek, G. Smits, E. Vladislavleva, in *Genetic Programming Theory and Practice V, Genetic and Evolutionary Computation*, chap. 12, ed. by R.L. Riolo, T. Soule, B. Worzel. Trustable symoblic regression models (Springer, Ann Arbor, 2007) pp. 203–222.
13. J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection* (MIT Press, Cambridge, MA, USA, 1992)
14. A. Krogh, J. Vedelsby, in *Advances in Neural Information Processing Systems*, vol. 7, ed. by G. Tesauro, D. Touretzky, T. Leen. Neural network ensembles, cross validation, and active learning (The MIT Press, Cambridge, MA, USA, 1995) pp. 231–238.
15. J. Landry, L.D. Kosta, T. Bernier, Discriminant feature selection by genetic programming: Towards a domain independent multi-class object detection system. J. Syst. Cybernet. Inform. **3**(1), 76–81 (2006)
16. X. Li, in *Lecture Notes in Computer Science*, vol. 2723/2003. A non-dominated sorting particle swarm optimizer for multiobjective optimization (Springer, Berlin, 2003), pp. 37–48.
17. Y. Liu, X. Yao, in *PPSN VII: Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*. Learning and evolution by minimization of mutual information (Springer, London, UK, 2002), pp. 495–504
18. Y. Liu, X. Yao, T. Higuchi, Evolutionary ensembles with negative correlation learning. IEEE Trans. Evol. Comput. **4**(4), 380–387 (2000)
19. H.R. Moskowitz, R. Bernstein, Variability in hedonics: Indications of world-wide sensory and cognitive preference segmentation. J. Sens. Stud. **15**(3), 263–284 (2000)
20. S. Mukherjee, V. Vapnik, in *NIPS 12*. Multivariate density estimation: a support vector machine approach (1999), pp. 1–8
21. K. Neshatian, M. Zhang, M. Johnston, in *Australian Conference on Artificial Intelligence, Lecture Notes in Computer Science*, vol. 4830, ed. by M.A. Orgun, J. Thornton. Feature construction and dimension reduction using genetic programming (Springer, Berlin, 2007), pp. 160–170
22. G. Paris, D. Robilliard, C. Fonlupt, in *Artificial Evolution 5th International Conference, Evolution Artificielle, EA 2001, LNCS*, vol. 2310, ed. by P. Collet, C. Fonlupt, J.K. Hao, E. Lutton, M. Schoenauer. Applying boosting techniques to genetic programming (Springer, Creusot France, 2001), pp. 267–278.
23. R. Poli, in *Evolutionary Computing, 1143*, ed. by T.C. Fogarty. Genetic programming for feature detection and image segmentation (Springer, University of Sussex, UK, 1996), pp. 110–125.
24. R.E. Schapire, The strength of weak learnability. Mach. Learn. **5**(2), 197–227 (1990)
25. M.D. Schmidt, H. Lipson, Coevolution of fitness predictors. IEEE Trans. Evol. Comput. **12**(6), 736–749 (2008)
26. J.R. Sherrah, R.E. Bogner, A. Bouzerdoum, in *Genetic Programming 1997: Proceedings of the Second Annual Conference*, ed. by J.R. Koza, K. Deb, M. Dorigo, D.B. Fogel, M. Garzon, H. Iba, R.L. Riolo. The evolutionary pre-processor: Automatic feature extraction for supervised classification using genetic programming (Morgan Kaufmann, Stanford University, CA, USA, 1997), pp. 304–312.
27. G. Smits, A. Kordon, K. Vladislavleva, E. Jordaan, M. Kotanchek, in *Genetic Programming Theory and Practice III, Genetic Programming*, vol. 9, chap. 6, ed. by T. Yu, R.L. Riolo, B. Worzel. Variable selection in industrial datasets using pareto genetic programming (Springer, Ann Arbor, 2005), pp. 79–92.
28. G. Smits, M. Kotanchek, in *Genetic Programming Theory and Practice II*, chap. 17, ed. by U.M. O'Reilly, T. Yu, R.L. Riolo, B. Worzel. Pareto-front exploitation in symbolic regression (Springer, Ann Arbor, 2004), pp. 283–299.
29. P. Sun, X. Yao, in *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*. Boosting kernel models for regression (IEEE Computer Society, Washington, DC, USA 2006), pp. 583–591
30. J.S. Taylor, A. Dolia, in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, ed. by N. Lawrence. A framework for probability density estimation. Journal of Machine Learning Research (2007), pp. 468–475
31. K. Veeramachaneni, K. Vladislavleva, M. Burland, J. Parcon, U.M. O'Reilly, in *GECCO '10: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, ed. by J. Branke, M. Pelikan, E. Alba, D.V. Arnold, J. Bongard, A. Brabazon, J. Branke, M.V. Butz, J. Clune, M. Cohen, K. Deb, A.P. Engelbrecht, N. Krasnogor, J.F. Miller, M. O'Neill, K. Sastry, D. Thierens, J. van Hemert, L. Vanneschi, C. Witt. Evolutionary optimization of flavors (ACM, Portland, Oregon, USA, 2010), pp. 1291–1298

32. E. Vladislavleva, *Model-based problem solving through symbolic regression via pareto genetic programming*. Ph.D. thesis (Tilburg University, Tilburg, the Netherlands, 2008). http://arno.uvt.nl/show.cgi?fid=80764
33. K. Vladislavleva, K. Veeramachaneni, M. Burland, J. Parcon, U.M. O'Reilly, in *GECCO '10: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, ed. by J. Branke, M. Pelikan, E. Alba, D.V. Arnold, J. Bongard, A. Brabazon, J. Branke, M.V. Butz, J. Clune, M. Cohen, K. Deb, A.P. Engelbrecht, N. Krasnogor, J.F. Miller, M. O'Neill, K. Sastry, D. Thierens, J. Hemert, L. Vanneschi, C. Witt. Knowledge mining with genetic programming methods for variable selection in flavor design (ACM, Portland, Oregon, USA, 2010), pp. 941–948.
34. K. Vladislavleva, K. Veeramachaneni, U.M. O'Reilly, in *Proceedings of the 13th European Conference on Genetic Programming, EuroGP 2010, LNCS*, vol. 6021, ed. by A.I. Esparcia-Alcazar, A. Ekart, S. Silva, S. Dignum, A.S. Uyar. Learning a lot from only a little: Genetic programming for panel segmentation on sparse sensory evaluation data (Springer, Istanbul, 2010), pp. 244–255.
35. D.H. Wolpert, Stacked generalization. Neural Netw. **5**(2), 241–259 (1992)
36. J. Yu, J. Yu, A.A. Almal, S.M. Dhanasekaran, D. Ghosh, W.P. Worzel, A.M. Chinnaiyan, Feature selection and molecular classification of cancer using genetic programming. Neoplasia **9**(4), 292–303 (2007)