Mach Learn (2017) 106:1599–1620
DOI 10.1007/s10994-017-5651-7

Knowledge elicitation via sequential probabilistic inference for high-dimensional prediction

Pedram Daee[1] · **Tomi Peltola**[1] · **Marta Soare**[1] · **Samuel Kaski**[1]

Received: 5 February 2017 / Accepted: 20 June 2017 / Published online: 12 July 2017

Abstract Prediction in a small-sized sample with a large number of covariates, the "small n, large p" problem, is challenging. This setting is encountered in multiple applications, such as in precision medicine, where obtaining additional data can be extremely costly or even impossible, and extensive research effort has recently been dedicated to finding principled solutions for accurate prediction. However, a valuable source of additional information, domain experts, has not yet been efficiently exploited. We formulate knowledge elicitation generally as a probabilistic inference process, where expert knowledge is sequentially queried to improve predictions. In the specific case of sparse linear regression, where we assume the expert has knowledge about the relevance of the covariates, or of values of the regression coefficients, we propose an algorithm and computational approximation for fast and efficient interaction, which sequentially identifies the most informative features on which to query expert knowledge. Evaluations of the proposed method in experiments with simulated and real users show improved prediction accuracy already with a small effort from the expert.

Keywords Bayesian methods · Experimental design · Human-to-machine transfer learning · Interactive machine learning · Statistics in high dimensions

1 Introduction

Datasets with a small number of samples n and a large number of variables p are nowadays common. Statistical learning, for example regression, in these kinds of problems is ill-posed,

Pedram Daee, Tomi Peltola and Marta Soare have contributed equally to this work.

Editors: Kurt Driessens, Dragi Kocev, Marko Robnik-Šikonja, Myra Spiliopoulou.

✉ Samuel Kaski
Samuel.Kaski@aalto.fi

[1] Helsinki Institute for Information Technology HIIT and Department of Computer Science, Aalto University, Konemiehentie 2, Espoo, Finland

and it is known that statistical methods have limits in how low in sample size they can go (Donoho and Tanner 2009). A lot of recent research in statistical methodology has focused on finding different kinds of solutions via well-motivated trade-offs in model flexibility and bias. These include strong assumptions about the model family, such as linearity, low rank, sparsity, meta-analysis and transfer learning from related datasets, efficient collection of new data via active learning, and, less prominently, prior elicitation.

There is, however, a certain disconnect between the development of state-of-the-art statistical methods and their application in challenging data analysis problems. Many applications have significant amounts of previous knowledge to incorporate into the analysis, but this is often unstructured and tacit. Building it into the analysis would require tailoring the model and eliciting the knowledge in a suitable format for the analysis, which would be burdensome for both experts in statistical methods and experts in the problem domain. More commonly, new methods are developed to work well in some broad class of problems and data, and domain experts use default approaches and apply their previous knowledge post-hoc for interpretation and discussion. Even when experts in both fields are directly collaborating, the feedback loop between the method development and application is often slow.

We propose to directly integrate the expert into the modelling loop by formulating knowledge elicitation as a probabilistic inference process. We study a specific case of sparse linear regression with the aim of solving prediction problems where the number of available samples (*training data*) is insufficient for statistically accurate prediction. A core characteristic of the formulation is that it adapts to the feedback obtained from the expert and it sequentially integrates every piece of information before deciding on the next query for the expert. In particular, the predictive regression model and the feedback model are subsumed into a joint probabilistic model, the related uncertainties of which can be sequentially updated, after each expert interaction. The query selection is then naturally formulated as an *experimental design problem*, aiming at maximizing the information gained from the expert in a limited number of queries. This efficiently reduces the burden on the expert, since the most informative queries will be asked first, redundant queries can be avoided via the sequential updating, and the expert's effort is not wasted on aspects of the model, where the training data already provides strong information. Notably, compared to pure prior elicitation, the reduction in the number of interactions makes knowledge elicitation for high-dimensional parameters (such as the regression weights in large p models) practicable. This paper contributes to the interactive machine learning literature, focusing on probabilistic modelling for interactively eliciting and incorporating expert knowledge. The other important aspect, of designing the user interfaces, will be focused on in future work.

1.1 Contributions and outline

After discussing related work (Sect. 2), we rigorously formulate expert knowledge elicitation as a probabilistic inference process (Sect. 3). We study a specific case of sparse linear regression, and in particular, consider cases where the expert has knowledge about the relevance of the covariates or the values of the regression coefficients (Sect. 4). We present an algorithm for efficient interactive sequential knowledge elicitation for high-dimensional models that makes knowledge elicitation in "small n, large p" problems feasible (Sect. 4.3). We describe an efficient computational approach using deterministic posterior approximations allowing real-time interaction for the sparse linear regression case (Sect. 4.4). Simulation studies are presented to demonstrate the performance and to gain insight into the behaviour of the approach (Sect. 5). Finally, we demonstrate that real users are able to improve the predictive performance of sparse linear regression in a proof-of-concept experiment (Sect. 5.4).

2 Related work

The problem we study relates to several topics studied in the literature, either by the method, goal, or by the considered setting. In this section, we highlight the main connections.

2.1 Interactive learning

Interactive machine learning includes a variety of ways to employ user's knowledge, preferences, and human cognition to enhance statistical learning (Ware et al. 2001; Fails and Olsen 2003; Amershi 2012; Robert et al. 2016). These methods have been used successfully in several applications, such as learning user intent (Ruotsalo et al. 2014) and preferential clustering. For instance, the semi-supervised clustering method in Lu and Leen (2007); Balcan and Blum (2008) uses feedback on pairs of items that should or should not be in the same cluster, to learn user preferences. In addition to the differences coming from the learning task, one notable contrast between these works and our method is that their aim is to identify user preferences or opinions, whereas our goal is to use expert knowledge as an additional source of information for an improved prediction model, by integrating it with the knowledge coming from the (small n) data. As a probabilistic approach, our work relates to Cano et al. (2011) and House et al. (2015), where expert feedback is used for improved learning of Bayesian networks and for visual data exploration, respectively. In Sect. 3.3, we show how these works can be seen as instances of the general approach we propose.

2.2 Active learning and experimental design

The method we propose for efficiently using expert feedback is related to active learning techniques [see, for instance, Settles (2010)], where the algorithms actively select the most informative data points to be used in prediction tasks. Our method similarly queries the expert for information with the goal of maximising the information gain from each feedback and thus learning more accurate models with less feedback. The same definition of efficiency with respect to the use of samples also connects our work with experimental design techniques (Kiefer and Wolfowitz 1959; Chaloner and Verdinelli 1995), which considers designing informative experiments for collecting data in settings with limited resources. This has been recently considered in sparse linear settings by Seeger (2008), Hernández-Lobato et al. (2013), Ravi et al. (2016) and is important in many application fields (Busby 2009; Ferreira and Gamerman 2015; Martino et al. 2017). Our task, however, is different as we do not aim at collecting new data samples, but the additional information comes from a different source, the expert, with its respective bias and uncertainty. Indeed, our method will be most useful in cases where obtaining additional input samples would be too expensive. Active learning has also been used to query feature labels rather than new data points in natural language processing applications (Druck et al. 2009; Settles 2011; Raghavan et al. 2006). The difference between these works and our paper comes from the task (they consider classification rather than prediction), the model assumptions (they did not consider sparse models which are suitable for "small n, large p" settings), and the feedback type.

2.3 Prior elicitation

Many works have studied approaches for efficient elicitation of expert knowledge. Typically, the goal of prior elicitation techniques (O'Hagan et al. 2006) is to use expert knowledge to construct a prior distribution for Bayesian data analysis and restrict the range of parameters to be later used in learning models. In particular, Garthwaite and Dickey (1988), Kadane et al. (1980) study methods of quantifying subjective opinion about the coefficients of linear regression models through the assessment of credible intervals. These elicitation methods were shown to obtain prior distributions that represent well the expert's opinion. Similar elicitation methods have been employed in a wide range of application settings, in which expert knowledge elicitation techniques have been studied, for instance *preference model elicitation* (Azari Soufiani et al. 2013), or *software development processes* (Hickey and Davis 2003). Our approach goes beyond pure prior elicitation as the training data is used to facilitate efficient user interaction. The concurrent works by Micallef et al. (2017), Afrabandpey et al. (2016), and Soare et al. (2016) also use elicited expert knowledge for improving prediction. Micallef et al. (2017) use a separate multi-armed bandit model to facilitate the elicitation and directly modify the regression model priors, Soare et al. (2016) consider predictions for a target patient in a simulated user setting, while Afrabandpey et al. (2016) consider pairwise similarity feedback on features and uses those to create a better covariance matrix for the prior of coefficients of ridge regression. Contrary to our work, these works do not formulate an encompassing probabilistic model for the expert knowledge and prediction, a crucial feature of our approach. Moreover, they do not consider a sparse regression model and the type of feedback is different.

3 Knowledge elicitation as interactive probabilistic modelling

In the following, we formulate expert knowledge elicitation as a probabilistic inference process.

3.1 Key components

Let y and x denote the outputs (target variables) and inputs (covariates), and θ and ϕ_y the model parameters. Let f encode input (*feedback*) from the user, presumably a domain expert, and ϕ_f be model parameters related to the user input. We identify the following key components:

1. An observation model $p(y \mid x, \theta, \phi_y)$ for y.
2. A feedback model $p(f \mid \theta, \phi_f)$ for the expert's knowledge.
3. A prior model $p(\theta, \phi_y, \phi_f)$ completing the hierarchical model description.
4. A query algorithm and user interface that facilitate gathering f iteratively from the expert.
5. Update process of the model after user interaction.

The observation model can be any appropriate probability model. It is assumed that there is some parameter θ, possibly high-dimensional, that the expert has knowledge about. The expert's knowledge is encoded as (possibly partial) feedback f that is transformed into information about θ via the feedback model. Of course, there could be a more complex hierarchy tying the observation and feedback models, and the feedback model can also be used to model more user-centric issues, such as the quality of or uncertainty in the knowledge or user's interests.

The feedback model, together with a query algorithm and a user interface, is used to facilitate an efficient interaction with the expert. The term "query algorithm" is used here in a broad sense to describe any mechanism that is used to intelligently guide the user's focus in providing feedback to the system. This enables considering a high-dimensional f without overwhelming the expert as the most useful feedbacks can be queried first. Crucially, this enables going beyond pure prior elicitation as the observed data can be used to inform the queries via the dependence of the feedback and observation models. For example, the queries can be formed as solutions to decision or experimental design tasks that maximize the expected information gain from the interaction.

Finally, as the expert feedback is modelled as additional data, Bayes theorem can be used to sequentially update the model during the interaction. For real-time interaction, this may present a challenge as computation in probabilistic models can be demanding. It is known that slow computation can impair effective interaction (Fails and Olsen 2003) and, thus, efficient computational approaches are important.

3.2 Overall interaction scheme

Figure 1 depicts the information flow. First, the posterior distribution $p(\theta|\mathcal{D})$ given the observations $\mathcal{D} = \{(y_i, x_i) : i = 1, \ldots, n\}$ is computed. Then, the expert is queried iteratively for feedback via the user interface and the query algorithm. The feedback is used to sequentially update the posterior distribution. The query algorithm has access to the latest beliefs about the model parameters and the predicted user behaviour, that is, the posterior predictive distribution of f, $p(f_{t+1} \mid \mathcal{D}, f_1, \ldots, f_t)$, where the f_j are possibly partial observations of f. Based on this information, the query algorithm chooses the most informative queries, or more generally interactions in the user interface.

3.3 Examples

The goal in this paper is to use the interaction scheme (Fig. 1) to help solve prediction problems in the "small n, large p" setting. The approach as described above is, however, more general and applicable to other problems as well. We briefly describe two earlier works that can be seen as instances of it.

Cano et al. (2011) present a method for integrating expert knowledge into learning of Bayesian networks. The observation model is a multinomial Bayesian network with Dirichlet priors. The expert provides answers to queries about the presence or absence of edges in the network and the feedback model assumes the answers to be correct with some probability. Which edge to query about next is selected by maximising the information gain with regard to the inclusion probability of the edges. Monte Carlo algorithms are used for the computation.

House et al. (2015) present a framework for interactive visual data exploration. They describe two observation models, principal component analysis and multidimensional scal-

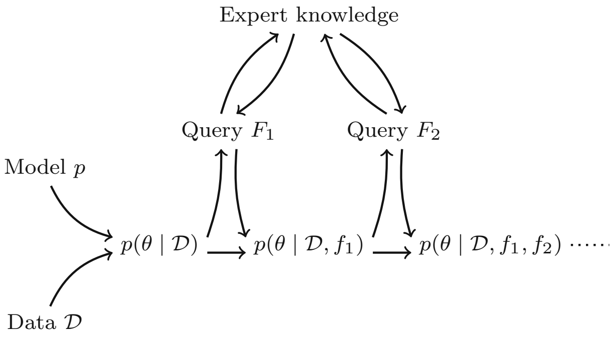

Expert knowledge
Query F_1
Query F_2
Model p
$p(\theta \mid \mathcal{D})$
$p(\theta \mid \mathcal{D}, f_1)$
$p(\theta \mid \mathcal{D}, f_1, f_2)$ ……
Data $\mathcal{D}$

Fig. 1 Information flow. The parameters ϕ_y and ϕ_f are omitted from the posterior distributions for brevity

ing, that are used for dimensionality reduction to visualise the observations in a two dimensional plot. They do not have a query algorithm, but their user interface allows moving points in a low-dimensional plot closer or further apart. A feedback model then transforms the feedback into appropriate changes in the parameters shared with the observation model to allow exploration of different aspects of the data. Their model affords closed form updates.

4 Feedback models and query algorithm for sparse linear regression

We next introduce the knowledge elicitation approach for sparse linear regression.

4.1 Sparse regression model

Let $\boldsymbol{y} \in \mathbb{R}^n$ be the observed output values and $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ the matrix of covariate values. We assume a Gaussian observation model for the regression[1]:

$$\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{X}\boldsymbol{w}, \sigma^2 \mathbf{I}),$$

where $\boldsymbol{w} \in \mathbb{R}^m$ are the regression coefficients and σ^2 is the residual variance. We assume a gamma prior on the inverse of σ^2 (that is, residual precision; or equivalently, inverse-gamma prior on the variance):

$$\sigma^{-2} \sim \mathrm{Gamma}(\alpha_\sigma, \beta_\sigma).$$

Other tasks, such as classification, could be accommodated by changing the assumption about the observation model to an appropriate generalized linear model.

A sparsity-inducing spike-and-slab prior (George and McCulloch 1993) is put on the regression coefficients $\boldsymbol{w}$:

$$\begin{aligned} w_j &\sim \gamma_j \, \mathrm{N}(0, \psi^2) + (1 - \gamma_j)\delta_0, && j = 1, \ldots, m, \\ \gamma_j &\sim \mathrm{Bernoulli}(\rho), && j = 1, \ldots, m, \end{aligned}$$

where the γ_j are latent binary variables indicating inclusion or exclusion of the covariates in the regression. For covariates *included* in the model, $\gamma_j = 1$ and w_j is drawn from a zero-mean Gaussian distribution with variance ψ^2. For covariates *excluded* from the model $\gamma_j = 0$ and $w_j = 0$ via the Dirac delta point mass at zero, δ_0. We will also refer to covariates included in the model as *relevant* for the regression and covariates excluded as *not-relevant*. The prior inclusion probability of the covariates ρ controls the expected number of covariates included (i.e., the sparsity of model). The α_σ, β_σ, ψ^2, and ρ are fixed hyperparameters.

After observing a *training dataset* $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$, the posterior distribution of the regression model is computed using the Bayes theorem[2] as

$$p(\boldsymbol{w}, \boldsymbol{\gamma}, \sigma^2 \mid \mathcal{D}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}, \sigma^2) p(\sigma^2) p(\boldsymbol{w} \mid \boldsymbol{\gamma}) p(\boldsymbol{\gamma})}{p(\boldsymbol{y} \mid \boldsymbol{X})}.$$

The predictive distribution for a new data point $\tilde{\boldsymbol{x}}$ is

$$p(\tilde{y} \mid \tilde{\boldsymbol{x}}, \mathcal{D}) = \int p(\tilde{y} \mid \tilde{\boldsymbol{x}}, \boldsymbol{w}, \sigma^2) p(\boldsymbol{w}, \sigma^2 \mid \mathcal{D}) d(\boldsymbol{w}, \sigma^2).$$

[1] The parametrizations of the distributions follow Gelman et al. (2014, Appendix A).

[2] We use the generic $p(\cdot)$ notation, where it is understood that the parameters identify the particular distribution. See, e.g., Gelman et al. (2014, p. 6).

4.2 Feedback models

Feedback models are used to incorporate the expert knowledge into the regression model. They extend the regression model such that both parts are subsumed into a single probabilistic model, where information flows naturally between the parts (following the Bayesian modelling paradigm).

The feedback model is naturally dependent on the available type of expert knowledge in the targeted application. For our formulation we consider two simple and natural feedback models encoding knowledge about the individual regression coefficients:

- Expert has knowledge about the value of the coefficient ($f_{w,j} \in \mathbb{R}$):

  $$f_{w,j} \sim \mathrm{N}(w_j, \omega^2). \tag{1}$$

  We do not assume that the expert can give exact values for the coefficients, but that there is some uncertainty in the expert's estimates, the amount of which is controlled by the variance ω^2. The smaller the ω^2, the more accurate the knowledge is assumed a priori, and the stronger the change in the model in response to the feedback.
- Expert has knowledge about the relevance of coefficient ($f_{\gamma,j} \in \{0, 1\}$ for *not-relevant*, *relevant*):

  $$f_{\gamma,j} \sim \gamma_j \,\mathrm{Bernoulli}(\pi) + (1 - \gamma_j)\,\mathrm{Bernoulli}(1 - \pi). \tag{2}$$

  Here, π models uncertainty of the knowledge (akin to ω^2 above). *A priori*, the expert feedback is expected to be 1 (*relevant*) with probability π if $\gamma_j = 1$ (the covariate is *included* in the regression). In other words, π can be thought of as the probability of the expert being correct in his or her feedback relative to the state of the covariate inclusion γ_j.

The posterior distribution after getting a set $\mathcal{F} = (\boldsymbol{f}_{\boldsymbol{w}}, \boldsymbol{f}_\gamma)$ of expert feedback, where, with some abuse of notation, $\boldsymbol{f}_{\boldsymbol{w}}$ and $\boldsymbol{f}_\gamma$ collect the given feedback (not assumed to be necessarily available for all covariates) is

$$p(\boldsymbol{w}, \boldsymbol{\gamma}, \sigma^2 \mid \mathcal{D}, \mathcal{F}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}, \sigma^2)p(\sigma^2)p(\boldsymbol{w} \mid \boldsymbol{\gamma})p(\boldsymbol{\gamma})p(\boldsymbol{f}_\gamma \mid \boldsymbol{\gamma})p(\boldsymbol{f}_{\boldsymbol{w}} \mid \boldsymbol{w})}{p(\boldsymbol{y}, \boldsymbol{f}_\gamma, \boldsymbol{f}_{\boldsymbol{w}} \mid \boldsymbol{X})}.$$

The predictive distribution follows as before but with the new posterior distribution. The posterior distribution can be updated sequentially when more feedback is collected, as explained in Sect. 3. In the following, we will assume only one type of feedback at a time in the modelling, but this could be extended to multiple simultaneous types.

To relate the work to prior elicitation, we can think of first updating the posterior distribution with only the expert feedback $\mathcal{F}$, and then, using the posterior as the prior distribution for updating the model with the training set observations $\mathcal{D}$. However, incorporating the expert knowledge through the feedback models (instead of directly as priors) is crucial for the sequential knowledge elicitation, as it allows computing the predictive distributions for the feedback. Moreover, the sequential elicitation exploits the information in the training data to facilitate an efficient elicitation process for high-dimensional parameters.

4.3 Query algorithm

Our aim is to improve prediction. Thus, the user interaction should focus on aspects of the model (here, predictive covariates or features; we use the terms interchangeably) that would be most beneficial towards this goal. We use the query algorithm to rank the features for choosing which one to ask feedback about next. The ranking is formulated as a Bayesian experimental design task (Chaloner and Verdinelli 1995) given all information collected thus far.

The utility function used for scoring the alternative queries can be tailored according to the application. In this paper we use information gain in the prediction, defined as the Kullback–Leibler divergence (KL) between the current posterior predictive distribution $p(\tilde{y} \mid \tilde{\boldsymbol{x}}, \mathcal{D}, \mathcal{F})$ and the posterior predictive distribution with the new feedback f_j, $p(\tilde{y} \mid \tilde{\boldsymbol{x}}, \mathcal{D}, \mathcal{F}, f_j)$. The bigger the information gain, the bigger impact the new feedback has on the predictive distribution. More specifically, the feature j^* that maximizes the expected information gain is chosen next:

$$j^* = \arg\max_{j \notin \mathcal{F}} \mathbb{E}_{p(\tilde{f}_j \mid \mathcal{D}, \mathcal{F})} \left[\sum_i \mathrm{KL}[p(\tilde{y} \mid \boldsymbol{x}_i, \mathcal{D}, \mathcal{F}, \tilde{f}_j) \parallel p(\tilde{y} \mid \boldsymbol{x}_i, \mathcal{D}, \mathcal{F})] \right],$$

where j indexes the features, $\mathcal{F}$ is the set of feedbacks that have already been given, and the summation over i goes over the training dataset. Since the feedback itself will only be observed after querying the expert, we take the expectation over the posterior predictive distribution of the feedback $p(\tilde{f}_j \mid \mathcal{D}, \mathcal{F})$. More details about the Bayesian experimental design are provided in Appendix B.

As a side note, if the predictive distribution of y was Gaussian, the problem would be simple. The expected information gain would be independent of y and the actual values of the feedbacks (when feedback is on values of the regression coefficients), and would only depend on the x and on which features the feedback was given (Seeger 2008). The sparsity-promoting prior, however, makes the problem non-trivial.

4.4 Computation

The model does not have a closed-form posterior distribution, predictive distribution, or solution to the information gain maximization problem. To achieve fast computation, we use deterministic posterior approximations. Expectation propagation (Minka 2001) is used to approximate the spike-and-slab prior (Hernández-Lobato et al. 2015) and the feedback models, and variational Bayes (e.g., Bishop 2006, Chapter 10) is used to approximate the residual variance σ^2. The form of the posterior approximation for the regression coefficients $\boldsymbol{w}$ is Gaussian. The posterior predictive distribution for y is also approximated as Gaussian. Details are provided in Appendix A.

Expectation propagation has been found to provide good estimates of uncertainty, which is important in experimental design (Seeger 2008; Hernández-Lobato et al. 2013; Hernández-Lobato et al. 2015). In evaluating the expected information gain for a large number of candidate features, running the approximation iterations to full convergence for each is too slow, however. We follow the approach of Seeger (2008), Hernández-Lobato et al. (2013) in computing only a single iteration of updates on the essential parameters for each candidate. We show in the results that this already provides a good performance for the query algorithm in comparison to random queries. Details on the computations are provided in Appendix B.

Markov chain Monte Carlo (MCMC) methods could alternatively be used for computation, but sampling efficiently over the binary space of size 2^m for $\boldsymbol{\gamma}$ can be difficult (Peltola et al. 2012; Schäfer and Chopin 2013) and naive approaches would be slow. Sequential Monte Carlo (SMC) algorithms (Del Moral et al. 2006; Schäfer and Chopin 2013) are designed to move between distributions that change in steps and could provide a feasible alternative to deterministic approximations here. However, designing efficient SMC algorithms for the spike and slab model is not trivial (see Schäfer and Chopin (2013) for an approach). We have not evaluated using MCMC computation in this work.

5 Experiments

The performance of the proposed method is evaluated in several "small n, large p" regression problems on both simulated and real data.[3] A proof-of-concept user study is presented to demonstrate the feasibility of the method with real users. We compare our sequential design algorithm (Sect. 4.3) to two baselines in the experiments:

– Random feature suggestion,
– The non-sequential version of our algorithm, which chooses the sequence of features to be queried before observing any expert feedback.

Additionally, to provide a yardstick, we plot the results of an "oracle," which knows the relevant features beforehand, and which is obviously not available in practice:

– Query first on the relevant features, and then choose at random from the features not already selected.

We compare the performance of these strategies with synthetic and real data, and with both simulated and real users providing feedback.

5.1 Synthetic data

We use synthetic data to study the behaviour of the approach in a wide range of controlled settings.

5.1.1 Setting

The covariates of n training data points are generated from $\boldsymbol{X} \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$. Out of the m regression coefficients $w_1, \ldots, w_m \in \mathbb{R}$, m^* are generated from $w_j \sim \mathrm{N}(0, \psi^2)$ and the rest are set to zero. The observed output values are generated from $\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{X}\boldsymbol{w}, \sigma^2\mathbf{I})$. We consider cases where the user has knowledge about the value of the coefficients (Eq. 1 with noise value $\omega = 0.1$) and where the user has knowledge about whether features are relevant or not (Eq. 2 with $\gamma_j = 1$ if w_j is non-zero, and $\gamma_j = 0$ otherwise, and $\pi = 0.95$). For a generated set of training data, all algorithms query feedback about one feature at a time. Mean squared error (MSE) is used as the performance measure to compare the query algorithms. For the simulated data setting, we use the known data-generating values for the fixed hyperparameters, namely: $\psi^2 = 1,\ \rho = m^*/m,$ and $\sigma^2 = 1$ (instead of the inverse-gamma prior of σ^2 used in the rest of the experiments).

5.1.2 Results

Sequential experimental design requires only few feedbacks to improve the MSE In Fig. 2, we consider a "small n, large p" scenario, with $n = 10, m^* = 10$ and with increasing dimensionality (hence also increasing sparsity) from $m = 12, \ldots, 200$. The heatmaps show the average MSE values over 100 runs (repetitions of the data generation) for both feedback

[3] All codes and data are available in https://github.com/HIIT/knowledge-elicitation-for-linear-regression.

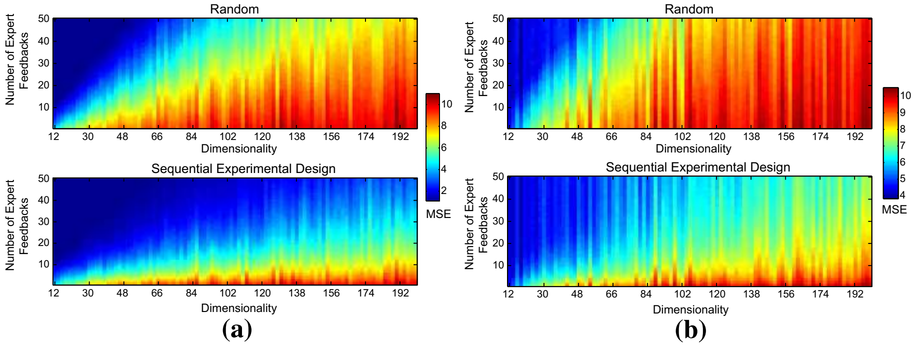


Fig. 2 Mean squared errors in simulated settings with increasing dimensionality. The number of relevant coefficients $m^* = 10$ and the number of training data points $n = 10$. The MSE values are averages over 100 independent runs. **a** Feedback on coefficient values, **b** feedback on coefficient relevances

models, as obtained by our sequential experimental design algorithm and by a strategy that randomly selects the sequence of features on which to ask for expert feedback. The result shows that our method achieves a faster improvement in the prediction, starting from the very first expert feedbacks, for both feedback types, and at all the dimensionalities. Notably, in the case of the random strategy, the performance decreases rapidly with the growing dimensionality (even with 50 feedbacks, in the setting with 200 dimensions, the prediction error for random strategy stays high), while the expert feedback via the sequential experimental design is informative enough to provide good predictions even in large dimensionalities. Comparing the two types of feedback, the feedback on the coefficient values gives better performance for both strategies.

Sequential experimental design requires only few training data points to identify informative queries Figure 3 shows heatmaps for the same setting but with a fixed dimension $m = 100$ and increasing numbers of training data points $n = 5, \ldots, 50$. For very small sample sizes ($n < 10$), a difference between the performance of the two methods starts being visible after 20-30 feedbacks. For larger training samples sizes ($10 < n < 30$), the MSE reduction in our method is more visible from the first feedbacks, while for $n > 30$, both strategies have much smaller MSE after receiving the first feedbacks. Thus, the experiments

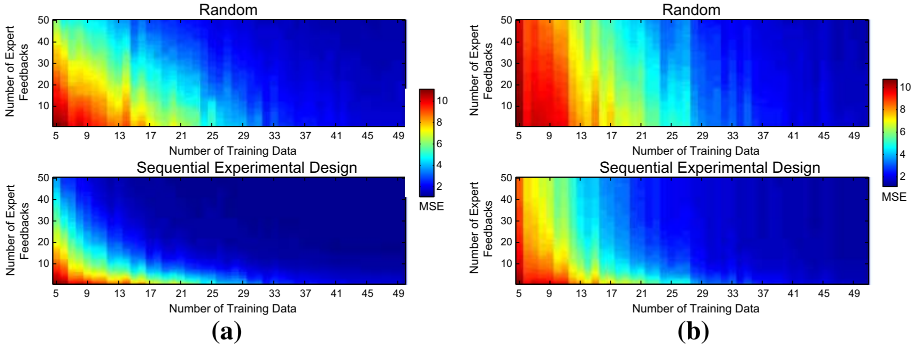


Fig. 3 Mean squared errors as a function of the number of training data points (*horizontal axis*) and number of expert feedbacks (*vertical axis*). The number of relevant coefficients $m^* = 10$ and the number of dimensions $m = 100$. The MSE values are averages over 100 independent runs. **a** Feedback on coefficient values, **b** feedback on coefficient relevances

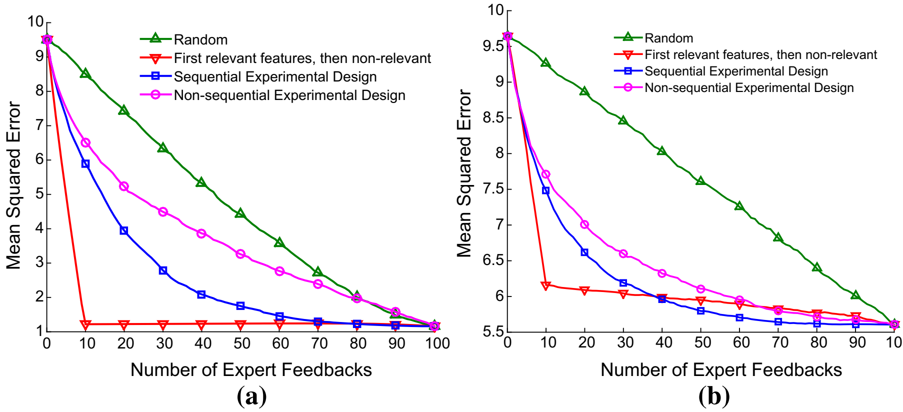


Fig. 4 MSE for all query algorithms, with simulated data, for feedback on coefficient values and relevance. Note that the *red* strategy is not available in practice. **a** Feedback on coefficient values, **b** feedback on coefficient relevances (Color figure online)

show that for all values of n there is an improvement in the prediction error, from the very first expert feedback. An important observation is that the largest improvements come when the training data does not alone provide enough information.

Expert feedback affects the next query We now study the difference between our method and its non-sequential version for the two feedback models. The non-sequential version chooses the sequence of features to be queried before observing any expert feedback. In Fig. 4, we consider a "small n, large p" scenario, with $n = 10$, $m = 100$, $m^* = 10$, and we report the average MSE value over 500 runs. The figure shows that with both experimental design methods the prediction loss decreases rapidly already in the first iterations. In other words, both methods manage to rapidly identify the most informative coefficients and ask about them. This is more evident in the feedback model about coefficient relevance (Fig. 4b). It is also clearly visible that the sequential version is able to reduce the prediction error faster. Also, as expected, the difference between the sequential and non-sequential experimental designs is larger in the case of the stronger feedback model on coefficient values (Fig. 4a).

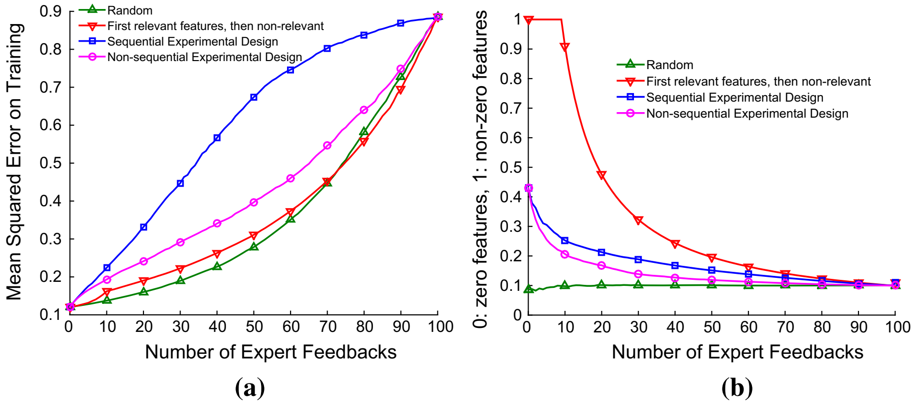


Fig. 5 MSE on the training data and average suggestion behaviour for all query algorithms, with simulated data, for the case where feedback is on coefficient values. **a** MSE on training data, **b** average query behaviour

Expert feedback improves the generalization performance We can get some insight into the behaviour of the approach by comparing the training and test set errors shown in Figs. 4a and 5a for the simulated data scenario described in the previous section with feedback on the coefficient values. The training set error begins to increase as a function of the number of expert feedbacks. This happens because the model without any feedbacks has exhausted the information in the training data (to the extent allowed by the regularizing priors) and fits the training data well. The expert feedback, however, moves the model away from the training data optimum and towards better generalization performance. Indeed, the MSE curves for the training and test errors converge close to each other as the number of feedbacks increases. Moreover, Fig. 5b shows the average (over the replicate runs) query behaviour of each method, indicating whether the methods have queried non-zero features (value 1 in the vertical axis) or zero features (value 0 in the vertical axis). A comparison of Figs. 4a and 5b shows that the convergence is faster for the query algorithms that start by suggesting the non-zero features, implying that these features are more informative.

5.2 Real data: review rating prediction

We test the proposed method in the task of predicting review ratings from textual reviews in subsets of Amazon and Yelp datasets. Each review is one data point, and each distinct word is a feature with the corresponding covariate value given by the number of appearances of the word in the review. Sparse linear regression models have been shown suitable for this task in previous studies, for instance, in Hernández-Lobato et al. (2015).

Amazon data The Amazon data is a subset of the sentiment dataset of Blitzer et al. (2007). This dataset[4] contains textual reviews and their corresponding 1-5 star ratings for Amazon products. Here, we only consider the reviews for products in the *kitchen appliances* category, which amounts to 5149 reviews. The preprocessing of the data follows the method described by Hernández-Lobato et al. (2015), where this dataset was used for testing the performance of a sparse linear regression model. Each review is represented as a vector of features, where the features correspond to unigrams and bigrams, as given by the data provided by Blitzer et al. (2007). For each distinct feature and for each review, we created a matrix of occurrences and only kept for our analysis the features that appeared in at least 100 reviews, that is, 824 features.

Yelp data The second dataset we use is a subset of the YELP (academic) dataset.[5] The dataset contains 2.7 million restaurant reviews with ratings ranging from 1 to 5 stars (rounded to half-stars). Here, we consider the 4086 reviews from the year 2004. Similarly to the preprocessing done for Amazon data, each review is represented as a vector of features (distinct words). After removing non-alphanumeric characters from the words and removing words that appear less than 100 times, we have 465 words for our analysis.

5.3 Simulated expert feedback

For all experiments on Amazon and Yelp datasets, we proceeded as follows: First, each dataset was partitioned in three parts: (1) a training set of 100 randomly selected reviews, (2) a test set of 1000 randomly selected reviews, and (3) the rest as a "user-data set" for

[4] https://www.cs.jhu.edu/~mdredze/datasets/sentiment/.

[5] https://www.yelp.com/dataset_challenge.

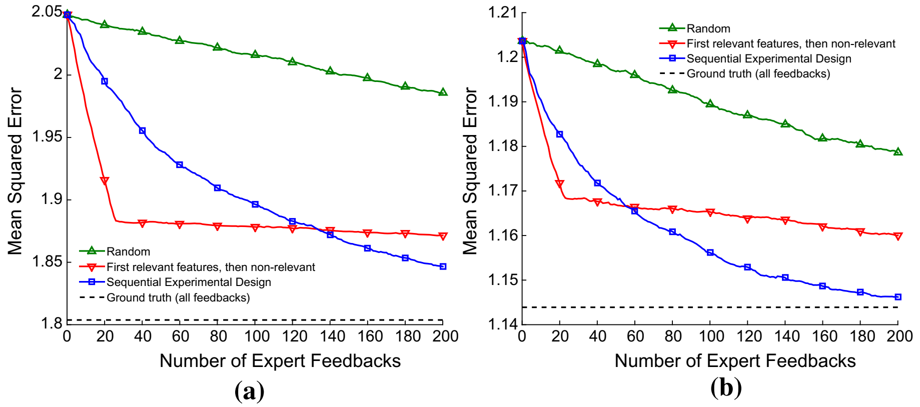


Fig. 6 Mean squared errors when user feedback is on relevance of features for Amazon and Yelp data. The MSE values are averages over 100 independent runs. Note that the *red* strategy is not available in practice. **a** Amazon data, **b** Yelp data (Color figure online)

constructing simulated expert knowledge. The data were normalised to have zero mean and unit standard deviation on the training and user-data sets. The simulated expert feedback was generated based on the posterior inclusion probabilities $\mathbf{E}[\gamma]$ in a spike-and-slab model trained on the user-data partition. We only considered the more realistic case where the expert can give feedback about the relevance of the words. For a word j selected by the algorithm, the expert gives feedback that the word is *relevant* if $\mathbf{E}[\gamma_j] > \pi$, *not-relevant* if $\mathbf{E}[\gamma_j] < 1 - \pi$, and *uncertain* otherwise. The intuition is that if the user-data indicate that a feature is zero/non-zero with high probability, then the simulated expert would select that feature as *not-relevant*/*relevant*. However, for *uncertain* words, the feedback iteration passes without receiving any feedback. For both datasets, the model hyperparameters were set as $\alpha_\sigma = 1$, $\beta_\sigma = 1$, the prediction parameters were tuned based on cross-validation before observing any feedback to $\psi^2 = 0.01$, and $\rho = 0.3$, and the probability that the simulated expert feedback is correct was set to $\pi = 0.9$. All algorithms query feedback about one feature at a time and MSE is used as the performance measure. The ground truth line represents the MSE after receiving expert feedback for all words in each dataset.

5.3.1 MSE improvement with feedback on feature relevances

A first observation of Fig. 6 is that the use of additional knowledge coming from the simulated expert reduces the prediction errors, for all algorithms and on both datasets. The second observation is that the reduction in the prediction error differs significantly depending on whether the methods manage to query feedback on the most informative features first. Indeed, the goal is to make the elicitation as little burdensome as possible for the experts. To reach the goal, a strategy needs to rapidly extract a maximal amount of information from the expert, which here amounts to the careful selection of the features on which to query feedback. As expected, the random query selection strategy has a constant and slow improvement rate, as the number of feedbacks grows, leaving a big gap from the ground-truth performance in both datasets, even after 200 expert feedbacks. If an “oracle” was available to tell which features are relevant, the (unrealistic) strategy to first ask about relevant features would produce a

Table 1 Number of samples/feedbacks needed to reach a particular MSE level in Yelp dataset

MSE	More samples		More feedback	
	Random	Active (Seeger 2008)	Random	SeqExpDes
1.20	21	3	30	3
1.19	55	6	96	11
1.18	94	12	185	25
1.17	146	22	266	46
l.16	241	44	324	85

The values are averages over 100 independent runs

steep increase in performance for the first iterations (26 words for Amazon and 23 for Yelp are marked as relevant, as computed from the full dataset); then it would continue with a very slow improvement rate coming from asking not-relevant words. Our method manages to identify the informative features rapidly and thus has a higher improvement compared to random from the first expert feedbacks. In the case of Yelp data, our strategy manages to be very close to the “oracle” in the initial feedbacks and then converges very close to the ground truth after 200 interactions. Furthermore, there is a significant gap compared to the random strategy for all amounts of feedbacks. In the more difficult (in terms of rating prediction error and size of dimensions) Amazon dataset, the gap to the random strategy is clear but our strategy exceeds the information gain obtained in the 26 non-zero features only after 140 feedbacks.

5.3.2 Expert knowledge elicitation versus collecting more samples

We next contrast the improvements in the predictions brought by eliciting the expert feedback to improvements gained if additional samples could be measured. In this experiment, we do have additional samples, and for choosing them we use two alternative strategies: randomly selecting a sequence of reviews to be included in the training set, and an active learning strategy, which selects samples based on maximizing expected information gain [an adaptation of the method by Seeger (2008)].

Tables 1 and 2 show how many *feedbacks* (for the knowledge elicitation strategies in the last two columns: random and our method; see Sect. 4.3) and respectively how many *additional samples* (that is, additional reviews to be included in the train set) are needed to reach *set levels of MSE*, noting that all strategies have the same “small n, large p” regression setting as a starting point, with $n = 100$.

The number of expert feedbacks required for a given performance level is of the same order of magnitude as the number of additional data needed (Table 1). This slightly surprising finding is even more remarkable since the expert feedback is of a weak type (feedback on the relevance of features). For instance, in Yelp dataset (Table 1) the same level of MSE = 1.18 is obtained either by asking an expert about the relevance of 25 features and by actively selecting 12 extra samples. When the active selection is not possible, we can see that the same information gain requires 94 additional randomly selected samples. Naturally, the results obtained are specific for this Yelp data and for the feedback model we assume. Nevertheless, the comparison shows the potential of expert knowledge elicitation in prediction for settings where getting additional samples is impossible or very expensive. The same observations and intuitions hold for the Amazon data (see Table 2).

Table 2 Number of samples/feedbacks needed to reach a particular MSE level in Amazon dataset

MSE	More samples		More feedback	
	Random	Active (Seeger 2008)	Random	SeqExpDes
2.025	8	4	73	9
2	15	7	152	19
1.975	29	12	>200	31
1.95	44	43	>200	43
1.925	59	71	>200	64
1.9	98	92	>200	95
1.875	>200	144	>200	136

The results are averages over 100 independent runs

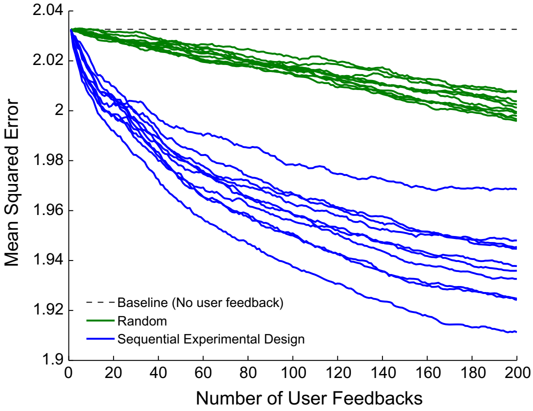
5.4 User study

The goal of the user study is to investigate the prediction improvement and convergence speed of the proposed sequential method based on human feedback. Our focus is on testing the accuracy of feedback from real users on the easily interpretable Amazon data rather than on details of the user interface. We asked ten university students and researchers to go through all the 824 words and give us feedback in the form of *not-relevant*, *relevant*, or *uncertain*. This allowed for a fast collection of feedbacks and we could use the pre-given feedback to test the effectiveness of several query algorithms. We assumed that the algorithms had access to 100 training data points and at each iteration they could query the pre-given feedback of the participant about one word. The whole process was repeated for 40 runs, where training data were randomly selected. The hyperparameters of the model were set to the same values as in the simulated data study with the only difference that the strength of user knowledge was lowered to $\pi = 0.7$.

Figure 7 shows the average MSE improvements for each of the ten participants, when using our proposed method and the random query order. From the very first feedbacks, the sequential experimental design approach performs better for all users and captures the expert knowledge more efficiently. The random strategy exhibits a constant rate of performance improvement with increasing number of feedbacks, while the sequential experimental design shows faster improvement rate in the beginning of the interaction. To further quantify the statistical evidence for the difference, we computed the paired-sample t tests between the random suggestion and the proposed method at each iteration (green and blue curves in Fig. 7). Already after the first feedback, the difference between the methods is significant at the Bonferroni corrected level $\alpha = 0.05/200$.

We complement the analysis of the results of the user study with two illustrations. First, to compare the convergence speed of different methods, we normalised the MSE improvements at each iteration by the amount of total improvement obtained by each of the users, when considering all their individual feedback. Figure 8a depicts the convergence speed of methods based on this measure. As can be seen from the figure, for all participants, the proposed method was able to capture most of the participants's knowledge with small budget of feedback queries (stabilizing at around 200 out of the total 824 features in the considered subset of Amazon data). Then, in Fig. 8b, we show the average suggestion behaviour of the methods. One can notice that our algorithm started by favoring queries about relevant words and after exhausting them, the suggestion behaviour moved to querying not-relevant words. The

Fig. 7 Mean squared errors for ten participants (average values over 40 independent runs)



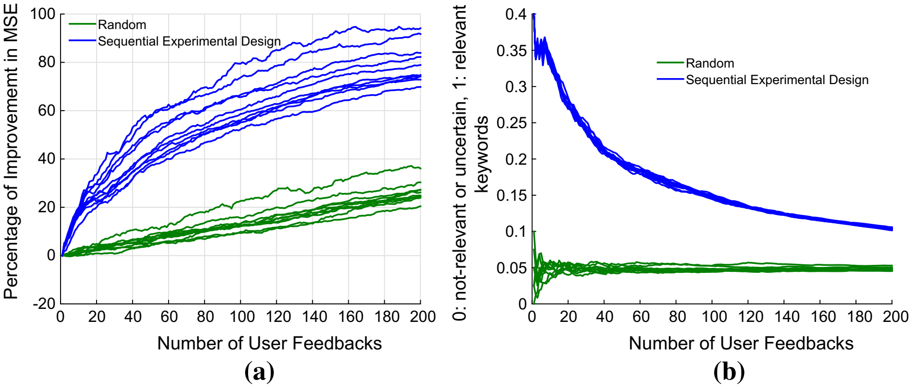


Fig. 8 **a** Convergence speed of different methods to reach the performance achieved by considering all the individual user feedbacks. **b** Average suggestion behaviour of the methods.

relevant words were identified by considering all the data in Amazon dataset and training an spike and slab model and then choosing words with $\mathbf{E}[\gamma_j] > 0.7$ (words with high posterior inclusion probability). Based on this threshold, 39 out of the total of 824 words were considered as relevant.

6 Conclusion and future work

We introduced an interactive knowledge elicitation approach for high-dimensional sparse linear regression. The results for "small n, large p" problems in simulated and real data with simulated and real users, and with expert knowledge on the regression weight values and on the relevance of features, showed improved prediction accuracy already with a small number of user interactions. The knowledge elicitation problem was formulated as a probabilistic inference process that sequentially acquires and integrates expert knowledge with the training data. Compared to pure prior elicitation, the approach can facilitate richer interaction and be used in knowledge elicitation for high-dimensional parameters without overwhelming the expert.

As a by-product of our study, we noticed that even for the rather weak feedback on the relevance of features, the number of expert feedbacks and the number of randomly acquired additional data samples needed to reach a certain level of MSE reduction were of the same order. Although this observation was obtained on a noisy dataset and for a simplifying user interaction setting, the fact that the considered feedback type was rather weak highlights that elicitation from experts is promising.

The presented knowledge elicitation method is widely applicable also beyond the specific assumptions made in this paper. Since all assumptions have been explicated as a probabilistic model, the assumptions can rigorously tailored to match specifics of other data, feedback models, and knowledge elicitation setups, and hence the approach can be applied more generally. The presented results show that it is possible to improve predictions even with preliminary types of feedback, and can be seen as a proof-of-concept of the approach. An important thing to keep in mind is that the amount of improvement in different applications naturally depends on the knowledge experts of that domain have, and their willingness to give the feedback. While our work dwells on improving prediction by using expert knowledge and facilitating elicitation, appropriate interface and visualization techniques are also required for a complete and effective interactive elicitation. These considerations are left for future work.

As for the applications where our method can be employed, of particular interest for future work are the medical settings, where because of the implied risks it is not always possible to increase the sample size. On the other hand, an expert may give feedback on the relevance of certain variables, or an expert might know how much some clinical and behavioural variables explain of the risk. For instance, we plan to test our methods in genomic cancer medicine cases, where the feature size is in the order of thousands (typically including gene expression, mutation data, copy number variation, and cytogenetic marker measurements), while the sample size (number of patients with known measurements) is in the order of hundreds. For different types of cancer, features that are indicative of the drug response (i.e., biomarkers) are well known in the literature (see for instance Garnett et al. 2012). Thus, based on the updated domain knowledge and experience, experts can identify and provide feedback on the relevance of some features.

Acknowledgements This work was financially supported by the Academy of Finland (Finnish Center of Excellence in Computational Inference Research COIN; Grants 295503, 294238, 292334, and 284642), Re:Know funded by TEKES, and MindSee (FP7ICT; Grant Agreement No. 611570). We acknowledge the computational resources provided by the Aalto Science-IT Project. We thank Juho Piironen for comments that improved the article.

Appendix A: Posterior approximation

The posterior distribution of the model and its approximation are

$$p(\boldsymbol{w}, \sigma^{-2}, \boldsymbol{\gamma} \mid \mathcal{D}) \propto p(\boldsymbol{f}_{\gamma} \mid \boldsymbol{\gamma}) p(\boldsymbol{f}_{w} \mid \boldsymbol{w}) p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}, \sigma^{2}) p(\sigma^{-2}) p(\boldsymbol{w} \mid \boldsymbol{\gamma}) p(\boldsymbol{\gamma}) \approx q(\boldsymbol{w}, \sigma^{-2}, \boldsymbol{\gamma}),$$

where $\mathcal{D} = (\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{f}_{\gamma}, \boldsymbol{f}_{w})$ are the training data observations together with the sets of observed expert feedback. The individual terms are approximated as

$$
\begin{aligned}
p(\boldsymbol{f}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma}) &= \prod_{j \in \mathcal{F}_\gamma} \left[\gamma_j \operatorname{Bernoulli}(f_{\gamma,j} \mid \pi) + (1-\gamma_j) \operatorname{Bernoulli}(f_{\gamma,j} \mid 1-\pi)\right] \\
&\approx \prod_{j \in \mathcal{F}_\gamma} \tilde{t}_{\text{Bernoulli}}(\gamma_j \mid \tilde{\rho}_j^{f_\gamma}), \\
p(\boldsymbol{f}_{\boldsymbol{w}} \mid \boldsymbol{w}) &= \prod_{j \in \mathcal{F}_w} \mathrm{N}(f_{w,j} \mid w_j, \omega^2) = \prod_{j \in \mathcal{F}_w} \tilde{t}_{\mathrm{N}}(w_j \mid \tilde{\mu}_j^{f_w}, \tilde{\tau}_j^{f_w}), \\
p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}, \sigma^2) &= \mathrm{N}(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{w}, \sigma^2 \mathbf{I}) \approx \tilde{t}_{\mathrm{N}}(\boldsymbol{w} \mid \tilde{\boldsymbol{\mu}}^y, \tilde{\boldsymbol{\Gamma}}^y) \tilde{t}_{\text{Gamma}}(\sigma^{-2} \mid \tilde{\alpha}^y, \tilde{\beta}^y), \\
p(\sigma^{-2}) &= \operatorname{Gamma}(\sigma^{-2} \mid \alpha_\sigma, \beta_\sigma) \\
&= \tilde{t}_{\text{Gamma}}(\sigma^{-2} \mid \alpha_\sigma - 1, -\beta_\sigma), \\
p(\boldsymbol{w} \mid \boldsymbol{\gamma}) &= \prod_j \left[\gamma_j \mathrm{N}(w_j \mid 0, \psi^2) + (1-\gamma_j)\delta_0(w_j)\right] \\
&\approx \prod_{j \in \mathcal{F}_\gamma} \tilde{t}_{\mathrm{N}}(\gamma_j \mid \tilde{\mu}_j^w, \tilde{\tau}_j^w) \tilde{t}_{\text{Bernoulli}}(\gamma_j \mid \tilde{\rho}_j^w), \\
p(\boldsymbol{\gamma}) &= \prod_j \operatorname{Bernoulli}(\gamma_j \mid \rho) = \prod_j \tilde{t}_{\text{Bernoulli}}(\gamma_j \mid \operatorname{logit}(\rho)).
\end{aligned}
$$

Here, $\mathcal{F}_\gamma$ and $\mathcal{F}_w$ denote the sets of indices of the features that have received relevance feedback and weight feedback, respectively. π, ω^2, α_σ, β_σ, and ψ^2 are assumed fixed hyperparameters. $\tilde{t}.$ denote the exponential family forms of the corresponding distributions parametrized by the precision-adjusted mean and precision for normal distribution, and the natural parameters for Bernoulli and gamma distributions. Note that the terms $p(\sigma^{-2})$, $p(\boldsymbol{f}_{\boldsymbol{w}} \mid \boldsymbol{w})$, and $p(\boldsymbol{\gamma})$ need not be approximated as they are already of the correct exponential family form. The full posterior approximation follows as $q(\boldsymbol{w}, \sigma^{-2}, \boldsymbol{\gamma}) = q(\boldsymbol{w})q(\sigma^{-2})q(\boldsymbol{\gamma})$ with

$$
\begin{aligned}
q(\boldsymbol{w}) &= \mathrm{N}(\boldsymbol{w} \mid \bar{\boldsymbol{m}}, \bar{\boldsymbol{\Sigma}}), \\
q(\sigma^{-2}) &= \operatorname{Gamma}(\sigma^{-2} \mid \bar{\alpha}_\sigma, \bar{\beta}_\sigma), \\
q(\boldsymbol{\gamma}) &= \prod_j \operatorname{Bernoulli}(\gamma_j \mid \bar{\rho}_j),
\end{aligned}
$$

where the parameters can be identified from the products of the corresponding site term approximations and are

$$
\begin{aligned}
\bar{\boldsymbol{m}} &= \bar{\boldsymbol{\Sigma}}(\tilde{\boldsymbol{\mu}}^y + \tilde{\boldsymbol{\mu}}^w + \tilde{\boldsymbol{\mu}}^{f_w}), \\
\bar{\boldsymbol{\Sigma}} &= (\tilde{\boldsymbol{\Gamma}}^y + \operatorname{diag}(\tilde{\boldsymbol{\tau}}^w) + \operatorname{diag}(\tilde{\boldsymbol{\tau}}^{f_w}))^{-1}, \\
\bar{\alpha}_\sigma &= \alpha_\sigma + \tilde{\alpha}^y, \\
\bar{\beta}_\sigma &= \beta_\sigma - \tilde{\beta}^y, \\
\bar{\rho}_j &= \frac{1}{1 + \exp(-(\tilde{\rho}_j^w + \operatorname{logit}(\rho) + \tilde{\rho}_j^{f_\gamma}))},
\end{aligned}
$$

where $\operatorname{diag}(\cdot)$ is a diagonal matrix with the parameter as the diagonal and feedback term approximation parameters are zero for feedbacks that have not been observed.

Expectation propagation (EP) and variational Bayes (VB) inference are used to find the parameters of the posterior approximation (Minka 2001, 2005; Bishop 2006). Expectation propagation for linear regression with spike and slab prior has been introduced by Hernández-Lobato et al. (2008) (see Hernández-Lobato et al. (2015) for a more extensive treatment). We update the $\tilde{t}_{\mathrm{N}}(\boldsymbol{w} \mid \tilde{\boldsymbol{\mu}}^y, \tilde{\boldsymbol{\Gamma}}^y)$ and $\tilde{t}_{\mathrm{Gamma}}(\sigma^{-2} \mid \tilde{\alpha}^y, \tilde{\beta}^y)$ term approximations using VB and all other terms using EP. The parameter update steps in the algorithm, to be iterated until convergence, are

1. $p(\boldsymbol{w} \mid \boldsymbol{\gamma})$ approximation using parallel EP update.
2. $p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}, \sigma^2)$ approximation using VB update.
3. $p(\boldsymbol{f}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma})$ approximation using parallel EP update.

All of the computations have closed form solutions. The VB update is used, because there is no closed form EP update for the term. Importantly, a full covariance matrix in the posterior approximation of the regression weights $\boldsymbol{w}$ is retained. Alternatively, an approximate EP update, following Hernandez-Lobato et al. (2015), would be possible.

Appendix B: Bayesian experimental design

The task is to find the feedback that maximises the expected information gain:

$$j^* = \underset{j \notin \mathcal{F}}{\arg\max}\, \mathbb{E}_{p(\tilde{f}_j \mid \mathcal{D})}\left[\sum_i \mathrm{KL}[p(\tilde{y} \mid \boldsymbol{x}_i, \mathcal{D}, \tilde{f}_j) \parallel p(\tilde{y} \mid \boldsymbol{x}_i, \mathcal{D})]\right],$$

where $\mathcal{F}$ is the set of feedbacks that have already been given (to simplify notation, those are here assumed included in $\mathcal{D}$) and the summation over i goes over the training dataset. The evaluation of the expected information gain is described in the following.

The posterior predictive distribution is approximated as Gaussian:

$$p(\tilde{y} \mid \tilde{\boldsymbol{x}}, \mathcal{D}) \approx \mathrm{N}(\tilde{y} \mid \tilde{\boldsymbol{x}}^\top \bar{\boldsymbol{m}}, \tilde{\boldsymbol{x}}^\top \bar{\boldsymbol{\Sigma}} \tilde{\boldsymbol{x}} + \bar{s}^2),$$

where $\bar{s}^2 = \frac{\bar{\beta}_\sigma}{\bar{\alpha}_\sigma}$ is the posterior mean approximation for the residual variance. Similarly, the posterior predictive distributions of the feedbacks for the two feedback types follow as approximate Gaussian and Bernoulli distributions:

$$\begin{aligned} p(\tilde{f}_{w,j} \mid \mathcal{D}) &\approx \mathrm{N}(\tilde{f}_{w,j} \mid \bar{m}_j, \bar{\Sigma}_{jj} + \omega^2), \\ p(\tilde{f}_{\gamma,j} \mid \mathcal{D}) &\approx \mathrm{Bernoulli}(\tilde{f}_{\gamma,j} \mid \pi \bar{\rho}_j + (1-\pi)(1-\bar{\rho}_j)). \end{aligned}$$

The information gain between the predictive distributions is

$$\begin{aligned} &\mathrm{KL}[p(\tilde{y} \mid \tilde{\boldsymbol{x}}, \mathcal{D}, \tilde{f}_j) \parallel p(\tilde{y} \mid \tilde{\boldsymbol{x}}, \mathcal{D})] \\ &\quad = \frac{1}{2}\left[\log \frac{\tilde{\boldsymbol{x}}^\top \bar{\boldsymbol{\Sigma}} \tilde{\boldsymbol{x}} + \bar{s}^2}{\tilde{\boldsymbol{x}}^\top \bar{\boldsymbol{\Sigma}}_{\tilde{f}} \tilde{\boldsymbol{x}} + \bar{s}^2_{\tilde{f}}} + \frac{\tilde{\boldsymbol{x}}^\top \bar{\boldsymbol{\Sigma}}_{\tilde{f}} \tilde{\boldsymbol{x}} + \bar{s}^2_{\tilde{f}} + (\tilde{\boldsymbol{x}}^\top \bar{\boldsymbol{m}}_{\tilde{f}} - \tilde{\boldsymbol{x}}^\top \bar{\boldsymbol{m}})^2}{\tilde{\boldsymbol{x}}^\top \bar{\boldsymbol{\Sigma}} \tilde{\boldsymbol{x}} + \bar{s}^2} - 1\right]. \end{aligned}$$

As running the EP algorithm to full convergence would be too costly for evaluating a large number of candidates, we approximate the posterior distribution with the new feedback with partial EP updates. This is similar to the approach of Seeger (2008) and Hernández-Lobato et al. (2013) for experimental design for sparse linear model. We consider the two types of feedback separately.

In the case of feedback directly on the regression weight, we add the corresponding site term (which is already of Gaussian form and does not need approximation, as noted above) and do not update the approximations of the other site terms (including assuming $\bar{s}^2_{\tilde{f}} = \bar{s}^2$). The new posterior approximation of $\boldsymbol{w}$ with these assumptions is

$$\bar{\boldsymbol{\Sigma}}_{\tilde{f}_{w,j}} = (\bar{\boldsymbol{\Sigma}}^{-1} + T\boldsymbol{e}\boldsymbol{e}^\top)^{-1},$$
$$\bar{\boldsymbol{m}}_{\tilde{f}_{w,j}} = \bar{\boldsymbol{\Sigma}}_{\tilde{f}_{w,j}}(\bar{\boldsymbol{\Sigma}}^{-1}\bar{\boldsymbol{m}} + h\boldsymbol{e}), \tag{3}$$

where $\boldsymbol{e}$ is a vector of zeros except for 1 at jth element, $T = \frac{1}{\omega^2}$, and $h = \frac{\tilde{f}_{w,j}}{\omega^2}$. Notably, $\bar{\boldsymbol{\Sigma}}^{-1}$ and $\bar{\boldsymbol{\Sigma}}^{-1}\bar{\boldsymbol{m}}$ are the precision and the precision-adjusted mean of the posterior approximation without the new feedback and are directly available from the previous EP approximation. The new posterior covariance is independent of the value of the feedback $\tilde{f}_{w,j}$ and it can be efficiently evaluated using the matrix inversion lemma as $\bar{\boldsymbol{\Sigma}}_{\tilde{f}} = \bar{\boldsymbol{\Sigma}} - \frac{1}{T^{-1}+\bar{\Sigma}_{jj}}\bar{\boldsymbol{\Sigma}}\boldsymbol{e}\boldsymbol{e}^\top\bar{\boldsymbol{\Sigma}}$. Furthermore, the expectation over the feedback in the expected information gain affects only the term with the squared difference of the means. This is

$$\mathbb{E}_{p(\tilde{f}_j|\mathcal{D})}\left[(\tilde{\boldsymbol{x}}^\top\bar{\boldsymbol{m}}_{\tilde{f}} - \tilde{\boldsymbol{x}}^\top\bar{\boldsymbol{m}})^2\right] = \mathbb{E}_{p(\tilde{f}_j|\mathcal{D})}\left[\left(\frac{T_{jj}}{1+T\bar{\Sigma}_{jj}}\tilde{\boldsymbol{x}}^\top\bar{\boldsymbol{\Sigma}}\boldsymbol{e}\right)^2\left(\frac{h}{T} - \bar{m}_j\right)^2\right]$$
$$= \left(\frac{T}{1+T\bar{\Sigma}_{jj}}\tilde{\boldsymbol{x}}^\top\bar{\boldsymbol{\Sigma}}\boldsymbol{e}\right)^2(\bar{\Sigma}_{jj} + \omega^2),$$

where the first equality follows from substituting the Eq. 3 and using the matrix inversion lemma, and the second equality from $\frac{h}{T} = \tilde{f}_{w,j}$ and the remaining expectation being equal to the variance of the predictive distribution of the feedback.

In the case of relevance feedback, we add the corresponding site term for the feedback and run single EP update on it and the corresponding prior term $p(w_j \mid \gamma_j)$. These updates are purely scalar operations and do not require any costly matrix operations. Other site term approximations are not updated. The new posterior approximation of $\boldsymbol{w}$ with these assumptions is

$$\bar{\boldsymbol{\Sigma}}_{\tilde{f}_{\gamma,j}} = (\bar{\boldsymbol{\Sigma}}^{-1} + T\boldsymbol{e}\boldsymbol{e}^\top)^{-1},$$
$$\bar{\boldsymbol{m}}_{\tilde{f}_{\gamma,j}} = \bar{\boldsymbol{\Sigma}}_{\tilde{f}_{\gamma,j}}(\bar{\boldsymbol{\Sigma}}^{-1}\bar{\boldsymbol{m}} + h\boldsymbol{e}),$$

where $T = [\bar{\boldsymbol{\Sigma}}^{-1}_{\tilde{f}_{\gamma,j}}]_{jj} - [\bar{\boldsymbol{\Sigma}}^{-1}]_{jj}$ and $h = [\bar{\boldsymbol{\Sigma}}^{-1}_{\tilde{f}_{\gamma,j}}\boldsymbol{m}_{\tilde{f}_{\gamma,j}}]_j - [\bar{\boldsymbol{\Sigma}}^{-1}\bar{m}]_j$. That is, now T and h are the changes in the precision and the precision adjusted mean in the jth feature and these are available with cheap scalar operations. The expectation over the value of the feedback in the expected information gain is in this case a sum of two terms and we evaluate both of the terms separately using the above scheme. Again, we use the matrix inversion lemma to avoid full inversions in computing the new posterior covariance.

References

Afrabandpey, H., Peltola, T., & Kaski, S. (2016). Interactive prior elicitation of feature similarities for small sample size prediction. In *Proceedings of the 25th conference on user modelling, adaptation and personalization (UMAP2017)* (to appear). arXiv preprint arXiv:1612.02802.

Amershi, S. (2012). *Designing for effective end-user interaction with machine learning*. PhD thesis, University of Washington.

Azari Soufiani, H., Parkes, D. C., & Xia, L. (2013). Preference elicitation for general random utility models. In *Uncertainty in artificial intelligence: Proceedings of the 29th conference* (pp. 596–605). AUAI Press.

Balcan, M. F., & Blum, A. (2008). Clustering with interactive feedback. In *Proceedings of the 19th international conference on algorithmic learning theory* (pp. 316–328).

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Berlin: Springer.

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics (ACL)* (pp. 187–205).

Busby, D. (2009). Hierarchical adaptive experimental design for Gaussian process emulators. *Reliability Engineering & System Safety*, *94*(7), 1183–1193.

Cano, A., Masegosa, A. R., & Moral, S. (2011). A method for integrating expert knowledge when learning Bayesian networks from data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *41*(5), 1382–1394.

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, *10*(3), 273–304.

Del Moral, P., Doucet, A., & Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(3), 411–436.

Donoho, D., & Tanner, J. (2009). Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. *Philosophical Transactions of the Royal Society A*, *367*, 4273–4293.

Druck, G., Settles, B., & McCallum, A. (2009). Active learning by labeling features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 81–90).

Fails, J. A., & Olsen Jr., D. R. (2003). Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces (IUI)* (pp. 39–45).

Ferreira, G. S., & Gamerman, D. (2015). Optimal design in geostatistics under preferential sampling. *Bayesian Analysis*, *10*(3), 711–735. doi:10.1214/15-BA944.

Garnett, M. J., Edelman, E. J., Heidorn, S. J., Greenman, C. D., Dastur, A., Lau, K. W., et al. (2012). Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature*, *483*(7391), 570–575.

Garthwaite, P. H., & Dickey, J. M. (1988). Quantifying expert opinion in linear regression problems. *Journal of the Royal Statistical Society Series B (Methodological)*, *50*, 462–474.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2014). *Bayesian data analysis* (3rd ed.). Boca Raton: Chapman & Hall/CRC.

George, E. I., & McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, *88*(423), 881–889.

Hernández-Lobato, D., Hernández-Lobato, J. M., & Dupont, P. (2013). Generalized spike-and-slab priors for bayesian group feature selection using expectation propagation. *Journal of Machine Learning Research*, *14*(1), 1891–1945.

Hernandez-Lobato, D., Hernandez-Lobato, J. M., & Ghahramani, Z. (2015). A probabilistic model for dirty multi-task feature selection. In F. Bach, D. Blei (Eds.), *Proceedings of the 32nd international conference on machine learning, PMLR, Lille, France, proceedings of machine learning research* (Vol. 37, pp. 1073–1082).

Hernández-Lobato, J. M., Dijkstra, T., & Heskes, T. (2008). Regulator discovery from gene expression time series of malaria parasites: A hierarchical approach. In *Advances in neural information processing systems 20 (NIPS)* (pp 649–656).

Hernández-Lobato, J. M., Hernández-Lobato, D., & Suárez, A. (2015). Expectation propagation in linear regression models with spike-and-slab priors. *Machine Learning*, *99*(3), 437–487.

Hickey, A. M., & Davis, A. M. (2003). Requirements elicitation and elicitation technique selection: A model for two knowledge-intensive software development processes. In *Proceedings of the 36th annual Hawaii international conference on system sciences (HICSS'03)—Track 3* (Vol. 3).

House, L., Scotland, L., & Han, C. (2015). Bayesian visual analytics: Bava. *Statistical Analysis and Data Mining*, *8*(1), 1–13.

Kadane, J. B., Dickey, J. M., Winkler, R. L., Smith, W. S., & Peters, S. C. (1980). Interactive elicitation of opinion for a normal linear model. *Journal of the American Statistical Association*, *75*(372), 845–854.

Kiefer, J., & Wolfowitz, J. (1959). Optimum designs in regression problems. *The Annals of Mathematical Statistics*, *30*(2), 271–294. doi:10.1214/aoms/1177706252.

Lu, Z., & Leen, T. K. (2007). Semi-supervised clustering with pairwise constraints: A discriminative approach. In *Proceedings of the eleventh international conference on artificial intelligence and statistics (AISTATS)* (pp. 299–306).

Martino, L., Vicent, J., & Camps-Valls, G. (2017). Automatic emulator and optimized look-up table generation for radiative transfer models. In *Proceedings of IEEE international geoscience and remote sensing symposium (IGARSS)*.

Micallef, L., Sundin, I., Marttinen, P., Ammad-ud-din, M., Peltola, T., Soare, M., Jacucci, G., & Kaski, S. (2017). Interactive elicitation of knowledge on feature relevance improves predictions in small data sets. In *Proceedings of the 22nd international conference on intelligent user interfaces (IUI'17)*.

Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. In *Proceedings of the seventeenth conference on uncertainty in artificial intelligence (UAI)* (pp. 362–369).

Minka, T. P. (2005). *Divergence measures and message passing*. Tech. rep., Microsoft Research.

O'Hagan, A., Buck, C. E., Daneshkhah, A., Eiser, J. R., Garthwaite, P. H., Jenkinson, D. J., et al. (2006). *Uncertain judgements. Eliciting experts' probabilisties*. Chichester: Wiley.

Peltola, T., Marttinen, P., & Vehtari, A. (2012). Finite adaptation and multistep moves in the Metropolis–Hastings algorithm for variable selection in genome-wide association analysis. *PloS One*, *7*(11), e49,445.

Raghavan, H., Madani, O., & Jones, R. (2006). Active learning with feedback on features and instances. *Journal of Machine Learning Research*, *7*(Aug), 1655–1686.

Ravi, S. N., Ithapu, V. K., Johnson, S. C., & Singh, V. (2016). Experimental design on a budget for sparse linear models and applications. In *Proceedings of the 33nd international conference on machine learning (ICML)* (pp. 583–592).

Robert, S., Büttner, S., Röcker, C., & Holzinger, A. (2016). Reasoning under uncertainty: Towards collaborative interactive machine learning. In A. Holzinger (Ed.), *Machine learning for health informatics* (pp. 357–376). Berlin: Springer.

Ruotsalo, T., Jacucci, G., Myllymäki, P., & Kaski, S. (2014). Interactive intent modeling: Information discovery beyond search. *Communications of the ACM*, *58*(1), 86–92.

Schäfer, C., & Chopin, N. (2013). Sequential Monte Carlo on large binary sampling spaces. *Statistics and Computing*, *23*, 163–184. doi:10.1007/s11222-011-9299-z.

Seeger, M. W. (2008). Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, *9*, 759–813.

Settles, B. (2010). *Active learning literature survey*. Computer Sciences technical report 1648, University of Wisconsin, Madison.

Settles, B. (2011). Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 1467–1478).

Soare, M., Ammad-ud-din, M., & Kaski, S. (2016). Regression with n → 1 by expert knowledge elicitation. In *Proceedings of the 15th IEEE ICMLA international conference on machine learning and applications* (pp. 734–739).

Ware, M., Frank, E., Holmes, G., Hall, M., & Witten, I. H. (2001). Interactive machine learning: Letting users build classifiers. *International Journal of Human-Computer Studies*, *55*(3), 281–292.