Human-in-the-loop Active Covariance Learning for Improving Prediction in Small Data Sets

Homayun Afrabandpey , **Tomi Peltola** and **Samuel Kaski**
Helsinki Institute for Information Technology HIIT
Department of Computer Science, Aalto University
{homayun.afrabandpey, tomi.peltola, samuel.kaski}@aalto.fi

Abstract

Learning predictive models from small high-dimensional data sets is a key problem in high-dimensional statistics. Expert knowledge elicitation can help. A strong line of research focuses on directly eliciting informative prior distributions for parameters. This either requires considerable statistical expertise or is laborious, as the emphasis has been on accuracy and not on efficiency of the process. Another line of work queries about importance of features one at a time, assuming them to be independent and hence missing covariance information. In contrast, we propose eliciting expert knowledge about pairwise feature similarities, to borrow statistical strength in the predictions, and using sequential decision making techniques to minimize the effort of the expert. Empirical results demonstrate improvement in predictive performance on both simulated and real data, in high-dimensional linear regression tasks, where we learn the covariance structure with a Gaussian process, based on sequential elicitation.

1 Introduction

Data sets with a small number of observations n and large number of variables p, a.k.a "small n, large p", are common in many applications and pose big challenges for statistical methods [Fan and Li, 2006]. Potential approaches to alleviate these challenges are either to provide more samples, which could be very expensive or even impossible in some applications, or to regularize models using additional sources of information. An example of such an additional source is a human expert who may have substantial subject-specific knowledge that can be used to improve data analysis.

In Bayesian statistics, expert-provided information can be formulated as a prior distribution over the parameters to regularize the model. Extracting and incorporating experts' knowledge into statistical models is known as *knowledge elicitation*. When designing an elicitation method, two important choices will affect the success. First is what is assumed of the knowledge of the expert: only domain expertise or also statistical expertise. In Bayesian statistics, most of the existing knowledge elicitation approaches require the domain expert to be expert, or at least knowledgeable, in statistics as well. The second choice is the assumptions made on the content of knowledge that is elicited, which also translates into the structure of the model used for the elicitation. Most existing methods for eliciting importance of variables for prediction tasks assume independence among the parameters to make the elicitation easier, at the cost of ignoring expert knowledge on dependencies.

In this paper, we propose a novel elicitation approach that avoids both problems: only domain expertise is required, and the underlying assumptions are generalized from independence of the parameters to pairwise dependencies. Technically, we formulate the problem as a sequential decision-making task of choosing which queries to ask to maximally improve predictions with minimal expert effort. The queries will be about pairwise relationships of features, to expand beyond the earlier assumption of their independence.

With experiments on synthetic and real data, we demonstrate that the approach is able to improve the predictive performance by constructing a more informative covariance matrix using expert feedback.

1.1 Contribution

In summary, the main contributions of the paper are: **(1)** A method for learning a prior with a full covariance matrix (without assuming the features to be independent) using information provided by a domain expert, **(2)** an algorithm for actively eliciting expert's knowledge on pairwise similarities of the features, to maximally improve predictive performance while minimizing the expert's workload.

2 Related Work

There exists a rich literature on knowledge elicitation for improving the performance of statistical models. A number of studies have proposed to elicit scores or point estimates of an unknown parameter or quantity of interest directly from the expert [Garthwaite *et al.*, 2008; Haakma *et al.*, 2011; O'Hagan, 1998]. These approaches typically either assume that the expert has a high level of statistical training, to be able to assess and formulate prior distributions, or is assisted by a human facilitator in an elicitation session. The main goal is accuracy of the knowledge instead of time-efficiency, which makes the elicitation burdensome. In an attempt to make knowledge elicitation more automatic and resource-efficient,

[Micallef *et al.*, 2017] proposed an interactive approach to elicit the expert's knowledge on the relevance of individual features to improve the predictive accuracy of an underlying model. They used a sequential decision making method, multi-armed bandits, to guide the interaction toward features that would likely benefit from the expert's input. [Daee *et al.*, 2017] formulated knowledge elicitation as a probabilistic inference problem where expert knowledge about individual features is sequentially extracted to improve predictions in a sparse linear regression task. The work was extended by [Sundin *et al.*, 2018] by considering information about the direction of a relevant feature. In all these works, independence among features is assumed in the prior distribution. These approaches could be used to complement our method.

In a similar scenario, [Xiao *et al.*, 2018] proposed an approach for extracting expert knowledge on influential relationships between pairs of variables in a directed acyclic graph, i.e., whether a variable a is likely to be up- or downstream of variable b. Their goal, however, is not prediction but to learn the structure of the graph. [Afrabandpey *et al.*, 2017] proposed an interactive method for eliciting knowledge on pairwise feature similarities based on an interactive display and used it to improve predictive performance of linear regression. This work is similar to ours, but differs in two important aspects: First, their method is passive – the queries do not change according to what has been elicited so far – which results in an inefficient and exhaustive elicitation process. Second, the expert feedbacks are only post-hoc connected to the prediction problem through a metric learning process, while our approach adopts expert feedback directly by constructing an informative prior distribution for the underlying model.

Our method builds upon two further works. [Krupka and Tishby, 2007] studied cases where there are meta-features that capture prior knowledge on features. Using these meta-features, they proposed a method to improve generalization of a classification model by constructing a prior covariance (or a Gaussian process) on the parameters without independence assumptions among the variables. Their approach, however, is not a human-in-the-loop system and does not elicit expert knowledge. [Yang *et al.*, 2007] proposed a Bayesian framework for actively learning a similarity metric among samples using pairwise constraints labeled by a human agent. The method aims to improve pairwise distances by querying the unlabeled pairs of samples with the greatest uncertainty in relative distance. In contrast, our approach is designed to optimize the predictive performance of the model using feedback on pairs of features and we query pairs based on their expected improvement on predictive performance.

3 Proposed Approach

We consider the task of learning a probabilistic predictive function $y = h_{\boldsymbol{\theta}}(\boldsymbol{x})$, parametrized by $\boldsymbol{\theta}$, given a data set $\mathcal{D} = \{(y_i, \boldsymbol{x}_i); i = 1, \ldots, n\}$ of n outcomes y_i and feature vectors $\boldsymbol{x}_i = [x_{i1}, \ldots, x_{ip}]^{\mathrm{T}}$. We assume that n is small and cannot be increased, so that it is worthwhile to spend effort on collecting and encoding prior knowledge or other external information. Here, we particularly focus on using knowledge elicitation from domain expert to learn the dependencies between the elements of $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_L]^{\mathrm{T}}$, where L is the total number of parameters of the predictive function. This approach is complementary to many of the previous prior and expert knowledge elicitation approaches.

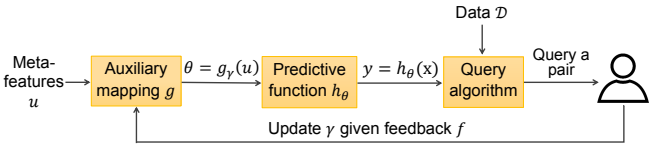


Figure 1: Components of the proposed approach. To improve the predictive function $h_{\boldsymbol{\theta}}$, the auxiliary mapping $g_{\boldsymbol{\gamma}}$ models dependencies between the parameters $\boldsymbol{\theta}$ using the meta-features $\boldsymbol{u}$ and the current value of $\boldsymbol{\gamma}$. The query algorithm optimizes the workload of the expert by choosing queries that maximize the expected gain in the predictive performance. The expert feedback f then updates $\boldsymbol{\gamma}$.

In addition to the data $\mathcal{D}$, we assume the availability of meta-features $\boldsymbol{u}_l \in \mathbb{R}^d$. Meta-features are vectorial description of the features which provide detailed information about their properties. Our approach to improve the predictive model $h_{\boldsymbol{\theta}}$ is then formulated in three components as shown in Figure 1: (1) An auxiliary mapping g from the meta-features $\boldsymbol{u}_l$ to the parameters θ_l, $\theta_l = g_{\boldsymbol{\gamma}}(\boldsymbol{u}_l)$, with parameters $\boldsymbol{\gamma}$. (2) An observation model for expert feedback that encodes the expert knowledge to information about g and consequently to the dependencies between the elements of $\boldsymbol{\theta}$. (3) A query algorithm that chooses questions to the expert to optimize the effort of the expert for improving the predictions. Each component is described in more detail in the following sections.

To concretize the approach, consider as an example a sentiment analysis task where the goal is to predict a rating assigned by a reviewer to a product from the text of the review. The training data is the set of all reviews given by different reviewers together with their corresponding ratings. The feature vector $\boldsymbol{x}$ describes occurrences of keywords in the review text (bag-of-words). Assume linear regression for prediction, $h_{\boldsymbol{\beta}}(\boldsymbol{x}) = \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}$, with $\boldsymbol{\theta} = \boldsymbol{\beta}$ being the regression coefficients. A simple approach to constructing a meta-feature vector $\boldsymbol{u}_l$ for each keyword l is to take the vector of occurrences of the keyword in the training set reviews. This implies that features that do not occur in the same documents are dissimilar and vice versa. Using a Gaussian process as the auxiliary mapping g then corresponds to learning a full covariance matrix for the prior of the regression coefficients $\boldsymbol{\beta}$ based on the meta-features. Knowledge about the pairwise similarity of the role of the keywords in the rating prediction is sequentially queried from the expert. The rationale is that the expert has knowledge about the correlation among regression coefficients of at least some keywords, i.e., either that two keywords positively affect the output (both cause an increase in the rating) or they negatively affect it (decrease the rating). The extracted knowledge will then be incorporated into the model to improve the prior covariance matrix and thus predictions. The case of prediction task with linear regression is described in detail in Section 3.4.

3.1 Modelling Parameter Dependencies

The auxiliary mapping g is used to model the dependencies between the parameters $\boldsymbol{\theta}$. We model it with a Gaussian process with a zero mean and covariance function $k_{\boldsymbol{\gamma}}$, where $\boldsymbol{\gamma}$ is a vector of parameters [Krupka and Tishby, 2007; Rasmussen and Williams, 2006]. In particular, this implies that $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K})$, where $\boldsymbol{K}$ is a covariance matrix (kernel). The element $K_{ij} = k_{\boldsymbol{\gamma}}(\boldsymbol{u}_i, \boldsymbol{u}_j)$ describes the prior covariance between the parameters θ_i and θ_j based on the similarity of their meta-features $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ (according to the covariance function).

To illustrate the effect of the auxiliary mapping on learning $\boldsymbol{\theta}$, consider a two-dimensional special case, with θ_1 and θ_2. If the prior correlation implied by $\boldsymbol{K}$ is equal to one, then $\theta_1 = \theta_2$; that is, g maps the two meta-feature vectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ to the same point and, effectively, there is only one parameter to be learned from the data set $\mathcal{D}$. If the prior correlation is zero, learning information about one parameter has no effect on the other, and we need to estimate both of the parameters from $\mathcal{D}$. Of course, often we would be somewhere between these two extremes, such that some statistical strength can be shared between learning of the two parameters.

The covariance function $k_{\boldsymbol{\gamma}}$ defines many of the properties (smoothness, flexibility, magnitude, etc.) of the auxiliary mapping based on the meta-features $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$. Many different covariance functions are possible (see, e.g., [Rasmussen and Williams, 2006]). We use the Gaussian kernel

$$k_{\boldsymbol{\gamma}}(\boldsymbol{u}_i, \boldsymbol{u}_j) = \exp\left(-\frac{\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2_{\boldsymbol{A}_{\boldsymbol{\gamma}}}}{2}\right), \tag{1}$$

where $\boldsymbol{A}_{\boldsymbol{\gamma}} \in \mathbb{R}^{d\times d}$ is a diagonal matrix defining the distance metric with elements γ_l, such that $\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2_{\boldsymbol{A}_{\boldsymbol{\gamma}}} = \sum_l \gamma_l (u_{il} - u_{jl})^2$. Here, the $\boldsymbol{\gamma}$ are inverse length scales: they determine how influential each dimension of the meta-feature is in the distance metric; $\gamma_i = 0$ means that the i^{th} dimension of the meta-features does not influence the distance. Increasing the value from 0 means that it is increasingly important in defining the distance. Inference about $\boldsymbol{\gamma}$ is done by defining a half-normal prior distribution $\boldsymbol{\gamma} \sim \mathcal{N}^+(\gamma_0 \mathbf{1}_{d\times 1}, \delta \boldsymbol{I}_{d\times d})$. The γ_0 and δ are constant values.

The auxiliary mapping g defines a prior on the parameters $\boldsymbol{\theta}$, based on the meta features. It would be attractive to learn g from the data $\mathcal{D}$, but that would require more data than available in the "small n, large p" case we are targeting. That is why we turn next to learning it based on knowledge elicitation instead.

3.2 Feedback Model

A feedback model is used to incorporate expert's knowledge into the model. In this work, we consider feedback on the pairwise similarities of the roles of the parameters $\boldsymbol{\theta}$ in the prediction. For example, if the $\boldsymbol{\theta}$ are linear regression weights, the expert tells the model which pairs of features affect the prediction output similarly (positively and/or negatively). Expert's feedback will be either "Similar", "Dissimilar", or "I don't know". The latter will be ignored by the model and has no effect on prior covariance and consequently on the predictions. Similar to [Yang *et al.*, 2007], we define the following likelihood for the feedbacks:

$$f_{ij} \sim \text{Bernoulli}(q_{ij}), \tag{2}$$

where f_{ij} is the feedback given on the similarity of the features i and j ($f_{ij} = 1$ for similar and $f_{ij} = -1$ dissimilar feedback), $q_{ij} = \frac{1}{1+\exp\left(\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2_{\boldsymbol{A}_{\boldsymbol{\gamma}}} - \xi\right)}$ is the parameter of the Bernoulli distribution, and $\xi > 0$ is the threshold determining the similarity/dissimilarity of the two features. Two features are more likely to be similar, $f_{ij} = 1$, when their distance is less than the threshold ξ and vice versa. To avoid setting the value of the threshold manually, we define a half-normal prior distribution over it, $\xi \sim \mathcal{N}^+\left(m_\xi, \sigma^2_\xi\right)$.

The expert knowledge is transferred to the data model through sharing the parameter $\boldsymbol{\gamma}$ in the feedback model and the auxiliary mapping. In the feedback model, there is an inverse relationship between probability of two features being similar, i.e. q_{ij}, and their scaled distance parameterized by $\boldsymbol{\gamma}$. Via the auxiliary mapping, this distance also determines the strength of the prior covariance of the corresponding parameters of the predictive model (Eq. 1).

3.3 Query Algorithm

Given the set of pairs of features, we would like to query first the similarity of the pair that will give us maximal improvement on predictive performance. Our query algorithm chooses the pair of features that maximizes the expected utility, which is a standard criterion in decision theory. The utility is task dependent – in our prediction task it is natural to choose the utility to be information gain of the predictions. The same rationale was earlier used successfully by [Daee *et al.*, 2017], for querying individual variables. The larger the information gain, the greater impact the feedback will have on the posterior predictive distribution. This is defined as the expected Kullback–Leibler divergence (KL) between the posterior predictive distribution before seeing the feedback and after giving the feedback:

$$\mathbb{E}_{\pi(\tilde{f}_{ij}|\mathcal{D},\mathcal{F})}\left[\sum_n \text{KL}\left[\pi\left(\tilde{y} \mid \boldsymbol{x}_n, \mathcal{D}, \mathcal{F}, \tilde{f}_{ij}\right) \| \pi(\tilde{y} \mid \boldsymbol{x}_n, \mathcal{D}, \mathcal{F})\right]\right] \tag{3}$$

where $\mathcal{F}$ is the set of feedbacks given up to the current step, and the summation goes over the training data. The expectation is over the unknown feedbacks, given everything observed so far. The posterior predictive distribution of the feedback, $\pi\left(\tilde{f_{ij}} \mid \mathcal{D}, \mathcal{F}\right)$, is obtained using the data and all the feedback given by the expert up to the current iteration.

Computational Complexity

In each iteration, we need to compute the utilities for all pairs of features, except those to which the expert already gave feedback in the previous iterations. The complexity is $O(p^2)$ which is expensive if p is large. To reduce this complexity, we used the idea presented in [Xiong *et al.*, 2015]: we construct a smaller pool of the pairs by randomly sampling from the original set of feature pairs, excluding the pairs to which the

expert gave feedback previously. We then select the most informative pair from among the pairs in the smaller pool based on their utilities. Denoting the randomly selected pool by R_p, the objective function for the query algorithm will change to

$$(ij)^* = \arg \max_{(ij) \in R_p} (\text{Expected Information Gain}), \quad (4)$$

where the expected information gain is computed using Equation 3. Although the optimal pair of R_p might not necessarily be optimal for the full set, the amount of degradation in the performance is negligible. This is due to the fact that in real data sets with large number of features, there will be significant redundancies among the pairs. The complexity reduces from $O(p^2)$ to $O(\mid R_p \mid)$, where $\mid R_p \mid$ denotes the cardinality of R_p.

The above strategy is effective only if the selected pair from R_p can be guaranteed to be near optimal with high probability. A pair is near-optimal if it is among the ϵ top-ranked pairs in the original set, where ϵ is a small scalar (e.g. 0.001). The near-optimality of the selected pair from the smaller pool can be proven by the following proposition:

Proposition 1. The best pair in R_p according to Equation 4 is among the ϵ top-ranked pairs of the original set with probability $1 - (1 - \epsilon)^{|R_p|}$.

Proof. Based on the definition of near-optimal pairs, the probability that a pair does not belong to the near-optimal pairs is $1 - \epsilon$. Since the size of the smaller pool is $\mid R_p \mid$, the probability of obtaining a near optimal query is $1-(1-\epsilon)^{|R_p|}$.

By setting $\mid R_p \mid$ to a reasonable multiple of p, e.g. $10p$, we reduce the complexity from $O(p^2)$ to $O(p)$, while with high probability (probability of ~ 0.99 for a moderate size $p = 400$), the selected pair from R_p will be among the top 0.1 percent pairs in the original set. To increase computational speed, we implemented computation of the utilities in parallel. This is straightforward since utilities of different pairs are independent.

3.4 Application To Linear Regression

The described approach is general and could, for example, be applied directly to Gaussian process regression where $\boldsymbol{\theta}$ would be the predictive function evaluations at the training data points. In this paper, we apply the method to the important case of linear regression with "small n, large p" data sets, which occurs in a wide array of practical applications. In particular, we take $\boldsymbol{\theta} = \boldsymbol{\beta}$ to be the regression weights. We assume availability of an expert who has prior knowledge on the pairwise similarities of the role of the features in the prediction task.

The linear regression model is defined as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad (5)$$

where $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ is the matrix of predictors, $\boldsymbol{y} \in \mathbb{R}^{n \times 1}$ is the vector of all dependent variables, $\epsilon \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ is the vector of residuals, and $\boldsymbol{\beta} \in \mathbb{R}^{p \times 1}$ is the regression coefficient vector to be inferred. With these assumptions, $\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\sigma}^2 \sim \mathcal{N}\left(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}\right)$. The goal is to learn the posterior distribution of $\boldsymbol{\beta}$ given the training data and the expert feedback. The proposed approach assumes the following prior distribution for $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} \mid \sigma^2, \tau^2, \boldsymbol{\gamma} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma^2 \tau^2 \boldsymbol{K}\right), \quad (6)$$

where $\boldsymbol{K}$ is the covariance matrix defined by the Gaussian covariance function $k_{\boldsymbol{\gamma}}$ and we have also introduced a scalar magnitude parameter τ^2. The expert knowledge is used to construct the covariance matrix $\boldsymbol{K}$ through which it affects the prediction.

Formulation of the model is completed by defining the conjugate inverse gamma distributions for the hyper-parameters $\sigma^{-2} \sim \text{Gamma}\,(a_\sigma, b_\sigma)$ and $\tau^{-2} \sim \text{Gamma}\,(a_\tau, b_\tau)$, where a_σ, b_σ, a_τ and b_τ are constant values. Setting the values of the hyper-parameters is discussed in Section 4.

3.5 Posterior Inference And Prediction

Parameters of the proposed model are $\{\boldsymbol{\beta}, \sigma^2, \tau^2, \boldsymbol{\gamma}, \xi\}$. Due to the complexity of the model, there is no closed-form posterior distribution. Instead, we obtain maximum a posteriori (MAP) estimates of the parameters by implementing the model in the probabilistic programming language Stan [Carpenter *et al.*, 2016].To reduce computational burden of computing the MAP estimate of $\boldsymbol{\beta}$, which requires the inversion of the covariance matrix $\boldsymbol{K}$, we marginalized the likelihood over $\boldsymbol{\beta}$ and σ^2. This results in a Student's t-distribution with $2a_\sigma$ degrees of freedom for the marginal likelihood, $\boldsymbol{y} \sim \text{MVSt}_{2a_\sigma}\left(0, \frac{b_\sigma}{a_\sigma}\boldsymbol{\Sigma}\right)$, with $\boldsymbol{\Sigma} = \boldsymbol{I} + \tau^2 \boldsymbol{X}\boldsymbol{K}\boldsymbol{X}^T$. The matrix $\boldsymbol{\Sigma}$ is of dimension $n \times n$ which is much smaller than $p \times p$. MAP estimate of $\boldsymbol{\gamma}$ is used to compute $\boldsymbol{K}$. The joint posterior distribution of $\boldsymbol{\beta}$ and σ^2 is then obtained as a Normal-Inverse-Gamma distribution, $\text{NIG}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*, a^*, b^*)$, with parameters

$$\begin{aligned}
\boldsymbol{\mu}^* &= \left(\left(\tau^2 \boldsymbol{K}\right)^{-1} + \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{y}, \\
\boldsymbol{\Sigma}^* &= \left(\left(\tau^2 \boldsymbol{K}\right)^{-1} + \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}, \\
a^* &= a_\sigma + \frac{N}{2}, \\
b^* &= b_\sigma + \frac{1}{2}\left[\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - (\boldsymbol{\mu}^*)^{\mathrm{T}}(\boldsymbol{\Sigma}^*)^{-1}\boldsymbol{\mu}^*\right].
\end{aligned}$$

Finally, predictions for the future input $\tilde{\boldsymbol{x}}$ can be done using the posterior predictive distribution:

$$\begin{aligned}
\pi(\tilde{y} \mid \boldsymbol{y}) &= \int \pi(\tilde{y} \mid \boldsymbol{\beta}, \sigma^2)\pi(\boldsymbol{\beta}, \sigma^2 \mid \boldsymbol{y}) d\boldsymbol{\beta} d\sigma^2 \\
&= \mathrm{t}_{2a^*}\left(\tilde{\boldsymbol{x}}^T \boldsymbol{\mu}^*, \frac{b^*}{a^*}(1 + \tilde{\boldsymbol{x}}^T \boldsymbol{\Sigma}^* \tilde{\boldsymbol{x}})\right), \quad (7)
\end{aligned}$$

where t_{2a^*} denotes the univariate Student's t-distribution with $2a^*$ degrees of freedom.

4 Experimental Results

In this section, the performance of the proposed method is evaluated in several "small n, large p" regression problems on both simulated and real data with a simulated user. To the best of our knowledge, no other method has been proposed for precisely the same task, i.e., improving generalization by constructing a prior distribution using feedback on pairs of features. Therefore, we compare with two natural alternatives:

- random query,

- non-sequential version of our algorithm, which computes the utilities once, before observing user feedback, and never updates the utilities.

4.1 Synthetic Data

Data Generation

We simulate data[1] from a linear regression model given by $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ and $\sigma^2 = 5$. The data are generated similarly to [Li *et al.*, 2010], with the difference that instead of manually constructing the optimal regression coefficient, $\boldsymbol{\beta}_{opt}$, we construct the optimal $\boldsymbol{\gamma}$, denoted by $\boldsymbol{\gamma}_{opt}$, and the meta-features. They are used to construct the optimal covariance structure, $\boldsymbol{K}_{opt}$, using which we sample $\boldsymbol{\beta}_{opt}$ from $\mathcal{N}\left(\boldsymbol{0}, \sigma^2\tau^2\boldsymbol{K}_{opt}\right)$, assuming $\tau^2 = 1$. The idea is to check whether or not the model can learn $\boldsymbol{\gamma}_{opt}$ and consequently $\boldsymbol{\beta}_{opt}$ using the feedback given by an oracle. We set the number of features to 25 which are divided into four groups of sizes 10, 5, 5, 5. Features within each group are highly correlated while features across groups are independent. Meta-features are vectors of size 9 and dimensions 1, 2, 4, 6 are assumed to be important for learning the optimum covariance structure, i.e. $\boldsymbol{\gamma}_{opt} = [1, 1, 0, 1, 0, 1, 0, 0, 0]^T$. The optimum covariance matrix constructed using $\boldsymbol{\gamma}_{opt}$ and meta-features is shown in Figure 2.a.

Training and test outputs are generated by randomly sampling $\boldsymbol{X}_{tr} \in \mathbb{R}^{15\times25}$ and $\boldsymbol{X}_{te} \in \mathbb{R}^{1000\times25}$ from a standard normal distribution, multiplying them by $\boldsymbol{\beta}_{opt}$ and adding normally distributed noise with variance σ^2.

Learning Of The Model And Results

The hyperparameters of the model are $a_\sigma = 2$ and $b_\sigma = 7$ for σ^2 and $a_\tau = 2$ and $b_\tau = 4$ for τ^2, to reflect relatively vague information on the residual variance and scaling parameter, respectively. The prior for $\boldsymbol{\gamma}$ is set to $\mathcal{N}^+(1, 0.5)$ emphasizing that most of its mass is in the range $[0, 2]$, allowing the similarity to become insensitive to some features (with γ near 0) and sensitive to others (with large γ values) if data supports it. *A priori* we expect almost all dimensions of the meta-features to be equally important, i.e., $\boldsymbol{\gamma}_0 \cong [1, 1, 1, 1, 1, 1, 1, 1, 1]^T$. In other words, *a priori* any difference in any dimensions of the meta-feature will affect the (dis)similarity of the features. The hyperparameters of the threshold variable are $\mu_\xi = 20$ and $\sigma_\xi^2 = 10$ to reflect vague information on the threshold and at the same time cover wide ranges of values.

The model was first trained to obtain an estimate of the regression coefficients without any feedback ($\boldsymbol{\beta}_0$). The posterior of the regression coefficients is then used to estimate the utility of each pair (to be used in the next round) and to compute the output of the test data. In each round, the simulated user gives one feedback for a pair queried using three different algorithms, i.e., random, non-sequential (using the utilities computed in round 0) and sequential (updates utilities of each pair, except those to which the user already gave feedback, using estimated parameters in the previous round).

[1]Codes and data are available in https://github.com/homayunafra/Human-in-the-loop-Active-Covariance-Learning-for-Improving-Prediction-in-Small-Data-Sets

Figures 2.b-d show how $\boldsymbol{K}$ evolves using feedback on pairs queried by the sequential model. The total number of possible feedbacks is $\frac{25\times24}{2} = 300$; using only one third of the feedbacks, the sequential model learned almost the optimal covariance structure.

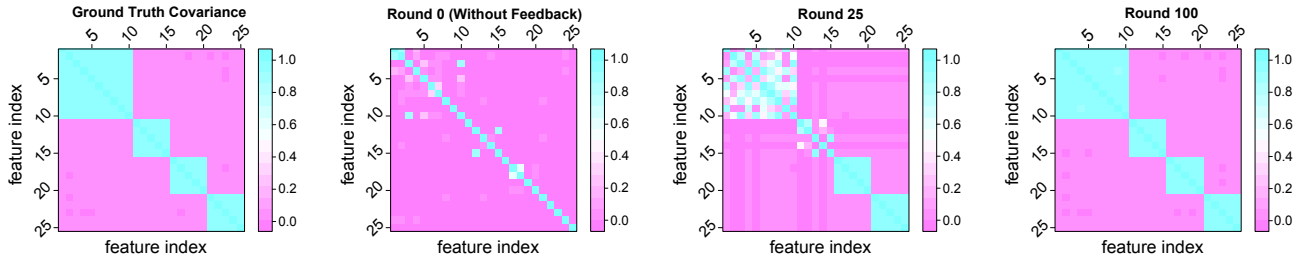
We ran the model 50 times with different randomly generated data with the setting explained above. The average predictive performances of the model with different query algorithms are shown in Figure 3. The sequential model obtains the same amount of improvement faster than the random and non-sequential models. Both the random and non-sequential models also perform well, which is due to the simple structure of the data.

4.2 Real Data With Simulated Expert

We tested the proposed method in the task of review ratings prediction from textual reviews, using subsets of Amazon review data and Yelp data. Both data sets contain textual reviews represented as bag-of-words and their corresponding ratings, integer values in the range $1 - 5$. Features correspond to keywords (unigrams and bigrams). For Amazon, we use the *kitchen appliances* subset which contains 5149 reviews [Blitzer *et al.*, 2007]. For *Yelp*, we consider 4086 reviews from the year 2004. In our analysis of both data sets, we removed stop words [Salton, 1971] and kept features that appeared in at least 100 reviews. This resulted in data sets containing 340 features for Amazon and 241 features for Yelp. From each data set, 1000 randomly selected reviews were used for training and tesing of the model and the rest as an "expert data set" for constructing the simulated expert. Among the 1000 reviews for the model, 10% were randomly selected for training and the remaining 90% for testing. This results in a training set of size 100×340 for Amazon and of size 100×241 for Yelp, both of which small n, large p. Training data were normalized to have zero mean and unit variance. As meta-features, we used the transpose of the unnormalized training data. This is the simplest possible meta-feature where features (keywords) that are not in the same documents are dissimilar and vice versa. Better meta-features can be defined using word2vec [Mikolov *et al.*, 2013] or more advanced representation of the keywords; here we show that the model works fairly well even with such naive meta-features.

Simulated Expert Feedback

To construct the simulated expert, we trained a sparse regression model using LASSO [Friedman *et al.*, 2010] on the expert data set. Features with positive regression coefficients are considered pairwise similar. Features with negative coefficients are also similar to each other and dissimilar to those with positive coefficients. Features with zero weight are assumed to be difficult for the simulated expert to evaluate and the feedback will be "I don't know" for any pair containing at least one of those features. Since our goal is to evaluate the predictive performance of the models and feedback on these features has no effect on models' performance, we remove all pairs containing any of these features to make the runs faster. For both data sets, there are almost 20 features in each of the two groups, resulting in $(40 \times 39)/2 = 780$ pairs in total.



(a) (b) (c) (d)

Figure 2: Evolution of $\boldsymbol{K}$ w.r.t. different number of feedbacks queried by the sequential model.

Using proposition 1 for the sequential model, in each iteration we randomly selected 400 pairs among which the most informative one will be selected to query to the expert.

The hyperparameters of the model were set to the same values as in the synthetic data, with the only difference that for both data sets we set $\tau^2 = 0.01$, obtained by cross-validation.

Results

Figure 4 shows the predictive performance of query algorithms on each data set. Results are averaged over 50 runs with randomly selected training, test, and expert data. Query algorithms ask one question at a time and the model was ran for 100 rounds. For all algorithms and on both data sets, additional knowledge from the simulated expert reduces the prediction error. Yet the amount of reduction depends on the query algorithm. The proposed method in sequential mode obtains fastest improvement. The random model performs better than the non-sequential model. This is expected since the non-sequential model is prone to querying redundant pairs; for instance when there is a cluster of similar features that are highly informative, asking a few of them is sufficient to inform of their grouping. However, the non-sequential model will query all the possible pairs since it does not update the utilities. To compare the effect of learned covariances using expert feedback on regularizing the model, we also compare to the baselines that regularize the model: LASSO, Ridge and Elastic net. It can be seen that a relatively small number of expert feedback could regularize the model well and achieve good prediction performance compared to the commonly used regularization methods.

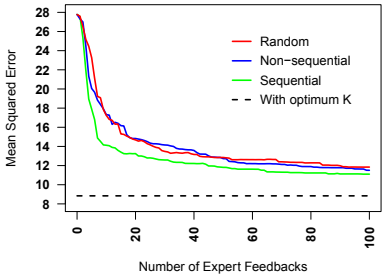
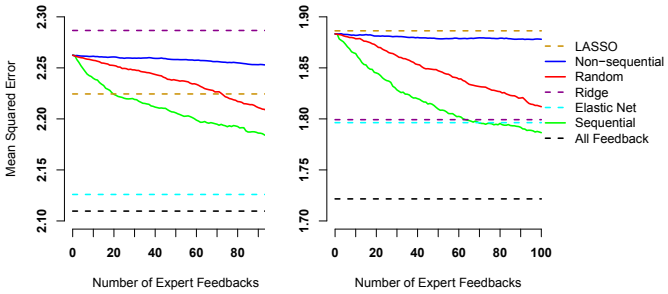


Figure 3: Comparison of predictive performances of different query algorithms in synthetic data.



(a) Amazon (b) Yelp

Figure 4: Mean squared error w.r.t. the number of feedbacks on pairs of features. MSE values are averages over 50 independent runs.

Knowledge Elicitation vs. Collecting More Samples

In this subsection, we contrast the improvements in the predictions brought by feedback on feature similarities to improvements gained if additional samples could be measured. Additional samples are selected from the expert data set to be used as training data, and for choosing them we use two strategies: random selection of samples (shown by **RND** in Table **??**), and active learning strategy (shown by **ACT**), which selects samples based on maximizing expected information gain (similar to [Seeger, 2008]). For feedback collection, we adopt our sequential knowledge elicitation approach (shown by **SEQELC** in Table **??**). All strategies have the same "small n, large p" setting as starting points with $\boldsymbol{X}_{tr} \in \mathbb{R}^{100\times 241}$ for Yelp and $\boldsymbol{X}_{tr} \in \mathbb{R}^{100\times 340}$ for Amazon.

Table **??** shows how many *feedback* and respectively, how many *additional samples* are required to reach preset levels of MSE. According to the table, a particular performance is obtained by a comparable number of expert feedback and additional data. It should be noted that values obtained for the sequential elicitation model are for its best case scenario since we already removed "I don't know"s which results in a smaller search pool.

5 Discussion and Conclusion

We proposed a knowledge elicitation approach that incorporates pairwise constraints provided by a human expert for constructing an informative prior distribution. The knowledge elicitation problem is formulated as sequential prob-

		More Samples		More Feedbacks
	MSE	**RND**	**ACT**	**SEQELC**
Yelp	1.85	5	5	13
	1.80	15	9	60
Amazon	2.20	20	14	55
	2.18	27	17	90

Table 1: Comparison of the number of required samples and feedbacks to reach a particular MSE level. Values are averages over 50 independent runs. Initial MSE values for Yelp and Amazon are 1.88 and 2.26, respectively.

abilistic inference which combines expert knowledge with training data. To avoid overwhelming the expert, we proposed a query selection approach that facilitates efficient interaction. To further increase the efficiency of the interaction, we formulated a theoretically motivated subsampling approach. Results for "small n, large p" problems in simulated and real data with simulated users showed improved prediction accuracy already with a small number of feedbacks.

From our experience, the proposed algorithm works well for moderately high-dimensional problems (e.g., up to 500 predictors). For very high-dimensional problems, with more than thousand predictors, the computations become infeasible due to the large number of pairs. However, the presented approach is general and can be used to complement other existing knowledge elicitation techniques where the user is asked about relevance of features. In this case we can query only pairs of relevant features which significantly improves the feasibility of the model.

Acknowledgments

This work was financially supported by the Academy of Finland (grants 294238, 319264, and 313195), by the Vilho, Yrjö and Kalle Väisälä Foundation of the Finnish Academy of Science and Letters and by Foundation for Aalto University Science and Technology. The authors acknowledged the computational resources provided by the Aalto Science-IT Project.

References

[Afrabandpey *et al.*, 2017] Homayun Afrabandpey, Tomi Peltola, and Samuel Kaski. Interactive prior elicitation of feature similarities for small sample size prediction. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 265–269. ACM, 2017.

[Blitzer *et al.*, 2007] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, 2007.

[Carpenter *et al.*, 2016] Bob Carpenter, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Michael A Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 20:1–37, 2016.

[Daee *et al.*, 2017] Pedram Daee, Tomi Peltola, Marta Soare, and Samuel Kaski. Knowledge elicitation via sequential probabilistic inference for high-dimensional prediction. *Machine Learning*, 106(9-10):1599–1620, 2017.

[Fan and Li, 2006] Jianqing Fan and Runze Li. Statistical challenges with high dimensionality: feature selection in knowledge discovery. In *Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, pages 595–622, 2006.

[Friedman *et al.*, 2010] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[Garthwaite *et al.*, 2008] Paul H Garthwaite, James B Chilcott, David J Jenkinson, and Paul Tappenden. Use of expert knowledge in evaluating costs and benefits of alternative service provisions: A case study. *International Journal of Technology Assessment in Health Care*, 24(3):350–357, 2008.

[Haakma *et al.*, 2011] Wieke Haakma, Laura Bojke, Lotte Steuten, and Maarten IJzerman. Pmd4 expert elicitation to populate early health economic models of medical diagnostic devices in development. *Value in Health*, 14(7):A244–A245, 2011.

[Krupka and Tishby, 2007] Eyal Krupka and Naftali Tishby. Incorporating prior knowledge on features into learning. In *Artificial Intelligence and Statistics*, pages 227–234, 2007.

[Li *et al.*, 2010] Qing Li, Nan Lin, et al. The Bayesian elastic net. *Bayesian Analysis*, 5(1):151–170, 2010.

[Micallef *et al.*, 2017] Luana Micallef, Iiris Sundin, Pekka Marttinen, Muhammad Ammad-ud din, Tomi Peltola, Marta Soare, Giulio Jacucci, and Samuel Kaski. 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, pages 547–552. ACM, 2017.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[O'Hagan, 1998] Anthony O'Hagan. Eliciting expert beliefs in substantial practical applications. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):21–35, 1998.

[Rasmussen and Williams, 2006] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[Salton, 1971] G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

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

[Sundin *et al.*, 2018] I Sundin, T Peltola, L Micallef, H Afrabandpey, M Soare, M Majumder Mamun, P Daee, C He, B Serim, A Havulinna, et al. Improving genomics-based predictions for precision medicine through active elicitation of expert knowledge. *Bioinformatics*, 34(13):i395–i403, 2018.

[Xiao *et al.*, 2018] Cao Xiao, Yan Jin, Ji Liu, Bo Zeng, and Shuai Huang. Optimal expert knowledge elicitation for Bayesian network structure identification. *IEEE Transactions on Automation Science and Engineering*, 2018.

[Xiong *et al.*, 2015] Sicheng Xiong, Yuanli Pei, Romer Rosales, and Xiaoli Z Fern. Active learning from relative comparisons. *IEEE Transactions on Knowledge and Data Engineering*, 27(12):3166–3175, 2015.

[Yang *et al.*, 2007] Liu Yang, Rong Jin, and Rahul Sukthankar. Bayesian active distance metric learning. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, pages 442–449. AUAI Press, 2007.