Automatic Choice of Control Measurements

Gayle Leen, David R. Hardoon, and Samuel Kaski

[1] Helsinki University of Technology
Department of Information and Computer Science
P.O. Box 5400, FIN-02015 TKK, Finland
gleen@cis.hut.fi,samuel.kaski@tkk.fi
[2] University College London
Dept. of Computer Science, Gower Street, London WC1E 6BT U.K.
D.Hardoon@cs.ucl.ac.uk

Abstract. In experimental design, a standard approach for distinguishing experimentally induced effects from unwanted effects is to design control measurements that differ only in terms of the former. However, in some cases, it may be problematic to design and measure controls specifically for an experiment. In this paper, we investigate the possibility of *learning to choose* suitable controls from a database of potential controls, which differ in their degree of relevance to the experiment. This approach is especially relevant in the field of bioinformatics where experimental studies are predominantly small-scale, while vast amounts of biological measurements are becoming increasingly available. We focus on finding controls for differential gene expression studies (case vs control) of various cancers. In this situation, the ideal control would be a healthy sample from the same tissue (the same mixture of cells as the tumor tissue), under the same conditions except for cancer-specific effects, which is almost impossible to obtain in practice. We formulate the problem of learning to choose the control in a Gaussian process classification framework, as a novel paired multitask learning problem. The similarities between the underlying set of classifiers are learned from the set of control tissue gene expression profiles.

1 Introduction

We approach the problem of learning to choose suitable control measurements for an experiment from a database of potential controls using a novel multi-task learning formulation. We begin by motivating the problem from a bioinformatics standpoint, and then formulate it in machine learning terms in Section 1.1.

Microarray technologies enable the simultaneous interrogation of the expression of thousands of genes, revealing the intricate workings of a cell on a molecular level. The ability to study the entire genomic profile in this way opens up many exciting research possibilities; biologists can characterise a cell in terms of its gene expression levels, and analyse how its profile varies between different conditions, leading to insights which could potentially benefit drug development, disease diagnosis, functional genomics, and many other fields. Due to the potential of this research, vast amounts of gene expression measurements under different experimental conditions have been collected, and many public databases are available such as ArrayExpress [1] and the Gene Expression Omnibus [2].

A typical experimental set-up to investigate the effect of some factor, for instance a disease or drug treatment, is to compare each gene's expression level in the affected sample with a control sample. These differential gene expression studies can lead to identification of possible gene targets for further analysis, biomarkers for a disease etc. However, this procedure is prone to error; gene expression data is inherently noisy, due to factors such as measurement noise, patient-specific and laboratory-specific variation. Additionally, in general, these experiments only consider a small set of samples, since often there are only a few test cases (e.g. patients) available to the laboratory carrying out the analysis. The presence of these potential sources of noise makes it especially crucial to select a good set of control samples for a differential gene expression study.

However, designing controlled experiments may not be a straightforward task. Ideally, a large set of control samples would be measured by the same laboratory conducting the experimental study, but in practice, there is only a small control set (or none) available, which has to be augmented through selecting samples from public repositories of gene expression data. This task is problematic; in addition to the bias induced in samples due to laboratory and patient-specific effects, there is no established ontology for sample / tissue annotation, resulting in vague or missing labels, or terminology that is inconsistent between experiments. Furthermore, there may only be a very small number of the desired control samples available. One typical way to resolve this problem is to average over a large set of available samples, which are only partially related to the correct type of control sample. Obviously this approach would be suboptimal if there is a large number of unrelated samples, and a more sensible solution, which we address in this paper, would be to weight the pool of samples according to their relevance to the study.

There have been recent studies that propose a number of methodologies for gene expression analysis: clustering tissue and cell samples into a number of groups according to overlapping feature similarities [3–5], classifier methodology as an exploration technique to identify mislabeled and questionable tissue samples [6] and analysing the origin of tissue samples by explicitly modeling each tissue as a probabilistic sample from a population of related tissues [7]. However none of the current gene/tissue analysis studies, to the knowledge of the authors, explore the issue of learning how to identify suitable controls to affected samples when they cannot be specifically designed.

1.1 Control sample selection and multitask learning approaches

This work proposes a novel approach to a frequently occurring and complex problem in experimental design for bioinformatics. We focus on the learning task of how to identify suitable controls for case samples, by using a novel paired multi-task learning framework. We formulate the problem as follows: The suitable controls for each experiment form a group of controls. These groups will be considered as classes, and the task is to classify each case sample to one of these classes. In learning the classification, we need to use knowledge about the relationships between the case and the control samples. This pairing will in effect be transferred to new pairs.

Suppose that we have N_H control samples $\mathbf{Y} = \{\mathbf{y}_1, ...\mathbf{y}_{N_H}\}$, which we can classify into one of K control classes: $t_y \in \{1, ..., K\}$, so that we have a labeled data set $\mathcal{D}_Y = \{\mathbf{y}_n, t_{y,n}\}_{n=1}^{N_H}$. We also have N_C case samples $\mathbf{X} = \{\mathbf{x}_1, ...\mathbf{x}_{N_C}\}$, for which there are known mappings to control classes $\mathcal{D}_X = \{\mathbf{x}_n, t_{x,n}\}_{n=1}^{N_C}$. For a new case sample (and case type not contained in the training set), $\mathbf{x}_m$, we want to predict the control class, given the preexisting mappings $\mathcal{D}_X$ and $\mathcal{D}_Y$, guided by the relationships between the different control classes i.e.

$$p(t_m \mid \mathbf{x}_m, \mathcal{D}_X, \mathcal{D}_Y) \tag{1}$$

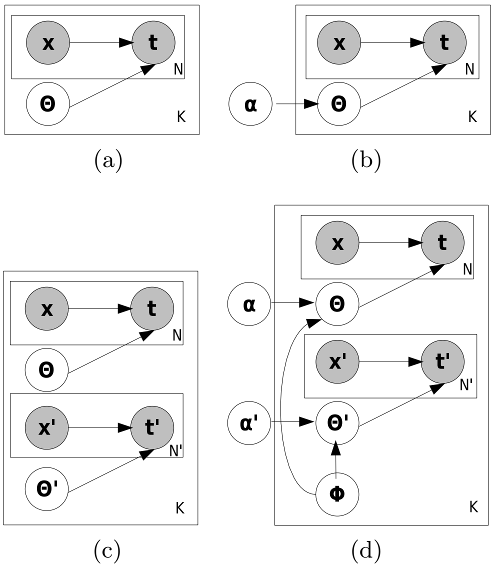
This is a multiclass (K classes) classification problem, which borrows statistical strength from $\mathcal{D}_Y$ about the relatedness of the classes. This represents the idea that if two control classes a and b are similar (found from the relationship between sets $\mathbf{Y}_a$ and $\mathbf{Y}_b$), then they are both likely to be used as control for the same case profile. In effect, $\mathbf{Y}$ augments the labeling for the control classes.

Our formulation of the control selection problem has resonance in several related subfields of machine learning, which address the issue of augmenting the data set for a learning problem with other partially related sources of information. The unifying concept is that the joint distribution of the inputs $\mathbf{x}$ and outputs t differs between the desired learning problem and the auxiliary learning problems; the existing approaches differ in the way that this shift is characterised. Transfer learning and multitask learning approaches [8–10] assume that information can be transferred from auxiliary, partially relevant tasks to the task(s) of interest, and generally assume the same input distributions $p(\mathbf{x})$ between tasks, with a task specific $p(t \mid \mathbf{x})$. Another family of approaches assumes that $p(t \mid \mathbf{x})$ remains unchanged between different tasks while the input domain $p(\mathbf{x})$ differs; they include learning under covariate shift [11, 12] and domain adaptation [13]. In these terms, the novel problem that we address in this paper can be called *paired multitask learning.*

1.2 Paired Multitask Learning

In a traditional multitask learning scenario, there is a set of K related tasks [3] which we will here call *primary tasks.* For example, given a set of inputs and labels $\{x_n, t_n\}_{n=1}^{N}$, $t_n \in 1, ..., K$, eash task could be to classify the samples to one of the K classes, learned by finding $\{p(t_i \mid \mathbf{x}, \theta_i)\}_{i=1}^{K}$, where θ_i is the parameterisation for the ith classifier (see Figure 1a). Information could be shared among tasks, for instance through a shared parameter α (Figure 1b).

[3] In this paper we consider situations where all tasks have the same set of inputs and outputs



(a) (b)

(c) (d)

Fig. 1: Schematic illustration of statistical strength sharing in multitask learning scenarios. Learning a set of K tasks as in (a) amounts to finding different parameterisations $\theta_i, i = 1, ..., K$ for the tasks. If the tasks are assumed to be related, multitask learning approaches assume some shared structure across all K tasks through a common parameterisation via α (b). We consider the situation where there are K pairs of tasks (c), and propose the structure in (d) to share information between the tasks. There is shared structure within each task set's parameterisation θ, θ' through α, α' and across each of the K pairs through ϕ.

In our framework, we consider an additional level of dependencies: we have an auxiliary set of tasks $\{x'_n, t'_n\}_{n=1}^{N'}$, $t'_n \in 1, ..., K$, where $p(\mathbf{x}') \neq p(\mathbf{x})$ (Figure 1c). We transfer information about the *relatedness of the auxiliary set of tasks* to the set of tasks of interest, by adding one more level of parameterisation, $\varPhi$ (Figure 1d). This is achieved by finding a corresponding set of classifiers $\{p(t'_i \mid \mathbf{x}, \theta'_i)\}_{i=1}^{K}$, coupled through $p(\theta'_1, ..., \theta'_K \mid \alpha')$. Information is shared between the two task sets, auxiliary and primary, through the shared parameterisation $\varPhi$ which couples the pairs of corresponding tasks. The proposed model uses flexible assumptions about the shift between the auxiliary tasks and the primary tasks, since we assume different sets of conditional distributions, linked only through shared parameterisation $\varPhi$.

Our approach is remotely related to [9, 10, 14] in that the Gaussian process framework is used to capture inter-task similarity, and partially to the recent transfer learning approach [15] where a sample from any task is weighted to match the joint distribution of the target task $p(\mathbf{x}, t)$. The weights are derived

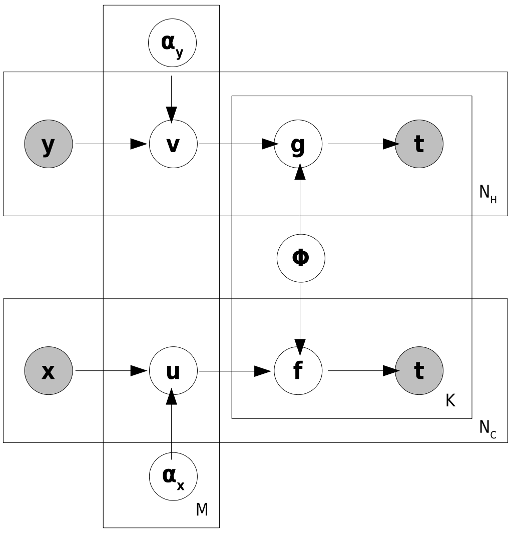


Fig. 2: Graphical model. The functions $\mathbf{f}_t$ and $\mathbf{g}_t$, which classify the control and case profiles $\mathbf{x}$ and $\mathbf{y}$, respectively, to the tth (of K) control classes, are related through parameters $\boldsymbol{\Phi}_t$.

from an input-output pair's probability of belonging to the target task, which is calculated from a multiclass classifier learned on the pool of samples. Whereas in [15] this is a prior step to the actual learning of tasks, in our work we learn intertask information jointly with learning the tasks, through learning an auxiliary set of tasks.

The rest of this paper is organised as follows: In the following section, we discuss our proposed model for addressing the problem of selecting appropriate control samples with our modeling assumptions and model inference. We continue to give our experiments using the proposed model in Section 3 and finally, we give our concluding discussion in Section 4.

2 Gaussian process classification for paired multitask learning

In this section we introduce a framework for paired multitask learning. There are K pairs of tasks, where the ith pair consists of learning to classify cancer profiles and control profiles to the ith control class. The graphical model is shown in Figure 2, and we next explain its structure.

We can classify the nth profile $\mathbf{y}_n$ from the control sample set into one of K classes, by learning the mappings to class labels; we assume that the probability of the labels depends on a set of K functions $\{f_1(\mathbf{y}_n), ..., f_K(\mathbf{y}_n)\}$ evaluated at $\mathbf{y}_n$. Using a multinomial probit link function, we define the mapping between the function values and class labels as:

$$P(t_n = c \mid \{f_1(\mathbf{y}_n), ..., f_K(\mathbf{y}_n)\}) = E_{p(z)}\left(\prod_{i \neq c} \Phi(z + f_c(\mathbf{y}_n) - f_i(\mathbf{y}_n))\right) \quad (2)$$

where $\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(u \mid 0, 1)du$ is the cumulative normal density function. Similarly, we assume that an expression profile $\mathbf{x}_n$ from the case samples can be classified into the K classes, where the probability of the class labels also depend on a set of corresponding underlying functions $\{g_1(\mathbf{x}_n), ..., g_K(\mathbf{x}_n)\}$ in an analagous link function to (2). We denote the jth functions evaluated at the data points as $\mathbf{f}_j = [f_j(\mathbf{y}_1), ..., f_j(\mathbf{y}_{N_H})]^\top$ and $\mathbf{g}_j = [g_j(\mathbf{x}_1), ..., g_j(\mathbf{x}_{N_C})]^\top$, and across all K classes as $\mathbf{f} = \left[\mathbf{f}_1^\top, ..., \mathbf{f}_K^\top\right]^\top$ and $\mathbf{g} = \left[\mathbf{g}_1^\top, ..., \mathbf{g}_K^\top\right]^\top$.

In this work, we are interested in transferring information about the interrelatedness of the classes from one task (mapping control samples to tissue classes) to the main task (mapping cancer samples to tissue classes). For a standard multiclass Gaussian process classification task, the K functions are given Gaussian process priors, which are assumed to be uncorrelated across classes, i.e. $p(\mathbf{f}) = \mathcal{N}(\mathbf{f} \mid \mathbf{0}, \mathbf{K})$, where $\mathbf{K}$ is block diagonal in the class specific covariance functions $\mathbf{K}_1, ..., \mathbf{K}_K$. We take a different approach, and model each function as a linear combination of M latent functions where $M < K$; for the ith pair of functions this is:

$$\mathbf{f}_i = \sum_j \Phi_{i,j}\mathbf{u}_j, \qquad \mathbf{g}_i = \sum_j \Phi_{i,j}\mathbf{v}_j \quad (3)$$

where $\Phi_{i,j}$ is the weight of the jth latent function in the ith function, and $\mathbf{u}_j$ and $\mathbf{v}_j$ are the jth pair of latent functions. This formulation models dependencies between the functions for each multi-class classifier via $\Phi \in \Re^{K \times M}$, and this structure is shared between the two classification tasks. If we place Gaussian process priors over each latent function, $p(\mathbf{u}_j) = \mathcal{N}(\mathbf{u}_j \mid \mathbf{0}, \mathbf{K}_{u,j}), p(\mathbf{v}_j) = \mathcal{N}(\mathbf{v}_j \mid \mathbf{0}, \mathbf{K}_{v,j})$, then the distributions over $\mathbf{f}$ and $\mathbf{g}$ are:

$$p(\mathbf{f} \mid \Phi) = \int p(\mathbf{f} \mid \mathbf{u}, \Phi)p(\mathbf{u})d\mathbf{u} = \mathcal{N}\left(\mathbf{f} \mid \mathbf{0}, (\Phi \otimes \mathbf{I})\mathbf{K}_u(\Phi \otimes \mathbf{I})^\top\right) \quad (4)$$

$$p(\mathbf{g} \mid \Phi) = \int p(\mathbf{g} \mid \mathbf{v}, \Phi)p(\mathbf{v})d\mathbf{v} = \mathcal{N}\left(\mathbf{g} \mid \mathbf{0}, (\Phi \otimes \mathbf{I})\mathbf{K}_v(\Phi \otimes \mathbf{I})^\top\right) \quad (5)$$

where $\otimes$ denotes the Kronecker product, $\mathbf{K}_u$ and $\mathbf{K}_v$ are block diagonal in the class specific covariance functions $\mathbf{K}_{u,1}, ..., \mathbf{K}_{u,K}$ and $\mathbf{K}_{v,1}, ..., \mathbf{K}_{v,K}$ respectively. This prior captures correlations within the latent function sets; the cross covariance functions between $\mathbf{f}_i$ and $\mathbf{f}_j$ is given by $\sum_n \Phi_{i,n}\Phi_{j,n}\mathbf{K}_{u,n}$, and similarly for $\mathbf{g}_i$ and $\mathbf{g}_j$: $\sum_n \Phi_{i,n}\Phi_{j,n}\mathbf{K}_{v,n}$, and this relationship is shared across the two tasks via Φ. The model has similarities to the semiparametric latent factor model [14] in that statistical strength is shared across K GP's through a smaller set of M underlying functions. However in our approach, we use the learned relationship between one set of GP's on the control set to help train a set of GP's on the related set of case samples.

2.1 Inference in the model

We use the data augmentation strategy as detailed in [16], by introducing auxiliary latent variables $\mathbf{f}'$ and $\mathbf{g}'$ in (2) for each classifier i.e. so that we can rewrite the probit link functions (linking $\{f_{n1}, ..., f_{nK}\}$ to $t_{x,n}$ in (2)) as:

$$\begin{aligned} P(t_{x,n} = c \mid \{f_{ni}\}_{i=1}^{K}) &= \int P(t_{x,n} = c \mid \{f'_{ni}\}_{i=1}^{K}) \prod_{i=1}^{K} p(f'_{ni} \mid f_{ni}) df'_{ni} \\ &= \int \delta(f'_{nc} > f'_{nk} \forall k \neq c) \prod_{i=1}^{K} \mathcal{N}(f'_{ni} \mid f_{ni}, 1) df'_{ni} \end{aligned} \tag{6}$$

where we have denoted the ith function of $\mathbf{x}_n$, $f_i(\mathbf{x}_n)$, as f_{ni}. We similarly derive $P(t_{y,n} = c \mid \{g_{ni}\}_{i=1}^{K})$. To train the model we need to find the posterior distribution over $\Theta_x = \{\mathbf{f}', \mathbf{u}\}$, $\Theta_y = \{\mathbf{g}', \mathbf{v}\}$ and also the shared mixing matrix $\mathbf{\Phi}$, its hyperparameters ψ, and covariance function hyperparameters for both classifiers (which we will denote by α_x and α_y). The joint distribution over these quantities is given by

$$\begin{aligned} &p(\mathbf{t}_x, \mathbf{t}_y, \Theta_x, \Theta_y, \Phi, \alpha_x, \alpha_y \mid \mathbf{X}, \mathbf{Y}, \psi) = \\ &p(\mathbf{t}_x, \Theta_x, \Phi, \alpha_x \mid \mathbf{X}) p(\mathbf{t}_y, \Theta_y, \Phi, \alpha_y \mid \mathbf{Y}) p(\Phi \mid \psi) \end{aligned} \tag{7}$$

where

$$\begin{aligned} &p(\mathbf{t}_x, \Theta_x, \Phi, \alpha_x \mid \mathbf{X}) = \\ &\prod_{n=1}^{N_C} \left[\sum_{i=1}^{K} \delta(f'_{ni} > f'_{nk} \forall k \neq i) \delta(t_{x,n} = i) \right] p(\mathbf{f}' \mid \mathbf{u}, \Phi) p(\mathbf{u} \mid \alpha_x, \mathbf{X}) \end{aligned} \tag{8}$$

and

$$\begin{aligned} &p(\mathbf{t}_y, \Theta_y, \Phi, \alpha_y \mid \mathbf{Y}) = \\ &\prod_{n=1}^{N_H} \left[\sum_{i=1}^{K} \delta(g'_{ni} > g'_{nk} \forall k \neq i) \delta(t_{y,n} = i) \right] p(\mathbf{g}' \mid \mathbf{v}, \Phi) p(\mathbf{v} \mid \alpha_y, \mathbf{Y}). \end{aligned} \tag{9}$$

We employ a variational approximation to the above, by finding an ensemble of approximating posterior distributions $Q(\Theta_x)Q(\Theta_y)Q(\Phi)Q(\alpha_x)Q(\alpha_y)$ to $p(\Theta_x, \Theta_y, \Phi, \alpha_x, \alpha_y \mid \mathbf{t}_x, \mathbf{t}_y, \mathbf{X}, \mathbf{Y})$ that maximise the lower bound on the marginal likelihood

$$\begin{aligned} &\log p(\mathbf{t}_x, \mathbf{t}_y, \mid \mathbf{X}, \mathbf{Y}, \psi) \geq \\ &E_{Q(\Theta_x)Q(\Theta_y)Q(\Phi)Q(\alpha_x)Q(\alpha_y)} \{\log p(\mathbf{t}_x, \mathbf{t}_y, \Theta_x, \Theta_y, \Phi, \alpha_x, \alpha_y \mid \mathbf{X}, \mathbf{Y})\} \\ &-E_{Q(\Theta_x)Q(\Theta_y)Q(\Phi)Q(\alpha_x)Q(\alpha_y)} \{\log Q(\Theta_x)Q(\Theta_y)Q(\Phi)Q(\alpha_x)Q(\alpha_y)\}. \end{aligned} \tag{10}$$

The form of the $Q()'s$ are given below:

$$Q(\mathbf{u}) \propto \mathcal{N}(\tilde{\mathbf{f}}' \mid (\tilde{\Phi} \otimes \mathbf{I})\mathbf{u}, \mathbf{I})\mathcal{N}(\mathbf{u} \mid 0, \tilde{\mathbf{K}}) \tag{11}$$

$$Q(\mathbf{f}') \propto \mathcal{N}(\mathbf{f}' \mid (\tilde{\Phi} \otimes \mathbf{I})\tilde{\mathbf{u}}, \mathbf{I}) \prod_{n} [\delta(f'_{ni} > f'_{nk} \forall k \neq i)\delta(t_{x,n} = i)] \tag{12}$$

$$Q(\alpha_x) \propto \mathcal{N}(\tilde{\mathbf{u}} \mid 0, \mathbf{K})\mathcal{G}(\alpha_{x,i} \mid a_i, b_i) \tag{13}$$

where we use Gamma distributions over the hyperparameters of the covariance function a_i, $\{w_i, ..., w_{D_x}\}$, $\tilde{x}$ denotes the posterior mean of $Q(x)$, and similarly for $Q(\mathbf{v})$), $Q(\mathbf{g}')$), and $Q(\alpha_y)$). Finally,

$$Q(\Phi) \propto \mathcal{N}(\tilde{\mathbf{f}'} \mid (\Phi \otimes \mathbf{I})\tilde{\mathbf{u}}, \mathbf{I})\mathcal{N}(\tilde{\mathbf{g}'} \mid (\Phi \otimes \mathbf{I})\tilde{\mathbf{v}}, \mathbf{I}) \prod_{i=1}^{K} \mathcal{N}(\Phi_i \mid 0, \sigma_i \mathbf{I}) \tag{14}$$

where Φ_i denotes the ith row of Φ.

3 Experiments

In this section, we evaluate the model's performance on simulated data, and on a real world data set.

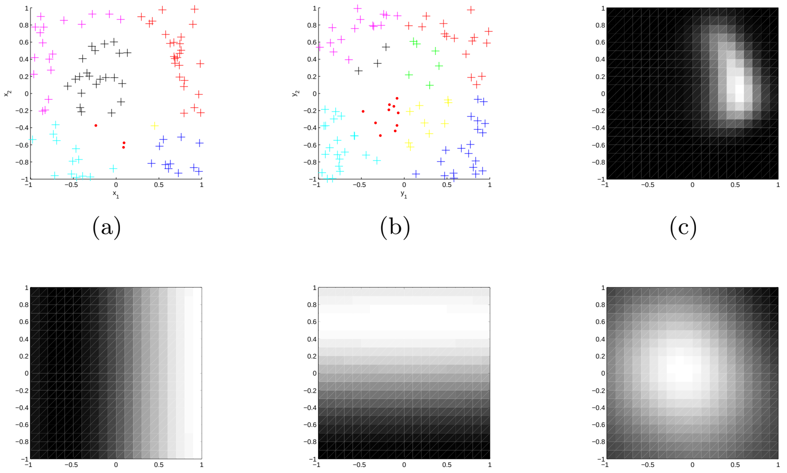
3.1 Experimental details

For all the experiments in this paper we use the following: squared exponential covariance function $k(\mathbf{x}_i, \mathbf{x}_j) = a \exp -\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{W}(\mathbf{x}_i - \mathbf{x}_j)$for all GP's, where a is a scale parameter, and $\mathbf{W}$ is a diagonal matrix with the inverse length scales $\{w_i, ..., w_D\}$ (D the dimension of the data) on the diagonal. We fix the noise level for each of the GP's to 1e-3, and use a distribution of $\mathcal{G}(1, 1)$ over the hyperparameters of the covariance functions. The optimization is sensitive to the initialisation; to initialise Φ, we first calculate a class similarity kernel between the means of each class in $\mathbf{Y}$, and then find the first M principal component vectors. We found that a linear kernel works best in practice.

3.2 Toy data

We demonstrate the model's ability to generalise to unseen classes for a new data point $\mathbf{x}_n$, based on the relationship of the unseen class with the other classes learned from $\mathbf{Y}$ (where all classes are present), for an 8- class classification problem.

We generated a pair of data sets, each containing eight classes. The classes for each data set are shown in Figure 3 (a) and (b). For the two classification tasks, there is the same underlying structure to the set of class boundaries; they



(a) (b) (c)

(d) (e) (f)

Fig. 3: Toy data experiment. Two sets of toy data: **X** in (a), and **Y** in (b) are from the same 8 classes (corresponding classes shown by the different coloured markers). The model learns an underlying set of functions for the first data set, as seen in (d) - (f), and can find a predictive distribution (c) over the missing class from the information learned from the auxiliary sets of tasks (b)

are a mixture of three latent functions, one for horizontal discrimination, one for vertical discrimination, and one that discriminates between inside and outside of the circle. We remove the data points belonging to one of the classes for the first data set, and use these as test data. Figure 3 (c) shows the correctly inferred distribution for the missing class (green in (b)) evaluated over a grid over the input space. The bottom row of the figure shows the three inferred latent functions.

3.3 Cancer profiles

Cancer is a complex disease, arising from genetic abnormalities which disrupt a cell's ordinary functioning, leading to uncontrolled cell proliferation. Identification of the mutated genes that drive the oncogenesis and gaining an understanding of cancer on a molecular basis is one of the key goals of cancer research; consequently there are ongoing coordinated efforts between clinicians, biologists and computer scientists to collect and analyse a large collection of genomic profiles over many different cancer types. While there is potentially a huge amount of useful information about cancer contained in such databases, its extraction presents many methodological challenges for bioinformaticians. Gene expression measurements are likely to contain bias due to factors such as patient-specific and laboratory-specific effects, and typically there are only a small number of samples available for each experimental condition. These factors make it problematic to select a set of pairs of control and normal tissue samples, such that the differential gene expression of the case samples is solely due to cancer-specific variation. However, we can exploit shared information between different sets of experiments: there are similarities (e.g. similar pathway activations) between different cancers, and similarities between normal tissue types.

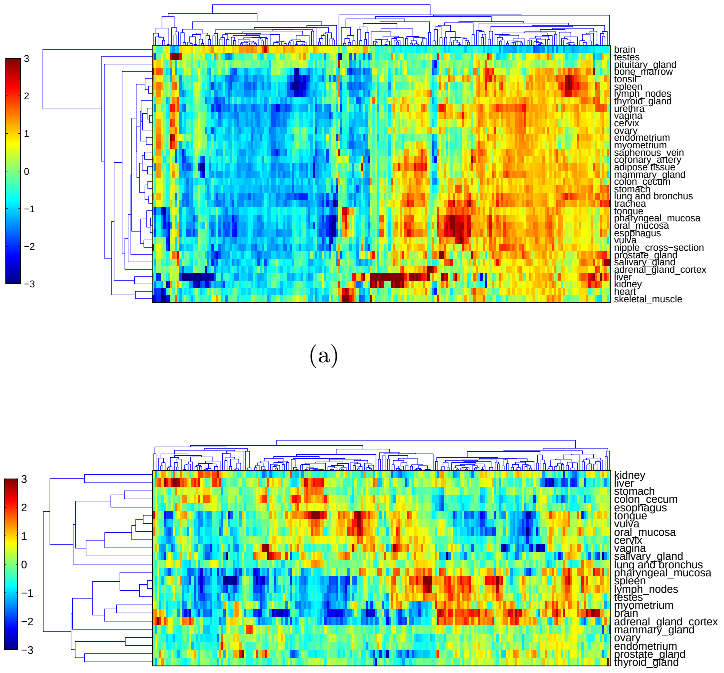
We propose a solution to the problem of control measurement selection for a set of case profiles through using our model to learn the relationship between a pool of controls and cancer profiles, over a wide range of control classes. We show how the model can predict control classes for new cancer samples.

Data. Two publicly available gene expression data sets were taken from the NCBI's Gene Expression Omnibus [2]. The first data set is a series of cancer portraits (accession GSE2109, `https://expo.intgen.org/geo/`), consisting of 1911 clinically annotated gene expression profiles taken from 82 different tumor types in humans. The second data set is a set of 353 gene expression profiles taken over 65 different tissues in the normal human body (accession GSE3526, Neurocrine Biosciences, Inc.). Both data sets were preprocessed using RMA [17]. For each sample, we constructed a feature vector where each element represented the genes' activation in a known biological pathway, according to KEGG[4] gene sets from the Molecular Signatures Database (MSigDB) [18], resulting in a 200-dimensional vector of pathway activations. To calculate each pathway activation, we used the mean of the expression levels of the genes in each pathway. These feature vectors were used as inputs to the model. A visualization of the two data sets is given in Figure 4 (see caption for details).

Based on the annotations for both data sets, we manually classified the normal tissue samples into 35 control classes. For each class we used a maximum of 10 samples. Annotations for the tumor samples include the classification tissue. We then assigned the tumor samples to the control classes for cases where the mapping was evident. For each class we also allowed a maximum of 10 samples, if available, for the training set. Many classes contained few, or even no samples. The rest of samples (where the mapping was known) were assigned to a first test set ('Test Set 1'). Samples where the potential control class was ambiguous, e.g. samples where the classification tissue did not have an equivalent in the control set, or vague: *connective and soft tissue*, *ill defined*, were assigned to a second test set ('Test Set 2').

Results We trained the model for different numbers of latent functions. Figure 5 shows the classification accuracy for the samples for each class in 'Test Set 1'. As a comparison method, we find control classes for each cancer sample by assigning it to the nearest control class using a K nearest neighbours classifier (optimal $K = 1$) trained over the control set. We picked the model with the

[4] A manually curated database of gene pathways from `http://www.genome.jp/kegg/`



(a)

(b)

Fig. 4: (a): Heat map visualisation of **Y**, the set of control samples and (b): **X**, the set of tumor samples. Each row corresponds to the mean of the pathway activations (columns) in a class, and the data was clustered using standard hierarchical two dimensional clustering. This shows the similarities between different classes in each data set (note that the ordering of the columns differs between the figures)

highest predictive likelihood over the test set for the next set of experiments. Figure 6 visualizes the performance of the model on some samples selected from 'Test Set 2'. For each sample (rows), the predictive distribution over the control classes is visualized as a heat map. From the figure, we see that the model is able to make some sensible predictions for some of the tumor samples, when the control class is ambiguous.

The *gastroesophagal junction adenocarcinoma* samples (where the ideal control sample should be taken from the junction between the esophagus and the stomach) are mapped to *esophagus* and *stomach. Leiomyosarcoma* (4th row) and *uterine sarcoma* are both uterine cancers, and are mapped to *endometrium*, and most of the *colorectal adenocarcinoma* samples map to *colon cecum. Malignant melanoma* samples, a type of skin cancer, are mapped to *adipose tissue* and *lymph*

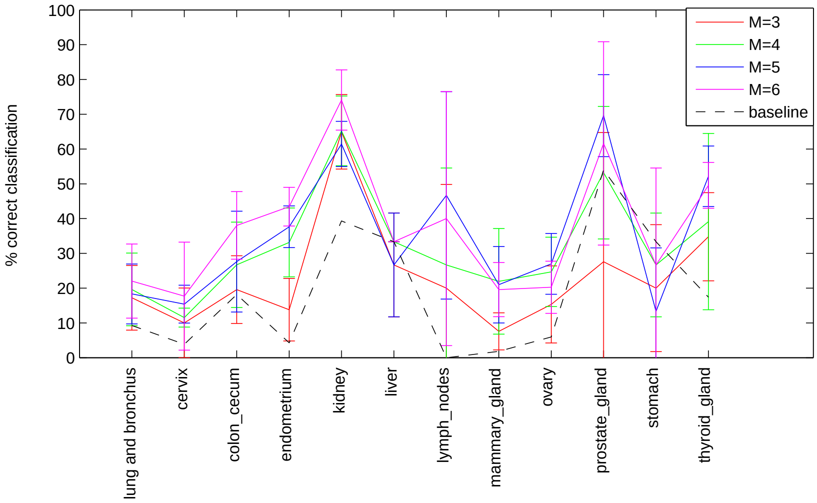


Fig. 5: Percentage of correct classifications for each control class in the test set, evaluated for different numbers of latent functions. The error bars correspond to ±1 s.d. over 10 runs.

nodes, which is a plausible prediction. Furthermore, the class *adipose tissue* was not in the training set, which verifies that the model was able to generalise to unseen classes. However, for cases such as the *bladder carcinomas* (rows 6 to 10) and the skin cancers (rows 11 to 15) where there does not appear to be a single appropriate control class, it is problematic to choose a suitable control.

4 Discussion

In this paper, we highlighted the relevant, and frequently occurring, problem of control measurement selection in experimental design, primarily focusing on differential gene expression studies for analysing cancer. The inherently biased nature of gene expression measurements towards many factors (such as patient, laboratory and tissue-specific) can cause erroneous findings for differential gene expression studies, if the control measurements do not match the biases of the case samples. The possibility of error could be minimised by using a large set of appropriate control measurements, but these are seldom available in practice. However, there are large volumes of publicly available gene expression measurements, which could potentially be used as controls for the experiment.

We proposed a novel approach which can automatically find control measurements for a given cancer profile, from a pool of control samples, and a set of existing mappings to cancer samples. We train a model which jointly learns

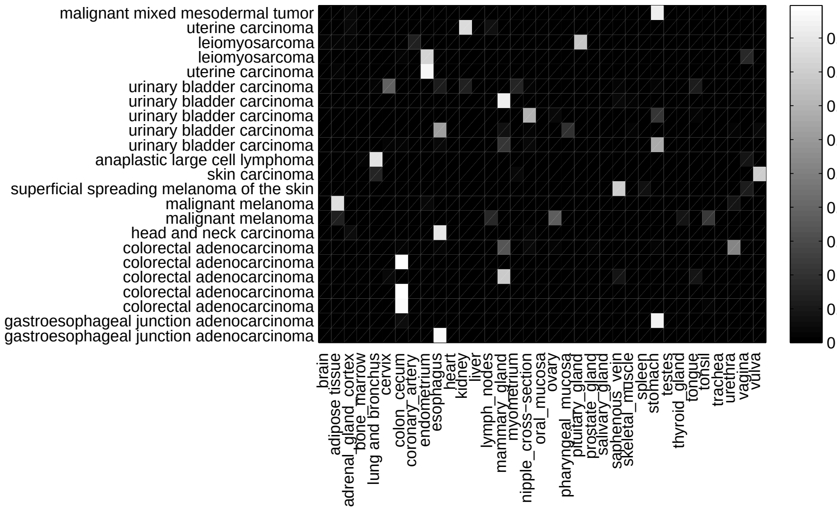


Fig. 6: Visualization of the probability distribution over the control classes (x axis) for some tumor samples (y axis) with unknown control classes

to classify the control and cancer samples into the K control classes, where the two sets of classifiers are constrained to be a mixture of underlying M functions where $M < K$. The functions can be different between the two sets, but the mixing matrix is the same. This approach can be viewed as a paired multitask learning problem, where the two sets of multiple tasks are the classification of control samples into control classes, and the classification of cancer samples into control classes. *Within* each task set, statistical strength is shared across the tasks by constraining the classifiers to be a mixture of underlying functions. Information about the relatedness of the tasks is shared *between* the two task sets, by constraining the mixing matrix to be the same.

We found that the model was able to give reasonable performance on finding control measurements for a set of cancer portraits. For test data where the control class was known, the model mapped cancer profiles to one of 35 control classes with a better accuracy than a nearest neighbour classifier, trained over the control samples. However, in some cases (see Figure 6), particularly when a single appropriate control class did not exist, the model did not find appropriate control classes. This could be due to the way in which the model is constrained to be a mixture of underlying functions. Consequently, the model could be wasting its modelling power on accurately discriminating for a small number of classes, while the accuracy in predicting the other classes remains low. This is likely, given the spread of classification accuracies in Figure 5.

We feel that this is a promising approach to a difficult problem in bioinformatics, and a new innovative framework for constructing and solving structured multitask learning problems. We aim to further develop this framework by imposing alternative structural constraints between the two sets of tasks.

Acknowledgments G.L. and S.K. belong to the Finnish Centre of Excellence on Adaptive Informatics Research. Funding: TEKES (grant no. 40101/07); PASCAL 2 Network of Excellence, ICT 216886 (partially to G.L. and S.K.).

References

1. A. Brazma, H. Parkinson, U. Sarkans, M. Shojatalab, J. Vilo, N. Abeygunawardena, E. Holloway, M. Kapushesky, P. Kemmeren, G. Garcia Lara, A. Oezcimen, P. Rocca-Serra, and S-A Sansone. ArrayExpress – A public repository for microarray gene expression data at the EBI. *Nucleic Acids Research*, 31(1):68–71, 2003.
2. R. Edgar, M. Domrachev, and A.E. Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–10, Jan 2002.
3. L. Knorr-Held and N. G. Best. Shared component models for detecting joint and selective clustering of two diseases, 2000. Sonderforschungsbereich 386, Paper 183.
4. K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309 – 318, 2001.
5. R. Ge, M. Ester, B. J. Gao, Z. Hu, B. Bhattacharya, and B. Ben-Moshe. Joint Cluster Analysis of Attribute Data and Relationship Data: The Connected k-Center Problem, Algorithms and Applications. *ACM Transactions on Knowledge Discovery from Data*, 2(2):Article 7, 2008.
6. T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):960 – 914, 2000.
7. G. K. Gerber, R. D. Dowell, T. S. Jaakkola, and D. K. Gifford. Automated Discovery of Functional Generality of Human Gene Expression Programs. *PLoS Comput Biol*, 3(8):e148. doi:10.1371/journal.pcbi.0030148, 2007.
8. Zhang J, Z. Ghahramani, and Y. Yang. Flexible Latent Variable Models for Multitask Learning. *Machine Learning*, 73(3):221–242, 2008.
9. E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams. Multi-task Gaussian Process Prediction. In *Neural Information Processing Systems*, 2008.
10. K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian Processes from Multiple Tasks. In *25th International Conference on Machine Learning*, 2008.
11. A. J. Storkey and M. Sugiyama. Mixture Regression for Covariate Shift. In *Advances in Neural Information Processing Systems 19*, 2007.
12. M. Sugiyama, M. Krauledat, and K-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, (8):985–1005, 2007.
13. H. Daumé and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artifical Intelligence Research*, (26):101–126, 2006.
14. Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric latent factor models. In *Proceedings of the Eighth Conference on Artificial Intelligence and Statistics (AISTATS)*, 2005.

15. S. Bickel, J. Bogojeska, T. Lengauer, and T. Scheffer. Multi-Task Learning for HIV Therapy Screening. In *22nd International Conference on Machine Learning*, 2008.
16. M. Girolami and S. Rogers. Variational Bayesian multinomial probit regression with Gaussian process priors. *Neural Computation*, 18(8):1790–1817, 2006.
17. R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed. Exploration, Normalization, and Summaries of High Density Oligonucleotide Array Probe Level Data. *Biostatistics*, 4(2):249–264, 2003.
18. A. Subramanian, P Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles . *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, October 2005.