Multi-Task Learning for HIV Therapy Screening

Steffen Bickel BICKEL@MPI-INF.MPG.DE
Jasmina Bogojeska JASMINA@MPI-INF.MPG.DE
Thomas Lengauer LENGAUER@MPI-INF.MPG.DE
Tobias Scheffer SCHEFFER@MPI-INF.MPG.DE
Max Planck Institute for Computer Science, Saarbrücken, Germany

Abstract

We address the problem of learning classifiers for a large number of tasks. We derive a solution that produces resampling weights which match the pool of all examples to the target distribution of any given task. Our work is motivated by the problem of predicting the outcome of a therapy attempt for a patient who carries an HIV virus with a set of observed genetic properties. Such predictions need to be made for hundreds of possible combinations of drugs, some of which use similar biochemical mechanisms. Multi-task learning enables us to make predictions even for drug combinations with few or no training examples and substantially improves the overall prediction accuracy.

1. Introduction

In *multi-task learning* one seeks to solve many classification problems in parallel. Some of the classification problems will likely relate to one another, but one cannot assume that the tasks share a joint conditional distribution of the class label given the input variables. The challenge of multi-task learning is to come to a good generalization across tasks: each task should benefit from the wealth of data available for the entirety of tasks, but the optimization criterion needs to remain tied to the individual task at hand.

Our work is motivated by the problem of predicting the therapeutic success of a given combination of drugs for a given strain of the *Human Immunodeficiency Virus-1* (HIV-1). HIV is associated with the *acquired immunodeficiency syndrome* (AIDS). Being a disease that claimed more than 25 million lives since 1981, AIDS is one of the most destructive epidemics in recorded history. Currently there are more than 33 million people infected with HIV (UNAIDS/WHO, 2007).

Antiretroviral therapy is hampered by HIV's strong ability to mutate and develop viral quasi-species that can quickly be dominated by resistant variants. In order to decide on a course of therapy, virus samples taken from each individual patient are tested for a set of resistance-relevant mutations. Given this set of identified mutations together with the patient's medication history, a medical practitioner needs to decide which combination of drugs to administer. The large number of genetic mutations and the wide array of available drug combinations render the process of predicting the success of a potential therapy difficult, at best, for a human doctor.

Historic treatment records of HIV patients cover only a small portion of all possible drug combinations. For many of these combinations, only few treatments have been recorded. This scarceness of training data precludes separate training of a powerful prediction model for each combination from only records of treatments which used the same drug combination. Distinct combinations can have similar effects when they intersect in jointly contained drugs, or when they include drugs that use similar mechanisms to affect the virus. Therefore, in order to predict the outcome of a given drug combination, it is desirable to exploit data from related combinations and thereby achieve generalization over both virus mutations and combinations of drugs.

We contribute a new multi-task learning model that can handle arbitrarily different data distributions for different tasks without making assumptions about the data generation process or the relation between tasks. We show that by appropriately weighting each instance in the pool of all examples, one can match the distribution that governs the pool of examples of all tasks to each of the single task distributions. We show

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

how appropriate weights can be obtained by discriminating the labeled sample for a given task against the pooled sample.

The rest of this paper is structured as follows. After formalizing the problem setting in Section 2, we review related transfer learning models in Section 3. We devise the model for multi-task learning by distribution matching in Section 4. In Section 5 we describe the data sets and the experimental setting and report on experimental results. Section 6 concludes.

2. Problem Setting

In *supervised multi-task learning*, each of several tasks z is characterized by an unknown joint distribution $p(\mathbf{x}, y|z)$ of features $\mathbf{x}$ and label y given the task z. The joint distributions of different tasks may differ arbitrarily but usually some tasks have similar distributions. A training sample $D = \langle(\mathbf{x}_1, y_1, z_1), \ldots, (\mathbf{x}_m, y_m, z_m)\rangle$ collects examples from all tasks. There may be tasks with no data. For each example, input attributes $\mathbf{x}_i$, class label y_i, and the originating task z_i are known. The entire sample D is governed by the mixed joint density $p(z)p(\mathbf{x}, y|z)$. The prior $p(z)$ specifies the task proportions.

The goal is to learn a hypothesis $f_z : \mathbf{x} \mapsto y$ for each task z. This hypothesis $f_z(\mathbf{x})$ should correctly predict the true label y of unseen examples drawn from $p(\mathbf{x}|z)$ for all z. That is, it should minimize the expected loss

$$\mathbf{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y|z)}[\ell(f_z(\mathbf{x}), y)]$$

with respect to the unknown joint distribution $p(\mathbf{x}, y|z)$ for each individual z.

This abstract problem setting models the HIV therapy screening application as follows. Input $\mathbf{x}$ describes the genotype of the virus that a patient carries, together with the patient's treatment history. Genotype information is encoded as a binary vector indicating the presence and absence of each out of a predefined set of resistance-relevance mutations, respectively. The treatment history can be represented as a binary vector indicating which drugs have been administered over the course of past treatments. A candidate combination of drugs plays the role of the task z: each task has an associated binary vector $\mathbf{z}$ that indicates a set of drugs that a medical practitioner is currently giving consideration. The binary class label y indicates whether the therapy will be successful.

In addition to training data, we may have prior knowledge on the similarity of tasks which is encoded in a kernel function $k(z, z')$. Prediction models for different drug combinations can be similar because the sets of drugs intersect (we will later refer to this as the drug feature kernel), or because similar sets of mutations in the virus render the drugs in the set ineffective (mutation table kernel).

3. Prior Work

One obvious strategy for multi-task learning is to learn independent models for each target task t by minimizing an appropriate loss function on the portion of $D_t = \{(\mathbf{x}_i, y_i, z_i) \in D : z_i = t\}$. The other extreme could be a one-size-fits-all model $f_*(\mathbf{x})$ trained on the entire sample.

In many applications, task-level descriptions or prior knowledge on task similarity encoded in a kernel are available. Bonilla et al. (2007) study an extension of the one-size-fits-all model and find that training with a kernel defined as the multiplication of an input feature kernel and a task-level kernel outperforms a gating network. Task-level features have also been utilized for task clustering and for a task-dependent prior on the model parameters (Bakker & Heskes, 2003).

Another simple extension to the one-size-fits-all model would be to train a model for a target task from all data with weighted examples from other tasks, using one fixed uniform weight for each task. Such a model is described by Wu and Dietterich (2004).

Our work is inspired by learning under covariate shift. In the covariate shift setting the marginals $p_{train}(\mathbf{x})$ and $p_{test}(\mathbf{x})$ of training and test distributions differ, but the conditionals are identical $p_{train}(y|\mathbf{x}) = p_{test}(y|\mathbf{x})$. If training and test distributions were known, then the loss on the test distribution could be minimized by weighting the loss on the training distribution with an instance-specific factor. Shimodaira (2000) illustrates that the scaling factor has to be $\frac{p_{test}(\mathbf{x})}{p_{train}(\mathbf{x})}$. Bickel et al. (2007) derive a discriminative expression for this marginal density ratio that can be estimated – without estimating the potentially high-dimensional densities of training and test distributions – by discriminating training against test data.

Hierarchical Bayesian models for multi-task learning are based on the assumption that task-specific model parameters are drawn from a common prior. The task dependencies are captured by estimating the common prior. Yu et al. (2005) impose a normal-inverse Wishart hyperprior on the mean and covariance of a Gaussian process prior that is shared by all task-specific regression functions. Mean and covariance of the Gaussian process are estimated using the EM algorithm. A Dirichlet process can serve as prior in a hierarchical Bayesian model and cluster the tasks (Xue

et al., 2007); all tasks in one cluster share the same model parameters. Evgeniou and Pontil (2004) derive a kernel that is based on a hierarchical Bayesian model with Gaussian prior (covariance matrix is scalar) on the parameters of a regularized regression.

Larder et al. (2007) tackle the problem of predicting virological response to a given HIV drug combination with neural networks. Lathrop and Pazzani (1999) apply combinatorial optimization to the same problem using features extracted from the viral genotype and the drugs in the combination. Altmann et al. (2007) approach the problem by including various phenotypic information and an estimate of future evolutionary development of the virus in the learning process.

4. Multi-Task Learning by Distribution Matching

In learning a classifier $f_t(\mathbf{x})$ for target task t, we seek to minimize the loss function with respect to $p(\mathbf{x}, y|t)$. Simply pooling the available data for all tasks would create a sample governed by $\sum_z p(z)p(\mathbf{x}, y|z)$. Our approach now is to create a task-specific resampling weight $r_t(\mathbf{x}, y)$ for each element of the pool of examples. The sampling weights match the pool to the target distribution $p(\mathbf{x}, y|t)$. The weighted sample is governed by the correct target distribution, but is still larger as it draws from the sample pool for all tasks.

Instead of sampling from the pool, one can weight the loss incurred by each instance by the resampling weight. The expected weighted loss with respect to the mixture distribution that governs the pool equals the loss with respect to the target distribution $p(\mathbf{x}, y|t)$. Equation 1 defines the resampling weights.

$$\begin{aligned} &\mathbf{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y|t)}[\ell(f(\mathbf{x},t),y)] \qquad (1)\\ &\quad = \mathbf{E}_{(\mathbf{x},y)\sim \sum_z p(z)p(\mathbf{x},y|z)}\left[r_t(\mathbf{x},y)\ell(f(\mathbf{x},t),y)\right] \end{aligned}$$

In the following, we will show that

$$r_t(\mathbf{x}, y) = \frac{p(\mathbf{x}, y|t)}{\sum_z p(z)p(\mathbf{x}, y|z)}$$

satisfies Equation 1. Equation 2 expands the expectation and introduces a fraction that equals one. Equation 3 expands the sum over z in the numerator to run over the entire expression because the integral over $(\mathbf{x}, y)$ is independent of z. Equation 4 is the expected loss over the distribution of all tasks weighted by $\frac{p(\mathbf{x},y|t)}{\sum_z p(z)p(\mathbf{x},y|z)}$.

$$\begin{aligned} &\mathbf{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y|t)}[\ell(f(\mathbf{x},t),y)] \qquad (2)\\ &= \int \frac{\sum_z p(z)p(\mathbf{x},y|z)}{\sum_{z'} p(z')p(\mathbf{x},y|z')} p(\mathbf{x},y|t)\ell(f(\mathbf{x},t),y)d\mathbf{x}dy\\ &= \int \sum_z \Bigg(p(z)p(\mathbf{x},y|z)\frac{p(\mathbf{x},y|t)}{\sum_{z'} p(z')p(\mathbf{x},y|z')} \qquad (3)\\ &\qquad\qquad \ell(f(\mathbf{x},t),y)\Bigg) d\mathbf{x}dy\\ &= \mathbf{E}_{(\mathbf{x},y)\sim \sum_z p(z)p(\mathbf{x},y|z)} \qquad (4)\\ &\qquad \left[\frac{p(\mathbf{x},y|t)}{\sum_{z'} p(z')p(\mathbf{x},y|z')}\ell(f(\mathbf{x},t),y)\right] \end{aligned}$$

Equation 4 signifies that we can train a hypothesis for task t by minimizing the expected loss over the distribution of all tasks weighted by $r_t(\mathbf{x}, y)$. This amounts to minimizing the expected loss with respect to the target distribution $p(\mathbf{x}, y|t)$.

Equation 4 leaves us with the problem of estimating the joint density ratio $r_t(\mathbf{x}, y) = \frac{p(\mathbf{x},y|t)}{\sum_z p(z)p(\mathbf{x},y|z)}$. One might be tempted to train density estimators for $p(\mathbf{x}, y|t)$ and $\sum_z p(z)p(\mathbf{x}, y|z)$. However, obtaining estimators for potentially high-dimensional densities is unnecessarily difficult because ultimately only a scalar weight is required for each example.

4.1. Discriminative Density Ratio Model

In this section, we derive a discriminative model that directly estimates the resampling weights $r_t(\mathbf{x}, y) = \frac{p(\mathbf{x},y|t)}{\sum_z p(z)p(\mathbf{x},y|z)}$ without estimating the individual densities. We reformulate the density ratio $\frac{p(\mathbf{x},y|t)}{\sum_z p(z)p(\mathbf{x},y|z)}$ in terms of a conditional model $p(t|\mathbf{x}, y)$. This conditional has the following intuitive meaning: Given that an instance $(\mathbf{x}, y)$ has been drawn at random from the pool $\cup_z D_z = D$ of samples for all tasks (including D_t); the probability that $(\mathbf{x}, y)$ originates from D_t is $p(t|\mathbf{x}, y)$. The following equations assume that the prior on the size of the target sample is greater than zero, $p(t) > 0$. In Equation 6 Bayes' rule is applied twice and in Equation 7 $p(\mathbf{x}, y)$ and $p(z)$ are canceled out. Equation 8 follows by $\sum_z p(z|\mathbf{x}, y) = 1$.

$$\begin{aligned} r_t(\mathbf{x}, y) &= \frac{p(\mathbf{x}, y|t)}{\sum_z p(z)p(\mathbf{x}, y|z)} \qquad (5)\\ &= \frac{p(t|\mathbf{x}, y)p(\mathbf{x}, y)}{p(t)} \frac{1}{\sum_z p(z)\frac{p(z|\mathbf{x},y)p(\mathbf{x},y)}{p(z)}} \qquad (6)\\ &= \frac{p(t|\mathbf{x}, y)}{p(t)\sum_z p(z|\mathbf{x}, y)} \qquad (7)\\ &= \frac{p(t|\mathbf{x}, y)}{p(t)} \qquad (8) \end{aligned}$$

The significance of Equation 8 is that it shows how the resampling weights $r_t(\mathbf{x}, y) = \frac{p(\mathbf{x},y|t)}{\sum_z p(z)p(\mathbf{x},y|z)}$ can be determined without knowledge of any of the task densities $p(\mathbf{x}, y|z)$. The right hand side of Equation 8 can be evaluated based on a model $p(t|\mathbf{x}, y)$ that discriminates labeled instances of the target task against labeled instances of the pool of examples for all tasks. Intuitively, $p(t|\mathbf{x}, y)$ characterizes how much more likely $(\mathbf{x}, y)$ is to occur in the target distribution than it is to occur in the mixture distribution of all tasks. Instead of potentially high-dimensional densities $p(\mathbf{x}, y|t)$ and $p(\mathbf{x}, y|z)$, a conditional distribution with a single variable needs to be modeled. One can apply any probabilistic classifier to model this conditional distribution.

4.2. Soft-Max Model for Density Ratio Estimation

We model $p(t|\mathbf{x}, y)$ of Equation 8 for all tasks jointly with a soft-max model (the multi-class generalization of the logistic model) with model parameters $\mathbf{v}$, displayed in Equation 9. The parameter vector $\mathbf{v}$ is a concatenation of task-specific subvectors $\mathbf{v}_z$, one for each task z. With this model an estimate for $p(t|\mathbf{x}, y)$ is given by $p(z = t|\mathbf{x}, y, \mathbf{v})$; this is the evaluation of the soft-max model with respect to task t.

$$p(z|\mathbf{x}, y, \mathbf{v}) = \frac{\exp(\mathbf{v}_z^\mathsf{T}\Phi(\mathbf{x}, y))}{\sum_{z'} \exp(\mathbf{v}_{z'}^\mathsf{T}\Phi(\mathbf{x}, y))} \quad (9)$$

Equation 9 requires a problem-specific feature mapping $\Phi(\mathbf{x}, y)$. Without loss of generality we define this mapping for binary labels $y \in \{+1, -1\}$ in Equation 10; δ is the Kronecker delta. In the absence of prior knowledge about the similarity of classes, input features $\mathbf{x}$ of examples with different class labels y are mapped to disjoint subsets of the feature vector.

$$\Phi(\mathbf{x}, y) = \begin{bmatrix} \delta(y, +1)\Phi(\mathbf{x}) \\ \delta(y, -1)\Phi(\mathbf{x}) \end{bmatrix} \quad (10)$$

With this feature mapping the models for positive and negative examples do not interact and can be trained independently.

For training the soft-max model we maximize the regularized log-likelihood of the data. Prior knowledge on the similarity of tasks in the form of a positive semi-definite kernel function $k(z, z')$ can be be encoded in the covariance matrix of a Gaussian prior $N(0, \Sigma)$ on parameter vector $\mathbf{v}$. We set all main diagonal entries of Σ to the scalar parameter $\sigma_\mathbf{v}^2$ and set the secondary diagonal entries corresponding to the covariances between $\mathbf{v}_z$ and $\mathbf{v}'_z$ to $k(z, z')\rho\sigma_\mathbf{v}^2$ (assuming kernel values $0 \leq k(z, z') \leq 1$). Parameter $\sigma_\mathbf{v}^2$ specifies the variance of each element in $\mathbf{v}$. $k(z, z')\rho$ is the correlation coefficient between elements of subvectors $\mathbf{v}_z$ and $\mathbf{v}'_z$; parameter ρ specifies the strength of this correlation. The covariance matrix Σ is required to be invertible and therefore $0 \leq \rho < 1$. All other entries of Σ are set to zero. When prior knowledge on the task similarities is encoded in the prior on the model parameters, then this prior knowledge dominates the optimization criterion for small samples while the data-driven portion of the criterion becomes dominant and overrides prior beliefs as more data arrives.

Optimization Problem 1 *Over parameters* $\mathbf{v}$*, maximize*

$$\sum_{(\mathbf{x}_i, y_i, z_i) \in D} \log(p(z_i|\mathbf{x}_i, y_i, \mathbf{v})) + \mathbf{v}^\mathsf{T}\Sigma^{-1}\mathbf{v}.$$

The solution of Optimization Problem 1 is a maximum *a posteriori* estimation of the soft-max model (Equation 9) over the model parameters $\mathbf{v}$ using a Gaussian prior with covariance matrix Σ. Tasks with no training examples are covered naturally in Optimization Problem 1. In this case, the Gaussian prior with the task kernel $k(z, z')$ encoded in the covariance matrix determines the model.

For our experiments we use a kernelized variant of Optimization Problem 1 by applying the representer theorem. Details on the kernelization of multi-class logistic regression can be learned from Zhu and Hastie (2002).

4.3. Weighted Empirical Loss and Target Model

The multi-task learning procedure first determines resampling weights $r_z(\mathbf{x}, y)$ for all tasks and instances by solving Optimization Problem 1. In this section we describe the second step of training an array of target models, one for each task, using weighted examples.

With the results of Optimization Problem 1 the discriminative expression for the weights of Equation 8 can be estimated. Using these weights we can evaluate the expected loss over the weighted training data as displayed in Equation 11. It is the regularized empirical counterpart of Equation 4.

$$\mathbf{E}_{(\mathbf{x},y)\sim D}\left[\frac{p(t|\mathbf{x}, y, \mathbf{v})}{p(t)}\ell(f(\mathbf{x}, t), y)\right] + \frac{\mathbf{w}_t^\mathsf{T}\mathbf{w}_t}{2\sigma_\mathbf{w}^2} \quad (11)$$

An instance of Optimization Problem 2 is solved for each task independently to produce a separate model for this task. Optimization Problem 2 minimizes Equation 11, the weighted regularized loss over the training data using a standard Gaussian log-prior with variance $\sigma_\mathbf{w}^2$ on the parameters $\mathbf{w}_t$. Each example is weighted by the discriminatively estimated density

fraction from Equation 8 using the solution of Optimization Problem 1.

Optimization Problem 2 *For task t: over parameters $\mathbf{w}_t$, minimize*

$$\sum_{(\mathbf{x}_i,y_i)\in D} \frac{p(t|\mathbf{x}_i,y_i,\mathbf{v})}{p(t)}\ell(f(\mathbf{x}_i,\mathbf{w}_t),y_i) + \frac{\mathbf{w}_t^\mathsf{T}\mathbf{w}_t}{2\sigma_\mathbf{w}^2}.$$

5. HIV Therapy Screening

We model HIV therapy screening as a multi-task learning problem. The input $\mathbf{x}$ to the prediction problem is given by attributes of the viral genotype and the patient's treatment history. The combination of drugs z plays the role of the task. Success or failure of the therapy constitutes class-label y.

In the next subsections we describe the data sets, reference methods, and the empirical results of our study.

5.1. Data Sets and Prior Knowledge on Task Similarity

We use data from the EuResist project (Rosen-Zvi et al., 2008). The data set comprises a total number of 52846 treatment records from the treatment histories of 16999 HIV patients treated in hospitals in the period of 1977 through 2007.

We use two different definitions of therapeutic success and failure to tag the data: *virus load labeling* and *multi-conditional labeling*.

According to our *virus load labeling* definition a therapy is successful if the viral load (number of virus copies per ml blood plasma, cp/ml) drops below the established level of virus detection of 400 cp/ml during the time of the treatment. Otherwise the treatment is a failure. In *multi-conditional labeling*, a therapy is successful if the viral load measured in the time range between 28 and 84 days after the start of the therapy decreases by at least 2 orders of magnitude compared to the most recent viral load measured one to three months before the start of the therapy, or the viral load drops below 400 cp/ml 56 days after the start of the therapy. A drawback of this definition is that due to the strict time intervals it imposes on the measurements, class labels that adhere to this labeling are only available for a small number of records. The *virus load labeling* does not require these strict time intervals by making use of any viral load measurement during the course of therapy to label it.

Out of all available treatment records we extract two different data sets using the two labelings. With the virus load labeling we extract 3260 and with the multi-conditional labeling 2011 treatment records with corresponding ratios of 65.7% and 64.1% successful treatments. The size of these data sets is much smaller than the size of the original data due to missing viral load measurements, or missing virus sequence information.

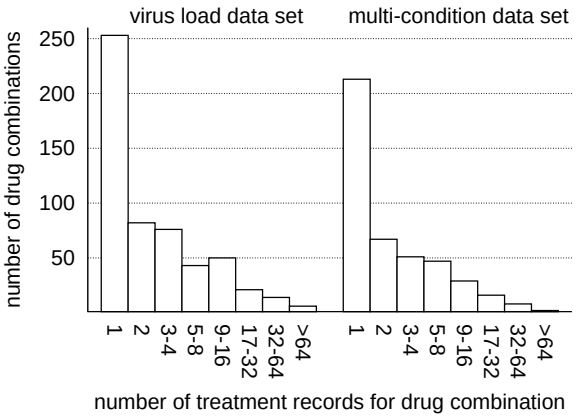

Figure 1. Histogram over number of treatment records for drug combinations (tasks) in the virus load data set (left) and multi-condition data set (right).

A number of 545 distinct drug combinations (tasks z) occur at least once in the virus load data set; 433 occur in the the multi-conditional data set. The histogram over sample sizes per task is displayed in Figure 1. For many combinations, only a few examples occur in the data. For instance, in the virus load data set we observe 253 out of 545 drug combinations with only one data point and 411 with less than 5 instances. Similarly, the multi-conditional data set has 213 out of 433 drug combinations with a single data point and 331 with less than 5 observations.

We extract two types of features for each instance: a genotypic description of the virus and information about the treatment history of the patient. We use the viral genotype taken from the patient shortly before the treatment and represent it by a binary vector indicating the presence of resistance-relevant mutations of the viral sequence (Johnson et al., 2007). Drug-resistant viral quasi-species evolve during the course of the treatment due to selective pressure imposed by the drug. As they remain in the patient's body, the treatment history plays an important role for predicting the outcome of a potential treatment. Hence, we extract all drugs given to the patient in previous treatments and use a binary vector representation with a one entry for each drug given to the patient in the treatment history. The 82-dimensional feature vector $\mathbf{x}$ for each data point results from the concatenation

Table 1. Classification accuracies with standard errors of differences to distribution matching method (ste.Δ). Symbols (•,○,∗,◇) indicate statistical significance according to a paired t-test with significance level $\alpha = 0.05$, (•) compared to separate baseline, (○) compared to pooled baseline, (∗) compared to hierarchical Bayesian kernel baseline, (◇) compared to hierarchical Bayesian Gaussian process baseline.

data set	prior knowledge	separate	ste.Δ	pooled	ste.Δ	hier. Bayes kernel	ste.Δ	hier. Bayes Gauss. proc.	ste.Δ	distribution matching
virus load	none	67.87%	1.80	75.00%	1.47	76.69%	1.39	76.53%	1.36	• ○ ∗ ◇ **79.14%**
	drug.feat.	67.87%	1.76	75.46%	1.39	75.31%	1.34			• ○ ∗ **77.91%**
	mut.table	67.87%	1.78	75.61%	1.37	76.84%	1.16			• ○ ∗ **79.29%**
multi-condition	none	64.64%	2.41	76.67%	1.13	77.17%	1.29	76.43%	1.44	• ○ ∗ ◇ **79.40%**
	drug.feat.	64.64%	2.29	**78.41%**	1.63	75.19%	1.44			• ∗ **78.16%**
	mut.table	64.64%	2.38	**78.66%**	1.11	**77.42%**	1.24			• **79.16%**

of 65 genotypic and 17 historic treatment features.

We have prior knowledge about the similarity of combinations and encode this knowledge into two different task similarity kernels $k(z, z')$. The binary drug indicator vector has an entry for each drug; entries of one indicate the presence of a drug in the combination. The *drug indicator kernel* is the inner product between the normalized drug indicator vectors of two combinations. The *mutation table kernel* is based on tables about the resistance-associated mutations of single drugs (Johnson et al., 2007). We construct binary vectors indicating resistance-relevant mutations for the set of drugs occurring in a combination. The kernel computes the normalized inner product between such binary vectors for two drug combinations.

5.2. Reference Methods

The first reference method is training of a separate logistic regression model for each task without any interaction ("separate"). Tasks without any training examples get a constant classifier that assigns each test example with 50% to each of both classes.

The next baseline is a one-size-fits-all model; all examples are pooled and only one common logistic regression is trained for all tasks ("pooled"). For the experiments with prior knowledge on task similarity we multiply the feature kernel with the task kernel values $k(\mathbf{x}, \mathbf{x}')(k(z, z') + 1)$ and train one model using this kernel (Bonilla et al., 2007). For task kernels that can have a value of zero we include a "+1" term to ensure that the feature kernel does not vanish.

The third reference method ("hier. Bayes kernel") is a logistic regression with the hierarchical Bayesian kernel $k_{hBayes}(\mathbf{x}, \mathbf{x}') = (\lambda + \delta(z, z'))k(\mathbf{x}, \mathbf{x}')$ of Evgeniou and Pontil (2004); $\delta(z, z')$ is the Kronecker delta and λ is a tuning parameter. For the experiments with task similarity kernel the hierarchical Bayes and the task kernel are multiplied. As second hierarchical Bayesian method ("hier. Bayes Gauss. proc.") we use the Gaussian process regression of Yu et al. (2005).

5.3. Experimental Setting and Results

In our experiments we study the benefit of distribution matching for HIV therapy screening compared to the reference methods described in Section 5.2. Optimization Problem 1 is solved with limited-memory BFGS and Optimization Problem 2 with Newton gradient descent using a logistic loss. For the prior term $p(t)$ required in Optimization Problem 2 we use a MAP estimate $\frac{|D_t|+\gamma}{\sum_z(|D_z|+\gamma)}$ with a symmetric Dirichlet prior. We use RBF kernels for all methods.

We apply a training-test split of the data consistent with the dates of the treatment records. We sort the treatment records by date and use the first 80% of the records as training data and the last 20% as test data. This procedure yields 653 and 403 test examples for the virus load and multi-conditional data set, respectively. The date consistent split is necessary because new drugs get approved over time, and under pressure of new drugs the viral population evolves. In such environments, the prediction models should be able to learn from data seen in the *past* and perform well on unseen data in the *future*.

We tune the prior and regularization parameters of all methods, the Dirichlet parameter γ, and the variance of the RBF kernels on tuning data resulting from a date consistent split of the training data.

The evaluation measure is the accuracy of predicting the correct label (success or failure of a treatment)

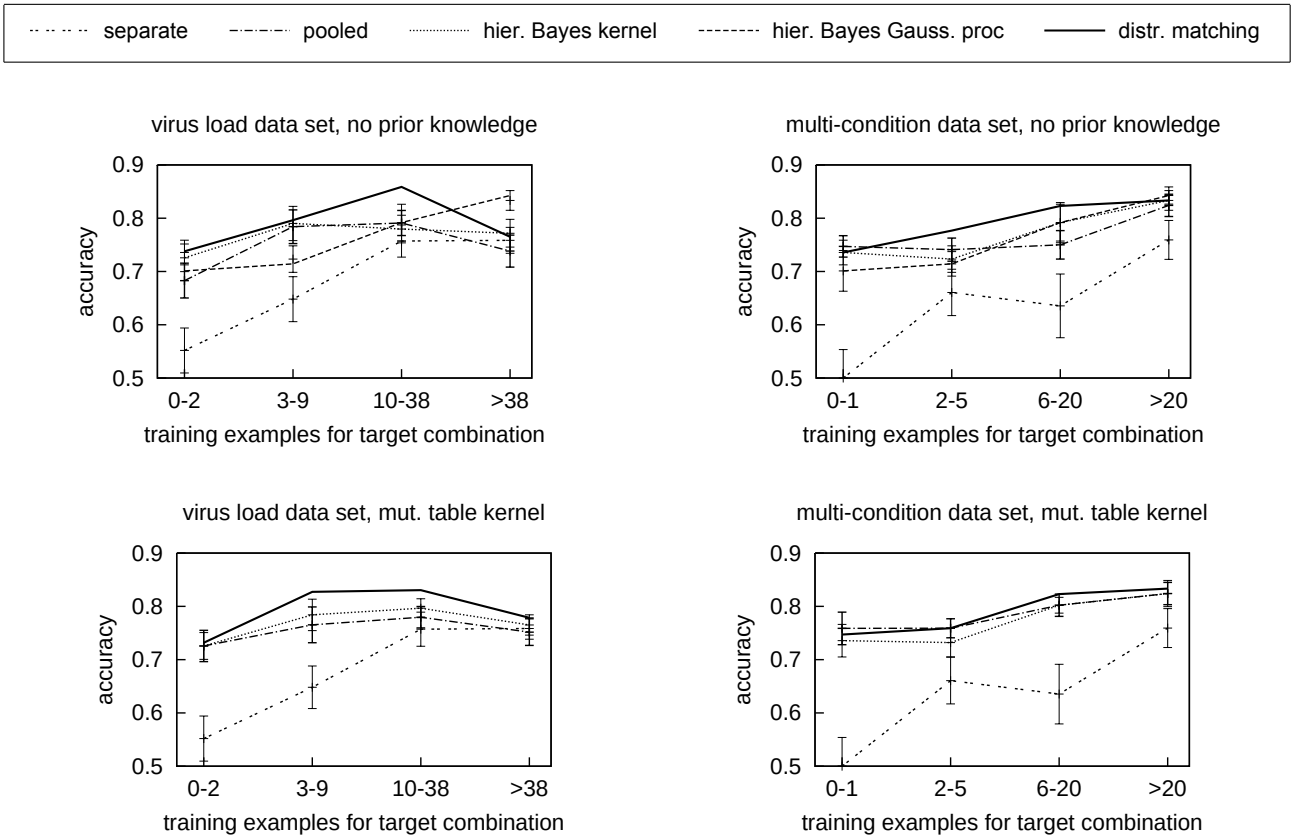

Figure 2. Accuracy over different number of training examples for target combination; virus load data set (left), multi-condition data set (right). Error bars indicate the standard error of the differences to distribution matching. The key can be found in the box right above the diagrams.

on the test set. Table 1 shows the results of the prediction accuracy for all methods over both data sets without and with two different types of prior knowledge on combination similarity. The columns "ste.Δ" placed next to the accuracy columns display the standard error of the differences to the distribution matching method.

Multi-task learning by distribution matching outperforms, or is as good as, the best alternative method in all cases. The improvement over the separate model baseline is about 10-14%. We can reject the null hypothesis that the pooled and the hierarchical Bayesian kernel baseline is at least as accurate as distribution matching in four and five cases respectively out of six according to a paired t-test at $\alpha = 0.05$.

For distribution matching, prior knowledge does not improve the accuracy. The pooled baseline benefits from prior knowledge for the multi-condition data set. For the case without prior knowledge we do not observe a statistically significant difference of the two hierarchical Bayesian methods, but they are both significantly worse than distribution matching according to the paired t-test. Note that the Gaussian process baseline is a regression model; all other methods are classification models.

Figure 2 displays the accuracy over the combinations in the test set grouped by the number of available examples for the settings without and with the mutation table kernel. For instance, an accuracy of 74% for the first group "0-2" means, that only test examples from combinations are selected that have zero, one, or two training examples each, and the accuracy on this subset of the test examples is 74%. Each of the four groups covers about the same number of test examples. The error bars indicate the standard error of the differences to the distribution matching method. Note, that the statistical tests described above are based on all test data and are not directly related to the group-specific error bars in the diagrams.

All methods benefit from larger numbers of training

examples per drug combination. The slightly decreasing accuracy for the virus load data set with ">38" training examples is surprising. Further analysis reveals that in this case there is an accumulation of test examples with history profiles very different from the training examples of the same combination.

For all methods that generalize over the tasks the benefit compared to the separate model baseline is the largest for the smallest group ("0-2" and "0-1" training examples respectively).

6. Conclusion

We devised a multi-task learning method that centers around resampling weights which match the distribution of the pool of examples of multiple tasks to the target distribution for a given task at hand. The method creates a weighted sample that reflects the desired target distribution and exploits the entire corpus of training data for all tasks. We showed how appropriate weights can be obtained by discriminating the labeled sample for a given task against the pooled sample. After weighting the pooled sample, a classifier for the given task can be trained. In our experiments on HIV therapy screening we found that the distribution matching method improves on the prediction accuracy over independently trained models by 10-14%. According to a paired t-test, distribution matching is significantly better than the reference methods for 17 out of 20 experiments.

A combination of drugs is the standard way of treating HIV patients. The accuracy to which the likely outcome of a combination therapy can be anticipated can therefore directly impact the quality of HIV treatments.

Acknowledgment

We thank Kai Yu for providing his Gaussian process implementation and Barbara Pogorzelska who adapted this code and conducted the experiments. We also thank the EuResist project with contract number EU-STREP IST-2004-027173 for providing the data. We gratefully acknowledge support from the German Science Foundation DFG.

References

Altmann, A., Beerenwinkel, N., Sing, T., Savenkov, I., Doumer, M., Kaiser, R., Rhee, S., Fessel, W., Shafer, W., & Lengauer, T. (2007). Improved prediction of response to antiretroviral combination therapy using the genetic barrier to drug resistance. *Antiviral Therapy*, *12*, 169–178.

Bakker, B., & Heskes, T. (2003). Task clustering and gating for Bayesian multitask learning. *The Journal of Machine Learning Research*, *4*, 83–99.

Bickel, S., Brückner, M., & Scheffer, T. (2007). Discriminative learning for differing training and test distributions. *Proceedings of the International Conference on Machine Learning*.

Bonilla, E., Agakov, F., & Williams, C. (2007). Kernel multi-task learning using task-specific features. *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

Evgeniou, T., & Pontil, M. (2004). Regularized multi–task learning. *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 109–117.

Johnson, V., Brun-Vezinet, F., Clotet, B., Günthrad, H., Kuritzkes, D., Pillay, D., Schapiro, J., Telenti, A., & Richman, D. (2007). Update of the drug resistance mutations in HIV-1: 2007. *Top HIV Med.*, *15*, 119–125.

Larder, B., Wang, D., Revell, A., Montaner, J., Harrigan, R., De Wolf, F., Lange, J., Wegner, S., Ruiz, L., Prez-Elas, M., Emery, S., Gatell, J., D'Arminio Monforte, A., Torti, C., Zazzi, M., & Lane, C. (2007). The development of artificial neural networks to predict virological response to combination HIV therapy. *Antiviral Therapy*, *12*, 15–24.

Lathrop, R., & Pazzani, M. (1999). Combinatorial optimization in rapidly mutating drug-resistant viruses. *Journal of Combinatorial Optimization*, *3*, 301–320.

Rosen-Zvi, M., Altmann, A., Prosperi, M., E., A., Neuvirth, H., Snnerborg, A., Schlter, E., Struck, D., Peres, Y., Incardona, F., Kaiser, R., Zazzi, M., & Lengauer, T. (2008). Selecting anti-HIV therapies based on a variety of genomic and clinical factors. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, *90*, 227–244.

UNAIDS/WHO (2007). AIDS Epidemic Update.

Wu, P., & Dietterich, T. (2004). Improving SVM accuracy by training on auxiliary data sources. *Proceedings of the International Conference on Machine Learning*.

Xue, Y., Liao, X., Carin, L., & Krishnapuram, B. (2007). Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, *8*, 35–63.

Yu, K., Tresp, V., & Schwaighofer, A. (2005). Learning Gaussian processes from multiple tasks. *Proceedings of the International Conference on Machine Learning*.

Zhu, J., & Hastie, T. (2002). Kernel Logistic Regression and the Import Vector Machine. *Advances in Neural Information Processing Systems 14*. MIT Press.