Aalto University
School of Science
Degree Programme in Engineering Physics and Mathematics

Transfer learning for molecular systems biology

Bachelor's Thesis

7.9.2011

Eemeli Leppäaho

Aalto University
School of Science
Degree Programme in Engineering Physics and Mathematics

ABSTRACT OF
BACHELOR'S THESIS

Author:	Eemeli Leppäaho
Title of thesis:	Transfer learning for molecular systems biology
Date:	September 7, 2011
Pages:	19
Major:	Systems Sciences
Code:	F3010
Supervisor:	Professor Harri Ehtamo
Instructor:	D.Sc. (Tech.) Elisabeth Georgii (Department of Information and Computer Science)

DNA microarrays measure transcriptional activity of genes in a biological sample, resulting in massive amounts of information. As transcriptional regulation plays a central role in all biological processes, such gene expression data has been studied intensively. However, huge dimensionality and a limited number of samples make it difficult to learn a well generalizable model for gene expression. While there exist ad hoc methods for selecting relevant features based on data set at hand or prior knowledge, this thesis exploits unlabeled background data from public databases to perform robust feature selection.

Two variants of transfer learning are compared to two standard machine learning methods in two different microarray classification tasks, showing that transfer of learned features can be advantageous. We also discuss in which situations transfer learning is not beneficial.

Keywords: transfer learning, principal component analysis, support vector machine, feature selection, self-taught learning, gene expression, DNA microarray

Aalto-yliopisto
Perustieteiden korkeakoulu
Teknillisen fysiikan ja matematiikan koulutusohjelma

KANDIDAATINTYÖN
TIIVISTELMÄ

Tekijä:	Eemeli Leppäaho
Työn nimi:	Siirto-oppiminen molekulaaribiologiassa
Päiväys:	7.9.2011
Sivumäärä:	19
Pääaine:	Systeemitieteet
Koodi:	F3010
Vastuuopettaja:	Professori Harri Ehtamo
Työn ohjaaja:	TkT Elisabeth Georgii (Tietojenkäsittelytieteen laitos)

DNA-mikrosiruilla mitataan biologisen näytteen geenien transkription aktiivisuutta, jolloin saadaan suuret määrät informaatiota. Tällaista geenien ilmentymistä kuvaavaa aineistoa on tutkittu intensiivisesti, sillä transkription säätely on keskeisessä roolissa kaikissa biologisissa prosesseissa. Aineiston suuri dimensio ja näytteiden rajallinen määrä hankaloittavat geenien ilmentymistä kuvaavan, hyvin yleistettävän mallin luomista. Dimensiota voidaan pienentää rajoittamalla tarkastelu aineiston merkityksellisiin ominaisuuksiin erilaisten ad hoc -menetelmien avulla. Nämä menetelmät perustuvat prioritietoon ja käytettävään aineistoon. Tässä työssä hyödynnetään ad hoc -menetelmien sijaan julkisesta tietokannasta saatavaa tausta-aineistoa robustissa ominaisuuksien valinnassa.

Työssä verrataan kahta siirto-oppimismenetelmää kahteen tavalliseen koneoppimismenetelmään mikrosiruaineiston luokittelutehtävissä ja osoitetaan, että siirto-oppimisesta voidaan hyötyä. Myös siirto-oppimisen mahdollisia haittoja analysoidaan.

Avainsanat:	siirto-oppiminen, pääkomponenttianalyysi, tukivektorikone, itseoppiminen, ominaisuuksien valinta, geenieksPressio, DNA mikrosiru

Contents

Symbols and Abbreviations

List of Symbols

$\|\|a\|\|_1$	L_1 norm of vector a
$\|\|a\|\|_2$	Euclidian norm of vector a
$a_j^{(i)}$	activation of basis vector b_j for sample $x^{(i)}$
b_j	the j:th basis vector
β	weight parameter enforcing sparsity
$\mathcal{C}_i$	class i
C_x	sample covariance matrix
$\mathcal{D}$	domain
$f(\cdot)$	objective predictive function
H	hyperplane
λ	Lagrange constant, eigenvalue
k	training set size
$P(X)$	marginal distribution
$\mathcal{T}$	task
$x_l^{(i)}$	labeled sample number i
$x_u^{(i)}$	unlabeled sample number i
$\mathcal{X}$	feature space
$\mathcal{Y}$	label space

List of Abbreviations

DNA	deoxyribonucleic acid
mRNA	messenger RNA
PCA	principal component analysis
RNA	ribonucleic acid
SVM	support vector machine

1 Introduction

In the traditional machine learning setup, a set of data is acquired from an experiment, and is then modelled to gain knowledge about a certain task. The performance of the model is for example measured by its predictive power on test data. However, it has been shown that it can be beneficial for a specific task to use data from other, related tasks [20, 2]. This idea is known as transfer learning (also inductive transfer and life-long learning).

In molecular systems biology one task of interest is to separate normal tissue samples from tissue samples that have a specific cancer. For solving this task, it can be useful to consider models and data sets of similar cancers, in addition to the data set at hand. There does not exist a general rule on how to transfer the knowledge or whether it will be beneficial. Thus, the performance of transfer learning models needs to be estimated carefully.

In this thesis, the experiment of interest comes from gene expression analysis, where features are related to genes. The feature space consists of around 13000 genes, which makes it hard to interpret the data. Even though the experiment is comparatively large, containing measurements for more than 300 samples, the number of samples is too small relative to the massive number of features. Models built from such data have a tendency to overfit, that is, they adapt too much to pecularities of the training set. This naturally makes the model weaker, that is, less generalizable. Usually this type of problem has been dealt with by selecting a smaller number of features according to prior knowledge or some other criteria, such as the amount of variance in the data set of interest [18]. The goal of this thesis is to find out whether transfer learning is beneficial in the context of feature selection in microarray classification tasks.

The structure of the thesis is as follows. First, relevant background knowledge is introduced in Section 2, both in the fields of biology and machine learning, including baselines used in the experimental section. Then, in Section 3, the field of transfer learning is presented, and specifically the feature selection models *transfer-PCA* and *self-taught learning* are introduced. In Section 4, classification of the molecular biological data is tested, comparing the transfer learning methods with the baselines. Finally in Section 5 the results are discussed and conclusions stated.

2 Background

This chapter gives an introduction to the fields of molecular systems biology and machine learning. Two machine learning methods that are used in our transfer learning approaches will be presented in detail: principal component analysis (PCA) and support vector machine (SVM). PCA is used for dimensionality reduction purposes and SVM for classification.

2.1 Molecular systems biology

2.1.1 Information flow in the cell

The cell is the functional unit of all living organisms, and the smallest unit classified as living [1]. Bacteria consist of a single cell only, whereas humans have approximately 100 trillion of them. Furthermore, each cell of a living organism contains its DNA (*deoxyribonucleic acid*), which stores the information used in its functional and developmental processes.

A gene is a stretch of DNA that encodes a functional product, typically protein, less frequently functional RNA (*ribonucleic acid*). The process of forming proteins, also called gene expression, works as follows: DNA is transcribed into messenger RNA (mRNA), and mRNA is translated into an amino acid sequence, which folds into an active protein. Identical mRNA can result in different proteins, since some parts of the mRNA may be removed before translation.

Every step of the gene expression process is regulated, but mRNA levels can be considered as an approximation for the importance of a gene product in a certain biological process. The next subsection will describe how to obtain and analyze such gene expression data.

2.1.2 Gene expression research

The most established way to measure gene expression is DNA microarray technology. DNA microarrays have been widely applied in molecular biological studies for instance in analyzing drug response in human leukemia cells and detecting new prostate cancer subtypes [3, 10].

A microarray consists of a large array with fragments of DNA, called probes, that are matched with mRNA in the cells. Cellular RNA samples are tagged with fluorescent molecules and exposed to the array, allowing the amounts of mRNA be measured in a high-throughput way, thousands of genes at once. The single probes are mapped into probe sets, which are ultimately mapped into genes. This information is used during

data normalization and for handling missing values.

One of the challenges in analyzing gene expression data is their high dimensionality. A data set may contain tens of thousands of gene probes, but at most a few hundred measurements; this makes it hard to do any modelling. Usually the genes whose expression values vary only a little between different cell samples are left out [18]. Thus the number of relevant genes in a study may be several thousands. Almost never a single experiment contains an adequate number of samples, resulting in the so called *"small n, large p"* -problem, where n denotes the number of samples and p the number of features (genes). For instance, it is not possible to reliably identify cancer genes if there are only a few patients in the study.

2.2 Machine learning

2.2.1 Unsupervised learning

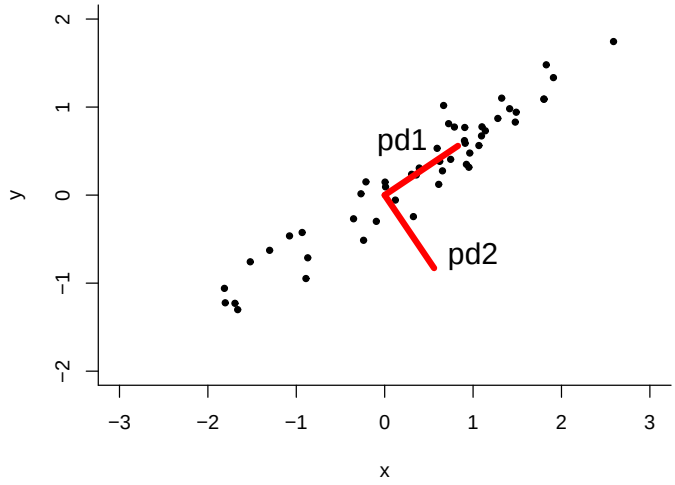
Unsupervised machine learning focuses on finding structure from unlabeled data, that is, data where we do not have additional information about the nature of samples. The focus lies on tasks such as clustering and dimensionality reduction.

The goals of the tasks can be briefly summarized as follows. In clustering, the data are intended to be divided into groups such that the data points in the same group have similar features and the data points in different groups have dissimilar features. In dimensionality reduction, we would like to define a small number of features that describe the structure of the data as faithfully as possible. The quality of the dimensionality reduction is evaluated by two measures: precision and recall. Good precision means that points that are close in the reduced space are also close in the original space. On the other hand, good recall means that points that are close in the original space are close in the reduced space, too. In general, it is impossible to maximize both at the same time, and trade-offs will have to be made.

An extremely popular model for unsupervised dimensionality reduction is principal component analysis (PCA) [14]. It is applied for centered data by simply rotating the coordinate axes such that the first axis (principal direction) corresponds to the direction with the largest variance, and each subsequent axis corresponds to the direction with the largest variance in the orthogonal subspace. This can be seen in figure 1 for a two-dimensional example.

Thus, the last principal components usually contain only a minimal part of the data variance and can be left out, leading to dimensionality reduction. Since PCA just projects the data linearly into its *principal subspace*, it has good recall. This is obvious because removing dimensions can only bring the data points closer to each other. However, this

Figure 1: Principal components of a data cloud. The first principal direction (pd1) corresponds to 98% of sample variance.



may apply also to distant points, resulting in bad precision.

Let us assume N data vectors $x_1, ..., x_N \in \mathbb{R}^M$. For solving the principal directions $w_1, ..., w_N$, the data set needs to be centered, after which the expectation value $\mathrm{E}[x_i]$ equals 0. First, we would like to find a linear projection that maximizes the variance of the first principal components $y = (y_1, ..., y_N)$, where $y_i = w_1^T x_i$.

$$\begin{aligned} \mathrm{Var}[y] &= \mathrm{Var}[w_1^T x_i] && (1)\\ &= E[(w_1^T x_i)^2] - E[w_1^T x_i]^2 && (2)\\ &= E[(w_1^T x_i)(x_i^T w_1)] - (w_1^T E[x_i])^2 && (3)\\ &= w_1^T E[x_i x_i^T] w_1 && (4) \end{aligned}$$

The last equation follows because $\mathrm{E}[x_i]$ equals zero. In order to obtain sensible results, the norm of w_1 is set to one, i.e. $w_1^T w_1$=1. Thus, we can form the following Lagrange function:

$$w_1^T E[xx^T] w_1 - \lambda(w_1^T w_1 - 1), \tag{5}$$

where λ is the Lagrange constant. Setting the gradient to zero, we get:

$$\frac{\partial}{\partial w_1}[w_1^T E[xx^T]w_1 - \lambda(w_1^T w_1 - 1)] = 0 \tag{6}$$

$$2E[xx^T]w_1 - \lambda(2w_1) = 0 \tag{7}$$

$$E[xx^T]w_1 = \lambda w_1 \tag{8}$$

We can see that equation (8) is the eigenvalue equation for matrix $E[xx^T]$. Knowing that there are N pairs of eigenvalues and -vectors, we still need to decide which one of them to choose. After multiplying equation (8) by w_1^T we get

$$w_1^T E[xx^T]w_1 = w_1^T \lambda w_1 = \lambda, \tag{9}$$

since $w_1^T w_1$=1. The largest eigenvalue should be chosen since $\mathrm{Var}[y] = w_1^T E[xx^T]w_1 = \lambda$. Thus the largest eigenvalue tells the variance captured by the first principal direction w_1, that is the eigenvector corresponding to the largest eigenvalue. For the next principal component we capture the maximum variance in an orthogonal direction, represented by the next largest eigenvalue and its corresponding eigenvector. Consequently PCA can be solved with the help of eigenvalues of $C_x = \mathrm{E}[xx^T]$. Furthermore we get the new features (principal components) $Y = W^T X$, where $W = (w_1, ..., w_P)$ is a $M \times P$ matrix formed by the P first principal directions and $X = (x_1, ..., x_N)$ is a $M \times N$ matrix. The dimensionality is reduced in a way that preserves as much variance as possible.

2.2.2 Supervised learning

Supervised learning is a machine learning setup where we have label information for the data. The most basic task is to divide the feature space in two parts: class 1 ($\mathcal{C}_1$) and class 2 ($\mathcal{C}_2$). A very popular choice for this kind of classification algorithm are support vector machines (SVM) [6]. A linear SVM simply finds two parallel hyperplanes that separate the two groups (samples with different labels) with a maximum margin between them. The hyperplanes are learned from training data, after which test data will be classified according to which side of the hyperplanes they lie on. Given that the classes actually are linearly separable, we can formulate the problem as follows:

$$w^T x(i) + w_0 \geq 1, \text{when } x(i) \in \mathcal{C}_1 \tag{10}$$

$$w^T x(i) + w_0 \leq -1, \text{when } x(i) \in \mathcal{C}_2, \tag{11}$$

where w is the normal vector of the hyperplanes H_1 and H_2:

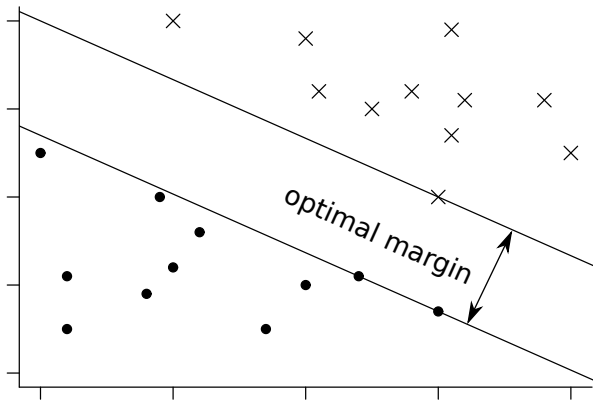
$$H_1 : w^T x(i) + w_0 = 1 \tag{12}$$

$$H_2 : w^T x(i) + w_0 = -1 \tag{13}$$

and $\frac{w_0 - 1}{||w||}$ and $\frac{w_0 + 1}{||w||}$ determine the offsets of the hyperplanes from the origin along the normal vector w.

The concept of SVM can be easily seen in figure 2. To deal with cases where the classes are not linearly separable, the optimization problem is slightly changed. The exact derivations are not relevant since most mathematical programs contain an SVM-implementation.

Figure 2: Two classes separated with an optimal margin.



In all models in this thesis, we use linear SVM for classification. Thus, the models differ only regarding feature selection.

3 Transfer learning

In this chapter, we introduce the concept of transfer learning in detail and divide it in different categories. Two models are presented to be used for feature selection.

3.1 Preliminaries

We are interested in a task and have one specific dataset at hand; it shall be called *target*. For some reason, for example due to a small amount of samples, target data may not allow us to form a good enough model, in which case transfer learning should be applied. Some knowledge is intended to be transferred from the *source*, which is somehow related to the target. The relation can vary from clear similarity to a scenario where the source might seem irrelevant to target data: for example, we might repeat a biological study with just different patients; on the other hand, in the concept of image recognition transfer learning has even been applied using totally random images as source data [16]. In the latter case labeling accuracy improved since even the random images share something to the target, for example the edges of patterns can be modelled better if there is previous knowledge of them.

Target and source data both can still be divided in two parts: *domain* and *task* [13]. A domain $\mathcal{D}$ is defined:

$$\mathcal{D} = \{\mathcal{X}, P(X)\}, \tag{14}$$

where $\mathcal{X}$ is feature space and $P(X)$ a marginal probability distribution for $X = \{x_1, ..., x_n\} \in \mathcal{X}$. Here X is the actual data, consisting of n samples. Marginal probability distribution $P(X)$ can be estimated from the data. Two domains are obviously different if their $\mathcal{X}$ or $P(X)$ differ. As a simple example we might have data from a set of patients, feature space containing height and weight, and $P(X)$ their marginal probabilities.

Given a specific domain $\mathcal{D}$, a task is:

$$\mathcal{T} = \{\mathcal{Y}, f(\cdot)\} \tag{15}$$

where $\mathcal{Y}$ is label space and $f(\cdot)$ an objective predictive function, i.e. a function for predicting labels from the features X. The objective predictive function is not observed, but it can be learned from the training data, given labels $Y \in \mathcal{Y}$. In the simple example labels could be binary, telling if the patient is male or female. Given the training data, we can learn to predict whether a patient is male or female, given his/her height and weight.

3.2 Transfer learning categories

Transfer learning can be divided into different fields using the data division to domain and task. We are interested in the target experiment, and will transfer knowledge from the source. If both source and target domains, and source and target tasks are the same, i.e. there is an earlier experiment just like ours, transfer can be done in traditional machine learning setup, where the source data is simply added to our model, resulting in a single study. On the other hand, if there are differences in domains or tasks, it will be important to find out what to transfer. Some knowledge may be source-specific and some common with the target, which requires transfer learning methods. Complete classification of the approaches can be found in table 1:

Table 1: Different learning settings, divided using similarities in domain and task space (taken from [13]).

Learning Settings	Source and Target Domains	Source and Target Tasks
Traditional Machine Learning	the same	the same
Inductive Transfer Learning /	the same	different but related
Unsupervised Transfer Learning	different but related	different but related
Transductive Transfer Learning	different but related	the same

In the *inductive transfer learning* setting, the target task is different from the source task, whereas the domains may be the same or not. Some labeled data in the target domain are required to induce an objective predictive model $f_T(\cdot)$. A special case of inductive transfer learning is multitask learning, where source and target tasks are learnt simultaneously, taking into account that they are different [2]. It has been one of the most popular approaches in transfer learning. Typical application examples include regression and classification [15].

In the *transductive transfer learning* setting, the source and target tasks are the same, while the domains are different. No labeled data in the target domain are available whereas a lot of labeled data are available in the source domain. A popular example of this type of setting is domain adaptation, where the aim is to transfer knowledge from source feature space to target feature space [8].

Finally, in the *unsupervised transfer learning* setting, similar to *inductive transfer learning*, the target task is different from but related to the source task. However, there are no labeled data available in both source and target domains. Thus, labels cannot be estimated, but the applications may contain for example clustering and dimensionality reduction [7, 21].

3.3 Transfer learning for feature selection

3.3.1 Transfer-PCA

Dimensionality of the target features can be easily reduced by calculating their principal components, as presented in Section 2.2.1. However, one needs to pay attention that an insufficient sample size will not result in a robust principal subspace: with ten samples, 100 percent of the sample variance will be captured with at most nine principal directions - no matter how large the actual dimensionality is. Thus, it would be very beneficial to have the sample size at least close to the number of dimensions.

Our approach is to estimate a more robust principal subspace from source data, which is naturally achieved via a greater sample size. Importantly the feature marginal distributions have to have something in common in order for knowledge to be transferred, which may allow us to learn a higher level representation of the target data. Technically this requires only projecting the target data into the principal subspace of the source and choosing the desired number of components. This feature selection approach shall be called *transfer-PCA*.

3.3.2 Self-taught learning

Self-taught learning is classified as an inductive transfer learning model. More specifically, it is used in situations where the feature spaces $\mathcal{X}$ are identical in source and target and marginal distribution $P(X)$ is different but related. Goal is to classify target data based on some labeled data and the source data, which contains no labels. It is assumed that the target consists only of rather few samples, making it hard to obtain a good classifier. Thus, source data should be used for help, even though it is assumed that the source samples may not be classified according to target labels at all. However, since source and target domains are related, some knowledge can be transferred. Raina et al. used the classification of images of elephants and rhinos as an example. Acquiring more images with the correct label information is considered expensive, and not necessarily plausible in many real applications. Thus, random images were used as the source, and out of them possibly none contained elephants or rhinos. Still there is something shared, since most pictures contain basic forms such as edges. This type of high-level information was transferred successfully, resulting in a better classification accuracy than without the seemingly irrelevant source data.

Self-taught learning algorithm begins with learning a higher-level representation of the source. Source data consists of unlabeled data $\{x_u^{(1)}, ..., x_u^{(k)}\}$ with each $x_u^{(i)} \in \mathbb{R}^n$. This is posed as the following optimization problem:

$$\min_{b,a} \sum_i ||x_u^{(i)} - \sum_j a_j^{(i)} b_j||_2^2 + \beta||a^{(i)}||_1 \tag{16}$$
$$\text{s.t. } ||b_j||_2 \leq 1, \forall j \in 1, ..., s$$

The optimization variables in this problem are the *basis vectors* $b = \{b_1, ..., b_s\}$ with each $b_i \in \mathbb{R}^n$, and the activations $a = \{a^{(1)}, ..., a^{(k)}\}$ with each $a^{(i)} \in \mathbb{R}^s$. Basis vectors b are used to capture the high-level representations, such as common forms or clusters. $a_j^{(i)}$ is the activation of basis b_j for input $x_u^{(i)}$. The optimization has two terms: (i) The first quadratic term encourages each input $x_u^{(i)}$ to be reconstructed well as a weighted linear combination of the bases b_j. (ii) The activations a should be sparse i.e. most elements of $a^{(i)}$ should be zero.

The optimization problem (16) can be solved iteratively over variables a and b by alternatingly holding one of them fixed and optimizing over the other one [11]. This stage of finding representations of unlabeled data is known as *sparse coding*, for which Lee et al. presented algorithms in their publication. In this thesis, problem (16) is solved using *feature-sign search algorithm*.[1]

After learning the basis vectors from source data the target data $\{x_l^{(1)}, ..., x_l^{(m)}\}$ with each $x_l^{(i)} \in \mathbb{R}^n$ are represented using them. This results in the following optimization problem:

$$\hat{a}(x_l^{(i)}) = \arg\min_{a^{(i)}} ||x_l^{(i)} - \sum_j a_j^{(i)} b_j||_2^2 + \beta||a^{(i)}||_1, \tag{17}$$

where the labeled target samples are assigned individually into basis b_j optimally, such that their reconstruction error is low and the activations are sparse. Later the activation vector $a^{(i)}$ will be used as a representation of sample $x_l^{(i)}$. As the final stage a classifier $\mathcal{C}$ will be learned by applying a supervised learning algorithm (e.g. SVM) to the labeled training set containing activations and the sample labels. Unconstrained minimization problem (17) can be solved using MATLAB's built-in minimizer, which can use more efficient methods if both gradient and Hessian matrix are provided. The cost function's matrix formulation is:

$$C(a^{(i)}) = ||x_l^{(i)} - Ba^{(i)}||_2^2 + \beta||a^{(i)}||_1, \tag{18}$$

where the bases b_j are the columns of matrix B and $a^{(i)}$ is a column vector containing the $a_j^{(i)}$s. The gradient can be computed in the following way:

[1] `http://ai.stanford.edu/~hllee/softwares/nips06-sparsecoding.htm`

$$\frac{\partial}{\partial a^{(i)}} C(a^{(i)}) \tag{19}$$

$$= \frac{\partial}{\partial a^{(i)}} [||x_l^{(i)} - Ba^{(i)}||_2^2 + \beta ||a^{(i)}||_1] \tag{20}$$

$$= 2 \frac{\partial}{\partial a^{(i)}} ||x_l^{(i)} - Ba^{(i)}||_2 \times ||x_l^{(i)} - Ba^{(i)}||_2 + \beta \text{sign}(a^{(i)}) \tag{21}$$

$$= 2 \frac{\partial}{\partial a^{(i)}} [Ba^{(i)}] \times - \frac{x - Ba^{(i)}}{||x_l^{(i)} - Ba^{(i)}||_2} \times ||x_l^{(i)} - Ba^{(i)}||_2 + \beta \text{sign}(a^{(i)}) \tag{22}$$

$$= -2B^T(x - Ba^{(i)}) + \beta \text{sign}(a^{(i)}) \tag{23}$$

Furthermore, the corresponding Hessian is:

$$\frac{\partial^2}{\partial a^{(i)} \partial a^{(i)^T}} C(a^{(i)}) = \frac{\partial}{\partial a^{(i)^T}} [-2B^T(x - Ba^{(i)}) + \beta \text{sign}(a^{(i)})] = 2B^T B \tag{24}$$

Using the gradient (23) and Hessian matrix (24) *fminunc* calculates the local minimum of (17) for all the target samples independently. The algorithm used is a subspace trust-region method and is based on the interior-reflective Newton method described in [4] and [5].

As a conclusion, knowledge transfer in self-taught learning is done by extracting high-level information from the source task in the form of a basis b, which is used to represent the target data. If the amount of data in the source is large, the learned basis will be robust, unlike for the potentially few samples in target. Transfer-PCA is based on the same idea, limiting to a linear projection instead of the freely optimized basis.

4 Experiments

The experimental section consists of two different classification tasks: in Section 4.2, the classes are *hepatitis C* and *normal*, whereas in Section 4.3 we classify *blood* versus *non-blood*. Both tasks were performed using data repository presented in Section 4.1.

4.1 Data repository

Margus Lukk et al. collected a repository containing 5372 human gene expression measurements [12]. They combined a total of 206 studies with variable goals, from searching a molecular explanation for different hepatitis C treatment results, to finding connections among diseases, genetic perturbation and drug action [19, 9]. All measurements were performed using Affymetrix microarray U133A, resulting in identical feature spaces $\mathcal{X}$, consisting of expression values for more than 22000 probe sets. However, since different phenomena are studied, marginal distributions $P(X)$ and thus the domains between different studies differ. There are thorough annotations for each sample, describing the tissue of origin, its disease state etc. Readily preprocessed data is acquired from the Gene expression atlas.[2] It shall be called *original data* from now on.

In their article, the authors presented "a global map of human gene expression", which was a visualization of the two first principal components of the data, with sample coloring according to different classes. However, this kind of huge data repository is a natural setup for applying transfer learning. Although all the samples in the repository are thoroughly labeled, we will discard the label information in source data. Naturally a better model could be built using all the information available, but in this thesis the interest lies in testing feature transfer from unlabeled source data. This is a more realistic setting because consistent annotation of large data repositories is tedious.

4.2 Predicting hepatitis C

As an example classification task, we used the second largest experiment in the repository of Lukk et al., which contains 308 samples from human blood cells - 192 of which are associated with hepatitis C and 116 of which are healthy [19]. The original data samples are divided randomly into two sets for training and testing. A linear SVM model is formed for the training set as described in Section 2.2.2 and the classes of test data (hepatitis C versus normal) are predicted according to the model. For transfer learning purposes all the other experiments in the data repository are used as source data.

Prediction accuracies are compared in four different settings, using:

[2] `http://www.ebi.ac.uk/gxa/experiment/E-MTAB-62`

Baseline	Original data
	Principal components of the raw data (PCA)
Transfer learning	Transfer-PCA
	Self-taught learning

The two first settings are traditional machine learning ones. Additionally in the second one calculation time is lower since SVM will be applied to a data set with approximately 40 percent less features (yet storing at least 96 percent of sample variance). As a downside, the estimated principal subspace and components will not be robust due to the small sample size. The third setting implements a simple knowledge transfer: the principal subspace estimated from the source will be more reliable since the sample size is over ten times larger, thus aiding in the *"small n, large p"*-problem. In the last setting, activations of the self-taught learning model will be used as features. The goal is to benefit more from the source data by acquiring high-level structural information through sparse coding.

Comparisons were made for different sizes of training sets and the results can be found in table 2.

Table 2: Predicting hepatitis C versus normal blood cell. Means and standard deviations of prediction accuracies for 200 different divisions to training and test sets. Tested with variable training set size k.

	Original	PCA	Transfer-PCA	Self-taught learning
k=20	**69.4% (4.7%)**	66.8%(5.4%)	**69.4% (5.0%)**	53.7%(4.4%)
k=50	76.8%(3.6%)	74.1%(4.1%)	**79.4% (3.0%)**	57.9%(3.5%)
k=100	83.4%(2.8%)	82.3%(3.2%)	**88.2% (3.0%)**	62.5%(3.7%)
k=250	93.0%(3.4%)	94.7%(2.7%)	**96.2% (2.6%)**	66.2%(6.0%)

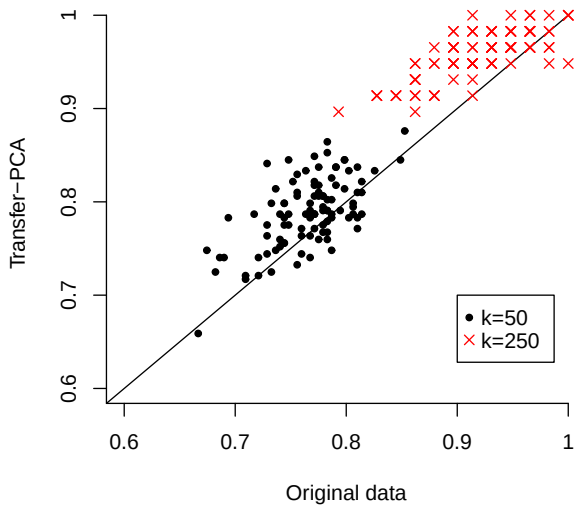
In table 2 the first column provides the baseline, where the original data are not modified. If no dimensionality reduction were done in the second case, results would be exactly the same, since both PCA and SVM are linear models. However, dimensionality was reduced such that at least 96 percent of sample variance was captured, thus discarding the directions with marginal variance. This produced worse prediction accuracies with all the training set sizes except 250.

Transfer learning was applied in the other two setups. Transfer-PCA resulted in the best prediction accuracies with all training set sizes. The much more complex self-taught learning model produced the worst prediction accuracies of all.

Furthermore, a pairwise comparison between the two settings with the best performance - original data and transfer-PCA - is visualized in figure 3. This allows us to compare

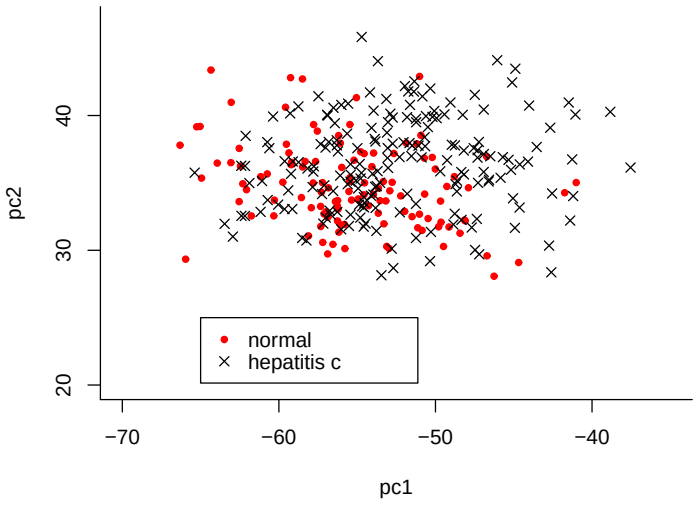
the best baseline against the best transfer learning method.

Figure 3: SVM prediction accuracies compared with the original data and transfer-PCA. Two training set sizes were used, and with the bigger one points are overlapping due to clearly discrete prediction accuracies.



A simple overview on the data can be obtained by visualizing them using the first two principal components of the source data. This is done in figure 4, showing that the different groups cannot be at least trivially separated.

Figure 4: The two first principal components of the target data, projected in the source's principal subspace. Normal and hepatitis C infected samples are marked differently.



4.3 Predicting cell type

In order to test further whether the performance of different SVM setups prevails in different data sets, another experiment was made. Here, the target task was chosen such that it contains samples from both blood and non blood cells. There were 25 blood samples and 299 non-blood. Due to the unbalanced classes, only random 50 of the non-blood samples were chosen for the experiment. All the other experiments were again left as source data. This time all the samples could actually be classified, but label information was discarded again. Four different feature spaces were tested with SVM, and the results can be found in table 3. Again, the prediction accuracies were tested with different training set sizes, leaving all the other target data into test set.

Table 3: Predicting non-blood versus blood cells. Means and standard deviations of prediction accuracies for 200 different divisions to training and test sets. Tested with variable training set size k.

	1	2	3	4
k=5	96.9%(5.2%)	**97.4% (5.7%)**	97.3%(6.6%)	78.4%(13.9%)
k=10	97.8%(3.4%)	97.2%(5.3%)	**98.3% (2.4%)**	82.3%(11.1%)
k=20	98.5%(0.8%)	98.5%(1.5%)	**98.6% (1.7%)**	83.6%(10.0%)
k=30	**98.7% (1.1%)**	98.6%(1.1%)	**98.7% (1.1%)**	84.5%(8.6%)

This time the classification task was much easier, resulting in close to 100 percent accuracies. The performance of different feature selection methods persisted, for most parts. Normal PCA resulted in relatively better accuracies, whereas transfer-PCA had a clear lead only with training sets containing 10 samples. Activations of self-taught learning were clearly outperformed by the other features.

5 Conclusions

The goal of this thesis was to compare the performance of transfer learning models to standard machine learning ones in classification tasks based on gene expression data. This was done using two different divisions of the data repository: one with target labels *hepatitis C* and *normal* and the other one with *blood* and *non-blood*. Labels of the source task were discarded, resulting in a more generally applicable learning problem. Classification was done using SVM with variable training set sizes.

Two standard machine learning setups used the original features and their principal components. Since the principal components resulted in slightly worse prediction accuracies, it seems that the discarded directions were helping in the division between normal and hepatitis C infected cells. Results for classifying blood versus non-blood cells in table 3 were practically the same, except for regular PCA performing slightly better.

In the transfer-PCA setup, transfer of knowledge was intended to be achieved with the help of large source data. Seemingly, a larger number of samples is advantageous for estimating a more robust principal subspace, and positive transfer of knowledge was achieved.

Self-taught learning, however, resulted in poor prediction accuracies. This could be due to several reasons. One mentioned in the original paper is that good performance is not achieved unless the activations a_j are sparse [16]. Optimization algorithms presented in section 3.3.2 resulted at maximum 81 percent of activations being below 0.001. Sparsities in the paper were greater and could not be achieved in this experiment no matter how the parameter were tweaked. In equation (17), the constant β was adjusted even 30 times as high as in the first optimization problem (16). Once set high enough a sensible optimum could not be found. Sparsity was even forced by rounding small activations to zero, resulting in problems with SVM. With only few non-zero activations many features were the same for most samples, thus not aiding at all in finding the classification boundary.

Other tweaks used in the original study were tried too. First we tried combining original features with the activations, which in this experiment performed still worse than the plain original features. They did perform better than the plain activations though, but the results acquired using plain activations were displayed, since combining them with original data for better results seems a bit *ad hoc*. More experiments on self-taught learning are needed to find out its performance in different data sets. As for the data used in this thesis, some crucial modifications would be needed in order to guarantee even an adequate performance.

Implementing transfer-PCA when possible is recommended, due to its improved prediction performance. Only requirement is a set of source data with the same feature space

and at least for some parts similar marginal distribution. At least with SVM the results of this thesis should be generalizable to different types of data sets. Further research is required to see how well transfer-PCA performs with other classification models.

The experiments showed that transfer can be advantageous, but differed from expectations. For one transfer-PCA performed pretty equal to the baselines with a very small amount of samples, but better otherwise. Expected behavior would have been the other way, and there is no apparent explanation for this phenomenon. Additionally the prediction performance of self-taught learning was surprisingly bad, providing an example of negative transfer [17].

References

[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walter. *Molecular biology of the cell*. fourth edition, 2002.

[2] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[3] M.H. Cheok, W. Yang, C.H. Pui, J.R. Downing, C. Cheng, C.W. Naeve, M.V. Relling and W.E. Evans. Treatment-specific changes in gene expression discriminate in vivo drug response in human leukemia cells. *Nature genetics*, 34(1):85–90, 2003.

[4] T.F. Coleman and Y Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996.

[5] T.F. Coleman and Y. Li. On the convergence of interior-reflective newton methods for nonlinear minimization subject to bounds. *Mathematical Programming*, 67:189–224, 1994. ISSN 0025-5610.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[7] W. Dai, Q. Yang, G.R. Xue and Y. Yu. Self-taught clustering. *Proceedings of the 25th international conference on Machine learning*, pages 200–207. ACM, 2008.

[8] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126, 2006.

[9] J. Lamb, E.D. Crawford, D. Peck, J.W. Modell, I.C. Blat, M.J. Wrobel, J. Lerner, J.P. Brunet, A. Subramanian, K.N. Ross et al. The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science*, 313 (5795):1929, 2006.

[10] J. Lapointe, C. Li, J.P. Higgins, M. Van De Rijn, E. Bair, K. Montgomery, M. Ferrari, L. Egevad, W. Rayford, U. Bergerheim et al. Gene expression profiling identifies clinically relevant subtypes of prostate cancer. *Proceedings of the National Academy of Sciences of the United States of America*, 101(3):811, 2004.

[11] H. Lee, A. Battle, R. Raina and A.Y. Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19:801, 2007.

[12] M. Lukk, M. Kapushesky, J. Nikkilä, H. Parkinson, A. Goncalves, W. Huber, E. Ukkonen and A. Brazma. A global map of human gene expression. *Nature biotechnology*, 28(4):322–324, 2010.

[13] S.J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1345–1359, 2009.

[14] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2(11):559–572, 1901.

[15] A. Quattoni, M. Collins and T. Darrell. Transfer learning for image classification with sparse prototype representations. *Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[16] R. Raina, A. Battle, H. Lee, B. Packer and A.Y. Ng. Self-taught learning: Transfer learning from unlabeled data. *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007.

[17] M.T. Rosenstein, Z. Marx, L.P. Kaelbling and T.G. Dietterich. To transfer or not to transfer. *NIPS Workshop on Inductive Transfer*. Citeseer, 2005.

[18] M.A. Shipp, K.N. Ross, P. Tamayo, A.P. Weng, J.L. Kutok, R.C.T. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G.S. Pinkus et al. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature medicine*, 8(1):68–74, 2002.

[19] M.W. Taylor, T. Tsukahara, L. Brodsky, J. Schaley, C. Sanda, M.J. Stephens, J.N. McClintick, H.J. Edenberg, L. Li, J.E. Tavis et al. Changes in gene expression during pegylated interferon and ribavirin therapy of chronic hepatitis c virus distinguish responders from nonresponders to antiviral therapy. *Journal of virology*, 81(7):3391, 2007.

[20] S. Thrun. Is learning the n-th thing any easier than learning the first? *Advances in neural information processing systems*, pages 640–646, 1996.

[21] Z. Wang, Y. Song and C. Zhang. Transferred dimensionality reduction. *Machine Learning and Knowledge Discovery in Databases*, pages 550–565, 2008.