Survival Prediction using Gene Expression Data - A Topic Modeling Approach

by

Luke Nitish Kumar

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Survival prediction is becoming a crucial part of treatment planning for most terminally ill patients. Many believe that genomic data will enable us to better estimate survival of these patients, which will lead to better, more personalized treatment options and patient care. As standard survival prediction models cannot cope with the high-dimensionality of such gene expression data, many projects use some dimensionality reduction techniques to overcome this hurdle. We introduce a novel methodology, inspired by topic modeling from the natural language domain, to derive expressive features from the high dimensional gene expression data. There, a document is represented as a mixture over a relatively small number of topics, where each topic is a distribution over the words; here, to accommodate the heterogeneity of a patient's cancer, we represent each patient (document) as a mixture over "(cancer) strains" (topics), where each strain is a mixture over gene expression values (words).

After using our novel discretized Latent Dirichlet Allocation (dLDA) procedure to learn these strains, we can then express each patient as a distribution over a small number of strains, then use this distribution as input to a learning algorithm. We then ran a recent survival prediction algorithm, MTLR, on this representation of the cancer dataset. Here, we focus on the METABRIC dataset, which describes each of n=1,981 breast cancer patients, using k=49,576 gene expression values. Our results show that our approach (dLDA followed by MTLR) provides survival estimates that are more accurate than standard models, in terms of the standard Concordance Index, as well as a relevant novel measure, D-calibration. We then validate this approach on the n=1082 TCGA BRCA dataset, over k=20532 gene expression values.

Holding onto anger is like drinking poison and expecting the other person to Die.

– Gautama Buddha.

Acknowledgements

First and foremost I would like to thank my advisor Prof. Russell Greiner for his continuous support, guidance and genuine interest in my academic and professional career. I greatly appreciate his counsel and suggestions throughout the program helping me to adapt to the new environment, culture and Edmonton.

I'm also grateful to my colleges at the Computing Science Department, especially Roberto Vega and Neil Borle for the lengthy discussions and their continuous support. I like to thank my examiners, Prof. Dale Schuurmans and Prof. David Wishart for their invaluable insights, comments, and appreciation. I'm also greatly thankful to Compute Canada for the computing resources through the West-Grid servers and Alberta Innovates Centre for Machine Learning, and NSERC for supporting this work. Finally, I would like to inscribe my gratitude to my parents, Donald & Vasanthy for their unconditional love and continuous support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

WHO's world report on cancers lists breast cancer as one of the common cancers among women around the globe (Stewart *et al.*, 2016) and one of the leading cause of death by any cancer (Koboldt and Network, 2012). Many analyses start with clinical features. Unfortunately, identifiers such as lymph node status and histological grade, which are highly predictive of metastases, do not appear to be sufficient to reliably categorize clinical outcome (Van't Veer *et al.*, 2002), and in general, the outcomes can vary widely for patients with similar diagnoses, who receive the same treatment regimen. This has led to many efforts to improve the prognosis for breast cancer, based on genomics data (*e.g.*, gene expression (GE) or copy number variation (aberrations) CNV) along with the clinical data (Margolin *et al.*, 2013; Parker *et al.*, 2009; Naderi *et al.*, 2007; Van't Veer *et al.*, 2002). This motivates efforts to find intrinsic cancer subtypes, by identifying new gene signatures. Parker *et al.* (2009) identified five subtypes of breast cancer, based on a panel of 50 genes (PAM50): luminal A, luminal B, HER2-enriched, basal-like, and normal-like. Later, Curtis *et al.* (2012) examined $\sim$2000 patients from a wide study combining clinical and genomic data, and identified around ten subtypes. Both of these studies showed that their respective subtypes produce significantly different Kaplan-Meier survival curves (Altman, 1990), suggesting such molecular variation does influence the disease progression.

More recently, many survival prediction models have been applied to breast cancer cohorts; some are based on standard statistical survival analysis tech-

niques, and others based on classic regression algorithms – *e.g.*, random survival forests (Ishwaran *et al.*, 2008), censored SVM (Shivaswamy *et al.*, 2008). With the growing number of gene expression experiments being cataloged for analysis, we need to develop survival prediction models that can utilize such high dimensional data. Our work describes such a system that can learn effective survival prediction models from high dimensional gene expression data.

The 2012 DREAM Breast Cancer Prognosis Challenge (BCC) was designed to focus the community's efforts to improve breast cancer survival prediction (Margolin *et al.*, 2013). Its organizers made available clinical and genomic data (GE & CNV) of $\sim$ 2000 patients from the Curtis *et al.* (2012) study. The winning model (Cheng *et al.*, 2013) performed statistically better than the state-of-the-art benchmark models (Margolin *et al.*, 2013).

Each submission to the BCC challenge identified each patient with a single real value (called "risk"), which is predicting that patients with higher risk should die earlier than those with lower risk. The entries were therefore evaluated based on the concordance measure: basically, the percentage of these pairwise predictions, that were correct (Kalbfleisch and Prentice, 2011). This is standard, in that most survival prediction tasks use the concordance as the primary measure to assess the performance of the survival predictors: *e.g.*, Breast Cancer Dream Challenge (Margolin *et al.*, 2013), Prostate Cancer Dream Challenge (Abdallah *et al.*, 2015). The concordance measure is appropriate if we need to order the survival times of the instances – *e.g.*, if we have one liver available for transplant and need to know which candidate patient will die first, without a transplant; or when we want to know which machine will fail first, to determine where to assign the single repair person.

However, such "risk scores" provide only a relative ordering of the instances – *i.e.*, it is a "discriminatory" score (Steyerberg *et al.*, 2010). For many tasks, however, it is more important to accurately predict a patient's actual survival times, or her chance of surviving at least 5 years, etc. In particular, (1) knowing the survival time for a specific breast cancer patient would help that patient plan her future actions – *e.g.*, hospice care versus making long-term plans – and (2) similarly it could help the physicians compare treatment options for a

patient. As we want models that can accurately estimate survival time, we will augment the standard (if often inappropriate) concordance measure with another "calibration" score: D-calibration; see Section 2.6.2.

This thesis proposes a novel topic modeling approach discretized Latent Dirichlet Allocation (dLDA) that can derive highly predictive covariates from the high-dimensional microarray (gene expression) data. This lets us map the microarray description into a much lower dimensional description (here, from $\sim$50K features to 30 in this BCC dataset), which can then be given as input to a recent non-parametric learning algorithm, multi-task logistic regression (MTLR), which is capable of producing a model that can then predict an individual's survival distribution (Yu *et al.*, 2011). We show that this predictor performs better than other standard survival analysis tools, in both discrimination (concordance) and calibration (D-calibration) evaluations.

Chapter 2 introduces and describes: the datasets used (both METABRIC from BCC, and BRCA from TCGA), the methods used in building the survival prediction models, various techniques for dimensional reduction, and the evaluation schemes. Then Chapter 3 presents our evaluation results, and finally Chapter 4 presents our discussion, conclusions and future work. In addition we have included three appendices with supplementary information: Appendix A presents concepts from survival prediction, Appendix B includes extended material on D-calibration and Appendix C provides a short tutorial on the LDA.

Chapter 2

Methods

Section 2.1 introduces the survival prediction task, Section 2.2 describes the datasets used in this study, Section 2.3 discusses the preprocessing stage, Section 2.4 presents the dimensionality reduction techniques we employed, and finally Sections 2.5 and 2.6 describe (resp.) the survival prediction algorithms and the evaluation schemes used in our experiments. Figure 2.1 gives the flow diagram of all the steps followed from the clinical and genomic data to survival predictions and evaluation.

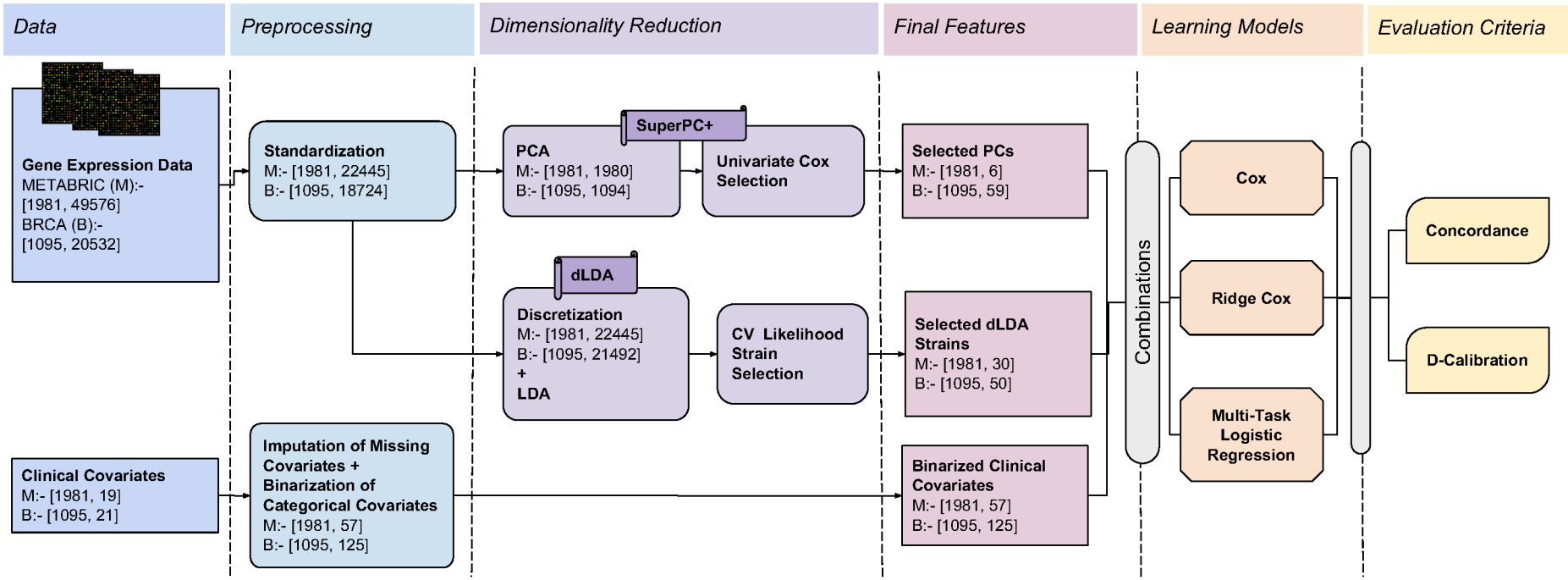


Figure 2.1: Data workflow diagram depicting all the stages from initial clinical and gene expression data, to final survival prediction and evaluation. METABRIC & BRCA = Datasets of Breast Cancer studies; associated pair of numbers are [#patients, #features], PCA = Principle Component Analysis, (d)LDA = (discretized) Latent Dirichlet Allocation, CV = Cross-Validation, PCs = Principle Components, Cox = Cox (1972), Discretization: see Section 2.4.2.

2.1 Survival Prediction

Survival prediction is similar to regression as both involve learning a model that regresses the covariates of an individual to produce an estimate of the value of a dependent real-valued response variable – here, the variable is "time to event" (where the standard event is "death"). But survival prediction differs from the standard regression task as its response variable is not fully observed in all training instances. In most real world cohorts, many of the instances are "right censored", in that we only see a *lower bound* of the response value. This might happen if a subject was alive when the study ended, meaning we only know that she lived *at least* 5 years, but do not know whether she actually lived 5 years and a day, or 30 years. This also happens if a subject drops out of a study, after say 2.3 years, and is then lost to follow-up; etc. Moreover, one cannot simply ignore such instances as it is common for many (or often, *most*) of the training instances to be right-censored; see Table 2.1. Such "partial label information" is problematic for standard regression techniques, which assume the label is completely specified for each training instance.

Fortunately, there are the survival prediction algorithms that can learn an effective model, from a cohort that includes such censored data. Each such dataset contains descriptions of a set of instances (*e.g.*, patients), as well as two "labels": one is the time, corresponds to the *time* from diagnosis to a final date (either death, or time of last follow-up) and the other is the *status* bit, which indicates whether the patient was alive at the last followup.

2.2 Data

We apply our methods to two large breast cancer datasets: METABRIC (Curtis *et al.*, 2012) and BRCA (Koboldt and Network, 2012). We initially focus on the METABRIC dataset, since it is one of the largest available survival studies that includes genomic information. In 2012, the Breast Cancer Prognostic Challenge (BCC) organizers released the METABRIC (Molecular Taxonomy of Breast Cancer International Consortium) dataset for training (Curtis *et al.*,

Table 2.1: Characteristics of the METABRIC and BRCA Cohorts

	METABRIC	BRCA
# Patients	1,981	1,082
# Censored	1,358 (∼68.5%)	930 (∼86%)
# Uncensored	623	152
Time span days (Uncensored)	3 – 8,941	26 – 7,455
# Clinical covariates	19	21
# Gene expression probes	49,576	20,532
Gender	Women (100%)	Women (99%), Men (1%)

2012). While they subsequently released a second dataset (OSLO) for final testing (Curtis *et al.*, 2012), we focus on only the METABRIC dataset, for several reasons: (1) METABRIC provided disease specific survival (DS), which considers only *breast cancer death* (BC death), rather than other causes of death also, non-BC deaths are considered censored here (Cheng *et al.*, 2013). By contrast, OSLO provides "overall survival" (OS), which does not distinguish BC-based deaths from others. As DS is clearly better for our purpose, it is better to evaluate on the METABRIC dataset. (2a) OSLO and METABRIC contained different sets of probes – and in particular, OSLO contains only ∼80% of the METABRIC probes. (2b) Similarly, the OSLO dataset is also missing some of the clinical covariates that are present in the METABRIC dataset – *e.g.*, menopausal status, group, stage, lymph nodes removed, etc.; see Table 1 from Margolin *et al.* (2013). This means a "METABRIC-OSLO study" would need to exclude some METABRIC features, and exclude some METABRIC probes.

We then wanted to use a second independent dataset, to verify the effectiveness of our approach. Here, we considered the OSLO dataset, but decided it had too few patients, and so instead used the BRCA dataset (Koboldt and Network, 2012) from TCGA (The Cancer Genome Atlas), which contains a sufficient number of patients. Table 2.1 lists some of the important characteristics of both datasets including both clinical and genomic information.[1] [2]

[1] METABRIC also included copy number variations (CNV) information for the patients, but as that did not improve performance in our preliminary analysis, we decided not to include CNV data in any of our models.

[2] We first removed two patients from the BRCA dataset because of patient ID mismatch with clinical and gene expression data, and further removed thirteen patients with zero

Note that METABRIC contains 49,576 probes compared to 20,532 in BRCA. In METABRIC, each gene may correspond to multiple probes targeting different DNA segments of the gene, therefore we have a one-to-many relation between the probes and the genes[3]. Throughout this study, we use will all the probes as given, without collating them to a one-to-a-gene relation.

Training and Validation

We apply the same experimental procedure to both datasets (METABRIC and BCRA): We partition each dataset into two sets with 80/20 split, with 80% of the data is used for training and the rest 20% for testing. Both partitions contains instances with comparable ranges of survival times and maintains the censored vs uncensored ratio. We trained on the 80% set (1586 patients in METABRIC and 865 in BRCA), then used the held-out validation set (of 395 patients in METABRIC and 217 patients in BRCA) for final testing. When necessary, we run internal cross-validation, within that training set, to find good settings for parameters, etc.

2.3 Preprocessing Steps

This section describes the steps applied to the initial data to produce a normalized dataset without any missing values, ready for the subsequent steps in the pipeline – See Figure 2.1.

We used standard steps to preprocess the clinical covariates: (1) impute missing real (resp., categorical) values for a feature with the mean (resp., mode) of the observed values for that feature; and (2) binarizing each categorical variable (aka "one-hot encoding") – *e.g.*, we encoded the 3-valued TumorGrade using three bits: grade one is [1, 0, 0], grade two is [0, 1, 0], and grade three is [0, 0, 1]. As we want to deal with the log of the initial gene expression value, we first $\log_2$-transformed the data, if necessary. (Below we use "gene expression" to refer to this transformed value.) We then discretized

censored survival time.

[3]METABRIC probes are designed using the RefSeq and UniGene databases.

these Gene Expression Values: Many genes have fairly constant expression values (typically near 0), which are therefore not very useful in learning topic models as they convey very little information – similar to stop words in natural language (Blei *et al.*, 2003). As a way to identify (and then ignore) such genes, we first translate all expression values into their "common z-scores". That is, we first compute the (common) mean and standard deviation over all the genes from the entire GE dataset: Letting $\hat{e}_i^j$ be the expression value of gene g_i of patient j, we compute the common mean $\mu = \frac{1}{n}\sum_{i,j}\hat{e}_i^j$, where n = 1,981×49,576 is the total number of entries (for METABRIC), and the variance $\sigma^2 = \frac{1}{(n-1)}\sum_{i,j}(\hat{e}_i^j - \mu)^2$. We then apply the Z-score transformation to each entry: $z_i^j = \frac{(\hat{e}_i^j - \mu)}{\sigma}$.[4]

We then eliminate gene g_i if its standardized (Z-score) expression values for all the patients falls within the range -1 to $+1$ (reflecting the first standard deviation) – *i.e.*, if $z_i^j \in (-1,\ +1)$ for all j. In METABRIC data, this filtering procedure eliminates 27,131 of the original 49,576 probes, leaving only 22,445 probes – *i.e.*, a $\sim 54.7\%$ reduction in features. The filtering process is unsupervised (*i.e.*, does not involve any labels) and is motivated by the assumption that any gene whose expressions does not change much across multiple patients, is unlikely to be directly related to the disease, while the genes that contribute to a specific cancer strain typically have significant variations in their expression levels across patients. While this filtering procedure is unsupervised, we anticipate that it will retain the probes that have the most prognostic ability. This can be confirmed by examining Table 1 from Cheng *et al.* (2013), which lists the top 100 probes (*i.e.*, with the highest concordance) in the METABRIC data, when each probe's expression value is used as a risk score. We found that our filtering process retains all the 100 probes listed in that table.

[4] Two notes: First, this standardization is done prior to dividing the data into train and validation sets. Second, using z-scores based only on a single gene would not be able to identify which genes did not vary much, as (after this transform) all genes would vary the same amount.

2.4 Dimensionality Reduction Methods for Feature Derivation

In this section we discuss the feature derivation methods employed in this study. Subsection 2.4.1 quickly describes a supervised dimensionality reduction procedure that extends the standard principal component analysis; then Subsection 2.4.2 presents our proposed approach based on topic modeling.

2.4.1 Feature Derivation via Principle Component Analysis (PCA)

There have been many methods proposed for survival prediction using gene expression data, such as hierarchical clustering, univariate gene selection, supervised PCA, penalized Cox regression and tree-based ensemble methods (Van Wieringen *et al.*, 2009). Some of these techniques first apply a procedure to reduce the dimensionality of the data, based on feature selection, feature extraction or a combination both, while others, such as random survival forests (Ishwaran *et al.*, 2008) and L1-penalized Cox (Goeman, 2010), include internal feature selection. The supervised principal component analysis (SuperPC) (Bair and Tibshirani, 2004) method first calculates the univariate Cox score statistic of each individual gene against the survival time, then retains just the subset of genes whose score exceeds a threshold, determined by internal cross-validation. It then computes PCA on the dataset containing only those selected genes, then projects each subject onto the first one (or two) components[5].

The main disadvantage of the SuperPC algorithm is that the individual genes selected from univariate selection might not perform the best in a multivariate (final) model, perhaps because many of these top-ranked genes may be highly correlated with one another – *i.e.*, it would be better having a more "diverse" set of genes (Van Wieringen *et al.*, 2009; Ding and Peng, 2005).

Instead, we propose an algorithm that initially applies PCA (on the standardized data z_i^j, see Section 2.3) to transform the data from the raw feature space into a different space, and then selects the top components based

[5]SuperPC does not specify why it chose to use these numbers of components.

on univariate Cox regression; we call this SuperPC+. Note this SuperPC+ is computationally efficient, as it is based on PCA, which is efficient: Even though microarray data is high dimensional ($p \gg n$, where p is the number of genes and n is the number of instances), the rank of the GE matrix will be (at most) $\min\{p, n\} = n$. Therefore, PCA can be performed without many computational restraints on the whole gene expression dataset, as PCA time complexity is $O(n^3)$. After performing PCA on the GE dataset, we can then identify the most important principal components by computing a Cox score statistic for the univariate association between each principal component and the survival time. In our experiments, we select the threshold τ for the p-value of the Cox score by internal cross-validation (wrt concordance), and retained all PCs having a p-value lower than τ – finding $\tau = 0.0005$ for the METABRIC dataset and $\tau = 0.05$ for BRCA.

2.4.2 Feature Derivation for Survival Prediction using discretized Latent Dirichlet Allocation (dLDA)

Latent Dirichlet Allocation (LDA) is a widely used generative model (Blei *et al.*, 2003), with many successful applications in natural language (NL) processing. LDA views each document as a distribution over multiple topics (document-topic distribution), where each topic is a distribution over a set of words (topic-word distribution) – that is, LDA assumes that each word in a document is generated by first sampling a topic from the document's document-topic distribution and then sampling a word from the selected topic's topic-word distribution. Given the set of topics, this means each document can be viewed as its distribution over topics, which is very low dimensional. The LDA learning process first identifies the latent topics – that is, the topic-word distributions corresponding to each latent topic – based on the words that frequently co-occur across multiple documents. This process depends on the distributional form of the document-topics and topic-word distributions – which here are typically Dirichlet. It also needs prior parameters for these distributions (typically initialized with a uniform prior), and also the number of latent topics, K. LDA then runs an unsupervised process (that does not

depend on the labels of the documents), to find the inherent structure present in the data – *i.e.*, a model (topic-word distributions for each of the k topics, and document-topic distributions for each document) that maximizes the likelihood of the training data.

LDA and other topic modeling techniques, such as the probabilistic Latent Semantic Analysis (pLSA), have been previously applied to microarray data for the gene classification task (Bicego *et al.*, 2012). Moreover, a probabilistic graphical model, which was inspired by LDA, has been proposed specifically for microarray data: Latent Process Decomposition (LPD) (Rogers *et al.*, 2005) has produced classification results comparable to the state-of-the-art. Bicego *et al.* (2012) reports that the LPD approach for gene classification using gene expression data to be very effective and also produces interpretable features as well.

This motivated us to apply a topic modeling approach to gene expression data, for the survival prediction task. While several projects used topic modeling techniques to categorize genes, very few have applied this technology to predict a patient's survival times (using gene expression data). Dawson and Kendziorski (2012) proposed a survival supervised LDA model, called survLDA, as an extension of supervised LDA (McAuliffe and Blei, 2008). survLDA uses a Cox model (Cox, 1972) to model the response variable (survival time) instead of the generalized linear model (McCullagh and Nelder, 1989) proposed by supervised LDA (McAuliffe and Blei, 2008). But Dawson and Kendziorski (2012) reported empirically that the topics learned from survLDA and from the general (unsupervised) LDA model to be similar.

Our work presents an analogue to the NL topic modeling that can be applied to our cohort of patients with the gene expression data, where a patient in the cohort corresponds to a document and the probes in the expression data correspond to the words that form the document. This task requires some modifications to the standard LDA model to be able to deal with the *real-valued* gene expression values: The standard NL topic models assume the observations are frequencies of words, which are non-negative integers that generally follow the Zipf distribution (Powers, 1998). By contrast, microarray

gene expression values are arbitrary real values, believed to follow a skewed Gaussian distribution (Wolfinger *et al.*, 2001). Some suggest dealing with this difference by shifting and scaling (Bicego *et al.*, 2012). LPD takes the novel approach of modeling the Gaussian distribution by estimating the mean and the variance as an additional set of parameters for each probe (Rogers *et al.*, 2005).

We follow an alternative approach, of applying an appropriate preprocessing to the gene expression, so the resulting values, basically, approximate a Zipf distribution. This involves a simple discretization process of gene expression values (described below), that adheres to the biological intuition behind the microarray data; we refer to our model as *dLDA* and the discretized gene expression values as *dGEVs*. We present empirical evidence (in Section 3.1) that our method performs better than the LPD technique for survival prediction.

As mentioned above, the LDA topic model decomposes each NL document into a Dirichlet mixture over the latent topics. This allows us to represent each document as this lower dimensional representation of the original text. We present a similar model that can be applied to our cohort of patients, each described using real-valued gene expression data: here, we represents each patient [document] as a mixture over "(cancer) strains" [topics], where each strain is a mixture over gene expression values [words]. In each strain, some genes will have "extreme" values – either much higher (or much lower) expression levels compared to the other genes in the data. This partially matches standard NL topic modeling, where some words have a high occurrence in a particular topic, but differs in that for gene expression, both over-expression and under-expression are rare and important. Moreover, since dLDA gives a soft clustering, each patient can still be modeled as a mixture of multiple cancer strains, showing promising compatibility to recent knowledge on cancer subclones (Deshwar *et al.*, 2015).

Here we propose our generative process for modeling the gene expression values – or actually, the discretized gene expression values (dGEV). (The next subsection explains how to transform the gene expression values into these dGEV's.) Figure 2.2 shows the generative process for generating the vector of

`Generate-Instance-dLDA(`$\alpha,\ \beta,\ M$`)`
% $\alpha \in \Re^{>0}$
% $\beta \in \Re^{K \times N}$ where K = #latent strains, N = #genes
% β_{ij} = probability of drawing gene j, from strain i.
% M is equivalent to word count of a document.
% Returns N-tuple of (discretized) gene expression values

1. Initialize `dGEV` := $\mathbf{0}$ (of dimension N)
2. Draw $\theta \sim Dirichlet(\alpha I_K)$
3. For $n = 1..M$
   (a) Choose a strain $z_n \sim$ Multinomial(θ)
   (b) Choose a probe $p_n \sim$ Multinomial($\beta[z_n, :]$)
   (c) Increment: `dGEV`$[p_n]$++
4. Return `dGEV`

Figure 2.2: Algorithm for producing an instance, from a specified dLDA model

gene expression values corresponding to a patient following the LDA model, given the distribution of strains (parameterized by α), the distributions over gene expression values corresponding to each strain (parameterized by β), of size M (corresponds to the number of words in a document).

Deriving discretized Gene Expression Values (dGEVs)

As described in Section 2.3, we first standardize all expression values in the data $\{z_i^j\}$, giving non-trivial values only to the probes that have high variance across patients – *i.e.*, we only consider probes with a non-trivial range. These normalized gene expression values can be arbitrary real values; however, LDA is designed to work with "counts" – that is, non-negative integers – and in particular, with word counts in documents where, in any given document, most words appear 0 times, then many fewer words appear once, then yet fewer words appear twice, etc. We therefore need a method for taking the real values z_i^j, and converting them to non-negative integers.

We consider two ways to do this. First, Technique A "discretizes" each non-trivial z_i^j standardized gene expression value by binning them into equal

`Learn-dLDA`(α, K, GE)
% $\alpha \in \Re^{>0}$
% K = *#latent strains*
% $GE \in \Re^{M \times N}$ – *Gene Expression,* N = *#genes,* M = *#patients*
% *Returns estimated* $\hat{\beta}$

1. % *Initialize*:

   $$\beta_{i,j} = \frac{1}{N} + \frac{U[0,1]}{N^2} \quad \forall_{i,j}; \quad \beta \in \Re^{K \times N}$$

2. % *Discretize GE*:

   `dGEVs = discretizeGE( GE )`

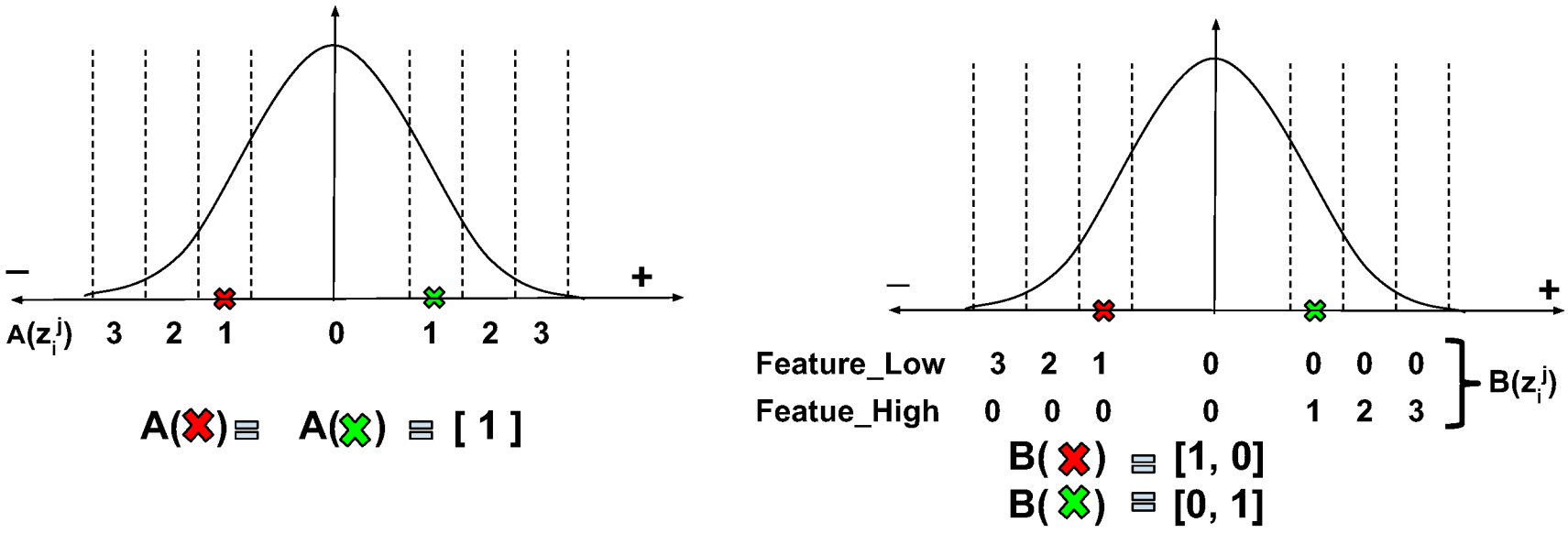
3. % *Blei* et al. *(2003)*:

   $\hat{\beta} =$ `LDA(` αI_K, β, `dGEVs )`

4. Return $\hat{\beta}$

Figure 2.3: Algorithm for learning a dLDA model

distance bins away from the first standard deviation. We bin all the non-trivial standardized gene expression values of each gene ($\{z_i^1, \ldots, z_i^j\}$, j–#patient) separately into 20 bins. Positives and negatives are binned separately into bins of size 10 each. The boundaries of the 10 equally spaced bins of the positives are computed given the minimum and maximum of the set of positive standardized gene expression values (of the gene) $\{z_i | z_i^j > 0\}$, similarly negatives are also binned into 10 bins. In Figure 2.4(left), the x-axis corresponds to the discretized values. Each z_i^j that fall within the first standard deviation is discretized as zero, the ones that fall in the first bin away from the first standard deviation are discretized as 1, and so on. However, this technique does not distinguish between overexpression and underexpression – *i.e.*, both 1.2 and -1.7 are binned into the same value $A(1.2) = A(-1.7) = 1$; See Figure 2.4(left). Alternatively the encoding Technique B, shown in Figure 2.4(right), distinguishes underexpression versus overexpression, by using two features (each a non-negative integer); where one encodes the negatives and the other encodes positives separately. Here, $B(1.2) = [0, 1]$ and $B(-1.7) = [1, 0]$.



Technique A *Technique B*

Figure 2.4: Gene expression encoding schemes to compute the discretized Gene Expression Values (dGEV). (left) Technique A discretizes the GEVs into a single count feature and (right) Technique B discretizes the GEVs into two count features representing overexpression and underexpression.

Note that both Techniques A and B match our requirements (non-negative integers) and basic conventions (many more 0s than 1s, etc). As Technique A uses just a single feature for each probe, while Technique B uses 2, clearly Technique B requires twice as many features. However, Technique B distinguishes the types of extreme situations (up vs down regulation), but Technique A does not.

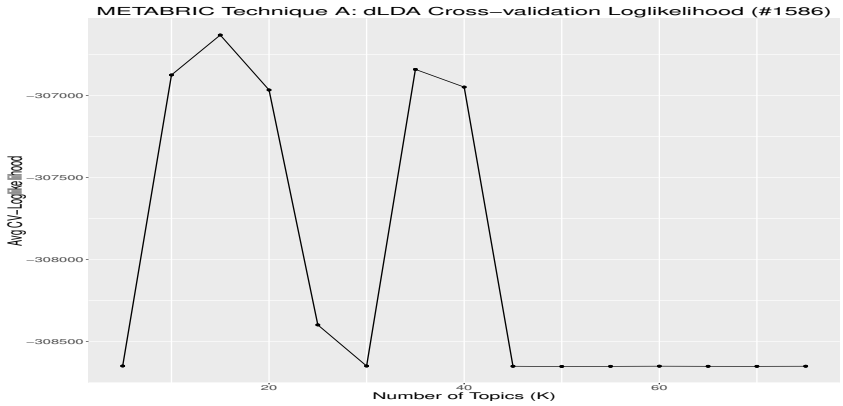
To train a LDA model with the new discretized gene expression values, we use the lda-c implementation. [6] As shown in Figure 2.3, the LDA model requires two input parameters from the user (and the GE dataset): (1) the Dirichlet prior for the patient-strains (document-topics) distribution α, (2) the number of latent strains K. Based on our experiments, we set a symmetric Dirichlet prior for the patient-strains distribution ($\alpha = 0.1$, note $\alpha 1_K = [0.1, \ldots, 0.1] \in \Re^K$)[7], and the strain-genes probabilities are initialized uniformly with some random values – have $\forall i, j,\ \beta[i, j] = \frac{1}{N} + \delta$ where $\delta \sim \frac{U[0,1]}{N^2}$. The next section below describes how we determined the appropriate number of latent strains K.

Determining the number of strains (K) for the LDA model

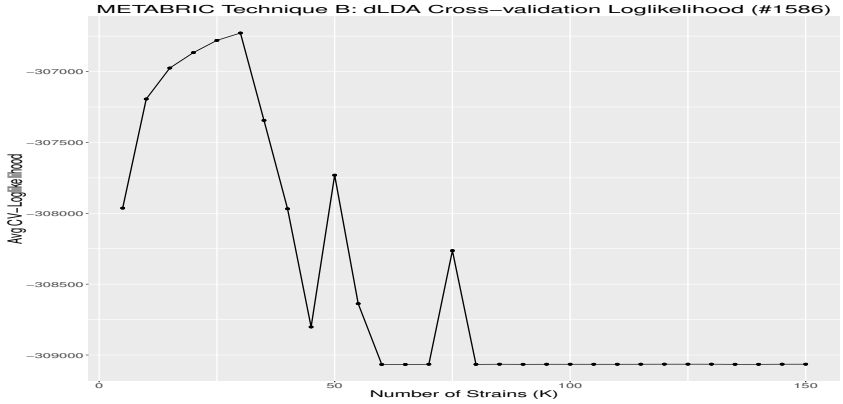
To find the appropriate number of strains (analogous to NL "topics"), we ran internal cross-validations with a range of probable values for K with each of the encoding schemes (A and B). Here, we searched over the range $K \in \{5,\ 10,\ 15, \ldots, 150\}$. For each technique {A, B} and each of the 30 values of K, we first computed the dLDA models over the training set, and then used these as covariates to learn a Cox model (Cox, 1972). We did this in-fold – using 4/5 of the training set to learn the dLDA strains and the Cox model, which we evaluated by computing the concordance (based on this learned model) on the remaining 1/5. Figure 2.5 shows the log-likelihood average over the 5 folds, with different number of strains and both feature encoding techniques A and B for METABRIC, and technique B for BRCA. We need to answer two questions to determine the best model: (1) which is the best discretization technique,

[6] Available from http://www.cs.columbia.edu/∼blei/lda-c/

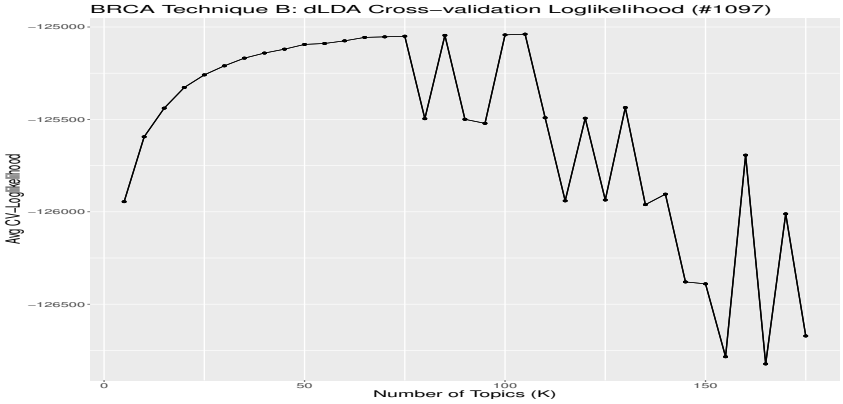
[7] We considered the base $\alpha_0 \in \{0.01,\ 0.1,\ 0.5,\ 1.0\}$, but found that the prior did not make much difference, since we allowed the model to estimate the prior internally.



METABRIC–Technique A



METABRIC–Technique B



BRCA–Technique B

Figure 2.5: Log-likelihood scores from the (internal) cross-validation with multiple values for K (number of strains) in METABRIC and BRCA. (top) METABRIC CV-likelihood of encoding Technique A; (middle) CV-likelihood of encoding Technique B: (bottom) BRCA CV-likelihood of encoding Technique B.

and (2) what is the appropriate K for that technique. To answer the first question, our strategy picked the encoding technique that gave the highest cross-validation concordance from all the combinations (30×2). Secondly, once decide on a encoding scheme, we identified the candidate set of K's whose cross-validation likelihood scores are within the first standard deviation of the largest (cross-validation) likelihood. From the candidates, we select the K that essentially gives the highest concordance – breaking ties by preferring smaller values for K when performances are almost equal.

2.5 Survival Prediction Algorithms Used

In this section we briefly describe two survival prediction algorithms. Subsection 2.5.1 introduces the classic Cox model (Cox, 1972) and Subsection 2.5.2 details a novel survival prediction model called Multi-Task Logistic Regression (Yu *et al.*, 2011).

2.5.1 Cox Proportional Hazard (PH) Model (Cox, 1972)

Cox regression model's the hazard function over time for an individual described by x the product of two components:

$$h_\beta(\, t \,|\, x \,) \quad = \quad h_0(t) \times \exp(x^T \beta) \tag{2.1}$$

where the baseline hazard $h_0(t)$ is independent of the covariates x and the covariates are (independently) multiplicatively related to the hazard. The above formulation simplifies modeling of the hazard function by limiting the contribution of the "time" variable t to the baseline hazard $h_0(\cdot)$, which means the hazard ratio (HR) between two patients

$$\mathrm{HR}(x_1,\, x_2) \;=\; \frac{h(t|x_1)}{h(t|x_2)} \;=\; \frac{\exp(x_1^T \beta)}{\exp(x_2^T \beta)} \;=\; \exp((x_1 - x_2)^T \beta)$$

does not depend on time and is linear (proportional) in the exponent. To estimate the coefficients of the model, Cox (1972) proposed a partial likelihood technique that eliminates the need to estimate the baseline hazard. This procedure allows the Cox proportional hazards model to be semi-parametric by

only using the survival times to rank the patients (Steck *et al.*, 2008). We can compute partial likelihood with all patients – both censored and uncensored:

$$L_c(\beta) \quad = \quad \prod_{i=1}^{N} \left(\frac{\exp({x_i}^T \beta)}{\sum_{k \in \mathcal{R}(y_i)} \exp({x_k}^T \beta)} \right)^{\delta_i} \tag{2.2}$$

- $\mathcal{R}(r_j)$ is the risk set at time y_j, which are the indices of individuals who are alive and not censored before time y_j
- $[x_i, r_i, \delta_i]$ describes the i^{th} subject, where
  x_i = vector of covariates
  r_i = response (survival or censor time)
  δ_i = censor bit
- N – total number of patients in the cohort
- β – coefficients (to be learned)

Note that only the uncensored likelihoods contribute, since for censored instances $\delta_i = 0$. Therefore the censored observations are only utilized in the denominator when summed over the instances in a risk set. In essence the partial likelihood only uses the patient's death times to rank them in the ascending order to find the risk sets and does not use the exact times explicitly (Steck *et al.*, 2008). Hence, the coefficients estimated by maximizing the partial likelihood depends only on the ordering of the patient's death times and the covariates, allowing for an implicit optimization for good concordance of the risk score. An in-depth study on the Cox proportional hazard model has revealed that the partial likelihood proposed by Cox (1972) is approximately equivalent to optimizing concordance (Steck *et al.*, 2008).

There are several extensions of the Cox proportional hazards model: some extend the initial model estimating the baseline hazard and others are based on the regularization methods imposed on the coefficients (β). Generally, regularization based on LASSO, ridge penalty or the elastic-net regularization (which allows both L1 and L2 penalties) are adopted to reduce overfitting. In our work, we use the glmnet R package (Simon *et al.*, 2011) with ridge

penalty (by setting $\alpha = 0$ in the glmnet function: from here onward referred to as RCox). We selected ridge penalty based on the internal cross validation concordance results where models with ridge penalty were better than those having no regularization (LASSO, elastic-net).

2.5.2 Patient Specific Survival Prediction via MTLR model

The Multi-Task Logistic Regression (MTLR) system (Yu *et al.*, 2011) learns a model (from survival data) that, given a description of a patient $\mathbf{x}$, produces a survival curve, which computes $Pr(D \geq t \,|\, \mathbf{x})$ vs t for $t \geq 0$. This survival curve is similar to a Kaplan–Meier curve, but incorporates all of the patient specific features $\mathbf{x}$. In more detail: MTLR first identifies m time points and then learns an extension of a logistic regression function, parameterized by $[\theta_i, b_i]$, for each of the m time points (a different such function for each of the times t_i), where the random variable D is the time of death (here of the patient described by x):

$$Pr_{\Theta}(\, D \in [t_{i-1}, t_i) \,|\, \mathbf{x}\,) \quad \propto \quad \exp\left(\sum_{j=i}^{m} ({\theta_j}^T \mathbf{x} \; + \; b_j) \right) \tag{2.3}$$

The MTLR model then combines these PMFs (probability mass functions) into a CMF (cumulative mass function) in a way that guarantees the resulting curve is monotonically decreasing for each patient $\mathbf{x}$, from the probability value of 1 at $t_0 = 0$ – *i.e.*, $Pr_{\theta}(\, D \geq 0 \,|\, \mathbf{x}\,) \; = \; 1$ – down to smaller values as the time t increases. The MTLR algorithm learns different coefficients for each time point – hence the parameters $\Theta = \{[\theta_i, b_i]\}$ is a matrix of size $m \times (r + 1)$, if there are r features. This requires encoding a patient's survival time d as a binary vector (of classification labels) $y(d) \; = \; [y_1(d), y_2(d), \ldots, y_m(d)]$, where each $y_i(d) \in \{0, 1\}$ encodes that patient's survival status at each time step t_i: $y_i(d) = 0$ (no death yet) for all i with $t_i < d$ and $y_i(d) = 1$ (death) for all $t_i \geq d$.

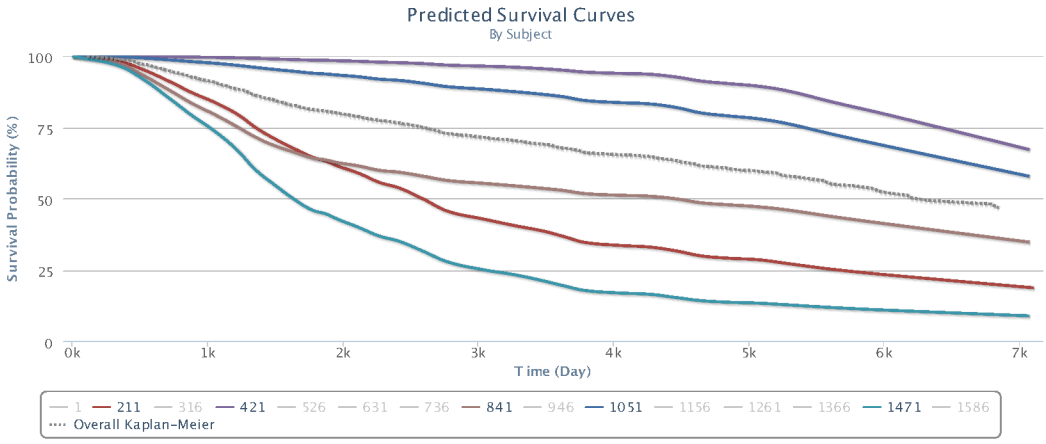


Figure 2.6: Patient specific survival functions from MTLR on the METABRIC data

The learning system attempts to optimize

$$\min_{\Theta} \frac{C}{2} \sum_{j=1}^{m} \|\theta_j\|^2 - \sum_{i=1}^{n} \left[\sum_{j=1}^{m} y_j(r_i)(\theta^T \mathbf{x}_i + b_j) - \log \sum_{k=0}^{m} \exp(f_\Theta(\mathbf{x}_i, k)) \right] \tag{2.4}$$

$$f_\Theta(\mathbf{x}_i, k) = \sum_{l=(k+1)}^{m} (\theta_l^T \mathbf{x}_i + b_l) \qquad \text{for } 0 \leq k \leq m$$

for uncensored data, with an obvious extension of Equation 2.4 to deal with censored instances. This overall equation includes a L2 regularization term to reduce the risk of overfitting. The MTLR parameter m (time points) is set to the square root of the number of instances in all our experiments.

MTLR differs from the Cox model as: (1) MTLR produces a survival function (See Figure 2.6), rather than just a risk score; and (2) MTLR does not make the proportional hazards assumption – *i.e.*, it allows effects of covariates to change with time. We will use the (negative of) the mean of the patient's specific predicted survival distribution as her risk score. Yu *et al.* (2011) presents more detailed explanations on model formulation, parameter learning (Θ), and the prediction task.

2.6 Evaluation Criteria

This section presents the evaluation criteria used to assess the various (learned) survival models. Following Steyerberg *et al.* (2010), we provide a discriminatory and a calibration measure.

2.6.1 Discriminatory: Concordance Index (CI)

This "CI" evaluation applies to any model that assigns a real number – a "risk score" – to each instance $f(\cdot)$. It considers all pairs of "comparable" instances, and determines which is predicted (by the model $f(\cdot)$) to die first, and also who actually died first. CI is the percentage (probability) of these pairs of instances whose actual pair-wise survival ordering, matches the predicted ordering, wrt. risk function $f(\cdot)$:

$$\text{CI}(f) \quad = \quad \frac{1}{|\Psi|} \sum_{(i,j)\in\Psi} I[\,(f(x_i) > f(x_j))\,] \tag{2.5}$$

where $I[\phi]$ is the indicator function, which is 1 if the proposition ϕ is true, and 0 otherwise. A pair of patients is "comparable" if we can determine which died first – *i.e.*, if both are uncensored, or when one patient censored after the observed death time of the other; this corresponds to the set Ψ. This CI(f) score is a real value between 0 to 1, where 1 means all comparable pairs are predicted correctly. CI can be viewed as a general form of the Mann–Whitne–Wilcoxon statistic and is similar to area under the ROC (AUC) of classification problems (Steck *et al.*, 2008).

2.6.2 D-Calibration

Calibration measures the deviation between the observed and the predicted events (death). While this is complicated for a risk score (like the basic Cox function), it can be computed for a survival distribution, like MTLR. In general, this involves computing the difference between the predicted versus observed probabilities in various subgroups – eg, if the predicted probability of surviving at least t =2576 days is 0.75 then we expect to observe around 75% of the patients to be alive at this time t.

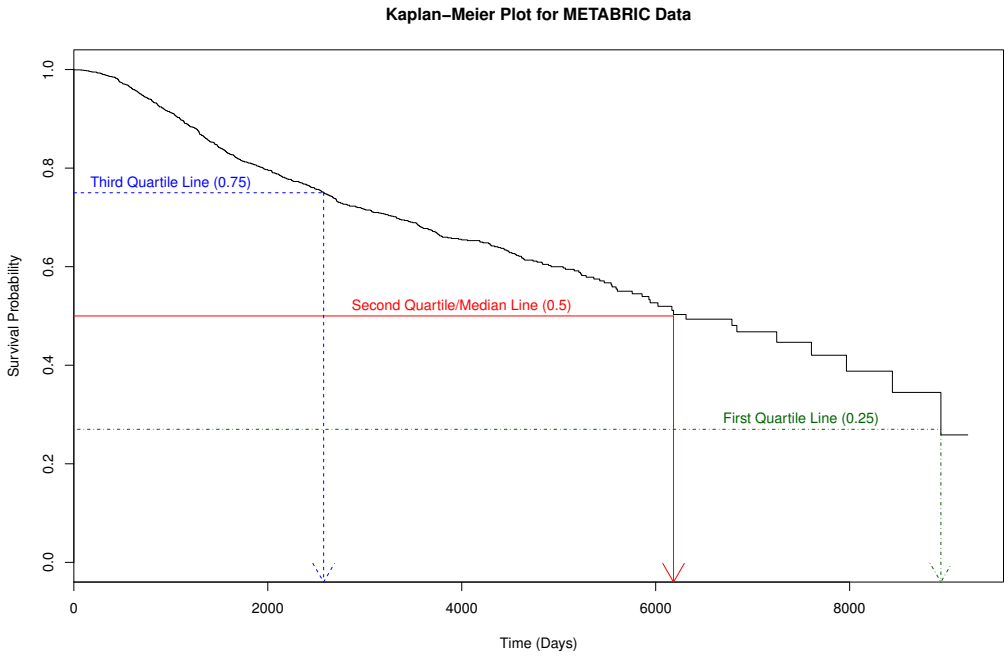


Figure 2.7: Kaplan–Meier survival function from METABRIC (training) data.

We consider a novel version, called Distribution-calibration. To motivate this, consider a standard Kaplan-Meier (KM) (Altman, 1990) Figure 2.7, which computes the set of points $(t,\ \mathrm{KM}(\,t\,))$ – *i.e.*, it predicts that the $\mathrm{KM}(\,t\,)$ fraction of patients will be alive at time t. Hence, the point (6184 days, 0.50) means the median survival time of the cohort is 6184 days (See Figure 2.7 (solid line)). We will use $\mathrm{KM}^{-1}(\,p\,)$ to be the time associated with the probability p – technically, $\mathrm{KM}^{-1}(\,p\,)$ is the earliest time when the KM curve hit p; hence $\mathrm{KM}^{-1}(\,0.5\,) = 6184$ days. If this plot is D-calibrated, then around 50% of the patients (from a hold-out set, not used to produce the KM curve) will be alive at this median time. So letting d_i be the time when the i^{th} patient died, consider the n values of $\{\,\mathrm{KM}(\,d_i\,)\,\}$. Here, we expect $\mathrm{KM}(\,d_i\,) > 0.5$ for $1/2$ of the patients. Similarly, as the curve includes (2576 days, 0.75) and (8941 days, 0.25), then we expect 75% to be alive at 2576 days, and 25% at 8941 days. (See Figure 2.7) Collectively, this means we expect 25% of the patients to die between $\mathrm{KM}^{-1}(\,1\,) = 0$ days and $\mathrm{KM}^{-1}(\,0.75\,) = 2576$ days, and another 25% between $\mathrm{KM}^{-1}(\,0.25\,)$ and $\mathrm{KM}^{-1}(\,0.5\,)$, etc. These are the predictions; we can also check, to see how many people actually died in each interval: in

the first quartile (between 0 and 2576 days), in the second (between 2576 and 6184 days), in the third (between 6184 and 8941 days), and the fourth (after 8941 days). If the KM plot is "correct" – *i.e.*, is D-calibrated – then we expect 1/4 of the patients will die in each of these 4 intervals. The argument above means we expect 1/4 of the $\{\,\mathrm{KM}(\,d_i\,)\,\}$ values to be in the interval [0, 0.25], and another quarter to be in [0.25, 0.5], etc. Stated more precisely,

$$\text{the values of } \{\,\mathrm{KM}(\,d_i\,)\,\} \text{ are uniform.} \tag{2.6}$$

A single KM curve is designed to represent a cohort of many patients. Our MTLR, however, computes a different survival curve for each patient – call it $Pr_i(\,\cdot\,) = Pr_\Theta(\,\cdot\,|\,\mathbf{x}_i\,)$ (from Equation 2.3). But the same ideas still apply: Each of these patients has a median predicted survival time – the time $Pr_i^{-1}(0.5)$ where its $Pr_i(\,\cdot\,)$ curve crosses 0.50. By the same argument suggested above, we expect (for a good model Θ) that 1/2 of patients will die before their respective median survival time – $d_i \leq Pr_i^{-1}(0.5)$; that is, $|\{i \;:\; Pr_i(d_i) \leq 0.5\}| \approx n/2$. Continuing the arguments from above, we therefore expect the obvious analogue to Equation 2.6:

$$\text{the values of } \{Pr_i(\,d_i\,)\} \text{ are uniform.} \tag{2.7}$$

We can now test whether a plot is D-calibrated by using the Hosmer-Lemeshow (HL) (Hosmer Jr *et al.*, 2013) goodness-of-fit test, which compares the difference between the predicted and observed events in the event subgroups:

$$\mathrm{HL}(\begin{bmatrix} [N_1,\ P_1,\ E_1] \\ \cdots \\ [N_g,\ P_g,\ E_g] \end{bmatrix}) \quad = \quad \sum_{g=1}^{G} \frac{(E_g - P_g)^2}{N_g \pi_g (1-\pi_g)} \tag{2.8}$$

where G is the number of subgroups (here 4), where the g^{th} subgroup has $N_g \in \mathbb{Z}^+$ events, with the empirical number of events $E_g \in \mathbb{Z}^+$ (which here is N/g, if there are $N = \sum_g N_g$ total patients), the corresponding predicted number of events $P_g \in \mathbb{Z}^{\geq 0}$, and $\pi_g = \frac{N_g}{N}$ (which here is $\frac{1}{G}$) is the proportion of the g^{th} subgroup. Under the null hypothesis (Equation 2.7), this HL statistics follows a Chi-Square distribution with $G-2$ degrees of freedom. If the predicted and empirical event rates are similar for the subgroups, the test statistic will fail to

reject the null hypothesis, providing evidence that the model's predictions are well D-calibrated (*i.e.*, we should have large p-values from the test statistic to accept the Null hypothesis).

Notes: (1) This evaluation criterion only applies to models that produce survival distributions, which means it directly applies to the MTLR models. For the Cox and RCox models, we used the Kalbfleisch-Prentice baseline hazard estimator (Kalbfleisch and Prentice, 2011) to produce personalized survival curves. (2) Rather than use quantiles, we mapped the $Pr_i(\, d_i \,)$ probabilities into 20 bins: [0, 0.05); [0.05, 0.1), ..., [0.95, 1.0]. (3) This analysis deals only with uncensored data; the Appendix B discusses how to extend this to deal with the censored instances.

Chapter 3

Evaluation

This chapter presents empirical results that show that our proposed dLDA model works effectively, using multiple learning algorithms and evaluation criteria. The earlier Figure 2.1 shows the steps involved in pre-processing, dimensionality reduction, our feature derivation procedures, and the possible model combinations (data and algorithm) that can be assessed. Section 2.3 describes preprocessing steps that eliminate genes that do not vary significantly with the standardization procedure and Section 2.4 describes two feature derivation methods: (1) SuperPC+ principle components and (2) the dLDA strains. Section 2.4.1 shows how we select the principal components from the microarray data to be used as covariates for the survival prediction task. We train multiple Cox models with each individual principal component as a univariate and select the components that exceeds the p-value threshold (τ) , which we determine from internal cross-validation on the training data.

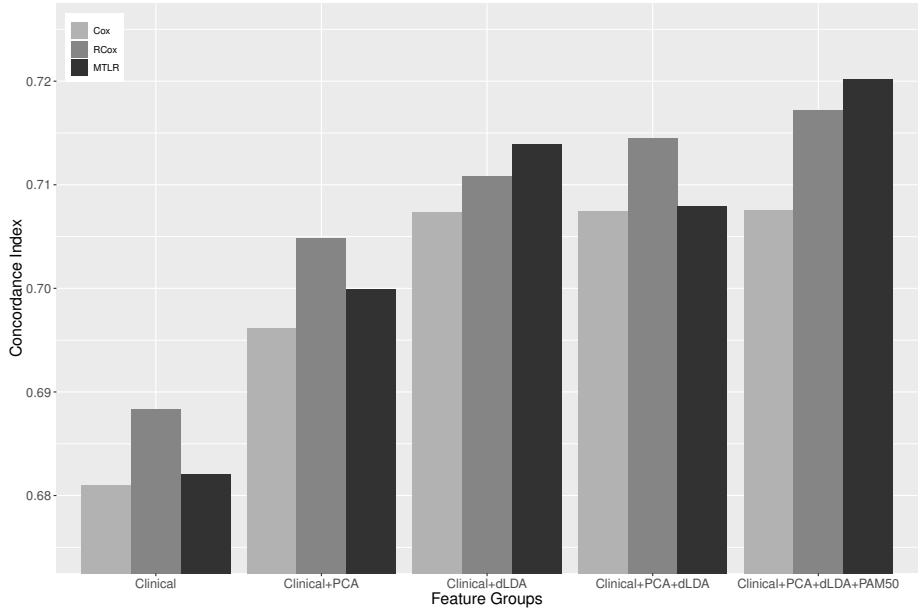
Finally this process selected 6 principle components (τ =5E-04) in METABRIC, and 59 in BRCA (τ =5E-2). Similarly, to determine the optimal number of strains for the dLDA model, we run internal cross-validation on the training set with a range of potential values for the number of strains; see Section 2.4.2. We selected the appropriate number of strains by picking the model that maximizes the concordance in the internal cross-validation. Based on our experiments we found that the discretization technique B, along with K =30 strains, produced the best dLDA algorithm for survival prediction in METABRIC; after deciding on technique B we followed the same approach on the BRCA

dataset to determine the number of strains and found K =50 strains were best.

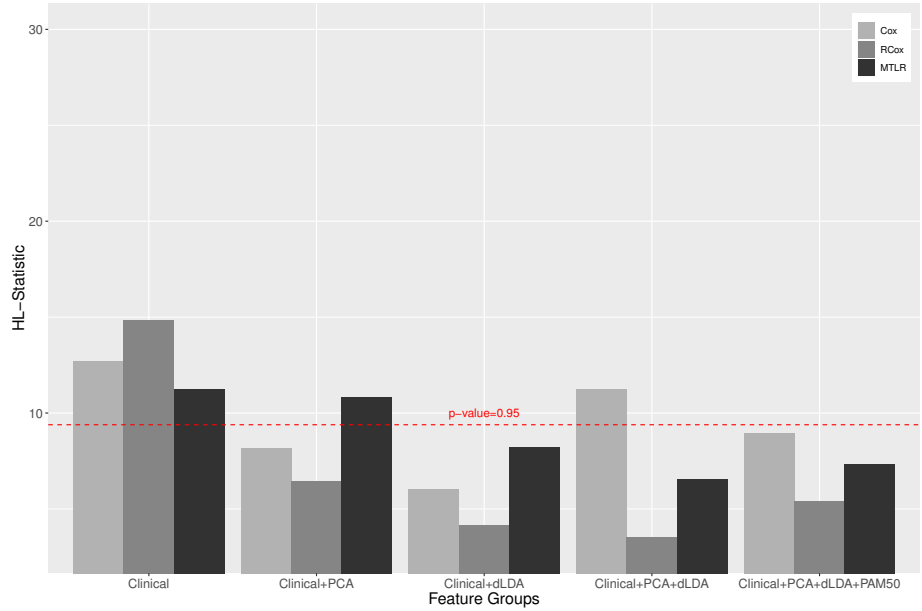
Section 3.1 below presents our empirical results from three different learning algorithms along with our two feature derivation methods. We evaluate these 3×2 learned models based on the two evaluation criteria: (1) Concordance Index, and (2) D-calibration. Section 3.2 discusses the reproduced results from the BCC Dream Challenge, Section 3.3 details the performance of the LPD procedure for the survival prediction task and finally Section 3.4 reports the predictive performance of the PAM50 genes in survival prediction.

3.1 Experimental Results

In this section, we present our empirical results from our experiments with various combinations of feature groups, learning models and evaluation criteria. As shown in the Figure 2.1, we considered various different combinations of features from the three main groups: (1) clinical features (includes PAM50 classification (Parker *et al.*, 2009) in METABRIC), (2) SuperPC+ principle components, and (3) the dLDA strains; we also considered three different learning algorithms: (a) Cox, (b) RCox, and (c) MTLR. Our goal in these experiments is to empirically evaluate the performance when genomic features are added to the clinical features to the survival models. Given this goal, we evaluate the performance using different (genomic) feature derivation methods by comparing their performance to a baseline model that only uses the clinical features with the Cox (1972) model. As we want this baseline to include only clinical information without any genomic information, it does not include the PAM50 intrinsic subtypes (Parker *et al.*, 2009); we therefore remove these features from the METABRIC clinical data. The other combinations include these clinical features as well as various different genomic features; each is trained using one of the three aforementioned survival prediction algorithms.



METABRIC Testset Concordance Index Results (higher better)



METABRIC Testset D-calibration Results (lower better)
HL-Statistic for p-value=0.95 given in dashed line

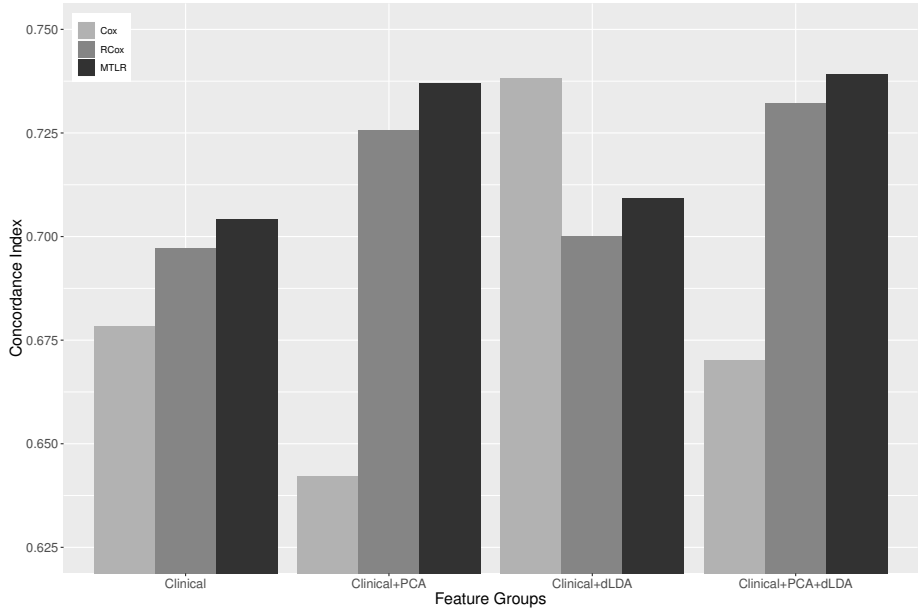
Figure 3.1: Held-out test data (n=395) concordance and D-calibration plots from METABRIC.

Table 3.1: Concordance and D-calibration results on the held-out test data (n=395) from multiple model combinations in METABRIC.

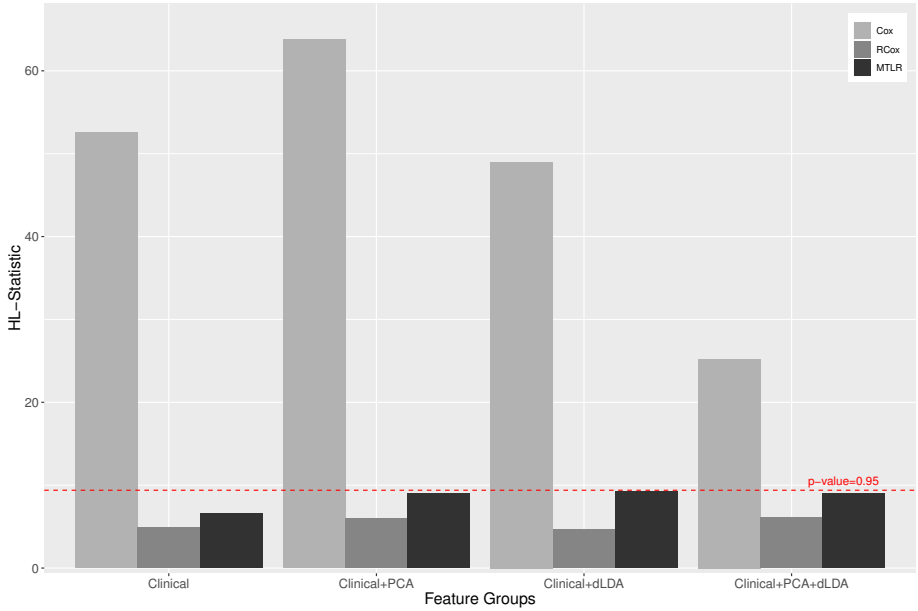
ID	Feature Groups				Learning Algorithm	Concordance	D-calibration	
	Clinical	PCA	dLDA	PAM50	Cox | RCox | MTLR		(HL-Statistic	p-value)
A (Baseline)	+	−	−	−	Cox	0.6810	12.6765	0.8104
B	+	−	−	−	RCox	0.6883	14.8143	0.6747
C	+	−	−	−	MTLR	0.6820	11.2300	0.8843
D	+	+	−	−	Cox	0.6961	8.1723	0.9760
E	+	+	−	−	RCox	0.7048	6.4499	0.9940
F	+	+	−	−	MTLR	0.6999	10.8421	0.9009
G	+	−	+	−	Cox	0.7073	6.0066	0.9962
H	+	−	+	−	RCox	0.7108	4.1344	0.9997
I	+	−	+	−	MTLR	0.7139	8.2000	0.9755
J	+	+	+	−	Cox	0.7074	11.2333	0.8842
K	+	+	+	−	RCox	0.7145	3.4981	0.9999
L	+	+	+	−	MTLR	0.7079	6.5158	0.9936
M	+	+	+	+	Cox	0.7075	8.9496	0.9609
N	+	+	+	+	RCox	0.7172	5.3902	0.9981
O	+	+	+	+	MTLR	**0.7202**	7.3263	0.9871

Table 3.2: Concordance and D-calibration results on the held-out test data (n=217) from multiple model combinations in BRCA.

ID	Feature Groups			Learning Algorithm	Concordance	D-calibration	
	Clinical	PCA	dLDA	Cox | RCox | MTLR		(HL-Statistic	p-value)
A (Baseline)	+	−	−	Cox	0.6782	52.5289	3.1e-05
B	+	−	−	RCox	0.6971	4.8816	0.9990
C	+	−	−	MTLR	0.7042	6.6300	0.9929
D	+	+	−	Cox	0.6422	63.7983	4.9e-07
E	+	+	−	RCox	0.7256	6.0013	0.9962
F	+	+	−	MTLR	0.7370	8.9895	0.9600
G	+	−	+	Cox	0.7381	48.9904	0.0001
H	+	−	+	RCox	0.6999	4.6031	0.9993
I	+	−	+	MTLR	0.7092	9.2421	0.9539
J	+	+	+	Cox	0.6700	25.2241	0.1189
K	+	+	+	RCox	0.7320	6.0813	0.9959
L	+	+	+	MTLR	**0.7391**	9.0211	0.9593



BRCA Testset Concordance Index Results (higher better)



BRCA Testset D-calibration Results (lower better)
HL-Statistic for p-value=0.95 given in dashed line

Figure 3.2: Held-out test data (n=217) concordance and D-calibration plots from BRCA.

As described in Section 2.3, missing values of 13 features in METABRIC and 13 in BRCA are imputed using the mean (resp., mode) for numerical (resp., nominal) features. We also binarize the nominal clinical features. As an additional feature selection step, we remove the covariate "Site" from the METABRIC clinical covariates, based on our experimental results (on the training data) which shows that its inclusion leading to worse concordance. We only experimented with a selected set of model combinations rather than experimenting all possible model combinations, to answer our major queries. They are (wrt CI unless specified):

(i) does adding genomic features improve survival prediction?

(ii) which is the best feature combination for superior survival prediction?

(iii) which is better representation of the genomic features: dLDA or SuperPC+?

(iv) how do the learning algorithms compare in D-calibration?

(v) are we deriving redundant genomic features?

We claim (based on Table 3.1): (i) Comparing the baseline (A) to the other models, we immediately see that adding genomic features from gene expression (using any of the dimensionality reduction technique) leads to better predictive models, in both evaluation schemes (See Figure 3.1). (ii) We also see that the best model in METABRIC is the one that includes all of the types of features derived from the gene expression – here O. (See Figure 3.1 (left).) We also performed student's t-tests on random bootstrap samples from the test data to validate the significance of our results. When we compare this best model O against models E (best model using only PCA genomic features) and I (best model using only dLDA genomic features), we find statistically significant difference between them (respective pairwise p-value: 4.8e-16, 1e-3), showing that model O is significantly better than its closest counterparts. (iii) Table 3.1 shows that, if you only use a single genetic feature set, the

dLDA (latent) strains perform better than the principle components; moreover, a model using all the features performs yet better (See Figure 3.1 (left)). (iv) Table 3.1 shows that all the models from the METABRIC data seem to have good D-calibration (Figure 3.1 (right) show MTLR and RCox is p-values well above 0.95). (v) Adding PAM50 subtypes as features to the baseline model in METABRIC improves the held-out test concordance. Indeed, Table 3.1 shows that our new genomic features have led to prognostic models that are better than PAM50 subtypes, by both evaluation criteria. It also shows that the performance of models that include PAM50 are marginally better than similar models that do not, suggesting that the features added by these different representations of microarray data are not redundant. Moreover, Figure 3.1, shows that in all feature groups, both RCox and MTLR clearly outperform Cox in both evaluation criteria.

We then tested the first four claims on the BCRA dataset; see Table 3.2. (As BRCA did not have PAM50 features, we could not test claim (v).) (i) As before, we found that adding genetic information improves over the baseline (See Figure 3.2 (left)). (ii) We again found that the best model (in terms of Concordance) was the one that included all of the features; here L. When we performed the same significance test, however, we found that model L was not significantly better than the models E or G. (But more important, it was not inferior *i.e.*, this model L was a top performer.) (iii) Table 3.2 shows that (again) models trained with only the dLDA-features performed better than PCA-features; but that including both features was yet better. (iv) The D-calibration results in Table 3.2 show that the Cox model fails in all combinations (See Figure 3.2 (right)). All Cox models score worse p-values, significantly below 0.95, where both RCox and MTLR models score well above. That result clearly suggests that, if we want models with both good discrimination and calibration, we should *not* rely solely on the Cox model rather we should use either RCox (Simon *et al.*, 2011) or MTLR (Yu *et al.*, 2011). Also, Figure 3.2 shows that RCox and MTLR outperforms Cox in almost all feature groups in both evaluation criteria (similar to METABRIC data). These studies also support our claim that our model, learned by running MTLR on

all genomic features, gives very good concordance scores – statistically better than other options in one dataset and equally good in another dataset.

3.2 BCC Dream challenge (METABRIC)

The model that won this competition (Cheng *et al.*, 2013) (i) leveraged prior knowledge of cancer biology to form meta-genes and (ii) trained an ensemble of multiple learners, fueled by the continuous insights from the challenge competitors via open sharing of code and trained models. To compare our performances with the BCC challenge winning team, we reproduced their model based on the DreamBox7 package.We re-trained their learners on our training split of the METABRIC data and tested on the held-out validation set. The resulting ensemble model achieved a concordance index of 0.7293 on the test data. All models were trained and tested only using the disease-specific censoring information. The performance from the winning team in our test split can also be viewed as an optimistic score since they made design choices for their models using the whole METABRIC cohort (all n=1981 instances) which might lead to an over-fitted test performance on the METABRIC test data. These results show that the performance (concordance) of our MTLR model is comparable with the winning team's performance, even though all of our tuning was performed solely on the training (n=1586) data

Our results all deal with individual "base learners", rather than ensembles. An ensemble system combining our approach with the winning team's model might lead to better performance (*i.e.*, yet higher concordance). We did not pursue this as it was not necessarily our goal, which is: providing empirical evidence that applying the topic modeling approach to microarray data, can discover features with strong predictive power, without any prior knowledge about the specific disease.

3.3 Latent Process Decomposition (LPD)

Rogers *et al.* (2005) introduced LPD as a topic model adaptation for microarray data and Bicego *et al.* (2012) later claimed that it could to produce accu-

Table 3.3: Concordance results on the METABRIC held-out test set, using the LPD features

Models	Concordance
Cox + LPD (10 topics)	0.6915
RCox + LPD (10 topics)	0.6977
MTLR + LPD (10 topics)	0.6995

rate results for gene classification. Table 3.3, however, shows that this complex adaptation of the LDA model for microarray data does not perform well for the survival prediction task. We ran the LPD model on a range of number of latent processes to find the optimal number of processes for the METABRIC data. Compared to LDA (Blei *et al.*, 2003), LPD has large time and memory requirements. Tables 3.1 and 3.3 clearly suggest that dLDA derives better features from the gene expression data for the survival prediction task. Moreover as our dLDA directly uses the LDA model, it can utilize all available off-the-shelf implementations, across several technology platforms with efficient and scalable implementation (Hoffman *et al.*, 2010).

3.4 PAM50 Genes for Survival Prediction

PAM50 intrinsic subtypes (Parker *et al.*, 2009) identifies five subtypes of breast cancer. PAM50 genes and the associated centroids for clustering the patients into these five subtypes are available publicly. Parker *et al.* (2009) claims that these subtypes have very different prognostic outcomes – *i.e.*, different survival times based on the Kaplan-Meier analysis (Altman, 1990). Hence it's very natural to use the 88 probes corresponding these 50 genes as features to make survival predictions.

We experimented this idea on the METABRIC dataset by using these 88 probes as features to build our models. We tried two different approaches: (1) use the 88 probes expression values as features (PAM50-Model-1), and (2) use the 88 probes as our new microarray data and train a topic model from them (followed our earlier proposed methods to discretize and learn a dLDA model: PAM50-Model-2). Both models performed worse than our earlier proposed models; see Table 3.4 for their performances (with respect to

Table 3.4: Concordance results on the METABRIC held-out test set, using the PAM50 classification genes

Models	Cox	RCox	MTLR
PAM50-Model-1	0.6920	0.7080	0.7071
PAM50-Model-2 (dLDA)	0.6939	0.7020	0.7052

concordance index) on the held out test data.

Chapter 4

Conclusion

Here we present our final conclusions based on our findings and interesting future research directions. Section 4.1 discusses our proposed survival prediction framework and the interpretability of the dLDA features. Section 4.2 lists our final conclusions and finally Section 4.3 presents suggested future research avenues.

4.1 Discussion

Given the growing number of gene expression experiments being cataloged for analysis to discover actionable knowledge, it would be very useful to develop survival prediction models that can utilize such high dimensional data. Therefore we have proposed a novel survival prediction methodology that can learn predictive yet (potentially) interpretable features from the gene expression data. We have limited this study to empirically evaluating whether the use of learned strains can help lead to models that can effectively predict survival, while acknowledging that the interpretation of these strains are high priority future work.

Second, we note that the Cox model (Cox, 1972) has been dominant in both survival analysis and prediction over the years, probably because this model is known to achieve competitive (or better) performance compared to other survival prediction models with respect to concordance. (See Section 2.5.1 for further discussion). However, while concordance is useful for determining whether the model can rank the patients, this is not the only task; it is often

useful to produce meaningful estimates of a patient's survival distribution – *i.e.*, a Calibration (rather than Discriminative) task (Steyerberg *et al.*, 2010).

We therefore endorse identifying and using the evaluation criteria, that is appropriate for the task; here D-calibation is arguably more relevant than concordance. We also note that models like Ridge-Cox and MTLR can perform well in both concordance and a calibration measure such as D-calibration – see the results in Tables 3.1 and 3.2. Note that the Ridge-Cox algorithm recently won Prostate Cancer Dream Challenge 9.5 (Abdallah *et al.*, 2015) (closely followed by MTLR).

Our evaluations show that the MTLR survival prediction model achieves comparatively (better in some cases) performance in both evaluation criteria across both datasets. Moreover, MTLR does not produce only a risk score, nor a single time point prediction, but rather a survival distribution, mapping each time to a probability. MTLR model is a novel addition to the suite of survival prediction techniques, that does not make the proportional hazards assumption and deals with the censored instances appropriately by summing over all possible future alternatives. More information can be found in Yu *et al.* (2011) and an online server of the MTLR algorithm can be found at http://pssp.srv.ualberta.ca.

4.2 Conclusion

Tables 3.1 and 3.2 collectively show that our proposed model, which uses MTLR to learn a model involving various type of the derived genomic features (dLDA strains and/or SuperPC+), performs best in concordance in two different breast cancer datasets. This shows that adding genomic features improves survival prediction and that including both dLDA and SuperPC+ genomic features gives the most consistent improvements across independent datasets – showing that the "framework" that produced the best model in METABRIC, was also the best in BRCA.

This validation on the BRCA dataset shows the robustness of our proposed prediction framework. Moreover the strains extracted by our dLDA procedure

(inspired by topic modeling) can be interpreted as collections of overexpressed or underexpressed gene sets; further analysis is needed to discover and validate the biological insights from these strains.

Our analysis used multiple evaluation criteria, based on our view that the survival models should not be only evaluated based on the concordance but also in other measures (*e.g.*, D-calibration), which elicit the significance of accurate survival predictions (showing that the predominant Cox model is not a silver bullet). As highlighted earlier, for this task, it is more relevant that a model be D-calibrated, rather than have a high concordance index. Our results show that our novel survival prediction model: MTLR coupled with our derived genomic features (dLDA strains and superPC+ components), is capable of better survival prediction compared to the standard survival models.

4.3 Future Work

We envision two major future directions of research from our current knowledge of the problem and the outcome of our research: (1) interpret what each strain from the dLDA represents, and (2) applying our dLDA adaptation to multiple cohorts of different types cancers, and applying to a single cohort of patients with different types of cancer.

(1) As discussed in Section 4.1, model interpretability is a possible advantage of the dLDA model, as it could allow us to understand what each latent strain represents. Moreover understanding (explaining) what each latent strain represents is also an interesting endeavor. But we must acknowledge that this is not straightforward as in the natural language domain. In natural language documents, once we find the latent topics from the corpus, then we can find the most frequent words representing each topic. Just by examining those word clouds, a human who is fluent in that language can see the abstract concept of a common theme across these words of each topic (*e.g.*, words like ball, score, strike would represent the concept of sports). But when we try to do a similar analysis of most probable genes (probes) representing a latent strain,

it is not trivial to find this common theme across the genes. Therefore to understand these strains we need to find (1) ontologies, (2) biological processes, and (3) pathways, etc. that overlap with these groups of genes. We propose enrichment analysis starting from the single genes to gene-sets enrichment. By analyzing the enriched terms of each strain from several databases (KEGG, BP, etc) an expert can identify whether there are any common themes within the enriched terms of a strain. These types of enrichment analysis will give a starting point to understand what each latent strain captures giving us necessary information to direct further analysis.

(2) Secondly, while our evaluation shows that the dLDA adaptation works well across these two breast cancer datasets it would be much more compelling if we can show that our model performs well across multiple cancers. Indeed, applying our model to multiple cohorts of different types cancers (*e.g.*, lung cancer, prostate cancer, etc.) might allow us to explore the generalization of our framework. Similarly applying our model to a single cohort of different cancer types might give us the opportunity to see what type of strains the dLDA model learns. When we do a study with a single cohort of different cancer types, it highly resembles the natural language corpus that contains documents from multiple themes (*e.g.*, sports, entertainment, politics, etc.). So when we apply the dLDA model to a cohort of different types of cancers, we expect to see the model identifying the difference across cancer types and mostly importantly to identify the similarities among cancers as well. So further studies on different cancer types can pave the path to yield the full potential of the dLDA adaptation framework.

Bibliography

Abdallah, K., Hugh-Jones, C., Norman, T., Friend, S., and Stolovitzky, G. (2015). The prostate cancer dream challenge: A community-wide effort to use open clinical trial data for the quantitative prediction of outcomes in metastatic prostate cancer. *The oncologist*, **20**(5), 459–460.

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., *et al.* (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, **403**(6769), 503–511.

Altman, D. G. (1990). *Practical statistics for medical research*. CRC press.

Bair, E. and Tibshirani, R. (2004). Semi-supervised methods to predict patient survival from gene expression data. *PLoS Biol*, **2**(4), e108.

Bicego, M., Lovato, P., Perina, A., Fasoli, M., Delledonne, M., Pezzotti, M., Polverari, A., and Murino, V. (2012). Investigating topic models' capabilities in expression microarray data classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, **9**(6), 1831–1836.

Blei, D. and Lafferty, J. (2006). Correlated topic models. *Advances in neural information processing systems*, **18**, 147.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, **3**, 993–1022.

Cheng, W.-Y., Yang, T.-H. O., and Anastassiou, D. (2013). Development of a prognostic model for breast cancer survival in an open challenge environment. *Science translational medicine*, **5**(181), 181ra50–181ra50.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, **34**(2), 187–220.

Curtis, C., Shah, S. P., Chin, S.-F., Turashvili, G., Rueda, O. M., Dunning, M. J., Speed, D., Lynch, A. G., Samarajiwa, S., Yuan, Y., *et al.* (2012). The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, **486**(7403), 346–352.

Dawson, J. A. and Kendziorski, C. (2012). Survival-supervised latent dirichlet allocation models for genomic analysis of time-to-event outcomes. *arXiv preprint arXiv:1202.5999*.

Deshwar, A. G., Vembu, S., Yung, C. K., Jang, G. H., Stein, L., Morris, Q., *et al.* (2015). Phylowgs: reconstructing subclonal composition and evolution from whole-genome sequencing of tumors. *Genome Biol*, **16**, 35.

Ding, C. and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, **3**(02), 185–205.

Goeman, J. J. (2010). L1 penalized estimation in the cox proportional hazards model. *Biometrical journal*, **52**(1), 70–84.

Harrington, D. P. and Fleming, T. R. (1982). A class of rank test procedures for censored survival data. *Biometrika*, **69**(3), 553–566.

Hoffman, M., Bach, F. R., and Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, **42**(1-2), 177–196.

Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied logistic regression*, volume 398. John Wiley & Sons.

Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, **15**(3), 651–674.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. (2008). Random survival forests. *The Annals of Applied Statistics*, **2**, 841860.

Kalbfleisch, J. D. and Prentice, R. L. (2011). *The statistical analysis of failure time data*, volume 360. John Wiley & Sons.

Koboldt, D. and Network, C. G. A. (2012). Comprehensive molecular portraits of human breast tumours. *Nature*, **490**(7418), 61–70.

Margolin, A. A., Bilal, E., Huang, E., Norman, T. C., Ottestad, L., Mecham, B. H., Sauerwine, B., Kellen, M. R., Mangravite, L. M., Furia, M. D., *et al.* (2013). Systematic analysis of challenge-driven improvements in molecular prognostic models for breast cancer. *Science translational medicine*, **5**(181), 181re1–181re1.

McAuliffe, J. D. and Blei, D. M. (2008). Supervised topic models. In *Advances in neural information processing systems*, pages 121–128.

McCullagh, P. and Nelder, J. A. (1989). *Generalized linear models*, volume 37. CRC press.

Naderi, A., Teschendorff, A., Barbosa-Morais, N., Pinder, S., Green, A., Powe, D., Robertson, J., Aparicio, S., Ellis, I., Brenton, J., *et al.* (2007). A gene-expression signature to predict survival in breast cancer across independent data sets. *Oncogene*, **26**(10), 1507–1516.

Park, M. Y. and Hastie, T. (2007). L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **69**(4), 659–677.

Parker, J. S., Mullins, M., Cheang, M. C., Leung, S., Voduc, D., Vickery, T., Davies, S., Fauron, C., He, X., Hu, Z., *et al.* (2009). Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of clinical oncology*, **27**(8), 1160–1167.

Powers, D. M. (1998). Applications and explanations of zipf's law. In *Proceedings of the joint conferences on new methods in language processing and computational natural language learning*, pages 151–160. Association for Computational Linguistics.

Rogers, S., Girolami, M., Campbell, C., and Breitling, R. (2005). The latent process decomposition of cdna microarray data sets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, **2**(2), 143–156.

Shivaswamy, P., Chu, W., and Jansche, M. (2008). A support vector approach to censored targets. In *ICDM 2007*, pages 655–660. IEEE.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2011). Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, **39**(5), 1–13.

Steck, H., Krishnapuram, B., Dehing-oberije, C., Lambin, P., and Raykar, V. C. (2008). On ranking in survival analysis: Bounds on the concordance index. In *Advances in neural information processing systems*, pages 1209–1216.

Stewart, B., Wild, C. P., *et al.* (2016). World cancer report 2014. *World*.

Steyerberg, E. W., Vickers, A. J., Cook, N. R., Gerds, T., Gonen, M., Obuchowski, N., Pencina, M. J., and Kattan, M. W. (2010). Assessing the performance of prediction models: a framework for some traditional and novel measures. *Epidemiology (Cambridge, Mass.)*, **21**(1), 128.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2012). Hierarchical dirichlet processes. *Journal of the american statistical association*.

Tibshirani, R. *et al.* (1997). The lasso method for variable selection in the cox model. *Statistics in medicine*, **16**(4), 385–395.

Van Wieringen, W. N., Kun, D., Hampel, R., and Boulesteix, A.-L. (2009). Survival prediction using gene expression data: a review and comparison. *Computational statistics & data analysis*, **53**(5), 1590–1603.

Van't Veer, L. J., Dai, H., Van De Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., *et al.* (2002). Gene expression profiling predicts clinical outcome of breast cancer. *nature*, **415**(6871), 530–536.

Wolfinger, R. D., Gibson, G., Wolfinger, E. D., Bennett, L., Hamadeh, H., Bushel, P., Afshari, C., and Paules, R. S. (2001). Assessing gene significance from cdna microarray expression data via mixed models. *Journal of computational biology*, **8**(6), 625–637.

Yu, C.-N., Greiner, R., Lin, H.-C., and Baracos, V. (2011). Learning patient-specific cancer survival distributions as a sequence of dependent regressors. In *Neural Information Processing Systems (NIPS)*, pages 1845–1853.

Appendix A

Survival Prediction

Survival prediction is similar to regression as both involve learning a model that regresses the covariates of an individual to produce an estimate of the value of a dependent real-valued response variable – here, the variable is "time to event" (where the standard event is "death"). But survival prediction differs from the standard regression task as its response variable is not fully observed "censored" at most training instances. The following sections describes in detail important facets of survival analysis. A.1 Initially we describe the nature of the response variable, A.2 introduces the survival and the hazard function, A.3 Kaplan-Meier, a survival function estimator, and finally A.4 prior related work on high dimensional survival prediction.

A.1 Censored Data

In many real world cohorts, most of the instances are "right censored", in that we only observe a *lower bound* on the response value. This might happen if a subject was alive when the study ended, meaning we only know that she lived *at least* 5 years, but do not know whether she actually lived 5 years and a day, or 30 years. This also happens if a subject drops out of a study or moves to a different location, say after enrolling 2.3 years, and is then lost to follow-up; etc. Even though right censoring is common in survival studies there are other censoring types as well such as: left censoring (unknown start) and interval censoring (both start and end not known). Figure A.1 shows different types of censoring events. In general, when dealing with cohorts of censored instances,

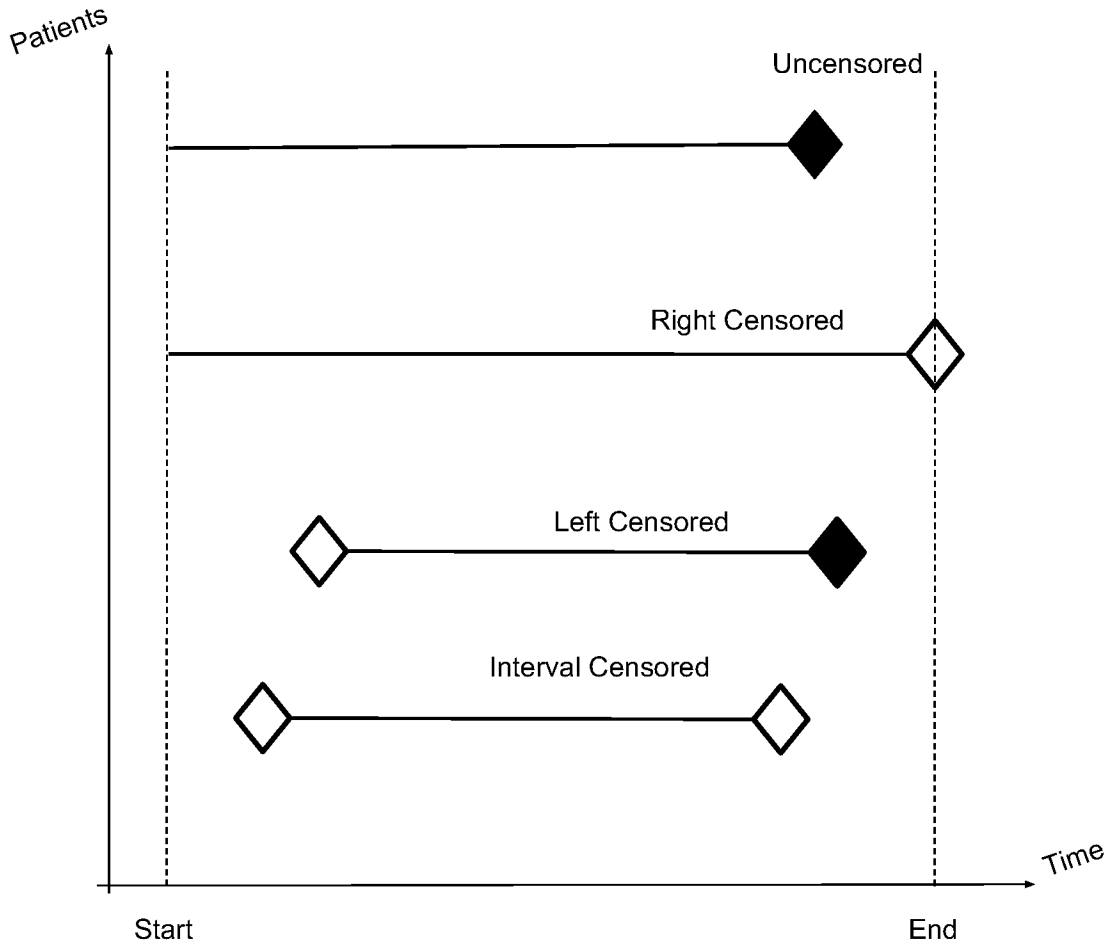


Figure A.1: Patients with different censoring types, solid diamond:–uncensored, hollow diamond:–censored; A–Uncensored, B–Right censored, C–Left censored and D–interval censored

one cannot simply ignore such instances as it is common for many (or often, *most*) of the training instances to be right-censored. Such "partial label information" is problematic for standard regression techniques, which assume the label is completely specified for each training instance. Fortunately, there are the survival prediction algorithms that can learn an effective model, from a cohort that includes such censored data. Each such dataset contains descriptions of a set of instances (patients), as well as two "labels" survival time and censor bit (t_i, d_i). Survival time of the i^{th} patient $t_i = \min(T_i, C_i)$ where T_i is true survival time of the patient and C_i is right censored time. Censor bit

$d_i = 0$ if censored $T_i > C_i$ and $d_i = 1$ if observed $T_i < C_i$.

A.2 Survival Function Vs Hazard Function

Survival function $S(t)$ is the probability that a patient is alive beyond a given time t; where $S(t) = P(T > t)$. The survival function is essential to survival analysis since it captures the probability of survival at every time step until the event of interest. Survival functions always start with probability 1.0 (patient is alive) and drops as time progress. Another crucial function predominantly used in survival analysis is the hazard function $h(t)$ which captures the instantaneous hazard of a patient.

$$h(t) \;=\; \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t \mid T \leq t)}{\Delta t}$$

Note that hazard function is *not* a probability rather the failure rate which can take any positive value (0 to ∞). Based on the definitions of the survival and the hazard function there is an interesting relationship between them (given in Equations A.1) allowing us to compute one from the other.

$$\begin{aligned} S(t) \;&=\; \exp\left[-\int_0^t h(u)du\right] \\ h(t) \;&=\; -\left[\frac{\frac{dS(t)}{dt}}{S(t)}\right] \end{aligned} \tag{A.1}$$

A.3 Kaplan-Meier (KM) Estimator

$$\hat{S}(t) \;=\; \prod_{t_i < t} \frac{N_i - D_i}{N_i} \tag{A.2}$$

KM is a non-parametric estimator of the survival function, given by Equation A.2, where N_i is number of subjects "at risk" *i.e.*, number of survivors immediately before time t_i when none of the patients are censored; but when we have censored patients, N_i = (survived patients − censored patients), where D_i = number of patients who died at time t_i.

KM plot is an elegant non-parametric way to analyze survival data, which is also widely used to compare several survival functions of different groups of patients (*e.g.*, control vs treatment). We can compute median (/mean) of the estimated KM survival function as summary statistics to describe the cohort of patients. Also, the median (/mean) survival days can be used as the prediction from the KM estimator in survival prediction tasks. Note that the mean of the estimated survival distribution is the area under the curve. Log-Rank (Harrington and Fleming, 1982) test is a common hypothesis test used to assess the similarity between two (or more) KM survival functions. Achieving smaller p-values from the log-rank test gives significant evidence to support the alternate hypothesis that the survival functions are different. This test also allows us to validate the existence of different hazard rates in the cohort.

A.4 Survival Prediction for High Dimensional Data

One recent challenge in survival prediction is the high dimensionality of the covariates (Van Wieringen *et al.*, 2009; Margolin *et al.*, 2013). Especially in the medical domain, with the introduction of novel genomic advances, recent survival studies include high dimensional data such as gene expression, copy number variations, etc. Therefore survival prediction models are expected to handle such high dimensional data. Here we discuss different approaches followed in literature to over come the high dimensionality in survival prediction. These proposed techniques follow two major themes: (1) a new dimensionality reduction step before survival prediction, or (2) a new (regularized) model capable of handling high dimensional data. Both approaches have been shown to have empirical success depending on the dataset and the evaluation criteria (Van Wieringen *et al.*, 2009). We present here the proposed techniques under each theme:

- **Dimensionality Reduction**

– *Clustering gene expression data*:

This technique was proposed by Alizadeh *et al.* (2000) where the gene expression data is initially clustered using hierarchical clustering and the cluster id's are used as covariate in the Cox proportional hazards model (Cox, 1972). There are several drawbacks in this approach: (1) its not clear whether a single variable cluster id could capture all the information relevant to survival prediction from gene expression, and (2) deciding the number of clusters is a common problem in clustering and often a very difficult one to address (Van Wieringen *et al.*, 2009)

Another hierarchical clustering approach called supervised harvesting of expression trees (Alizadeh *et al.*, 2000). Rather than using the cluster id's they compute the average expression values of each cluster. After the hierarchical clustering step, each of the cluster averages are used as covariates to train a Cox proportional hazards model (Cox, 1972) with forwards addition and backward deletion to select the final set of clusters. This model is highly sensitive to the clustering algorithm used in the initial step and it requires a large number of samples to have higher performance (Van Wieringen *et al.*, 2009)

– *Univariate gene selection*:

Is a straightforward procedure, where each gene in the expression data is ordered by the association between the gene's expression values and the survival times. Here each gene's expression values are used as a univariate with the Cox model (Cox, 1972) and the corresponding p-values of the Cox scores are used to rank the genes (Van Wieringen *et al.*, 2009). After computing the p-values of each gene, we remove all the genes whose p-value is higher than the threshold, being the set of genes with significant association with the survival time (We use cross validation to set the threshold). Finally these selected gene are used as covariates in the Cox

proportional hazards model to predict risk scores. A major drawback of this technique is that the resulting set of genes might be highly correlated with each other this might lead to poor predictions (Van Wieringen *et al.*, 2009).

- *Supervised principle component (SuperPC) analysis* :
  SuperPC (Bair and Tibshirani, 2004) and it is very similar to the unsupervised PCA, but differs by using only the genes selected from univariate gene selection. Then the resulting principle components are used as covariates in the Cox model (Cox, 1972). While SuperPC does not suggest how many principle components to use, in the implementations generally the first two components are used (Van Wieringen *et al.*, 2009).

- **Regularized Survival Models**

  - *L_1-penalized Cox*:
    The penalized Cox model (Park and Hastie, 2007) uses an L_1 penalty with the Cox model (Cox, 1972). This L_1 penalty on the coefficients makes most of the coefficients to shrink to zero, therefore resulting in an automatic feature selection (Van Wieringen *et al.*, 2009). Park and Hastie (2007) proposed an efficient algorithm to compute the coefficients for high dimensional datasets compared to the previous LASSO Cox model (Tibshirani *et al.*, 1997). They use a hyper-parameter λ to control the regularization factor, which starts from $\lambda = \infty$ (where all coefficients are zero) and decrements stepwise in each iteration allowing more and more coefficients to take non-zero values. The algorithm stops when the set of non-zero coefficients are stable (not changing) (Van Wieringen *et al.*, 2009). The final model will have fewer covariates contributing to the prediction with non zero coefficients.

  - *Tree Ensembles & Random Forests*:
    While these techniques work with high dimensional data sets users

often to apply an initial selection process to reduce the computational cost (Van Wieringen *et al.*, 2009). Survival trees are grown similar to decision tress, branching on each covariate and growing the tree no more splits based on "stopping criterion" (Hothorn *et al.*, 2006). Finally survival functions are estimated using the KM estimator from all the instances of a leaf node, resulting in a KM curve for every leaf node. After learning, a new instance is dropped through the tree, and the prediction is based on the KM curve associated with the leaf node it results (Often the mean or the median of the KM curve is used as the predicted survival time).

A bagging (bootstrap aggregation) procedure is proposed for survival trees by Hothorn *et al.* (2006), which produces one survival tree from each of the several bootstrap datasets. Afterwards, a new instance is dropped through all the trees, reaching a set of leaf nodes we then use all the instances from every leaf node to compute a KM curve for the final prediction. Hothorn *et al.* (2006) also proposed a random forest for survival prediction similar to tree ensembles but varying in the branching mechanism. In addition to the number of trees the random forest learner also uses a second parameter that specifies the number of candidate covariates sampled for each split. Random forests are known to be sensitive to these two parameters (Van Wieringen *et al.*, 2009). Both tree ensembles and random forests seem to have similar performances on multiple survival datasets (Van Wieringen *et al.*, 2009).

Appendix B

D-calibration Computation for Censored Instances

Section 2.6.2 describes D-calibration for uncensored patients. Based on the Equation 2.8, when computing P_g's, for uncensored patients, $Pr_i(d_i)$ can be computed using the individual's survival curve and the (event) death time d_i. The appropriate subgroup's P_g will be incremented (by one) based on the $Pr_i(d_i)$ value *e.g.*, if $Pr_i(d_i) = 0.55$ then the appropriate subgroup P_{11} (coressponds to $(0.55, 0.60]$) will be incremented; $P_{11} = P_{11} + 1$. If the patient is censored, we compute the $P_i(d_i)$ similar to an uncensored patient, but we increment all subgroups that are eligible to contain this patient, with a fractional weight (instead of one). These are the subgroups where the lower event probability threshold (a_g of $[a_g, b_g)$) is less than $Pr_i(d_i)$) *i.e.*, $\forall_g\ a_g \leq Pr_i(d_i)$. Our proposed heuristic weight is computed as $weight = \frac{1}{\max\left(1, \left\lceil Pr_i(d_i) * G \right\rceil\right)}$. This distributes the contribution by the censored patient among all subgroups where the lower event probability threshold (a_g) is less than $Pr_i(d_i)$ of the censored time, agreeing with the fact that the right censored time is a lower bound for the survival time and the patient might have died at any time point later.

Appendix C

Tutorial on Latent Dirichlet Allocation (LDA)

Many (natural language) topic models have been proposed over the years, including; LDA (Blei *et al.*, 2003), probabilistic Latent Semantic Allocation (Hofmann, 2001), Hierarchical Dirichlet Process (Teh *et al.*, 2012), etc. Here we present a short tutorial on LDA. LDA is a widely used (generative) topic model proposed by Blei *et al.* (2003), which has been successfully applied to natural language (NL) documents for multiple different tasks (Blei *et al.*, 2003). The LDA generative process assumes that each document corresponds to a distribution over multiple topics (document-topics) and every topic is a distribution over the vocabulary of words (topic-words). This allows us to represent a document with thousands of words with a small (in the order of hundreds) number of topics. LDA is an unsupervised technique that finds inherent structure present in the data, based on latent structures (topics) that capture frequent co–occurring words across multiple documents.

We use following notations throughout this tutorial to describe the LDA model (adopted from Blei *et al.* (2003)):

- w is a word in a NL document. Superscripts w^v where $v \in 1, \ldots, V$ denote the index of the word in the vocabulary vector; *i.e.*, $w^v = 1$ if v^{th} word in the vocabulary is present and $w^u = 0$ for all $u \; != \; v$

- $\mathbf{w}$:= $[w_1, w_2, \ldots, w_N]$ represents a document including a sequence of N words

- D is a collection of M documents (corpus)
- k is fixed number of latent topics
- z:= k-dimensional vector denoting the topic assignment of a word
- β:= matrix of size $k * V$
- $\beta_{i,j}$:= $P(\, w^j = 1 \,|\, z^i = 1\,)$ is the probability of word w^j occurring under topic i
- α:= k-dimensional vector of the Dirichlet parameters used to represent document-topics distribution for the corpus
- θ:= k-dimensional Dirichlet random variable, representing the random assignments of a document's topic components
- ε is a parameter for the Poisson distribution corresponding to the number of words in a document

LDA assumes every document in the corpus is generated by the following generative process:

- Draw $N \sim Poisson(\varepsilon)$
- Draw $\theta \sim Dirichlet(\alpha)$
- For $n = 1, \ldots, N$:
    1. Draw a topic $z_n \sim Multinomial(\theta)$
    2. Draw a word $w_n \sim Multinomial(\beta[z_n, :])$

From the generative process above we generate a document by first sampling a topic from the document-topics distribution and then sampling a word from the selected topic's topic-words distribution; once we sampled a word we can repeat this process for required number of words (N) to generate our document.

LDA's graphical model representation is given in Figure C.1. From the Figure C.1 we can see that only the words of the documents are observed

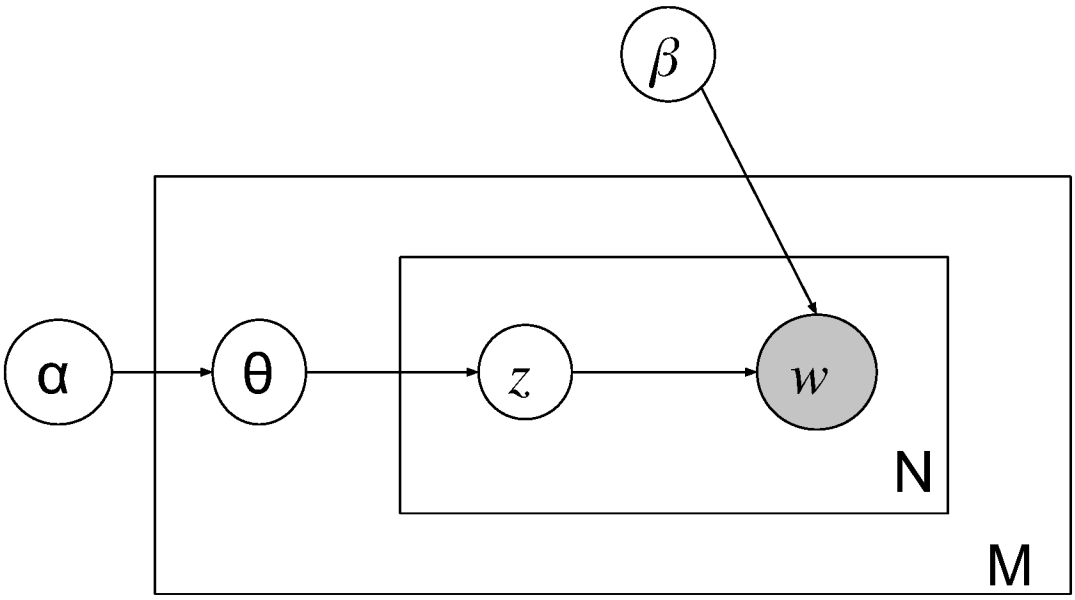


Figure C.1: LDA plate model, N-number of words in a document, M-number of documents in the corpus

and all other variables are inferred through the learning process. It has been wildly understood that the topics learned from the LDA model are generally non-correlated; note that from the plate model (in Figure C.1) we cannot make any explicit independence claims but implicitly the learned topics are prone to become independent because of the Dirichlet on the document-topics distribution (Blei and Lafferty, 2006).

LDA requires two input parameters from the user apart from the corpus: (1) the Dirichlet prior for the document-topics distribution (α), and (2) the number of latent topics (k). Note that in some LDA graphical representations, an additional variable as a prior on the topic-words distributions (influencing β) is added for smoothing (referred to as smoothed LDA), but for simplicity we have not included this prior.

Given the parameters the joint probability distribution of $\theta, \mathbf{z}, \mathbf{w}$ of a single

document is given in Equation C.1.

$$P(\,\theta, \mathbf{z}, \mathbf{w}\,|\,\alpha, \beta\,) \;=\; P(\,\theta\,|\,\alpha\,)\prod_{n=1}^{N} P(\,z_n\,|\,\theta\,)P(\,w_n\,|\,z_n, \beta\,) \tag{C.1}$$

Based on the Figure C.1 of the LDA plate model, we can observe the inter-dependencies between the random variables and derive the join probability distribution of a single document. To compute the joint probability of the corpus, we need to take product over the marginal probabilities of all the documents. We can obtain the marginal distribution from Equation C.1 by integrating out θ and summing over $\mathbf{z}$. Equation C.2 gives the join probability of the corpus D.

$$P(\,D\,|\,\alpha, \beta\,) \;=\; \prod_{d=1}^{M} \int P(\,\theta_d\,|\,\alpha\,) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} P(\,z_{dn}\,|\,\theta_d\,)P(\,w_{dn}\,|\,z_{dn}, \beta\,) \right) d\theta_d \tag{C.2}$$

Equation C.2 has parameters that operate at multiple levels. (1) $\alpha,\ \beta$, are corpus level parameters, which are initialized once for the whole corpus, (2) θ is a document level random variable (sampled), assigned for each document, and (3) $z_{d,n}$ is assigned for each word $w_{d,n}$ in a document d. From above Equations C.2, and C.1, we understand the relationships between the random variables in the joint probability distribution. But we need the posterior distribution given $\alpha,\ \beta$ and the document. Equation C.3 gives the posterior distribution over the latent variables $\theta, \mathbf{z}$.

$$P(\,\theta, \mathbf{z}\,|\,\mathbf{w}, \alpha, \beta\,) \quad = \quad \frac{P(\,\theta, \mathbf{z}, \mathbf{w}\,|\,\alpha, \beta\,)}{P(\,\mathbf{w}\,|\,\alpha, \beta\,)} \tag{C.3}$$

Equation C.3 is intractable for exact inference (Blei *et al.*, 2003). Therefore we use approximate inference algorithms such as Laplace approximation, variational approximation, and Markov chain Monte Carlo (Blei *et al.*, 2003). When following the variational inference procedure we can approximate the posterior distribution with a family of simpler distributions with more independence assumptions. Once we define the variational parameters for the approximate distribution we can then estimate these parameters by minimiz-

ing the Kullback–Leibler divergence between the approximate distribution and the posterior distribution for each document (Blei *et al.*, 2003). Note that we assumed that β (and α) is known when computing the variational parameters. Therefore we can compute both the variational parameters and the final β using an expectation maximization procedure (Blei *et al.*, 2003). Explanations on the variational expectation maximization procedure and more information on the variational inference procedure are given in Blei *et al.* (2003) for interested readers.