Kernel Learning Framework for Cancer Subtype Analysis with Multi-omics Data Integration

William Bradbury
wbradbur@stanford

Thomas Lau
thomklau@stanford

Shivaal Roy
shivaal@stanford

December 12, 2015

Abstract

Recent advances in Multiple Kernel Learning (MKL) and unsupervised clustering methods have each enabled large-scale analysis of integrated multi-omics data for cancer subtyping. However, efforts to combine the advantages of a kernel approach with the flexibility of unsupervised methods have not progressed due to the underconstrained nature of the problem. In this paper, we present a novel approach to solving this problem based on methods for Unsupervised Multiple Kernel Learning (UMKL). We present the possibility of constraining the problem with available data on clinically determined subtypes for cancer patients. We successfully demonstrate that modifying the constraints of the UMKL problem to include conditions for sparse clinical labelling data is tractable and thus presents a robust alternative for cancer subtype analysis by reducing it to an efficient alternating quadratic optimization problem.

1. Introduction

The expansion and increased availability of large-scale biomedical data in the past decade has led to the rising use of Machine Learning as an integral component for providing biological insights - especially in the domain of cancer research. Recently, with the increasing number of large scale genomic projects, such as The Cancer Genome Atlas (TCGA), the amount of openly accessible patient data is greater than ever. In order to make sense of this ever increasing quantity of data, numerous statistical approaches have been developed to analyze TCGA and other similar datasets.

Cancer subtyping is of particular clinical relevance because it has the potential to enable medical practitioners to design more finely tuned cancer therapies. Intuitively, the more precisely we are able to classify types of cancer, the more effective the treatments we design can be. Futhermore, cancer subtyping has the potential to more accurately predict treatment outcomes and survival rates based on rates of cancer progression and other clinical factors. Currently, doctors can identify cancer subtypes in patients by measuring biological features during the progression of their cancer. However, these subtype classifications are slow to perform, often very broad (Luminal A vs Luminal B for BRCA), and have weak prognostic ability, since subtyping by physicians can only be refined as the cancer progresses. Recent advances in statistical analyses suggest there is much potential to improve cancer subtyping.[1,5,4]

Previous approaches[1,5,7] suggest the potential for a UMKL solution which relies on sparse clinical data to analyze multi-omic data in the search of more specific and useful subtype characterizations - thereby empowering doctors to determine optimal methods of treatment. We propose a method for such an approach, combining the techniques of previous UMKL approaches with novel constraints based on clinical data.

TCGA contains information about methylation, copynumber variation (CNV), microRNA, mRNA, RPPA, and other factors relevant to the onset and progression of cancer. The interaction between these levels of genetic control is referred to as multi-omics. Intuitively, by looking at the genomic causes of cancer, we can more accurately predict future outcomes - as opposed to clinically looking at phenotypic cancer progression.

Our goal in this paper is to extend the existing unsupervised multiple kernel learning technique in *Zhuang et al.* to leverage sparse labeling for specific use with clinical and multi-omic data from TCGA. We wish to demonstrate that such an extension is still a tractable optimization problem.

2. Related Work

This paper builds off of *Zhuang et al.*, which proposes a method for UMKL that also integrates case-specific constraint functions. This paper demonstrates the feasibility of the original problem and provides cases for which the algorithm performs especially well. The algorithm presented in this paper, however, does not perform well as the number of features increases or as the number of sample points decreases.[7]

Supervised multiple kernel learning has been implemented in *Daemen et al.* to classify rectal cancer microarray data. The resulting Support Vector Machine accurately classified different outcomes, often reaching above 0.90. The model also performed better when using more than one genome-wide data set, suggesting that integrating multiple genome-wide data sources allows models to reach higher accuracy.[1]

The iCluster algorithm developed by *Shen et al.* was used to accurately group cancer outcomes based on multi-omic considerations found in TCGA data on breast cancer. The algorithm clusters different cancer subtypes using multiple genomic features such as DNA copy number changes and gene expression. iCluster, however, does not take advantage of any kernel methods.[5]

We build upon these papers to solve the problem of harnessing the power of kernels for unsupervised learning in a high dimensional feature space with a relatively small number of samples.

3. Background

3.1. Multiple Kernel Learning

Multiple Kernel Learning is the use of a linear combination of kernels to map points to a higher-dimensional feature space where they can be more easily separated by an SVM. We write this as

$$\mathcal{K}_{conv} = \sum_{t=1}^{m} \mu_t k_t$$

where μ_t is the weight assigned to each kernel. In MKL, we try to learn the kernel combination that linearly separates the data best.

3.2. Unsupervised Multiple Kernel Learning

Unsupervised Multiple Kernel Learning also implements a linear combination of kernels to create a distance metric in a higher-dimensional space. The distance metric is used to determine groupings using a k-means or alternate clustering algorithm.[7]

3.3. Sparse Labels

We take advantage of various labels, although sparse, to constrain our data and aid in the clustering process. We incorporate the clinical data mentioned earlier into our cost function to have the resultant combination of kernels reflect this added restriction. We also impose domain knowledge of the separation between cancer and non-cancer patients. This data can be included in our model as further priors on the subtypes.

4. Optimization Problem

4.1. Cluster-label alignment metric

We design an optimization problem to produce a kernel and clustering which together yield the best assignment of patients to subtypes. In order to do this, we employ the cost functions found in *Zhuang et al.* as well as two new cost functions:

- A good kernel should induce kernel values which place samples of the same subtype together in the feature mapping. In other words for each $\mathbf{x}_i$ we expect that the optimal kernel minimizes $\sum_{\mathbf{x}_j \in \mathcal{L}_i} k_{ii} - 2k_{ij} + k_{jj}$, where C_i is the set of all samples with the same known label as $\mathbf{x}_i$. $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. $C_i = \varnothing$ for $\mathbf{x}_i$ that do not have known labels.

- A good kernel should induce kernel values which place samples of different subtypes apart in the feature mapping. When this is the case, we expect the kernel to maximize $\sum_{\mathbf{x}_j \in \mathcal{L}_{i\neq j}} k_{ii} - 2k_{ij} + k_{jj}$. Where here $C_{i\neq j}$ is the set of sample points labeled with a different subtype than $\mathbf{x}_i$.

Each of these cost functions is designed to affect the optimization problem in such a way as to yield a kernel which places co-labeled points together and differently labeled points apart. These heuristics guide the search for an optimal kernel based on the prior sparse clinical information by encouraging the eventual adoption of a kernel which conforms to the clinical data as well as possible. Such a kernel will yield a clustering which places co-labled points in the same or nearby clusters, while placing differently labeled points in different clusters if possible.

4.2. Cost function

In order to obtain an overall cost function for the optimization problem, we combine these cluster-label alignment metrics with those found in *Zhuang et al.* to form the overall optimization problem:

$$\begin{aligned}\min_{\mathcal{B}, k \in \mathcal{K}_{conv}} \frac{1}{2} \sum_{i=1}^{n} & \left\| \mathbf{x}_i - \sum_{\mathbf{x}_j \in B_j} k_{ij} \mathbf{x}_j \right\|^2 \\ & + \gamma_1 \sum_{i=1}^{n} \sum_{\mathbf{x}_j \in \mathcal{B}_i} k_{ij} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2 \\ & + \gamma_2 \sum_{i} |\mathcal{B}_i| \\ & + \gamma_3 \sum_{i=1}^{n} \sum_{\mathbf{x}_j \in \mathcal{L}_i} (k_{ii} - 2k_{ij} + k_{jj}) \\ & - \gamma_4 \sum_{i=1}^{n} \sum_{\mathbf{x}_j \in \mathcal{L}_{i\neq j}} (k_{ii} - 2k_{ij} + k_{jj})\end{aligned}$$

where

$\mathcal{B}$ is a clustering assignment,

$$\mathcal{K}_{conv} = \{k(\cdot,\cdot) = \sum_{t=1}^{m} \mu_t k_t(\cdot,\cdot) : \sum_{t=1}^{m} \mu_t = 1, \mu_t \geq 0\},$$

γ_i control constraint trade-offs, and

$\mathcal{L}$ is given in §4.1.

This cost function is not–in its present form–feasible to optimize given the size of the data.

4.3. A simpler formulation

Instead of optimizing the above function with respect to $\kappa \in \mathcal{K}$ and $\mathcal{B}_i$ we instead formulate the problem as one of optimizing the cost function with respect to μ and $\mathbf{D}$. This allows us to eventually present the problem as one of solving a quadratic program on the vector μ and later $\mathbf{D}$.

As in *Zhuang et al.* we define matrices $\mathbf{D}, \mathbf{S} \in \{0,1\}^{n\times n}$ where each element is given as

$$\begin{aligned}[\mathbf{D}]_{ij} &= \mathbb{1}\{\mathbf{x}_j \in \mathcal{B}_i\} \\ [\mathbf{S}]_{ij} &= \mathbb{1}\{\mathbf{x}_j \in \mathcal{L}_i\} \\ [\mathbf{Q}]_{ij} &= \mathbb{1}\{\mathbf{x}_j \in \mathcal{L}_{i\neq j}\} \\ [\mathbf{M}]_{ij} &= \mathbf{x}_i^T \mathbf{x}_i + \mathbf{x}_j^T \mathbf{x}_j - 2\mathbf{x}_i^T \mathbf{x}_j\end{aligned}$$

so that we can write:

$$\begin{aligned}\min_{\mu, \mathbf{D}} \frac{1}{2} & \left\| \mathbf{X}(\mathbf{I} - \mathbf{K} \circ \mathbf{D}) \right\|_F^2 \\ & + \gamma_1 \mathrm{tr}\, \mathbf{K} \circ \mathbf{D} \circ \mathbf{M}(\mathbf{1}\mathbf{1}^T) + \gamma_2 \left\| D \right\|_{1,1} \\ & + \mathrm{tr}\, \left(\left[(\mathbf{K} \circ \mathbf{I})(\mathbf{1}\mathbf{1}^T) - 2\mathbf{K} + (\mathbf{1}\mathbf{1}^T)(\mathbf{K} \circ \mathbf{I})^T \right] \right. \\ & \qquad \left. \circ (\gamma_3 \mathbf{S} + \gamma_4 \mathbf{Q}) \right) (\mathbf{1}\mathbf{1}^T) \\ \text{s.t } [\kappa]_{ij} &= \sum_{t=1}^{m} \mu_t k^t(\mathbf{x}_i, \mathbf{x}_j), \\ & 1 \leq i, j \leq n, \\ & \mu^T \mathbf{1} = 1, \\ & \mu \geq 0, \\ & \mathbf{D} \in \{0,1\}^{n\times n}\end{aligned}$$

The function given is not quadratic and also not guaranteed to be convex. Because of this, there are no off-the-shelf tools which are able to generally solve this problem. This difficulty motivates our work to reformulate the problem as an alternating optimization problem in the next section.

4.4. Alternating Optimization Algorithm

In order to formulate the problem in such a way that it can be approximately solved, we break the problem into two components.[7] Consider the space parametrized by μ and $\mathbf{D}$. We can reduce the above problem to one of coordinate descent by alternatively optimizing μ and $\mathbf{D}$ while holding the other constant. If we can show that each of these individual problems is itself quadratic and therefore tractable, we will have broken the problem down into a tractable approximation.

4.4.1 Solving μ by fixing D

We accomplish this separation by first optimizing μ while holding $\mathbf{D}$ constant and ignoring the otherwise difficult constraints on $\mathbf{D}$. Fixing $\mathbf{D}$ we try to solve for μ by minimizing the cost function:

$$J(\mu) = \mu^T \left(\sum_{t=1}^{m} \sum_{i=1}^{n} \kappa_{t,i} \kappa_{t,i}^T \circ \mathbf{d}_i \mathbf{d}_i^T \circ P \right)^T \mu + \mathbf{z}^T \mu,$$

where

$$\begin{aligned} [\mathbf{z}]_t = & \sum_{i=1}^{n} (2\gamma_1 \mathbf{v}_i \circ \mathbf{d}_i - 2\mathbf{p}_i \circ \mathbf{d}_i \\ & + \gamma_3 (\mathbf{e}_i \sum_j S_{ij} + \mathbf{e}_i \sum_j S_{ji} - 2\mathbf{s}_i) \\ & - \gamma_4 (\mathbf{e}_i \sum_j Q_{ij} + \mathbf{e}_i \sum_j Q_{ji} - 2\mathbf{q}_i))^T \kappa_{t,i} \end{aligned}$$

$$\mathbf{P} = \mathbf{X}^T \mathbf{X}$$

$$\kappa_{t,i} = [k^t(\mathbf{x}_i, \mathbf{x}_1), ..., k^t(\mathbf{x}_i, \mathbf{x}_n)]^T$$

$\mathbf{p}_i$ is the i-th column of $\mathbf{P}$

$\mathbf{v}_i$ is the i-th column of $\mathbf{M}$

$\mathbf{s}_i$ is the i-th column of $\mathbf{S}$

$\mathbf{q}_i$ is the i-th column of $\mathbf{Q}$

Most importantly, this cost function is a Quadratic Program and so is tractable, even for large $\mathbf{X}$.

4.4.2 Solving D by fixing μ

Because we introduced new constraints on the kernel, but not the cluster matrix $\mathbf{D}$, the optimization step for $\mathbf{D}$ is unchanged from its presentation in *Zhuang et al.* and so we simply give the result for each column of $\mathbf{D}$:

$$J(\mathbf{d}) = \mathbf{d}^T \left(\kappa \kappa^T \circ \mathbf{P} \right) \mathbf{d} + \left(2\gamma_1 \kappa \circ \mathbf{v} - 2\kappa \circ \mathbf{p} \right)^T \mathbf{d},$$

which is of the form

$$J(\mathbf{d}) = \mathbf{d}^T \mathbf{W} \mathbf{d} + \mathbf{c}^T \mathbf{d}$$

and so is also tractable with a QP convex optimization by iterating over each sample point and using this minimization problem to find the optimal set of neighbors.

5. Discussion

In deriving a quadratic problem from our initial complex constrained optimization problem, we have shown that our new approach for incorporating partially labeled data is a solvable problem. Here, we explain the underpinnings of the initial constraints and derivation of its quadratic form.

5.1. Underlying motivations

The original UMKL algorithm presents two constraints on the behavior of the kernels. The first was that of a standard clustering optimization problem: minimize the sum of the distances between samples in the same cluster. In this case, however, the distances were computed by the kernel function being chosen. Thus, a good choice of kernel function would place points which end up in the same cluster together in the feature mapping. This is useful for learning an optimal kernel on a large quantity of 'training' data, only to apply it using something other than a clustering method. *Zhuang et al.* demonstrates this approach by learning an optimal kernel for a data set, and then applying that kernel to do classification with an SVM.

The second constraint on the behavior of the kernels was that of continuity with the given sample points. This constraint enforces, as far as possible, that the kernel should have a limited impact on the geometry of the data. Points should not be mapped so that they are distant from their neighbors in the original space. This constraint ensures that the kernel does not map points in an *ad hoc* manner such that essentially meaningless, but perfect, clusters form in the feature mapping space.

5.2. Our contributions

Building on this last constraint, this paper introduced two new constraints. The first specified that points which are given the same labels in the cluster priors are placed as close to possible in the feature mapping. Combined with the previous constraint, this ensures the learned kernel will highlight the underlying geometry in such a way as points with the same label are close together, but it will not mangle the space unnecessarily. This forces the kernel to 'learn' whatever is important to identifying each of the prior clusters.

In a similar vein, the second new constraint requires that points with different prior clusters are placed as far apart as possible. By the same mechanism as the previous constraint, this forces the kernel optimization algorithm to learn a kernel that highlights the features which distinguish clusters.

Both of these new constraints operate on the principle that the goal of a kernel in a situation in which the sample data is highly dimensional is to highlight the features of that data which are most important. We designed the optimization problem such that the optimal kernel does just this.

5.3. Deriving quadratic form

Injecting these additional constraints into the problem would help to incorporate sparse training data, but only if the new problem was solvable. In order to demonstrate that the new problem is solvable we showed that it could be reduced to an alternating optimization problem, each step of which was itself quadratic. There are numerous know methods for efficiently solving Quadratic Programming problems. An implementation of this algorithm would simply have to make use of an available convex optimization toolkit. This drastically reduces the complexity of the problem from a general mixed integer problem to a quadratic programming problem.

In order to demonstrate that the problem could be reduced to two quadratic programming problems, we primarily focused on the optimization of μ while holding the clusters (represented in $\mathbf{D}$) constant. This allowed us to forgo all the constraints on $\mathbf{D}$, and thus simplified the problem immensely.

We know that the the kernel matrix $\mathbf{K}$ is given by a summation over the tensor $\kappa_{t,i,j}$ in the dimension indexed by t, weighted by μ. Thus,

$$\mathbf{K} = \sum_{t=1}^{m} \mu_t \kappa_t.$$

Thinking about $\mathbf{K}$ in this fashion allows for interpreting the additional constraints as selection and scaling operations on columns of κ. We think of the problem as one of determining which kernel elements to include and which to avoid for each $\mathbf{x}_i$ and for each κ_t. For each frame of the tensor κ, which elements must be summed and with which coefficients?

In order to accomplish this filtering step, we translate the initial cluster priors into matrices which specify whether two elements have the same or different initial clusterings. Columns of these matrices can then be used as masks for selecting only kernel elements that refer to the distances between samples of the same lable or of different labels. These correspond to each of the two constraints.

5.4. Implications

We believe that the work presented in this paper will be beneficial to the bio-computation community by providing an additional tool for making sense of TCGA and other large-scale data sets. In particular, we think that applications such as cancer subtype analysis–which initially motivated this investigation–are particularly suited due to the the small number of patients, expansive feature space, complex geometry, and existence of sparsely labeled subtype data.

6. Acknowledgements

We thank Prof. Olivier Gevaert for his help in formulating and guiding this project and for his help in understanding and navigating MKL and TCGA. We also thank Prof. Serafim Batzoglou for his initial suggestions which led us to cancer subtype analysis.

References

[1] Anneleen Daemen et al. "A kernel-based integration of genome-wide data for clinical decision support." In: *Genome medicine* 1.4 (2009), p. 39.

[2] Anneleen Daemen et al. "Improved microarray-based decision support with graph encoded interactome data." In: *PloS one* 5.4 (2010), e10225.

[3] GRG Lanckriet and Nello Cristianini. "Learning the kernel matrix with semidefinite programming". In: *Journal of Machine Learning Research* 5 (2004), pp. 27–72.

[4] R. Shen, A. B. Olshen, and M. Ladanyi. "Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis". In: *Bioinformatics* 25.22 (2009), pp. 2906–2912.

[5] Ronglai Shen et al. "Integrative Subtype Discovery in Glioblastoma Using iCluster". In: *PLoS ONE* 7.4 (2012), e35236.

[6] Emily a. Vucic et al. "Translating cancer 'omics' to improved outcomes". In: *Genome Research* 22.2 (2012), pp. 188–195.

[7] Jinfeng Zhuang et al. "Unsupervised multiple kernel learning". In: *Proceedings of the Third Asian Conference on Machine Learning* 20 (2011), pp. 129–144.