Laurent Jacob
Supervisor: Jean-Philippe Vert

Master report

Multitask learning for epitope prediction

ENSMP/CBIO
Spring 2006

Contents

Acknowledgments **iii**

Introduction **v**

1 Context **1**
1.1 Intelligent vaccine design 1
1.1.1 Biological background 1
1.1.2 The importance of prediction 6
1.2 Methods 7
1.2.1 Leveraged logistic regression 7
1.2.2 Multitask kernel 8

2 Methods **13**
2.1 Comparative analysis 13
2.1.1 Logistic regression as a regularization problem 13
2.1.2 Leveraged LR as a special case of multitask learning 14
2.1.3 Regularization: the one-level tasks case 15
2.1.4 Regularization: the clustered tasks case 16
2.2 Controlled leveraged logistic regression model 17
2.2.1 First attempt 17
2.2.2 Controlled leveraged logistic regression 18
2.3 Further generalization 22
2.3.1 Non-linear single-task kernels 22
2.3.2 Non-Dirac task-specific kernels 22
2.4 Optimization methods 23
2.4.1 Conjugate gradient 23
2.4.2 Newton 24

3 Experiments **27**
3.1 Data 27
3.2 Effect of the parameters on the resulting classifiers 27
3.3 Parameter selection and comparison with leveraged LR 29
3.4 Discussion 31

Conclusion **33**

A Proof of the leveraged LR norm **37**

B Proof of the gllr norm **41**

C Proof of the cllr norm **43**

D Alternative proof for the cllr norm **45**

E Proof of proposition 5 **47**

Acknowledgments

First of all, I would like to thank Jean-Philippe Vert who gave me the opportunity to work on this project and supervised me during this semester. I am very grateful for all the time he patiently spent helping me with encouraging comments and fruitful discussions, and for giving me the chance to attend various interesting conferences.

I also acknowledge Armines for financing this work.

Lastly, I am very grateful to the whole team of the computational biology lab for creating such a good atmosphere. Many thanks to Mikhail Zaslavskiy for his insights into mathematics and for putting up with my intensive use of the cluster, to Pierre Mahé, Martial Hue, Franck Rapaport and Caroline Bernard Michel for sharing their graduate student knowledge with me, to Véronique Stoven who greatly enriched my biological knowledge with her discussions and reviews, and to Christian Lajaunie who organized a bicycle trip to remember for the lab.

Introduction

At the end of 2005, the United Nations estimated to 25 million the number of people who died from AIDS since 1981. Although a lot of progress has been made in the field of highly active antiretroviral therapy, the treatment is still very heavy, does not actually cure the patient, has a variable efficiency and is far too expensive for developing countries, which have the highest rates of HIV-infection.

A good alternative to halt the pandemic would be the development of a vaccine against the HIV, which is strongly believed to be the cause of AIDS. However, this is not a trivial issue either since among other problems, high variability of the HIV-I envelope together with HLA-allele epitope specificity make it very difficult to design a large population spectrum vaccine.

On the other hand, recent progress in machine learning and computational learning theory on multitask approaches both prove the dramatic improvement it can bring to prediction accuracy and propose new models for it. At the same time, contributions on HIV epitope prediction show that learning classifiers across HLA alleles improves allele-specific prediction.

The purpose of this work was to analyze current epitope prediction approaches from a multitask learning point of view in order to develop new efficient models. The first chapter presents the biological and mathematical background, the second one analyzes an already existing method and suggests the improvements that logically arise. The last chapter presents the experimental results obtained with these improvements.

Chapter 1

Context

This work intends to propose new efficient ways to design vaccines by using multitask learning approaches to predict epitopes. This section briefly presents the basic concepts of immunology and machine learning that motivate this approach.

1.1 Intelligent vaccine design

1.1.1 Biological background

1.1.1.1 Cytotoxic T Lymphocytes

Our *immune system* is the set of mechanisms that protect our organism against all kinds of infectious agents. These mechanisms are organized in two main branches: innate or non-specific and adaptive or specific immune system.

The idea of a vaccine is to artificially trigger active immunity to a disease, so we focus our interest on specific immune system, *i.e.*, lymphocytes and specific antibodies. Since this work focuses on HIV vaccines, for which the most efficient mechanism seems to be cytotoxic reactions, we will be even more specific and mostly describe this last mechanism.

T-cells are special lymphocytes that play a major role in the cell-mediated response, by contrast with the humoral immunity ruled by the antibodies. They all express the *T-Cell Receptor* (TCR). Cytotoxic T-cells are involved in the destruction of virally infected cells. Since most of them express the CD8 glycoprotein, they are also known as CD8+ T-cells.

A key step in the cell-mediated immune response is the *activation* of the T-cells through the interaction of the TCR with a specific MHC-antigen complex. This is illustrated on figure 1.1. Basically the T-cell "recognizes" an antigen, which is a peptide, *i.e.*, a fragment of protein, that is presented by a cell. Since the only viable T-cells are those who do not recognize the organism-specific peptides, the recognition means that the presenting cell is either not from the organism, like a bacteria, or has been infected by a virus and presents its proteins.

The naive T-cells that recognize an antigen both divide and mature into effector cells. Activated cytotoxic T lymphocytes (CTL) are then able to kill specifically the

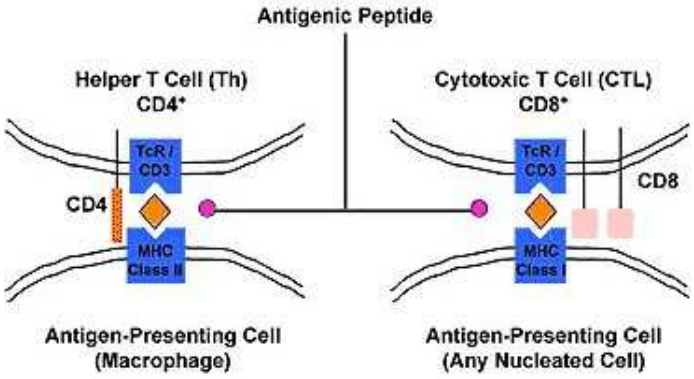

Figure 1.1: T cell – HLA molecule interaction

infected cells they recognize. This is done through the release of effector proteins such as perforin and granzymes, or via the binding of Fas in the target cell membrane by the Fas ligand that leads to activation of caspases. All these molecules induce apoptosis in the target cells. The killing process is illustrated on figure 1.3 and the whole process on figure 1.2.

1.1.1.2 The MHC-epitope binding

As we explained in the previous section, both the activation of the T-cells and their action imply the recognition of a specific MHC-antigen complex. We now describe this complex more precisely.

The MHC is a large gene family involving around 140 genes subdivided into three groups or classes. CD8+ T-cells recognize antigen bound with class I MHC molecules[1].

These molecules are heterodimers, consisting of a single transmembrane polypeptide chain (the α-chain) and a β_2 microglobulin (which is encoded elsewhere, not in the MHC). The schematic representation on figure 1.4 shows the peptide-binding groove formed by the two polymorphic domains α_1, α_2. This part of the molecule, whose shape depends on the corresponding MHC genes allele presents an antigen to the TCR of the T-cells. The binding mechanism is shown on figure 1.5.

As one can see on figure 1.6, the shape of the epitope must be compatible with the shape of the groove. In other words, the potential epitopes can be different if the molecules are different, which is likely to occur if the corresponding MHC gene alleles are different.

The problem is that the MHC harbors much allelic diversity, *i.e.*, one finds many different genotypes for the MHC genes. This implies different phenotypes, which means that one finds a large variety of MHC presenting molecules, each of them being able to complex with different peptides because of its different structure.

[1]In humans, the subset of the MHC genes that code for presenting molecules is also known as HLA for *human leukocyte antigen*.

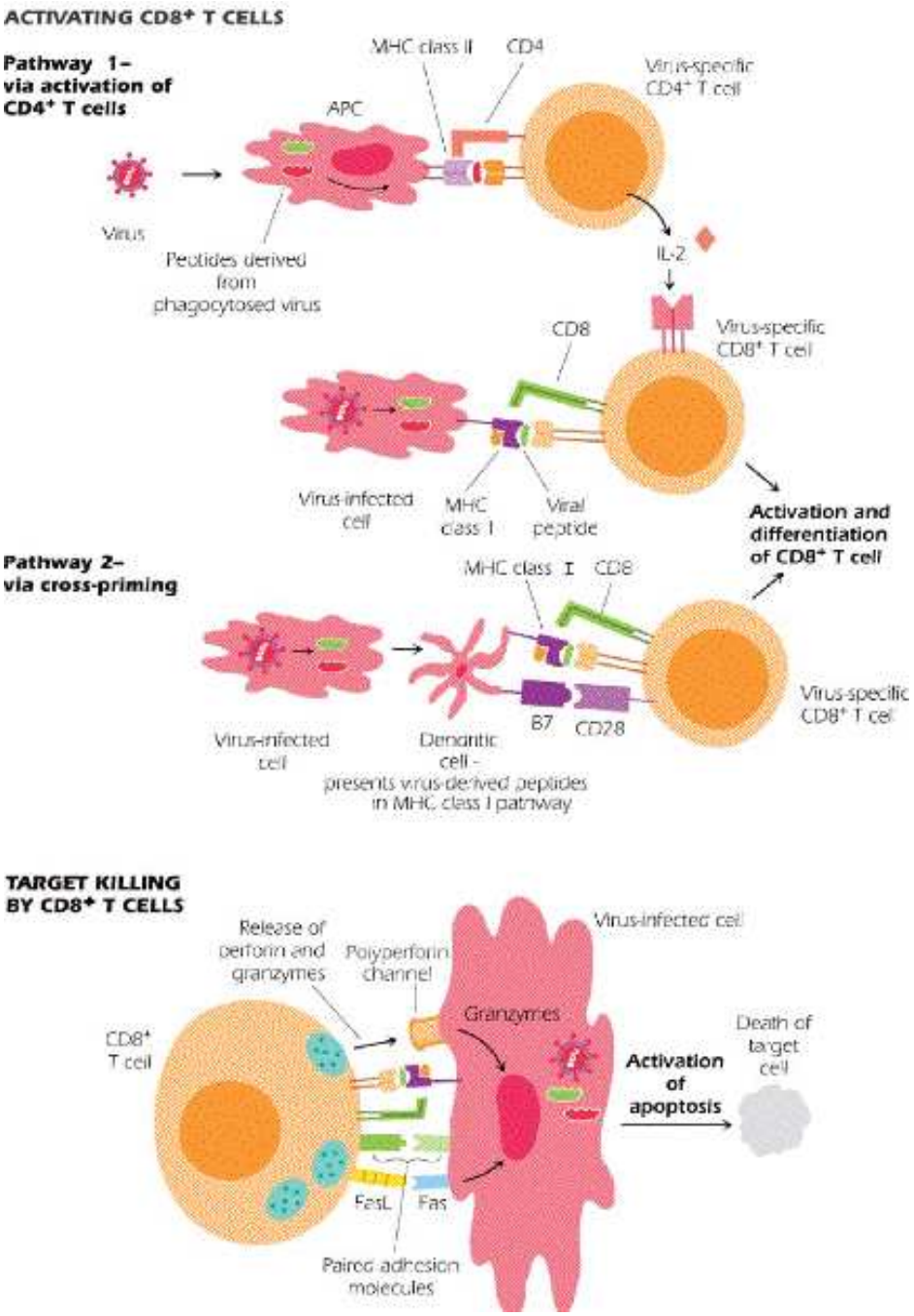

Figure 1.2: CD8+ T-cells mechanism from activation to effective killing

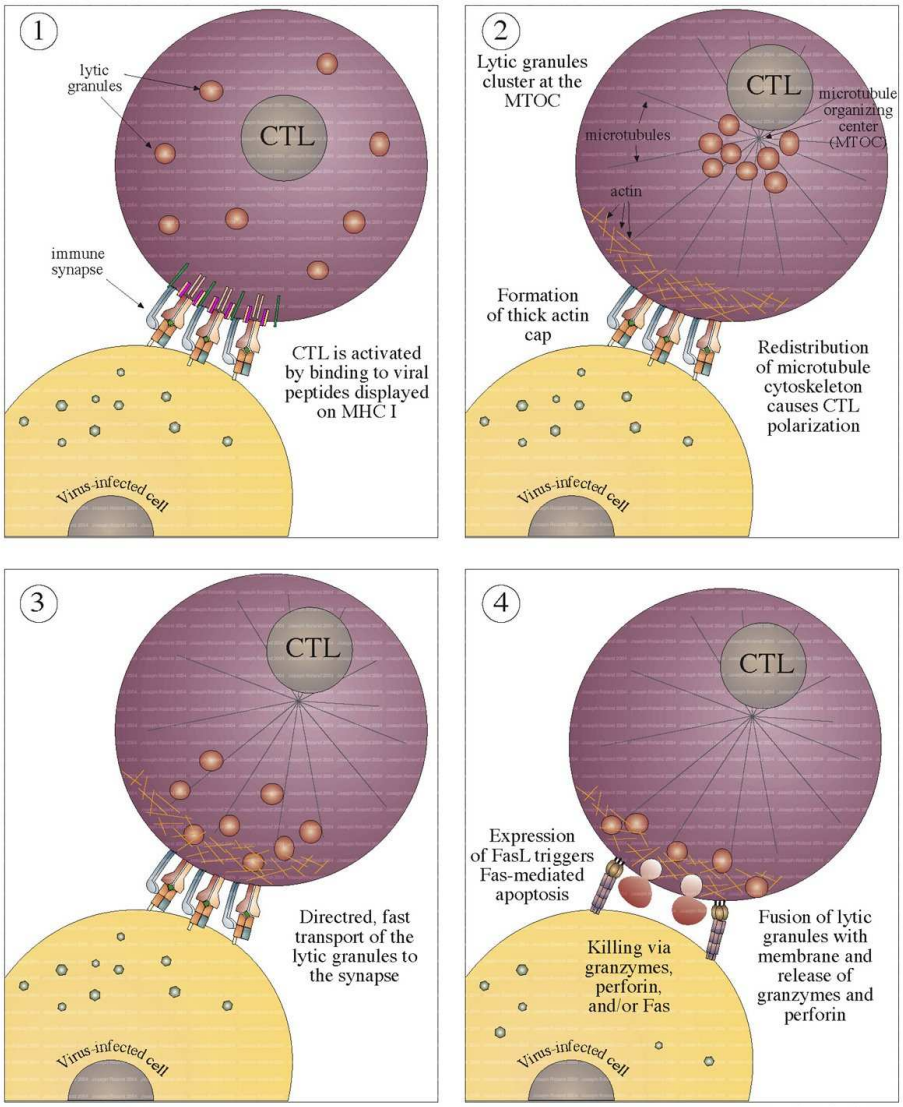

Figure 1.3: Killing by Cytotoxic T Lymphocyte

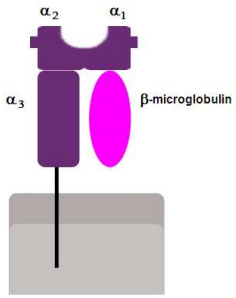

Figure 1.4: Schematic representation of MHC class I molecule

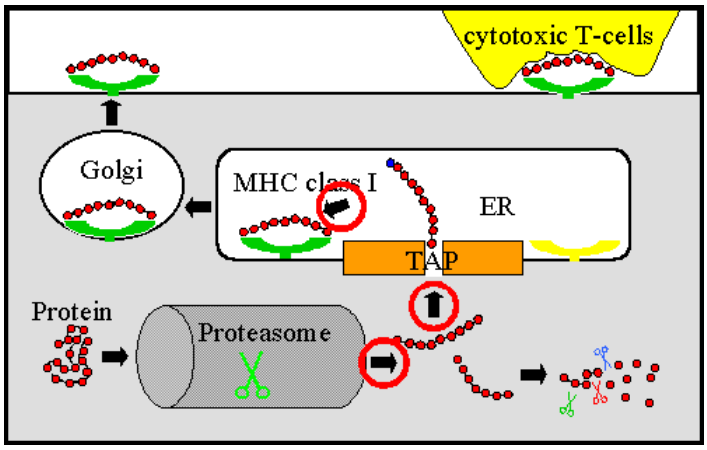

Figure 1.5: Construction of the MHC-antigen complex

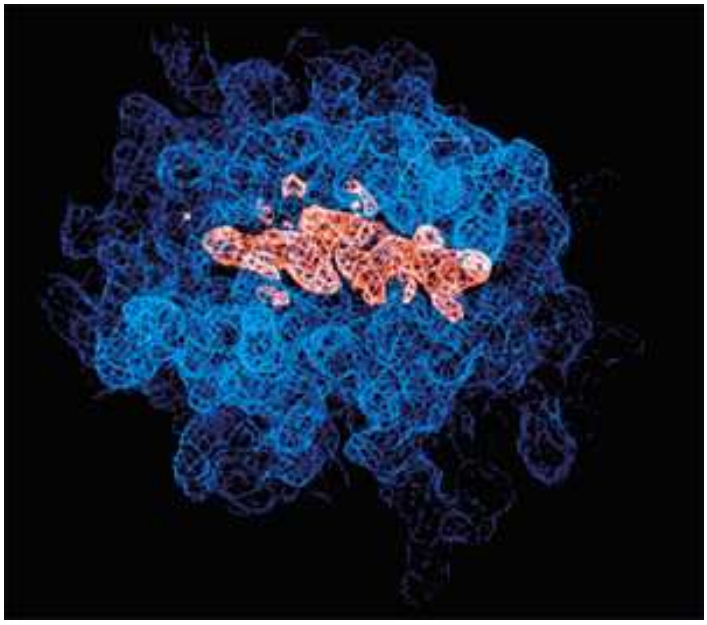

Figure 1.6: Epitope presented in the groove of a MHC class I molecule

1.1.2 The importance of prediction

One goal in intelligent vaccine design, and the goal of this work, is to predict whether or not a peptide will be bound by the MHC-I molecules. If one could predict this property, one just would have to inject the adequate peptide cocktail to a person to make sure the specific immunity against the virus the peptide comes from is triggered.

Of course the practical objective is not to make a perfect predictor, but rather to get a prediction as precise as possible that will finally be experimentally tested, avoiding random test of all the potential epitopes.

1.1.2.1 Machine learning

Since predicting MHC binding is so crucial, several contributions have been trying to find good models to make such a prediction, using classical machine learning methods. [1] sums up (and contributes to) the work done so far: weight-matrix-based methods such as [2, 3, 4] attempt to model the pattern of a sequence that can be bound, [2, 5, 6, 7, 8, 9] use artificial neural network approaches. Other used methods include SVM [2, 10, 11, 9], decision trees [9] and distance learning [12]. Besides, the SYFPEITHI databank [13] proposes its own hand-crafted classifier.

1.1.2.2 Multitask learning

The polymorphism of the MHC-I molecules forces any prediction to be allele-specific: one wants to predict whether or not the peptide will be bound for a specific allele, and the fact that it does is neither sufficient nor necessary for it to be bound for another allele.

On the other hand, as stated before, the MHC harbors much allelic diversity, making it difficult to have much data, if any, for some alleles. Now in order to design a peptide cocktail that works for as many alleles as possible, we still need to be able to make a prediction for these alleles.

The multitask learning approach addresses the problem by proposing models that learn from all the data yet keeping the ability to make allele-specific prediction. The basic idea is to use somehow the fact that even if the problems are not exactly the same, they should share some similarities.

Few work has been done so far to apply multitask learning methods to MHC binding prediction. Among them, [1] that was used as a starting point for this work, proposes to leverage between the HLA alleles and supertypes by adding allele-specific and supertype-specific features to the peptide features. The DistBoost algorithm [12] also allows for some kind of leveraging since it learns a distance among all the alleles.

From a more theoretical point of view *i.e.*, without any application to epitope prediction, various attempts have been made to propose models for and study simultaneous learning of similar problems, including among others computational learning theory results [14, 15, 16], artificial neural network [17, 18, 19], Bayesian [18, 19] or regularization based approaches [20, 21]. For the later, the regularization can be either norm-based or rank-based, following the ideas developed in [22, 23].

Supertype	HLAs
A1	A01, A25, A26, A32, A36, A43, A80
A2	A02, A6802, A69
A3	A03, A11, A31, A33, A6801
A24	A23, A24, A30
B7	B07, B1508, B35, B51, B53, B54, B55, B56, B67, B78
B27	B14, B1503, B1509, B1510, B1518, B27, B38, B39, B48, B73
B44	B18, B37, B40, B41, B44, B45, B49, B50
B58	B1516, B1517, B57, B58
B62	B13, B1501, B1502, B1506, B1512, B1513, B1514, B1519, B1521, B46, B52

Table 1.1: Supertypes for HLA alleles.

1.2 Methods

We start with a short presentation of two methods. The first was an epitope prediction model published in [1], the second a multitask learning formalism published in [20]. They are the basis of our work: we show that the first approach happens to be a special case of the second, and use this fact and this analyze to improve it.

1.2.1 Leveraged logistic regression

This method was specifically proposed for the problem of epitopes prediction, and aims to take advantage of potential common discriminant features among the alleles and allele supertypes, the later being families of similar alleles, see table 1.1.

Each original data point consists of a peptide, a HLA allele with its unique supertype and whether or not the peptide is bound by MHC molecules for this specific allele.

1.2.1.1 Features model

The leveraging in this method is done through the features that are used. Instead of simply using a description of the peptide, the authors add the label of the supertype, of the allele, and conjunctions of the peptide features with this information on the allele or the supertype.

More specifically, for each data x formed by a 9-mer, the allele and supertype for which the peptide is or not an epitope, the features vector $\Phi(x)$ contains the following type of binary elements:

- `AAi(Arg)`: is the ith element of the chain an Arginin?
- `AAi(aromatic)`: is the ith element of the chain an aromatic amino acid?
- `S == A1`: is the data an example for supertype A1?
- `A == A01`: is the data an example for allele A01?
- `AAi(aromatic)` $\wedge$ `(A == A01)`

- `AAi(Arg)` $\wedge$ `(S == A1)`

thus describing not only the peptide itself, but also the relations between the features of the peptide and the class for which it is and example. The table used to assign chemical properties[2] to each amino acid is available at `http://www.geneinfinity.org/rastop/manual/aatable.htm`.

This gives 334 "basic" features for each 9-mer and 334×(1+nb_alleles+nb_supertypes) features in total.

Therefore, if a property is useful for all the alleles/supertype, the corresponding feature will play a role in discrimination, and if it is useful only for one allele/supertype, the feature corresponding to the adequate conjunction `property` $\wedge$ `label` will play a role.

1.2.1.2 Learning model

The model used is a logistic regression. If y is the binary variable to be predicted and $\Phi(x) \in \mathbb{R}^q$ the features vector described above with an additional 1 added in first position, the model is

$$\log \frac{p(y=1|\Phi(x))}{1-p(y=1|\Phi(x))} = w_0 + \sum_{i=1}^{q-1} w_i.\Phi_i(x) = w^T\Phi(x), \tag{1.1}$$

where the w_i are the weights to be learned, which we assume to be iid Gaussian with prior $p(w_i|\sigma^2) = \mathcal{N}(0,\sigma^2)$. Given an iid learning data set of m points (x_i, y_i), $i = 1 \ldots m$, the problem is solved by taking the MAP estimator of $w = (w_0, \ldots, w_{q-1}) \in \mathbb{R}^q$:

$$\begin{aligned}
\arg\max_w p(w|\Phi(x), y) &= \arg\max_w \frac{p(y|\Phi(x), w)p(w)}{p(y)} \\
&= \arg\max_w p(y|\Phi(x), w)p(w) \\
&= \arg\max_w \sum_{i=1}^{m} \log p(y_i|\Phi(x_i), w) + \sum_{j=0}^{q-1} \log p(w_j|\sigma^2).
\end{aligned}$$

Parameter σ^2 is chosen by cross validation on the training data.

Besides, a feature selection is applied after the learning step by setting to 0 the Z weights with smallest absolute value. Z is also chosen by cross validation on the training data.

1.2.2 Multitask kernel

Multitask kernels is a formalism proposed in [20], which can be used in various "kernelizable" methods. It allows to re-write multitask problems as if they were single-task, thus giving access to many well known methods to solve them.

[2] Acidic, acyclic, aliphatic, aromatic, basic, buried, charged, cyclic, hydrophobic, large, medium, negative, neutral, polar, positive, small, surface.

1.2.2.1 The single case problem

Before we explain the re-writing of the multitask learning problem, we briefly review the single case, that will be used as a reference. Keeping the same notations as before, we learn a decision function f by minimizing on $f \in \mathcal{H}$:

$$S(f) = \frac{1}{m}\sum_{j=1}^{m} L(y_j, f(\phi(x_j))) + \gamma \|f\|^2_{\mathcal{H}},$$

where $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS), L a loss function and $\phi(x) \in \mathbb{R}^d$ the features of the single task data, in our case the features of the peptide.

The first term makes the function *fit* to the data while the second one is a *regularization* term and makes f as regular as possible in the sense of $\mathcal{H}$. Parameter γ deals with the compromise between fit and regularity.

We restrict ourselves to the case of linear functions $f(\phi(x)) = v^T\phi(x)$, $v \in \mathbb{R}^d$, for which the functional to be minimized can be re-written

$$S(v) = \frac{1}{m}\sum_{j=1}^{m} L(y_j, v^T\phi(x_j)) + \gamma v^T v. \tag{1.2}$$

The resolution of the problem depends on the loss function L, but the key point is generally the use of the *representer theorem*, according to which the solution at each point is a linear combination of the kernel functions taken between the point and each learning point.

1.2.2.2 The multitask problem

We now turn to the general problem of multitask learning, for which in our case each task will be, for example, a different allele. As stated in introduction, we could learn a function for each task/allele, but as some of them have few data example, we want to take advantage of the obvious similarity between the tasks. Noting $f_l(\phi(x)) = u_l^T\phi(x)$, $u_l \in \mathbb{R}^d$ the discriminative function for task l, the n-task problem is to estimate $u = (u_l : l \in \mathbb{N}_n)$ minimizing the functional

$$R(u) = \frac{1}{m}\sum_{i=1}^{m} L(y_i, u_{t(i)}^T\phi(x_i)) + \gamma J(u), \tag{1.3}$$

where $t(i)$ indicates the task to which data point i belongs and $J(u)$ is a homogeneous quadratic functional of u, that can be written

$$J(u) = u^T E u \ , E \in \mathcal{M}_{dn\times dn}.$$

This last term corresponds to the regularization term, and should represent both regularity of the functions and the relations between the tasks. If we choose for example $J(u) = \frac{1}{n}\sum_{l=1}^{n} \|u_l\|^2$, the problem becomes separable in n minimizations that are equivalent to single task problems.

1.2.2.3 Back to the single case formulation

The form (1.3) can be re-written as a single task learning problem. To obtain this result, we need to use an adequate kernel embedding the multitask aspect.

The idea is to re-write the expression in terms of a common parameter $v \in \mathbb{R}^p$ instead of the specific $u_l \in \mathbb{R}^d$, where $p \geq dn$.

For a $B_l \in \mathcal{M}_{pd}$, we write $u_l = B_l^T v$, which is possible for any u_l if the B_l are of full rank d. Therefore,

$$f_l(x) = u_l^T x = v^T B_l \phi(x).$$

In other words these new functions associate a scalar value to any task/point couple instead of n values for each point, by mapping each $\phi(x)$ to a task-specific space.

The RKHS associated to these $(\phi(x), l) \mapsto v^T B_l \phi(x)$ has the *linear multitask* reproducing kernel $K((\phi(x), l), (\phi(t), q)) = \phi(x)^T B_l^T B_q \phi(t)$.

Indeed,

$$f(\phi(x), l) = f_l(\phi(x)) = v^T B_l \phi(x) = \langle f, K_{\phi(x)l} \rangle \Leftrightarrow f = v \in \mathbb{R}^p \text{ and } K_{\phi(x)l} = B_l \phi(x) \in \mathbb{R}^p, \tag{1.4}$$

hence the kernel $K((\phi(x), l), (\phi(t), q)) = \langle K_{\phi(x)l}, K_{\phi(t)q} \rangle = \phi(x)^T B_l^T B_q \phi(t)$. In other words, we are left with a linear RKHS (and kernel) where the multitask issue has been hidden by a mapping B of $\phi(x) \in \mathbb{R}^d$ to a space of higher dimension p.

The authors establish that using this mapping, the functional (1.3) can be written

$$S(v) = \frac{1}{m} \sum_{i=1}^{m} L(y_i, v^T B_{t(i)} \phi(x_i)) + \gamma v^T v, \tag{1.5}$$

which is the same form as (1.2).

To show why this formulation is equivalent to (1.3), all we need to do is to write the relation between $B = (B_l,\ l \in \mathbb{N}_n)$ and E. Indeed, it is obvious in the formulation of (1.5) that for a chosen B, E is determined since we have $u_l = B_l^T v$ and we want $v^T v = u^T E u$.

The authors of [20] propose a proof that

$$S(v) = R(B^T v) \tag{1.6}$$

if and only if for a chosen B we define E as $E = (B^T B)^{-1}$ or if for a chosen E we take $B = T^T E^{-1}$ where T is a squared root of E.

Since (1.6) casts the multitask problem as a single-task one, the general resolution methods can be applied to solve it.

1.2.2.4 Choosing the mapping

The formalism proposed above allows an explicit formulation of the relation between the tasks through the specification of a matrix B, or equivalently a regularization functional J or a matrix E. Throughout the article, the authors of [20] propose various ways to model this relation. We will use one of them as a starting point for our comparative analysis in section 2.1.

If we choose B_l of the form

$$B_l^T = (I_d\sqrt{1-\lambda}, \underbrace{0, \ldots, 0}_{l-1}, I_d\sqrt{\lambda n}, \underbrace{0, \ldots, 0}_{n-l}) \tag{1.7}$$

with $0 \in \mathcal{M}_{dd}$ then E is defined by

$$E_{lq} = ((B^T B)^{-1})_{lq} = \frac{1}{n}\left(\frac{\delta_{lq}}{\lambda} - \frac{1-\lambda}{n\lambda}\right) I_d,$$

where E_{lq} is the lq-th $d \times d$ block of E. Finally the regularization functional is given by

$$J(u) = \frac{1}{n}\left(\sum_{l=1}^{n} \|u_l\|^2 + \frac{1-\lambda}{\lambda}\sum_{l=1}^{n} \|u_l - \frac{1}{n}\sum_{r=1}^{n} u_r\|^2\right). \tag{1.8}$$

This functional is a clear trade-off between individual regularity and low variance of the tasks, this trade-off being controlled by the parameter $\lambda \in [0, 1]$. If $\lambda = 1$, the tasks are learned independently whereas for a small λ we look for very related tasks.

Chapter 2

Methods

2.1 Comparative analysis

Although they are stated in quite different terms, it is interesting to investigate the relationship between the two approaches for learning many tasks simultaneously. Our analysis actually shows that leveraged logistic regression is a special case of the model proposed in [20].

2.1.1 Logistic regression as a regularization problem

We start with a brief review of how the MAP approach to logistic regression can be reformulated as a regularization problem.

First of all, model (1.1) is equivalent to

$$p(Y = y|\Phi(x)) = \frac{1}{1 + e^{-yw^T\Phi(x)}},$$

and the MAP estimator of w is

$$\begin{aligned}
&\arg\max_w \sum_{i=1}^{m} \log p(y_i|\Phi(x_i), w) + \sum_{j=0}^{d-1} \log p(w_j|\sigma^2) \\
&= \arg\min_w \sum_{i=1}^{m} \log(1 + e^{-y_i w^T \Phi(x_i)}) - \sum_{j=0}^{d-1} \log p(w_j|\sigma^2) \\
&= \arg\min_w \sum_{i=1}^{m} L(y_i, w^T\Phi(x_i)) - \sum_{j=0}^{d-1} \log p(w_j|\sigma^2)
\end{aligned}$$

if we take L to be the logistic loss.

Now since by definition $w_j \sim \mathcal{N}(0, \sigma^2)$, we can re-write

$$\sum_{j=0}^{d-1} \log p(w_j|\sigma^2) = -\sum_{j=0}^{d-1} \frac{1}{2\sigma^2} w_j^2 - \frac{d^2}{2} \log(2\pi\sigma^2),$$

and finally the MAP estimator of w is

$$\arg\min_w \sum_{i=1}^{m} L(y_i, w^T \Phi(x_i)) + \frac{1}{2\sigma^2}\|w\|^2,$$

which is of the same form as (1.2).

2.1.2 Leveraged LR as a special case of multitask learning

Proposition 1. *The leveraged LR model is a special case of the multitask learning formalism of [20].*

Proof. Some difference remains between the two formulations although they have been written as the same regularization problem. The data space is not the same since in the case of leveraged logistic regression we put conjunctions of the peptide features with the class label in our feature space whereas in the multitask kernel formalism our data point in (1.5) were the mapped peptide features $B_l\phi(x)$.

However, $\Phi(x)$ simply contains $\phi(x)$ as the peptide features part and zeros everywhere else except at the conjunction implying the right allele and supertype labels, where the features will be precisely $\phi(x)$. In other words, we can write $\Phi(x)$ as

$$\Phi(x)^T = \big(\underset{l}{\phi(x)}, \quad 0,\ldots,0, \quad \underset{(n+s+1)}{\phi(x)}, \quad 0,\ldots,0, \quad \phi(x), \quad 0,\ldots,0\big),$$

where $0 \in \mathbb{R}^d$, n is the total number of alleles, l and s are the allele and supertype labels respectively. We simply add a 1 to the peptide features in $\phi(x)$: in the first occurrence of $\phi(x)$, it will represent the offset and for the others it will represent the allele and supertype labels.

This $\Phi(x)$ is clearly a special case of the multitask formalism of [20], since

$$\Phi(x_i)^T = B_l\phi(x_i) \Leftrightarrow B_l^T = \big(I_d, \quad 0,\ldots,0, \quad \underset{l}{I_d}, \quad 0,\ldots,0, \quad \underset{(n+s+1)}{I_d}, \quad 0,\ldots,0\big) \triangleq (B_l^{lev})^T, \tag{2.1}$$

I_d being the $\mathcal{M}_{dd}$ identity. □

By definition, a similar decomposition can be applied to the weight vector w, with n alleles and C supertypes:

$$w^T = (w_c, w_{a_1}, \ldots, w_{a_n}, w_{S_1}, \ldots, w_{S_C}).$$

All the w_X are $\mathbb{R}^d$ vectors, w_c is the weight vector corresponding to the common features represented by the first $\phi(x)$, the w_{a_i} are the weight vectors for each allele-specific features and similarly the w_{S_k} are the weight vectors for each supertype-specific features. By allele-specific features, we mean the conjunctions of common features, which in our case are the peptide features, and the allele label.

Therefore, the discriminative function of the leveraged LR model for a given allele l is

$$f_l(x) = w^T.\Phi(x) = w^T B_l^{lev}\phi(x) = (w_c + w_{a_l} + w_{S_{s(l)}})\phi(x),$$

where $s(l)$ is the supertype label of the l-th allele, the sum $w_c + w_{a_l} + w_{S_{s(l)}}$ can be viewed as a task-specific function u_l just like the ones we used in the multitask kernel formalism, and the functional minimized by leveraged LR is exactly (1.3) with $\gamma = \frac{1}{2\sigma^2}$ and $J(u)$ given by the squared norm of w.

Eventually, the leveraged logistic regression model can be seen in the multitask learning formalism as the use of a special application B_l^{lev} that would map the features to a bigger space implying task/features conjunctions, or equivalently, that allows for each task to learn a function implying a task-specific part and a global part.

We now turn to the interpretation of the regularization associated to this B_l^{lev}.

2.1.3 Regularization: the one-level tasks case

We start with the simple case where there is no supertype, only alleles, *i.e.*, where there is only one level of tasks and

$$B_l^T = \big(I_d, \quad 0, \ldots, 0, \quad \underset{l}{I_d}, \quad 0, \ldots, 0\big).$$

This of course is a special case of 1.7 with $\lambda = \frac{1}{n+1}$ but writing the proof for the associated norm in this case is a first step for proving the one in general leveraged LR.

In this case,

$$\|w\|^2 = \|w_c\|^2 + \sum_{l=1}^{n} \|w_{a_l}\|^2 = \|w_c\|^2 + \sum_{l=1}^{n} \|u_l - w_c\|^2,$$

since by definition $u_l = w_c + w_{a_l}$.

Moreover, solving $\frac{\partial \|w\|^2}{\partial w_c} = 0$ shows that for any set of fixed u_l, the optimal w will be such that w_c be the mean of the u_l and 0, *i.e.*,

$$w_c = \frac{1}{n+1} \sum_{l=1}^{n} u_l = \frac{n}{n+1} \bar{u},$$

where $\bar{u}$ is the mean of the u_l.

Using this fact and decomposing the second term with respect to $\bar{u}$, we have for the optimal w:

$$\begin{aligned}
\|w\|^2 &= \left(\frac{n}{n+1}\right)^2 \|\bar{u}\|^2 + \sum_{l=1}^{n} \|u_l - \bar{u} + \bar{u} - w_c\|^2 \\
&= \left(\frac{n}{n+1}\right)^2 \|\bar{u}\|^2 + \sum_{l=1}^{n} \|u_l - \bar{u}\|^2 + n\|\bar{u} - w_c\|^2 + 2(\bar{u} - w_c) \underbrace{\sum_{l=1}^{n} (u_l - \bar{u})}_{=0} \\
&= \left(\frac{n}{n+1}\right)^2 \|\bar{u}\|^2 + \sum_{l=1}^{n} \|u_l - \bar{u}\|^2 + n \left\| \frac{\bar{u}}{n+1} \right\|^2 \\
&= \left(\frac{n}{n+1}\right) \|\bar{u}\|^2 + \sum_{l=1}^{n} \|u_l - \bar{u}\|^2.
\end{aligned}$$

It is interesting to compare this functional to the one obtained in (1.8) adequately re-written

$$\begin{aligned} J(u) &= \frac{1}{n}\left(\sum_{l=1}^{n}\|u_l\|^2 + \frac{1-\lambda}{\lambda}\sum_{l=1}^{n}\|u_l-\bar{u}\|^2\right) \\ &= \frac{1}{n}\left(\sum_{l=1}^{n}\|u_l-\bar{u}\|^2 + n\|\bar{u}\|^2 + \frac{1-\lambda}{\lambda}\sum_{l=1}^{n}\|u_l-\bar{u}\|^2\right) \\ &= \|\bar{u}\|^2 + \frac{1}{n\lambda}\sum_{l=1}^{n}\|u_l-\bar{u}\|^2, \end{aligned}$$

and minimizing $\|w\|^2$ is equivalent to minimizing $J(u)$ with $\lambda = \frac{1}{n+1}$. This is consistent since for this value, $\sqrt{\lambda n} = \sqrt{1-\lambda}$ and (1.8) is equivalent to our simplified model. A direct consequence is that there is one degree of freedom less in the leveraged LR formulation than in the multitask kernel formulation. In the former, the trade-off is implicitly controlled by the number of tasks while in the later it can be freely chosen.

In other words if we consider the space of the parameters we use to control $\gamma J(u)$ and denote $\alpha_1 = \gamma$ and $\alpha_2 = \frac{\gamma}{n\lambda}$, the multitask formulation allows us to choose any point of the space while the leveraged LR restricts the choice to the $\alpha_2 = \frac{n+1}{n}\alpha_1$ curve.

This could not be dramatic since the authors of [20] experimentally noticed that for this functional, one should choose a small λ for many tasks and a larger λ for few tasks. Nevertheless, one should be aware of this implicit restriction.

2.1.4 Regularization: the clustered tasks case

We now consider the full leveraged LR problem, in which each allele is assigned to a specific supertype, and which has already been shown to be equivalent to the model (2.1). This can be thought of like a clustering of the tasks: we add the prior information that some of the tasks are linked in some way.

Proposition 2. *The norm associated to the full logistic LR model is*

$$\|w\|^2 = \left(\frac{\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+1}}{1+\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+1}}\right)\|\hat{u}\|^2 + \sum_{l=1}^{n}\|u_l-\bar{u}_{S_{s(l)}}\|^2 + \sum_{k=1}^{C}\frac{|S_k|}{|S_k|+1}\|\bar{u}_{S_k}-\hat{u}\|^2, \quad (2.2)$$

with

$$\hat{u} = \left(\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+1}\right)^{-1}\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+1}\bar{u}_{S_k}.$$

The proof is postponed to appendix A.

A natural generalization of the one-level regularization functional $J(u)$ would have been the adding of a term penalizing the variance between the clusters

$$\begin{aligned} J_S(u) &= \lambda_1\sum_{l=1}^{n}\|u_l\|^2 + \lambda_2\sum_{l=1}^{n}\|u_l-\bar{u}_{S_{s(l)}}\|^2 + \lambda_3\sum_{k=1}^{C}|S_k|\|\bar{u}_{S_k}-\bar{u}\|^2 \\ &= n\lambda_1\|\bar{u}\|^2 + \lambda_2'\sum_{l=1}^{n}\|u_l-\bar{u}_{S_{s(l)}}\|^2 + \lambda_3'\sum_{k=1}^{C}|S_k|\|\bar{u}_{S_k}-\bar{u}\|^2, \end{aligned}$$

holding a new trade-off between individual regularity, closeness of the tasks and closeness of the clusters pondered by their sizes, this trade-off being controlled by three free parameters.

This form is quite similar to the $\|w\|^2$ of leveraged LR. In particular, the terms penalizing the within cluster variance are identical. There are two differences between these expressions. First, once again $\gamma\|w\|^2$ has only one degree of freedom that controls the overall importance of the function regularity which doesn't allow a control on the trade-off between the three components. If this was the only difference, using leveraged LR would be equivalent to using multitask linear kernel with logistic loss, J_S regularization term and choosing the λ_i parameters on a curve in the 3-dimensional parameter space.

The other difference is that, as mentioned above, $\|w\|^2$ doesn't imply $\|\bar{u}\|^2$ but $\|\hat{u}\|^2$, which means that instead of pondering the distance of a cluster by its size, it ponders it by the quantity $\frac{|S_k|}{|S_k|+1}$. The consequence is that after a certain cluster size, all cluster pondering will be very close thus giving the same importance to the penalization of supertypes with few alleles and the penalization of supertypes with many.

2.2 Controlled leveraged logistic regression model

As underlined in the previous section, the regularization term associated with leveraged logistic regression holds a trade off between individual, intra-cluster and inter-cluster regularization that can't be explicitly controlled.

We now present a simple variation that allows for explicit control of the regularization terms.

2.2.1 First attempt

Similarly to (1.7), we define:

$$(B_l^{gllr})^T \triangleq \big(\alpha_1 I_d, \quad 0,\dots,0, \quad \underset{l}{\alpha_2 I_d}, \quad 0,\dots,0, \quad \underset{(n+s+1)}{I_d}, \quad 0,\dots,0\big), \tag{2.3}$$

where *gllr* stands for generalized leveraged logistic regression. This means that

$$f_l(x) = w^T B_l^{gllr}\phi(x) = (\alpha_1 w_c + \alpha_2 w_{a_l} + w_{S_{s(l)}})\phi(x).$$

Proposition 3. *The norm associated to the multitask kernel induced by* B^{gllr} *is*

$$\begin{aligned}\|w\|^2 =& \frac{1}{1+\alpha_1^2\sum_{k=1}^C \frac{|S_k|}{|S_k|+\alpha_2^2}} \sum_{k=1}^C \frac{|S_k|}{|S_k|+\alpha_2^2}\|\bar{u}_{S_k}\|^2 + \frac{1}{\alpha_2^2}\sum_{l=1}^n \|u_l - \bar{u}_{S_{s(l)}}\|^2 \\ &+ \frac{\alpha_1^2\sum_{k=1}^C \frac{|S_k|}{|S_k|+\alpha_2^2}}{1+\alpha_1^2\sum_{k=1}^C \frac{|S_k|}{|S_k|+\alpha_2^2}} \sum_{k=1}^C \frac{|S_k|}{|S_k|+\alpha_2^2}\|\bar{u}_{S_k} - \hat{u}\|^2,\end{aligned} \tag{2.4}$$

with

$$\hat{u} = \left(\sum_{k=1}^C \frac{|S_k|}{|S_k|+\alpha_2^2}\right)^{-1} \sum_{k=1}^C \frac{|S_k|}{|S_k|+\alpha_2^2}\bar{u}_{S_k}.$$

A proof is proposed in appendix B.

If we set

$$\alpha_1 = \sqrt{\left(\frac{1-\lambda_1}{\lambda_1}\right)\left(\sum_{k=1}^{C}\frac{|S_k|}{\left(\frac{n}{c\lambda_2}\right)+|S_k|}\right)^{-1}}, \qquad \alpha_2 = \sqrt{\frac{n}{c\lambda_2}},$$

then

$$\|w\|^2 = \lambda_1\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\left(\frac{n}{c\lambda_2}\right)}\|\bar{u}_{S_k}\|^2+\frac{c\lambda_2}{n}\sum_{l=1}^{n}\|u_l-\bar{u}_{S_{s(l)}}\|^2+(1-\lambda_1)\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\left(\frac{n}{c\lambda_2}\right)}\|\bar{u}_{S_k}-\hat{u}\|^2,$$

and the two parameters λ_1, λ_2 allow for a (partial) explicit control of the trade off among the three terms.

Unfortunately, the incidence of α_2 in the formulation of $\hat{u}$ doesn't allow to control well the importance of the second term in the minimization of $\|w\|^2$. Indeed, $\alpha_2 \to 0$ allows one to make the second term arbitrarily important, with relative importance of the first term ruled by α_1, but $\alpha_2 \to \infty$ makes $\hat{u} \to \bar{u}$ and $\|w\|^2 \to \sum_{l=1}^{n}\|u_l\|^2$.

Besides, the three terms still aren't precisely what we want to control, *i.e.*, individual norm, within class and between class variance. The next part gives partial solutions to these issues.

2.2.2 Controlled leveraged logistic regression

2.2.2.1 Construction

We now choose a linear multitask kernel with mapping of the form

$$(B_l^{cllr})^T \triangleq \big(\alpha_1 I_d, \underset{l}{0,\ldots,0}, \alpha_2 I_d, 0,\ldots,0, \underset{(n+s+1)}{\alpha_3(s(l))I_d}, 0,\ldots,0\big), \tag{2.5}$$

that is,

$$f_l(x) = w^T B_l^{cllr}\phi(x) = (\alpha_1 w_c + \alpha_2 w_{a_l} + \alpha_3(s(l))w_{S_{s(l)}})\phi(x),$$

where $\alpha_3(k)$ is specific to each supertype k, which is possible since the mapping B_l is task-specific.

Notice that this is not an actual generalization of the B_l mapping corresponding to leveraged LR model anymore, since we can't have $\alpha_1 = \alpha_2 = \alpha_3(k) = 1 \forall k$.

Proposition 4. *The norm associated to the multitask kernel induced by B^{cllr} is such that*

$$\|w\|^2 \propto \frac{1}{n}\sum_{l=1}^{n}\|u_l\|^2+\frac{x+\alpha_1^2 n}{\alpha_2^2}\cdot\frac{1}{n}\sum_{l=1}^{n}\|u_l-\bar{u}_{S_{s(l)}}\|^2+\frac{\alpha_1^2 C}{x+\alpha_2^2}\cdot\frac{1}{C}\sum_{k=1}^{C}|S_k|\|\bar{u}_{S_k}-\bar{u}\|^2. \tag{2.6}$$

where the terms we wanted to be able to control appear explicitly. Since for any α_1, α_2 and any $x \geq 0$, the proportionality constant is positive in u, minimizing $\|w\|^2$ is

equivalent to minimizing this last expression. Proof of proposition 4 is postponed to appendix C.

Finally, if we set

$$\alpha_1 = \sqrt{\frac{x(1-\lambda_1)}{C\lambda_1(1-\lambda_2) - n\lambda_2(1-\lambda_1)}}, \qquad \alpha_2 = \sqrt{\frac{x\lambda_2(n+\lambda_1(C-n))}{C\lambda_1(1-\lambda_2) - n\lambda_2(1-\lambda_1)}},$$
$$\alpha_3(k) = \sqrt{\frac{x}{|S_k|}},$$

with $\lambda_1 \in [0,1]$ and $\lambda_2 \in \left[0, \frac{C\lambda_1}{n+\lambda_1(C-n)}\right]$, the regularizer is

$$\|w\|^2 \propto \frac{1}{n}\sum_{l=1}^{n} \|u_l\|^2 + \frac{1-\lambda_2}{\lambda_2}.\frac{1}{n}\sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \frac{1-\lambda_1}{\lambda_1}.\frac{1}{C}\sum_{k=1}^{C} |S_k|\|\bar{u}_{S_k} - \bar{u}\|^2.$$

λ_1 controls the relative importance of the BSS term in the norm, and λ_2 the relative importance of the WSS term. Choosing (λ_1, λ_2) couples is equivalent to choosing a multitask kernel of the form

$$K((x,l),(t,q)) = x^T t(\alpha_1^2 + \alpha_2^2\delta_{lq} + \alpha_3^2\delta_{s(l)s(q)}). \tag{2.7}$$

In the norm term of the S functional to minimize, x is simply a coefficient, more precisely

$$J(u) = \frac{nx}{(n\tilde{\alpha_1}^2 + \tilde{\alpha_2}^2 + 1)}\left(\frac{1}{n}\sum_{l=1}^{n} \|u_l\|^2 + \frac{1-\lambda_2}{\lambda_2}.\frac{1}{n}\sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \frac{1-\lambda_1}{\lambda_1}.\frac{1}{C}\sum_{k=1}^{C} |S_k|\|\bar{u}_{S_k} - \bar{u}\|^2\right),$$

where $\tilde{\alpha_i} = \alpha_i/\sqrt{x}$. In the fit term, one has

$$\log 1 + e^{-y_i w^T \Phi(x_i)} = \log 1 + e^{-\sqrt{x} y_i \phi(x_i)(\tilde{\alpha_1} w_c + \tilde{\alpha_2} w_{a_{t(i)}} + \tilde{\alpha_3} w_{S_{s(t(i))}})},$$

and x changes the slope of the logistic loss, therefore we choose $x = 1$.

2.2.2.2 Limitation

The only thing that can't be done with this kernel is to give more importance to the BSS than to the WSS. More precisely, for α_1, α_2 to be defined, we need

$$C\lambda_1(1-\lambda_2) - n\lambda_2(1-\lambda_1) \geq 0 \Leftrightarrow \frac{\frac{1-\lambda_2}{\lambda_2}}{\frac{1-\lambda_1}{\lambda_1}} \geq \frac{n}{C}.$$

For $n \gg C$, the WSS term will always have to be much more important than the BSS one. Intuitively, this could not be too problematic, since in this case we want an important penalization for WSS, but the information on clusters is less important. On the other hand, for $n \to C$, *i.e.*, one tends to one task per cluster, the limit curve in the (λ_1, λ_2) space tends to the $\lambda_1 = \lambda_2$ line, that gives equal importance to WSS and BSS.

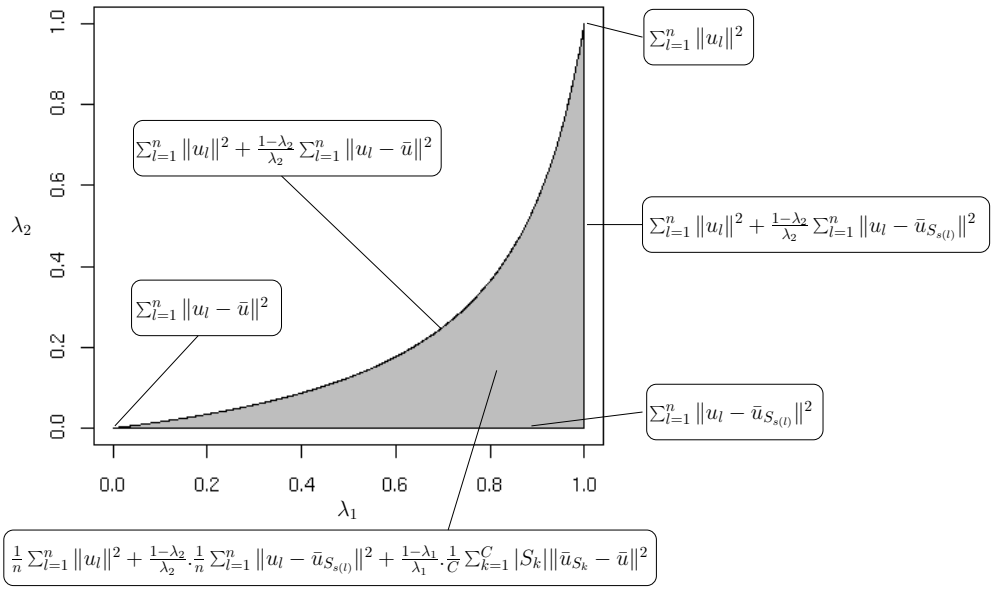

Figure 2.1: Regularizer as a function of the position in the parameter space. The gray area is the "licit" subspace. Each regularizer is given up to a positive constant.

Proposition 5. *The norm*

$$\frac{1}{n}\sum_{l=1}^{n} \|u_l\|^2 + \beta \frac{1}{n}\sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \alpha \frac{1}{C}\sum_{k=1}^{C} |S_k| \|\bar{u}_{S_k} - \bar{u}\|^2$$

can be obtained by a kernels restricted to the form (2.7)*, holding a "simple" natural block mapping of the form* (2.5) *only if* $\beta \geq \alpha \frac{n}{C}$.

A proof is proposed in appendix E. It is actually possible to use a kernel

$$K((x,l),(t,q)) = x^T t(\beta_1 + \beta_2 \delta_{lq} + \beta_3 \delta_{s(l)s(q)}),$$

with $\beta_i \leq 0$, which would hold the same regularizer without the constraints, yet being positive definite under some simple conditions, but the related mapping would be more complex to define, and it would be much more convenient to work in the dual space in this case.

2.2.2.3 Special cases

The choice of the parameters in the (λ_1, λ_2) space is limited by the $\lambda_2 = \frac{C\lambda_1}{n+\lambda_1(C-n)}$ curve, the subspace above the curve being unreachable for $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}^+$.

Under the curve, any point corresponds to a particular kernel with its RKHS norm. Low λ_1-coordinate points give norms with more relative importance for BSS, low λ_2-coordinate points give norms with more relative importance for WSS.

In particular, the points on the limit curve correspond to the kernels of [20] with regularizer (1.8):

$$K((x,l),(t,q)) = x^T t(1-\lambda+\lambda n\delta_{lq}). \tag{2.8}$$

This is also equivalent to having infinite α_1/α_3 and α_2/α_3, which is coherent.

The right limit of the space *i.e.*, the line $\lambda_1 = 1$ generates the kernels

$$K((x,l),(t,q)) = x^T t\left(\frac{1-\lambda_2}{\lambda_2}\delta_{lq} + \frac{1}{|S_{s(l)}|}\delta_{s(l)s(q)}\right), \tag{2.9}$$

whose RKHS are associated to the norms

$$\|w\|^2 \propto \frac{1}{n}\left(\sum_{l=1}^{n}\|u_l\|^2 + \frac{1-\lambda_2}{\lambda_2}\sum_{l=1}^{n}\|u_l - \bar{u}_{S_{s(l)}}\|^2\right), \tag{2.10}$$

the same as (1.8) but using the distance of each function to the mean of its cluster instead of the distance to the global mean. This is also a kernel proposed in [20].

The bottom limit, *i.e.*, the line $\lambda_2 = 0$ gives a regularizer

$$J(u) \propto \sum_{l=1}^{n}\|u_l - \bar{u}_{S_{s(l)}}\|^2. \tag{2.11}$$

When one tends to $\lambda_1 = 0$ on this line, the last two terms have the same importance and sum to $\sum_{l=1}^{n}\|u_l - \bar{u}\|^2$. This is also an extreme case of (1.8), which is coherent since the point is the intersection of the curves $\lambda_2 = \frac{C\lambda_1}{n+\lambda_1(C-n)}$ and $\lambda_2 = 0$.

The two others intersections also give coherent regularizers, *i.e.*, $\sum_{l=1}^{n}\|u_l\|^2$ for the upper right corner and $\sum_{l=1}^{n}\|u_l - \bar{u}_{S_{s(l)}}\|^2$ for the bottom right corner.

All these cases are summarized on figure 2.1.

2.2.2.4 Unclustered tasks

In real data, it can be the case that some tasks belong to clusters and others don't. In particular, this is the case in the data of [1], where we don't know the supertype for some alleles.

In this case, the regularizer is slightly different. It is quite straightforward to see from the formulation in appendix D that

$$J(u) \propto \sum_{l=1}^{n}\|u_l\|^2 + \frac{x+\alpha_1^2 n}{\alpha_2^2}\sum_{l\notin S_{\mathrm{NA}}}\|u_l - \bar{u}_{S_{s(l)}}\|^2 + \frac{\alpha_1^2 n}{x+\alpha_2^2}\left(\sum_{k=1}^{C}|S_k|\|\bar{u}_{S_k} - \bar{u}\|^2 + \sum_{l\in S_{\mathrm{NA}}}\|u_l - \bar{u}\|^2\right), \tag{2.12}$$

where the sum on $l \in S_{\mathrm{NA}}$ can be thought of like the equivalent of the BSS for unclustered tasks, each of them being a cluster itself[1].

[1]However, in terms of mapping, one should keep the same B_l^{cllr} without artificially adding clusters for unassigned tasks, otherwise the regularizer will not be the same.

If one takes

$$\alpha_1 = \sqrt{\frac{x(1-\lambda_1)(\lambda_2(n-|S_{\mathrm{NA}}|)+n(1-\lambda_2))}{n\left((C+|S_{\mathrm{NA}}|)\lambda_1(1-\lambda_2)-(n-|S_{\mathrm{NA}}|)\lambda_2(1-\lambda_1)\right)}},$$
$$\alpha_2 = \sqrt{\frac{x\lambda_2(n-|S_{\mathrm{NA}}|)(n+\lambda_1(C-n+|S_{\mathrm{NA}}|))}{n\left((C+|S_{\mathrm{NA}}|)\lambda_1(1-\lambda_2)-(n-|S_{\mathrm{NA}}|)\lambda_2(1-\lambda_1)\right)}},$$
$$\alpha_3(k) = \sqrt{\frac{x}{|S_k|}},$$

with $\lambda_1 \in [0,1]$ and $\lambda_2 \in \left[0, \frac{(C+|S_{\mathrm{NA}}|)\lambda_1}{(n-|S_{\mathrm{NA}}|)+\lambda_1(C-n+2|S_{\mathrm{NA}}|)}\right]$, the regularizer is

$$J(u) \propto \frac{1}{n}\sum_{l=1}^{n}\|u_l\|^2 + \frac{1-\lambda_2}{\lambda_2}.\frac{1}{n-|S_{\mathrm{NA}}|}\sum_{l\notin S_{\mathrm{NA}}}\|u_l-\bar{u}_{S_{s(l)}}\|^2$$
$$+\frac{1-\lambda_1}{\lambda_1}.\frac{1}{C+|S_{\mathrm{NA}}|}\left(\sum_{k=1}^{C}|S_k|\|\bar{u}_{S_k}-\bar{u}\|^2+\sum_{l\in S_{\mathrm{NA}}}\|u_l-\bar{u}\|^2\right).$$

2.3 Further generalization

The kernel of the form (2.7) can be seen, in a more general framework developed in [21], as the product of two kernels, which is also the kernel of the tensor product of the two RKHS corresponding to the kernels. Two natural generalization are then possible:

- Use a non-linear kernel instead of the dot product for the single-task-part kernel.
- Use an attribute kernel instead of the Dirac kernel for task-specific part of the product.

2.3.1 Non-linear single-task kernels

The dot product on the peptide-specific features proposed by Heckerman defines a kernel for peptides. Any other attribute kernel can be used instead, including all those specifically designed for sequences.

This generalization can also be seen directly in the formalism of [1], as a kernelization of the logistic regression.

2.3.2 Non-Dirac task-specific kernels

In (2.7), $(\alpha_1^2+\alpha_2^2\delta_{lq}+\alpha_3^2\delta_{s(l)s(q)})$ can be thought of as a particular kernel to compare tasks. It is quite poor though, since it doesn't involve any information on the tasks, only the fact that they are the same or not (or, in this case, that they belong to the same cluster or not). If more information is available, one could use it by replacing this Dirac kernel by any other kernel.

For example in our case of epitope prediction, the tasks are alleles. The reason why allele-specific prediction is crucial is that different HLA allele imply different

MHC molecules, for which we try to predict the binding with peptides. The variable part of these molecules is alpha-chains, and it seems natural to use kernels to compare these chains instead of just using their supertype.

The problem would then turn to classification for pairs, for which all the classical classification tools can be used by simply using a tensor product kernel.

2.4 Optimization methods

All the approaches previously presented are solved by minimizing a functional $S(v)$. In practice, this optimization is not trivial even if the functional is convex, since the objective function lives in a high dimensional space $\mathbb{R}^p, p \sim 25000$.

Fast optimization methods include Newton, which converges very fast but implies the resolution of a $p \times p$ system, and conjugate gradient, slower but only implying the gradient of the functional. We present now how these methods were worked out.

2.4.1 Conjugate gradient

The idea of this method is based on a classical gradient descent, in which the descent direction at each step is $-\nabla_v S(v)$, but speed up by ensuring that the descent direction at each step is conjugated with the previous one.

All we need is the gradient of the functional:

$$S(v) = \frac{1}{m}\sum_{i=1}^{m} \log\left(1 + e^{-y_i v^T B_{t(i)} x_i}\right) + \gamma\|v\|^2.$$

We have

$$\begin{aligned}\frac{\partial S(v)}{\partial v_j} &= \frac{1}{m}\sum_{l=1}^{n}\sum_{i\in I_l} \frac{-y_i \left(B_l x_i\right)_j e^{-y_i u_l^T x_i}}{1 + e^{-y_i u_l^T x_i}} + 2\gamma v_j \\ &= \frac{1}{m}\sum_{l=1}^{n}\sum_{i\in I_l} \frac{-y_i \left(B_l x_i\right)_j}{1 + e^{y_i u_l^T x_i}} + 2\gamma v_j,\end{aligned}$$

with

$$\left(B_l x_i\right)_j = \begin{cases} 1 & \text{if } j \in \{1,\dots,d\} \\ & \cup\{ld+1,\dots,(l+1)d\} \\ & \cup\{(n+s(l))d+1,\dots,(n+s(l)+1)d\} \\ 0 & \text{otherwise.}\end{cases}$$

so eventually,

$$\nabla_v S(v) = \frac{1}{m} \begin{pmatrix} \sum_{i=1}^{m} \frac{-y_i x_i}{1+e^{y_i u_{t(i)}^T x_i}} \\ \vdots \\ \sum_{i \in I_l} \frac{-y_i x_i}{1+e^{y_i u_l^T x_i}} \\ \vdots \\ \sum_{l \in S_c} \sum_{i \in I_l} \frac{-y_i x_i}{1+e^{y_i u_l^T x_i}} \end{pmatrix} + 2\gamma v.$$

2.4.2 Newton

The newton step dv is the solution of the system

$$\nabla_v^2 S(v) dv = -\nabla_v S(v).$$

A direct computation would imply the inversion of a $\mathcal{M}_{25000\times 25000}$ matrix, but fortunately, the structure of the Hessian allows for some simplifications that make the computation amenable.

Indeed, the second order partial derivatives of S are

$$\begin{aligned} \frac{\partial^2 S(v)}{\partial v_j \partial v_k} &= \frac{1}{m} \sum_{l=1}^{n} \sum_{i \in I_l} \frac{y_i^2 \left(B_l x_i\right)_j \left(B_l x_i\right)_k e^{y_i u_l^T x_i}}{(1+e^{y_i u_l^T x_i})^2} + 2\gamma\delta_{jk} \\ &= \frac{1}{m} \sum_{l=1}^{n} \sum_{i \in I_l} \frac{\left(B_l x_i\right)_j \left(B_l x_i\right)_k e^{y_i u_l^T x_i}}{(1+e^{y_i u_l^T x_i})^2} + 2\gamma\delta_{jk}, \end{aligned}$$

i.e., then only non-zero terms are the blocks corresponding to the interactions of global terms with anything else (so the first block line and block column are not empty), allele terms with themselves or with their supertype, and supertypes with themselves, and the Hessian matrix in the adequate basis has the form:

$$\nabla_v^2 S(v) = \frac{1}{m} \begin{pmatrix} D_1 & 0 & 0 & t_1 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & D_c & t_c \\ t_1^T & \dots & t_c^T & s \end{pmatrix},$$

where if we note

$$\Sigma(A) = \sum_{i \in I_A} \frac{x_i x_i^T e^{y_i u_{t(i)}^T x_i}}{(1+e^{y_i u_{t_i}^T x_i})^2} \in \mathcal{M}_{dd},$$

and $a_i(S_k)$ to be the i^{th} allele in the cluster S_k, we define

$$\begin{aligned} t_k^T &= \big(\Sigma(a_1(S_k)), \dots, \Sigma(a_{|S_k|}(S_k)), \Sigma(S_k)\big) \\ s &= \Sigma(\text{all}) + 2\gamma I_d \\ D_k &= \begin{pmatrix} \Sigma(a_1(S_k)) & 0 & 0 & \Sigma(a_1(S_k)) \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & \Sigma(a_{|S_k|}(S_k)) & \Sigma(a_{|S_k|}(S_k)) \\ \Sigma(a_1(S_k)) & 0 & \Sigma(a_{|S_k|}(S_k)) & \Sigma(S_k) \end{pmatrix} + 2\gamma I_{|S_k|+1}. \end{aligned}$$

For example if the first cluster contains two alleles a and b and the last one three alleles c, d, e, the Hessian in the adequate basis will be:

$$\nabla_v^2 S(v) = \frac{1}{m} \begin{pmatrix} \Sigma(a) & 0 & \Sigma(a) & 0 & 0 & 0 & 0 & 0 & \Sigma(a) \\ 0 & \Sigma(b) & \Sigma(b) & 0 & 0 & 0 & 0 & 0 & \Sigma(b) \\ \Sigma(a) & \Sigma(b) & \Sigma(S_1) & 0 & 0 & 0 & 0 & 0 & \Sigma(S_1) \\ 0 & 0 & 0 & \ddots & 0 & 0 & 0 & 0 & \vdots \\ 0 & 0 & 0 & 0 & \Sigma(c) & 0 & 0 & \Sigma(c) & \Sigma(c) \\ 0 & 0 & 0 & 0 & 0 & \Sigma(d) & 0 & \Sigma(d) & \Sigma(d) \\ 0 & 0 & 0 & 0 & 0 & 0 & \Sigma(e) & \Sigma(e) & \Sigma(e) \\ 0 & 0 & 0 & 0 & \Sigma(c) & \Sigma(d) & \Sigma(e) & \Sigma(S_c) & \Sigma(S_c) \\ \Sigma(a) & \Sigma(b) & \Sigma(S_1) & \dots & \Sigma(c) & \Sigma(d) & \Sigma(e) & \Sigma(S_c) & \Sigma(\text{all}) \end{pmatrix} + 2\gamma I_p.$$

Of course for gllr and cllr models the Hessian is slightly different, but the sparsity pattern doesn't change. The interest of such a re-ordering of the lines and columns of the Hessian, is that we are left with a system of the form

$$\begin{pmatrix} D & t \\ t^T & s \end{pmatrix} \begin{pmatrix} dv_1 \\ dv_2 \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix},$$

where D is block diagonal, each block having the same "arrow" sparsity structure as the full Hessian. This structure allows for efficient resolution, given by the following Cholesky decomposition, which is a generalization of an example given in [24]:

$$\begin{pmatrix} D^{1/2} & 0 \\ t^T D^{-1/2} & \sqrt{s - t^T D^{-1} t} \end{pmatrix} \begin{pmatrix} D^{1/2} & D^{-1/2} t \\ 0 & \sqrt{s - t^T D^{-1} t} \end{pmatrix} \begin{pmatrix} dv_1 \\ dv_2 \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix},$$

and finally all we need to solve is the system:

$$\begin{cases} \hat{w}_1 &= D^{-1} g_1 \\ dv_2 &= (s - t^T D^{-1} t)^{-1} (g_2 - t^T \hat{w}_1) \\ dv_1 &= \hat{w}_1 - D^{-1} t dv_2 \end{cases},$$

which is much simpler than the original one, and computationally amenable because all we need to invert (and store) is $d \times d$ matrices. An intermediate solution could be to invert directly the arrow blocks of the diagonal instead of using their sparsity pattern, but in our case using this pattern improved speed by a factor 10.

Chapter 3

Experiments

3.1 Data

We use the same data as in [1], *i.e.*, 3200 9-mer with allele, supertype and whether or not the 9-mer is bound by the MHC-I molecules for this allele. We also use the same features $\phi(x)$ as in [1]: a binary vector containing the amino-acid for each position and the chemical properties for each of them, plus an offset.

The classifiers are trained and tested on the same 5 folds as in [1] for comparison purpose.

3.2 Effect of the parameters on the resulting classifiers

We first attempt to verify the effect of controlled regularization on the resulting u_l, *i.e.*, whether or not we are able to penalize independently the individual norms $\|u_l\|^2$, the intra-cluster variance terms $\|u_l - \bar{u}_{S_{s(l)}}\|^2$ and the inter-cluster variance terms $\|\bar{u}_{S_k} - \bar{u}\|^2$. The first line of boxplots on figure 3.1 shows these terms for models trained with an arbitrary $\gamma = 10$ in order to highlight the regularization effect and with arbitrary λ_1, λ_2 parameters indicated above.

We abusively note WSS the $\|u_l - \bar{u}_{S_{s(l)}}\|^2$ values, BSS the $\|\bar{u}_{S_k} - \bar{u}\|^2$ values, TSS the $\|u_l - \bar{u}\|^2$ values and "Uncl. tasks" the same thing for the tasks that do not belong to any cluster.

The effect observed is the one that can be expected from the regularizers of figure 2.1, except that these data are in the partially unclustered case for which some tasks behave differently, as described in equation 2.12.

For a small value of both λ_1 and λ_2, the regularizer is close to $\sum_{l=1}^{n} \|u_l - \bar{u}\|^2$ that is, the individual norm is hardly penalized. As expected, the corresponding boxplot shows high values for individual norms and smaller ones for the other terms. Since we have to take $\lambda_2 < \lambda_1$, the WSS values are smaller than the BSS and unclustered ones. With this regularizer, all the tasks are learned jointly.

Now if we take a larger value of λ_1, we still give a small relative penalization to the individual norms but we also shrink the coefficient of the last term, that is,

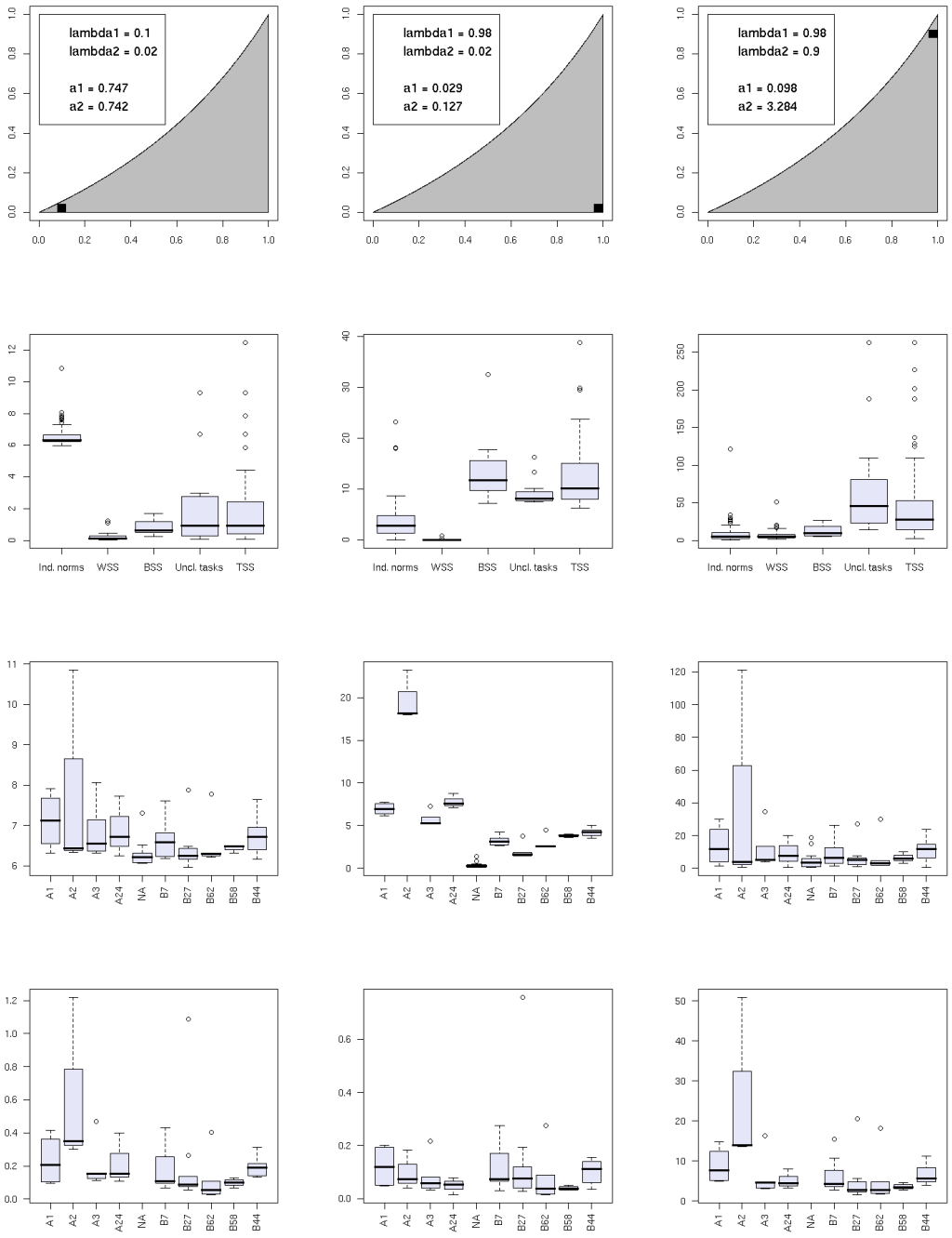

Figure 3.1: Effect of the parameters on the resulting classifier. The first line indicates the point in the parameter space, the second one gives the norms of the various parts of the induced regularizer for each case. The last two lines are the individual norms and WSS by supertype.

$\lambda_2 \backslash \lambda_1$	0.1	0.98
0.9	NA	56.1
0.02	45.3	1474.1

Table 3.1: $\|w\|^2$ for various choices of λ_i and $\gamma = 10$.

BSS and unclustered tasks. As a consequence, we are left with a regularizer close to $\sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2$. The corresponding boxplot shows that the WSS values stay very low and the individual norms quite high while the BSS terms, and therefore the total variance, increase substantially. With this regularizer, the tasks of each cluster are learned jointly but the clusters can be arbitrarily different.

For the last case, the last two terms are almost canceled and the regularizer is close to $\sum_{l=1}^{n} \|u_l\|^2$. The boxplot confirms that using this regularizer decreases dramatically the individual norms of the trained predictors. Like in the first case, we have to take $\lambda_2 < \lambda_1$ hence the lower values for WSS. Using this regularizer is almost equivalent to learning the tasks separately.

The last two lines give the detail of individual norms and distances to the cluster centers for each supertype. It is remarkable that the A2 supertype shows very atypical and high values. Since this is the supertype with the highest cardinal, it could be interesting to check the relevance of this supertype at least for binding prediction, and maybe to subdivide it or uncluster the alleles in this supertype for a better learning.

Finally, table 3.1 shows that for the same γ, the $\|w\|^2$ obtained are very different for different λ_i, which can be explained by more or less difficulty to fit the data for each value of the parametrization.

3.3 Parameter selection and comparison with leveraged LR

We now try to evaluate how using cllr instead of leveraged LR improves the prediction. We use the same data and folds as in [1] and the same criterion: area under the ROC curve (AUC), that is to be minimized.

For each fold, cllr is run on a grid of the (λ_1, λ_2) space and for each point of the parameter space, we keep the best $\gamma \in \{1, \ldots, 9\}.10^{\{-3,-2,-1\}}$ and $Z \in \{10000, 10100, \ldots, 21900, 22000\}$, thus over-fitting all the parameters. The results are presented on figure 3.2.

We also run the leveraged LR model on each fold, over-fitting on γ and Z. Unfortunately, we were not able to reproduce the results of [1] although we used exactly the same model and data. Therefore we use our runs of the model as a comparison. The best AUC obtained for each fold along with the corresponding selected γ and Z and with the improvement brought by cllr on leveraged LR are presented on table 3.2.

A first comment could be that cllr does not improve much the AUC: depending on the fold, we either win few percents or even loose one. Possible explanations are discussed in section 3.4.

It seems that for this data, the best regularization does not really make use of the supertypes structure since it is close to the $\sum_{l=1}^{n} \|u_l\|^2 + \frac{1-\lambda}{\lambda} \sum_{l=1}^{n} \|u_l - \bar{u}\|^2$ with a

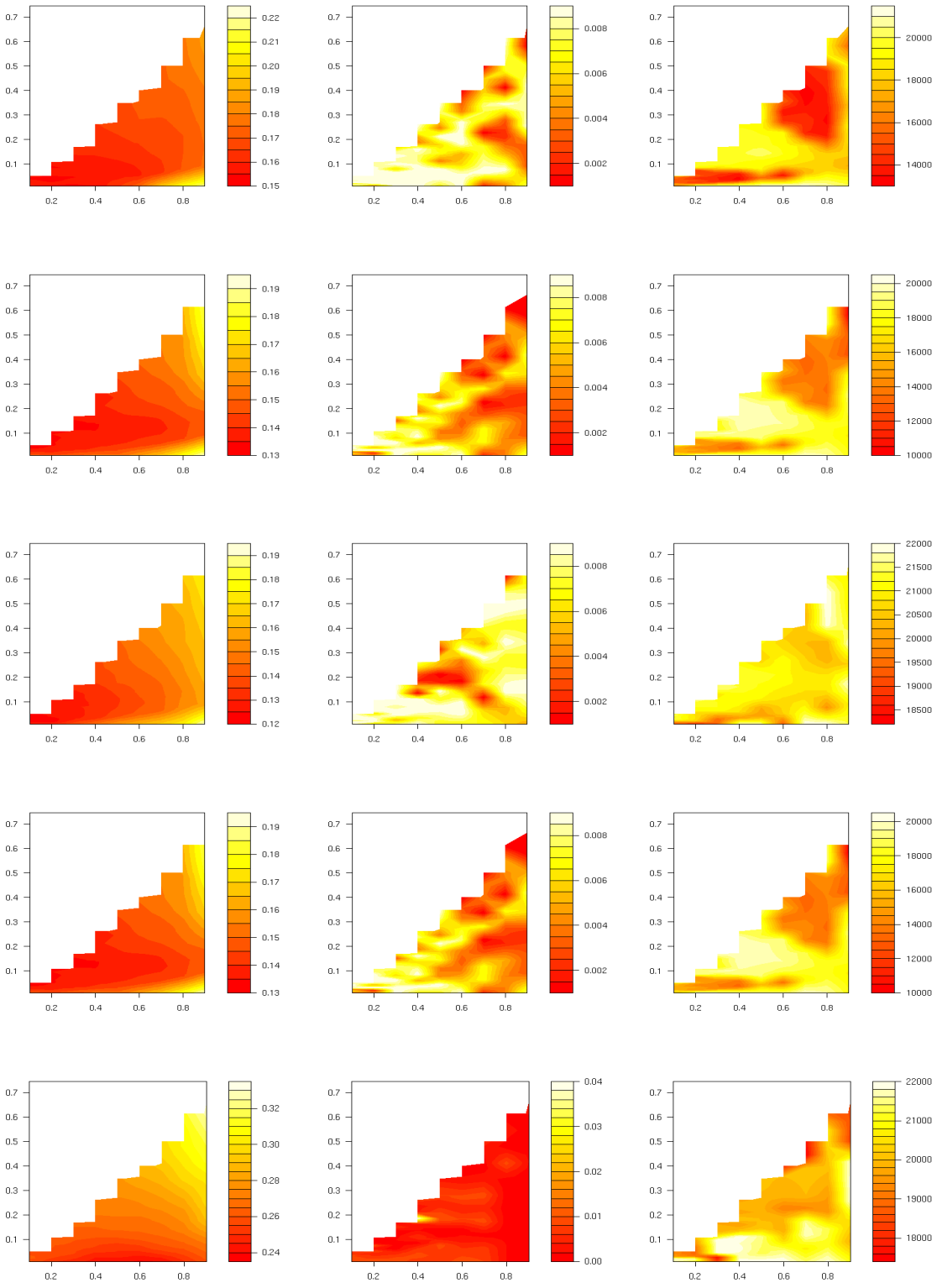

Figure 3.2: From left to right: best AUC, selected γ and selected Z for cllr in the (λ_1, λ_2) space for the 5 folds.

Fold	1	2	3	4	5
AUC	0.16	0.12	0.13	0.18	0.23
γ	0.04	4	0.6	7	0.5
Z	20900	21800	21200	21900	21800
$AUC_{cllr} - AUC_{llr}$	0.01	-0.01	0.01	0.06	-0.01

Table 3.2: Best AUC with corresponding selected parameters for leveraged LR and improvement of cllr on each fold.

small λ, which is one of the models of [20]. However, we observe a kind of saturation on this border of the parameter space, and maybe the supertype structure matters but more weight should be given to the BSS than to the WSS, which was not possible in our implementation, but should be implemented in our future work.

Except for fold 5, whose behavior is quite atypical, all the folds select low γ values with somehow similar, but irregular distributions in the parameter space. The same thing can be said for selected Z although the range is much broader. Interestingly, cllr seems to be more robust to irregular structures without having to regularize more, which could explain why the fold 4 for which leveraged LR has to regularize much is better learned by cllr.

3.4 Discussion

A possible explanation for the low improvement of cllr with respect to leveraged LR could be that the selected γ are to small and therefore the type of regularization is not very important. It can be the case if there are enough data point with respect to the regularity of their distribution. Indeed, numerical experiments on 2/3rd of the first fold data gives a better improvement: $AUC = 0.21$ for the best selected γ (which is more than with all the points) with leveraged LR and 0.19 with cllr, so it could be interesting to test the model on other data sets, maybe in cases where we do not have enough data.

Similarly, one could try the cllr model on data sets with more or less alleles and more or less supertypes. In [20], the selected λ parameter depends on the number of tasks: when there are more tasks, a smaller λ is selected which intuitively means that it is more efficient to learn all the tasks jointly when there are many of them, and individually when there are few. In our case, the improvement could be more important with other numbers of alleles and supertypes.

We could of course use the more general kernel that avoids the constraints of proposition 5

Finally, the model could be improved by using either a different kernel as proposed in section 2.3, or a different loss function, e.g. a hinge loss, replacing the LR model by a SVM one.

Conclusion

This work establishes that the leveraged logistic regression model can be thought of like a special case of multitask learning with kernels, using a specific Dirac kernel for the task part. This point of view allows us to understand better the action of the associated regularization, and to design new Dirac kernels giving better control on this regularization.

Experiments confirm the control given by the developed controlled leveraged logistic regression, although using it does not seem to improve much the prediction on the dataset that was used.

This analyze also highlights new improvement possibilities in terms of other loss function, non-linear individual kernels or non-Dirac task kernels. These possibilities, together with the application of multitask approaches to other similar problems of bioinformatics and chemoinformatics should be investigated during my thesis work.

Bibliography

[1] David Heckerman, Carl Kadie, and Jennifer Listgarten. Leveraging information across HLA alleles/supertypes improves HLA-specific epitope prediction, 2006.

[2] Manoj Bhasin and G. P S Raghava. Prediction of CTL epitopes using QM, SVM and ANN techniques. *Vaccine*, 22(23-24):3195–3204, Aug 2004.

[3] Pedro A Reche, John-Paul Glutting, Hong Zhang, and Ellis L Reinherz. Enhancement to the RANKPEP resource for the prediction of peptide binding to MHC molecules using profiles. *Immunogenetics*, 56(6):405–419, Sep 2004.

[4] Hai-Long Dong and Yan-Fang Sui. Prediction of HLA-A2-restricted CTL epitope specific to HCC by SYFPEITHI combined with polynomial method. *World J Gastroenterol*, 11(2):208–211, Jan 2005.

[5] S. Buus, S. L. Lauemøller, P. Worning, C. Kesmir, T. Frimurer, S. Corbet, A. Fomsgaard, J. Hilden, A. Holm, and S. Brunak. Sensitive quantitative predictions of peptide-MHC binding by a 'query by committee' artificial neural network approach. *Tissue Antigens*, 62(5):378–384, Nov 2003.

[6] Mette Voldby Larsen, Claus Lundegaard, Kasper Lamberth, Søren Buus, Søren Brunak, Ole Lund, and Morten Nielsen. An integrative approach to CTL epitope prediction: a combined algorithm integrating MHC class I binding, TAP transport efficiency, and proteasomal cleavage predictions. *Eur J Immunol*, 35(8):2295–2303, Aug 2005.

[7] M. Milik, D. Sauer, A. P. Brunmark, L. Yuan, A. Vitiello, M. R. Jackson, P. A. Peterson, J. Skolnick, and C. A. Glass. Application of an artificial neural network to predict specific class I MHC binding peptide sequences. *Nat Biotechnol*, 16(8):753–756, Aug 1998.

[8] Morten Nielsen, Claus Lundegaard, Peder Worning, Sanne Lise Lauemøller, Kasper Lamberth, Søren Buus, Søren Brunak, and Ole Lund. Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Sci*, 12(5):1007–1017, May 2003.

[9] Yingdong Zhao, Clemencia Pinilla, Danila Valmori, Roland Martin, and Richard Simon. Application of support vector machines for T-cell epitopes prediction. *Bioinformatics*, 19(15):1978–1984, Oct 2003.

[10] Manoj Bhasin and G. P S Raghava. SVM based method for predicting HLA-DRB1*0401 binding peptides in an antigen sequence. *Bioinformatics*, 20(3):421–423, Feb 2004.

[11] Pierre Dönnes and Arne Elofsson. Prediction of MHC class I binding peptides, using SVMHC. *BMC Bioinformatics*, 3:25, Sep 2002.

[12] Chen Yanover and Tomer Hertz. Predicting protein-peptide binding affinity by learning peptide-peptide distance functions. In *RECOMB*, pages 456–471, 2005.

[13] H. Rammensee, J. Bachmann, N. P. Emmerich, O. A. Bachor, and S. Stevanović. Syfpeithi: database for mhc ligands and peptide motifs. *Immunogenetics*, 50(3-4):213–219, Nov 1999.

[14] Jonathan Baxter. A bayesian/information theoretic model of bias learning. In *COLT '96: Proceedings of the ninth annual conference on Computational learning theory*, pages 77–88, New York, NY, USA, 1996. ACM Press.

[15] Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

[16] S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning, 2003.

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

[18] Tom Heskes. Empirical bayes for learning to learn. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 367–374, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[19] Bart Bakker and Tom Heskes. Task clustering and gating for bayesian multitask learning. *J. Mach. Learn. Res.*, 4:83–99, 2003.

[20] Theodoros Evgeniou, Charles A. Michelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:616–637, 2005.

[21] Jean-Philippe Vert, Francis Bach, and Theodoros Evgeniou. Low-rank matrix factorization with attributes, 2006.

[22] N. Srebro and T. Jaakkola. Weighted low rank approximation, 2003.

[23] Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *COLT*, pages 545–560, 2005.

[24] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

Appendix A

Proof of the leveraged LR norm

In this case, the leveraged LR regularization functional is

$$\|w\|^2 = \|w_c\|^2 + \sum_{l=1}^{n} \|w_{a_l}\|^2 + \sum_{k=1}^{C} \|w_{S_k}\|^2.$$

Similarly to the previous case, we know that $u_l = w_c + w_{a_l} + w_{S_{s(l)}}$. Besides if we note $\tilde{w}_{S_k} = w_c + w_{S_k}$, the optimal w will be such that for any $u_l, l = 1 \dots n$ and w_c fixed,

$$\forall k, \tilde{w}_{S_k} = \frac{1}{|S_k| + 1} \left(w_c + \sum_{l \in S_k} u_l \right), \tag{A.1}$$

and for any $\tilde{w}_{S_k}$ fixed,

$$w_c = \frac{1}{c+1} \sum_{k=1}^{C} \tilde{w}_{S_k}, \tag{A.2}$$

i.e., the $\tilde{w}_{S_k}$ should be the mean of the vectors of the cluster and w_c, and w_c should be the mean of the $\tilde{w}_{S_k}$ and 0.

The second term can be written

$$\begin{aligned} \sum_{l=1}^{n} \|w_{a_l}\|^2 &= \sum_{k=1}^{C} \sum_{l \in S_k} \|u_l - \tilde{w}_{S_k}\|^2 \\ &= \sum_{k=1}^{C} \left(\sum_{l \in S_k} \|u_l - \bar{u}_{S_k}\|^2 + |S_k| \|\bar{u}_{S_k} - \tilde{w}_{S_k}\|^2 \right), \end{aligned}$$

where $\bar{u}_{S_k}$ is the mean of the $u_l \in S_k$. From (A.1), we deduce that $\bar{u}_{S_k} - \tilde{w}_{S_k} = \frac{1}{|S_k|+1}(\bar{u}_{S_k} - w_c)$, so

$$\begin{aligned} \sum_{l=1}^{n} \|w_{a_l}\|^2 &= \sum_{k=1}^{C} \left(\sum_{l \in S_k} \|u_l - \bar{u}_{S_k}\|^2 + \frac{|S_k|}{(|S_k|+1)^2} \|\bar{u}_{S_k} - w_c\|^2 \right) \\ &= \sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \sum_{k=1}^{C} \frac{|S_k|}{(|S_k|+1)^2} \|\bar{u}_{S_k} - w_c\|^2. \end{aligned}$$

If we notice that (A.1) implies that $\tilde{w}_{S_k} = \frac{|S_k|}{|S_k|+1}\bar{u}_{S_k} + \frac{1}{|S_k|+1}w_c$, the third term can also be reformulated:

$$\begin{aligned}\sum_{k=1}^{C}\|w_{S_k}\|^2 &= \sum_{k=1}^{C}\|\tilde{w}_{S_k} - w_c\|^2 \\ &= \sum_{k=1}^{C}\left\|\frac{|S_k|}{|S_k|+1}\bar{u}_{S_k} + \frac{1}{|S_k|+1}w_c - w_c\right\|^2 \\ &= \sum_{k=1}^{C}\left(\frac{|S_k|}{|S_k|+1}\right)^2\|\bar{u}_{S_k} - w_c\|^2.\end{aligned}$$

Replacing these terms in the decomposition, we have

$$\begin{aligned}\|w\|^2 =&\|w_c\|^2 + \sum_{l=1}^{n}\|u_l - \bar{u}_{S_{s(l)}}\|^2 + \sum_{k=1}^{C}\frac{|S_k|}{(|S_k|+1)^2}\|\bar{u}_{S_k} - w_c\|^2 \\ &+ \sum_{k=1}^{C}\left(\frac{|S_k|}{|S_k|+1}\right)^2\|\bar{u}_{S_k} - w_c\|^2 \\ =&\|w_c\|^2 + \sum_{l=1}^{n}\|u_l - \bar{u}_{S_{s(l)}}\|^2 + \sum_{k=1}^{C}\frac{|S_k|}{|S_k|+1}\|\bar{u}_{S_k} - w_c\|^2.\end{aligned}$$

At this point intuitively we would like to decompose the last term with respect to the mean $\bar{u}$ of the cluster centers pondered by the cardinals of the clusters[1] to make appear a term penalizing the variance between the clusters, but we can't because the coefficient in front of the norm is $\frac{|S_k|}{|S_k|+1}$ instead of $|S_k|$, and because anyway we couldn't rewrite the first term like $\alpha\|\bar{u}\|^2$. Indeed, injecting (A.1) into (A.2) shows that

$$w_c = \left(1 + \sum_{k=1}^{C}\frac{|S_k|}{|S_k|+1}\right)^{-1}\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+1}\bar{u}_{S_k}, \tag{A.3}$$

so w_c is not proportional to $\bar{u}$.

The quantity that appears naturally is

$$\hat{u} = \left(\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+1}\right)^{-1}\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+1}\bar{u}_{S_k},$$

i.e., the mean of the cluster centers pondered by the quantities $\frac{|S_k|}{|S_k|+1}$. Using it, we can write

$$\|w\|^2 = \|w_c\|^2 + \sum_{l=1}^{n}\|u_l - \bar{u}_{S_{s(l)}}\|^2 + \sum_{k=1}^{C}\frac{|S_k|}{|S_k|+1}\|\bar{u}_{S_k} - \hat{u}\|^2 + \|\hat{u} - w_c\|^2\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+1}.$$

[1] Which, by the partial barycenter theorem, is also the mean of all the u_l.

(A.3) is equivalent to

$$w_c = \left(\frac{\sum_{k=1}^{C} \frac{|S_k|}{|S_k|+1}}{1 + \sum_{k=1}^{C} \frac{|S_k|}{|S_k|+1}} \right) \hat{u},$$

and finally we have

$$\begin{aligned}
\|w\|^2 &= \left(\frac{\sum_{k=1}^{C} \frac{|S_k|}{|S_k|+1}}{1 + \sum_{k=1}^{C} \frac{|S_k|}{|S_k|+1}} \right)^2 \|\hat{u}\|^2 + \sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \sum_{k=1}^{C} \frac{|S_k|}{|S_k|+1} \|\bar{u}_{S_k} - \hat{u}\|^2 \\
&\quad + \frac{\sum_{k=1}^{C} \frac{|S_k|}{|S_k|+1}}{\left(1 + \sum_{k=1}^{C} \frac{|S_k|}{|S_k|+1}\right)^2} \|\hat{u}\|^2 \\
&= \left(\frac{\sum_{k=1}^{C} \frac{|S_k|}{|S_k|+1}}{1 + \sum_{k=1}^{C} \frac{|S_k|}{|S_k|+1}} \right) \|\hat{u}\|^2 + \sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \sum_{k=1}^{C} \frac{|S_k|}{|S_k|+1} \|\bar{u}_{S_k} - \hat{u}\|^2.
\end{aligned}$$

□

Appendix B

Proof of the gllr norm

Following the same idea as for the proof of 2, we define $\tilde{w}_{S_{s(l)}} = \alpha_1 w_c + w_{S_{s(l)}}$ and we can show that the optimal w will be such that for any $u_l, l = 1 \dots n$ and w_c fixed,

$$\forall k, \tilde{w}_{S_k} = \frac{\alpha_2^2}{\alpha_2^2 + |S_k|} \left(\alpha_1 w_c + \frac{|S_k|}{\alpha_2^2} \bar{u}_{S_k} \right), \tag{B.1}$$

and for any $\tilde{w}_{S_k}$ fixed,

$$w_c = \frac{\alpha_1}{\alpha_1^2 c + 1} \sum_{k=1}^{C} \tilde{w}_{S_k}. \tag{B.2}$$

The overall regularization is still

$$\|w\|^2 = \|w_c\|^2 + \sum_{l=1}^{n} \|w_{a_l}\|^2 + \sum_{k=1}^{C} \|w_{S_k}\|^2.$$

We decompose the second term:

$$\begin{aligned} \sum_{l=1}^{n} \|w_{a_l}\|^2 &= \sum_{k=1}^{C} \sum_{l \in S_k} \left\| \frac{u_l - \tilde{w}_{S_k}}{\alpha_2} \right\|^2 \\ &= \frac{1}{\alpha_2^2} \sum_{k=1}^{C} \left(\sum_{l \in S_k} \|u_l - \bar{u}_{S_k}\|^2 + |S_k| \|\bar{u}_{S_k} - \tilde{w}_{S_k}\|^2 \right). \end{aligned}$$

From (B.1), we deduce that $\bar{u}_{S_k} - \tilde{w}_{S_k} = \frac{\alpha_2^2}{|S_k| + \alpha_2^2} (\bar{u}_{S_k} - \alpha_1 w_c)$, so

$$\begin{aligned} \sum_{l=1}^{n} \|w_{a_l}\|^2 &= \frac{1}{\alpha_2^2} \sum_{k=1}^{C} \left(\sum_{l \in S_k} \|u_l - \bar{u}_{S_k}\|^2 + \frac{\alpha_2^4 |S_k|}{(|S_k| + \alpha_2^2)^2} \|\bar{u}_{S_k} - \alpha_1 w_c\|^2 \right) \\ &= \frac{1}{\alpha_2^2} \sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \sum_{k=1}^{C} \frac{\alpha_2^2 |S_k|}{(|S_k| + \alpha_2^2)^2} \|\bar{u}_{S_k} - \alpha_1 w_c\|^2. \end{aligned}$$

If we notice that (B.1) implies that $\tilde{w}_{S_k} = \frac{|S_k|}{|S_k|+\alpha_2^2}\bar{u}_{S_k} + \frac{\alpha_1\alpha_2^2}{|S_k|+\alpha_2^2}w_c$, the third term can also be reformulated:

$$\begin{aligned}\sum_{k=1}^{C}\|w_{S_k}\|^2 &= \sum_{k=1}^{C}\|\tilde{w}_{S_k} - \alpha_1 w_c\|^2 \\ &= \sum_{k=1}^{C}\left\|\frac{|S_k|}{|S_k|+\alpha_2^2}\bar{u}_{S_k} - \frac{\alpha_1|S_k|}{|S_k|+\alpha_2^2}w_c\right\|^2 \\ &= \sum_{k=1}^{C}\left(\frac{|S_k|}{|S_k|+\alpha_2^2}\right)^2\|\bar{u}_{S_k} - \alpha_1 w_c\|^2.\end{aligned}$$

Replacing these terms in the decomposition, we have

$$\begin{aligned}\|w\|^2 =&\|w_c\|^2 + \frac{1}{\alpha_2^2}\sum_{l=1}^{n}\|u_l - \bar{u}_{S_{s(l)}}\|^2 + \sum_{k=1}^{C}\frac{\alpha_2^2|S_k|}{(|S_k|+\alpha_2^2)^2}\|\bar{u}_{S_k} - \alpha_1 w_c\|^2 \\ &+ \sum_{k=1}^{C}\left(\frac{|S_k|}{|S_k|+\alpha_2^2}\right)^2\|\bar{u}_{S_k} - \alpha_1 w_c\|^2 \\ =&\|w_c\|^2 + \frac{1}{\alpha_2^2}\sum_{l=1}^{n}\|u_l - \bar{u}_{S_{s(l)}}\|^2 + \sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}\|\bar{u}_{S_k} - \alpha_1 w_c\|^2.\end{aligned}$$

We now use the same trick as in the leveraged logistic regression model and define:

$$\hat{u} = \left(\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}\right)^{-1}\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}\bar{u}_{S_k},$$

so

$$w_c = \left(\frac{\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}}{1+\alpha_1^2\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}}\right)\alpha_1\hat{u},$$

and

$$\begin{aligned}\|w\|^2 =& \left[\left(\frac{\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}}{1+\alpha_1^2\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}}\right)\alpha_1\right]^2\|\hat{u}\|^2 + \frac{1}{\alpha_2^2}\sum_{l=1}^{n}\|u_l - \bar{u}_{S_{s(l)}}\|^2 \\ &+ \sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}\|\bar{u}_{S_k} - \hat{u}\|^2 + \|\hat{u} - \alpha_1 w_c\|^2\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2} \\ =&\frac{\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}}{1+\alpha_1^2\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}}\|\hat{u}\|^2 + \frac{1}{\alpha_2^2}\sum_{l=1}^{n}\|u_l - \bar{u}_{S_{s(l)}}\|^2 + \sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}\|\bar{u}_{S_k} - \hat{u}\|^2 \\ =&\frac{1}{1+\alpha_1^2\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}}\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}\|\bar{u}_{S_k}\|^2 + \frac{1}{\alpha_2^2}\sum_{l=1}^{n}\|u_l - \bar{u}_{S_{s(l)}}\|^2 \\ &+ \frac{\alpha_1^2\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}}{1+\alpha_1^2\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}}\sum_{k=1}^{C}\frac{|S_k|}{|S_k|+\alpha_2^2}\|\bar{u}_{S_k} - \hat{u}\|^2.\end{aligned}$$

□

Appendix C

Proof of the cllr norm

Calculation similar to those of the previous attempts leads to

$$\forall k, \tilde{w}_{S_k} = \frac{\alpha_2^2 \alpha_3^2(k)}{\alpha_2^2 + \alpha_3(k)|S_k|)} \left(\frac{\alpha_1}{\alpha_3^2(k)} w_c + \frac{|S_k|}{\alpha_2^2} \bar{u}_{S_k} \right), \tag{C.1}$$

for any $u_l, l = 1 \dots n$ and w_c fixed, and for any $\tilde{w}_{S_k}$ fixed,

$$w_c = \left(1 + \sum_{k=1}^{C} \frac{\alpha_1^2}{\alpha_3^2(k)} \right)^{-1} \left(\alpha_1 \sum_{k=1}^{C} \frac{\tilde{w}_{S_k}}{\alpha_3^2(k)} \right), \tag{C.2}$$

where $\tilde{w}_{S_{s(l)}} = \alpha_1 w_c + \alpha_3(s(l)) w_{S_{s(l)}}$.

The regularization term can be written

$$\|w\|^2 = \|w_c\|^2 + \frac{1}{\alpha_2^2} \sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \sum_{k=1}^{C} \frac{|S_k|}{\alpha_2^2 + \alpha_3^2(k)|S_k|} \|\bar{u}_{S_k} - \alpha_1 w_c\|^2,$$

and combining (C.1) and (C.2), we obtain

$$w_c = \frac{\alpha_1}{1 + \alpha_1^2 \left(\sum_{k=1}^{C} \frac{1}{\alpha_3^2} - \sum_{k=1}^{C} \frac{\alpha_2^2}{\alpha_3^2(\alpha_2^2 + \alpha_3^2|S_k|)} \right)} \sum_{k=1}^{C} \frac{|S_k| \bar{u}_{S_k}}{\alpha_2^2 + \alpha_3^2(k)|S_k|}. \tag{C.3}$$

If α_3 was a constant in k, it would be useless for control, which is why we didn't use it in the previous model. Now if we enforce a certain relation between $\alpha_3(k)$ and k, such that $\alpha_3^2(k)|S_k|$ be a constant in k x , we get the terms we were looking for.

This is equivalent to fixing x and choosing

$$\forall k, \alpha_3^2(k) = \frac{x}{|S_k|}. \tag{C.4}$$

Then

$$w_c = \frac{\alpha_1 n}{\alpha_2^2 + x + \alpha_1^2 x \sum_{k=1}^{C} \alpha_3^{-2}} \bar{u},$$

where $\bar{u} = \sum_{k=1}^{C} |S_k| \bar{u}_{S_k}$ is the overall barycenter of the tasks. Summing equation (C.4) on k, we get

$$\sum_{k=1}^{C} \alpha_3^{-2} = \frac{n}{x},$$

which injected in (C.3) gives

$$w_c = \frac{\alpha_1 n}{x + \alpha_1^2 n + \alpha_2^2} \bar{u}, \tag{C.5}$$

hence the regularizer

$$\begin{aligned}
\|w\|^2 =& \frac{n}{x + \alpha_1^2 n + \alpha_2^2} \|\bar{u}\|^2 + \frac{1}{\alpha_2^2} \sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \frac{1}{x + \alpha_2^2} \sum_{k=1}^{C} |S_k| \|\bar{u}_{S_k} - \bar{u}\|^2 \\
=& \frac{1}{x + \alpha_1^2 n + \alpha_2^2} \sum_{k=1}^{C} |S_k| \|\bar{u}_{S_k}\|^2 + \frac{1}{\alpha_2^2} \sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 \\
&+ \frac{\alpha_1^2 n}{(x + \alpha_2^2)(x + \alpha_1^2 n + \alpha_2^2)} \sum_{k=1}^{C} |S_k| \|\bar{u}_{S_k} - \bar{u}\|^2 \\
=& \frac{1}{x + \alpha_1^2 n + \alpha_2^2} \sum_{l=1}^{n} \|u_l\|^2 + \frac{x + \alpha_1^2 n}{\alpha_2^2 (x + \alpha_1^2 n + \alpha_2^2)} \sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 \\
&+ \frac{\alpha_1^2 C}{(x + \alpha_2^2)(x + \alpha_1^2 n + \alpha_2^2)} \cdot \frac{n}{C} \sum_{k=1}^{C} |S_k| \|\bar{u}_{S_k} - \bar{u}\|^2 \\
\propto& \frac{1}{n} \sum_{l=1}^{n} \|u_l\|^2 + \frac{x + \alpha_1^2 n}{\alpha_2^2} \cdot \frac{1}{n} \sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \frac{\alpha_1^2 C}{x + \alpha_2^2} \cdot \frac{1}{C} \sum_{k=1}^{C} |S_k| \|\bar{u}_{S_k} - \bar{u}\|^2.
\end{aligned}$$

□

Appendix D

Alternative proof for the cllr norm

The regularizer (2.6) can also be found by using proposition 1 of [20]. Indeed if $B_l = B_l^{cllr}$, then

$$B = \begin{pmatrix} \alpha_1 I_d & \alpha_1 I_d & \dots & \alpha_1 I_d \\ \alpha_2 I_d & 0 & \dots & 0 \\ 0 & \alpha_2 I_d & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \alpha_2 I_d \\ \alpha_3 I_d & \alpha_3 I_d & \dots & 0 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \alpha_3 I_d \end{pmatrix}, \tag{D.1}$$

and $B^T B$ is such that

$$\forall l, q, \ \left(B^T B\right)_{lq} = (\alpha_1^2 + \alpha_2^2 \delta_{lq} + \alpha_3^2 \theta_{lq}) I_d,$$

where we took the notation of [20], $\theta_{lq} = \delta_{s(l)s(q)}$. Therefore,

$$\begin{aligned} \forall l, q, \ E_{lq} =& \left(B^T B\right)^{-1}_{lq} \\ =& \Bigg(- \frac{\alpha_1^2}{(\alpha_2^2 + \alpha_3^2(s(l))|S_{s(l)}|)(n\alpha_1^2 + \alpha_2^2 + \alpha_3^2(s(q))|S_{s(q)}|)} + \frac{1}{\alpha_2^2}\delta_{lq} \\ & - \frac{\alpha_3^2(s(l))}{\alpha_2^2(\alpha_2^2 + \alpha_3^2(s(l))|S_{s(l)}|)}\theta_{lq} \Bigg) I_d, \end{aligned}$$

and we have

$$\begin{aligned} J(u) =& u^T E u \\ =& \frac{1}{\alpha_2^2} \left(\sum_{l,q=1}^{n} \delta_{lq} u_l u_q - \sum_{l,q=1}^{n} \theta_{lq} \frac{\alpha_3^2(s(l))}{\alpha_2^2 + \alpha_3^2(s(l))|S_{s(l)}|} u_l u_q - \sum_{l,q=1}^{n} \frac{\alpha_1^2 \alpha_2^2}{\alpha_2^2(\alpha_2^2 + \alpha_3^2(s(l))|S_{s(l)}|)} u_l u_q \right). \end{aligned}$$

We show that

$$\begin{aligned}
\sum_{l,q\in S_k} u_l u_q &= -|S_k| \left(\sum_{l\in S_k} \|u_l - \bar{u}_{S_k}\|^2 - \sum_{l\in S_k} \|u_l\|^2 \right) \\
\sum_{l,q=1}^{n} u_l u_q &= -n \left(\sum_{l=1}^{n} \|u_l - \bar{u}\|^2 - \sum_{l=1}^{n} \|u_l\|^2 \right) \\
&= -n \left(\sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \sum_{l=1}^{n} \|\bar{u}_{S_{s(l)}} - \bar{u}\|^2 - \sum_{l=1}^{n} \|u_l\|^2 \right),
\end{aligned}$$

so if we choose

$$\forall k, \alpha_3^2(k) = \frac{x}{|S_k|},$$

we get

$$\begin{aligned}
J(u) =& \frac{1}{\alpha_2^2} \Bigg[\left(1 - \frac{x}{\alpha_2^2 + x} - \frac{\alpha_1^2 \alpha_2^2}{(\alpha_2^2 + x)(n\alpha_1^2 + \alpha_2^2 + x)} \right) \sum_{l=1}^{n} \|u_l\|^2 \\
&+ \left(\frac{x}{\alpha_2^2 + x} + \frac{\alpha_1^2 \alpha_2^2}{(\alpha_2^2 + x)(n\alpha_1^2 + \alpha_2^2 + x)} \right) \sum_{l=1}^{n} \|u_l - u_{S_{s(l)}}\|^2 \\
&+ \frac{\alpha_1^2 \alpha_2^2}{(\alpha_2^2 + x)(n\alpha_1^2 + \alpha_2^2 + x)} \sum_{k=1}^{C} \|\bar{u}_k - \bar{u}\|^2 \Bigg] \\
=& \frac{1}{(n\alpha_1^2 + \alpha_2^2 + x)} \sum_{l=1}^{n} \|u_l\|^2 + \frac{n\alpha_1^2 + x}{\alpha_2^2 (n\alpha_1^2 + \alpha_2^2 + x)} \sum_{l=1}^{n} \|u_l - u_{S_{s(l)}}\|^2 \\
&+ \frac{n\alpha_1^2}{(\alpha_2^2 + x)(n\alpha_1^2 + \alpha_2^2 + x)} \sum_{k=1}^{C} \|\bar{u}_k - \bar{u}\|^2 \\
\propto& \sum_{l=1}^{n} \|u_l\|^2 + \frac{x + \alpha_1^2 n}{\alpha_2^2} \sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \frac{\alpha_1^2 n}{x + \alpha_2^2} \sum_{k=1}^{C} |S_k| \|\bar{u}_{S_k} - \bar{u}\|^2,
\end{aligned}$$

which is what we found in appendix C with decompositions from $\|w\|^2$. □

Appendix E

Proof of proposition 5

Using the same mechanism as in D, we can prove proposition 5: if we want

$$J(u) \propto \sum_{l=1}^{n} \|u_l\|^2 + \beta \sum_{l=1}^{n} \|u_l - \bar{u}_{S_{s(l)}}\|^2 + \alpha \sum_{k=1}^{C} |S_k| \|\bar{u}_{S_k} - \bar{u}\|^2,$$

with $\alpha, \beta \in \mathbb{R}^+$, then necessarily

$$E_{lq} \propto \left(\delta_{lq}(1+\beta) - \theta_{lq} \left(\frac{\beta - \alpha}{|S_{s(l)}|} \right) - \alpha \right) I_d,$$

and

$$\forall l, q,\ E_{lq}^{-1} \propto \left(\frac{\alpha}{1+\alpha} + \frac{1}{1+\beta}\delta_{lq} + \frac{\beta - \alpha}{|S_{s(l)}|(1+\beta)(1+\alpha)}\theta_{lq} \right) I_d.$$

Since we must have $BEB^T = I_p$, it is necessary for B to be a squared root of E^{-1}. Such a root exists because E is positive definite.

Now if we want the root to have the form of (D.1), we need to set

$$\alpha_1 = \sqrt{\frac{\alpha}{1+\alpha}},\ \alpha_2 = \sqrt{\frac{1}{1+\beta}},\ \alpha_3 = \sqrt{\frac{\beta - \alpha}{|S_{s(l)}|(1+\beta)(1+\alpha)}},$$

and for such a mapping to be defined, one must have $\beta - \alpha \geq 0$. □