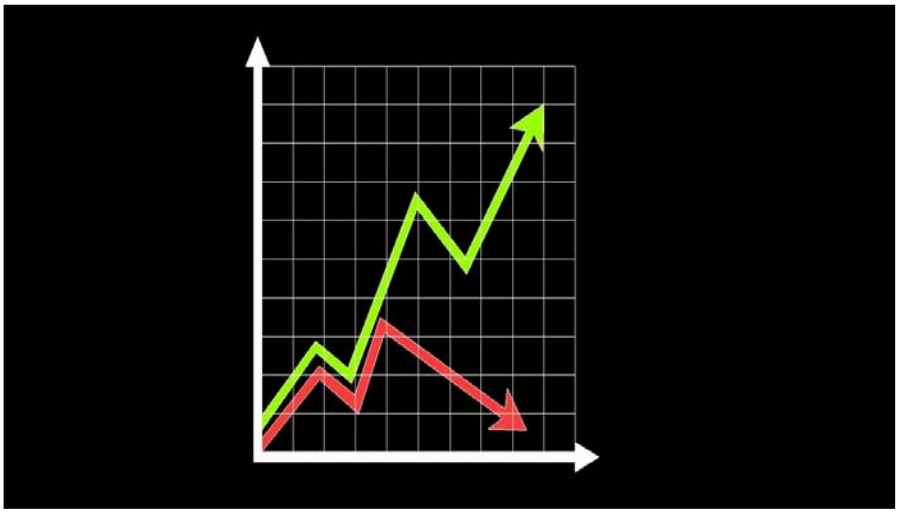
New methods for multi-objective learning

Shameem Ahamed Puthiya Parambath

Par **Shameem Ahamed PUTHIYA PARAMBATH**

New methods for multi-objective learning

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 16 décembre 2016
Spécialité : Computer Science : Unité de recherche Heudyasic (UMR-7253)

D2322

Université de Technologie de Compiègne

Heudiasyc

New methods for multi-objective learning

Submitted in partial satisfaction of the requirements for the degree of Doctor

in Computer Science

by

Shameem Ahamed Puthiya Parambath

Thesis presented on 16-12-2016 to the Jury:

Prof.	Marie Szafranski	ENSIIE, Evry	(Examinateur)
Prof.	Massih-Reza Amini	Université Grenoble Alpes	(Rapporteur)
Dr.	Nicolas Usunier	Facebook AI Research	(Directeur)
Prof.	Patrick Gallinari	Université Pierre et Marie Curie	(Rapporteur)
Prof.	Thierry Denoeux	Université de Technologie de Compiègne	(Presidént)
Dr.	Yves Grandvalet	CNRS	(Directeur)

In loving memory of my father Ahamed Naduviloth, to my mother Bushra, my wife Regina and my daughter Evana

Acknowledgements

This thesis was done at the Heudiasyc laboratory of the Université de Technologie de Compiègne, France. It was carried out in the framework of the Labex MS2T (Control of Technological Systems of Systems) which was funded by the French Government, through the program "Investment for the Future" managed by the National Agency for Research (ANR-11-IDEX-0004-02) and Hauts-de-France regional council.

First and foremost, I would like to express my sincere gratitude to both of my supervisors Nicolas Usunier and Yves Grandvalet, to whom I am forever indebted for the realization of this thesis, teaching me and helping me to think like a researcher. I would also like to thank the researchers of Heudiasyc, Philippe Xu and Sébastian Destercke, with whom I had many discussion about many aspects of the research.

I would also like to use this opportunity to thank the administrative and support staff of UTC, in particular I would like to thank Berengere Guermonprez, Laurie Herlin and Nathalie Alexandre for thier help and support during the course of this thesis. I am also very thankful to current and past fellow PhD students and friends, Alberto García Durán, Nguyen Vu Linh, Alia Chebly, Hafida Mouhagir, Chunlei Yu, Bihao Wang, Subber Rangra, Neeraj Maheshwari, Dingfu Zhou, Batoul Abdelaziz, Elwan Hery and Clément Dubos and all others for the wonderful time we had together and making my stay in France an unforgettable experiance. I also thank Prof. Sanjay Chawla for very valuable suggestions and for his support.

I thank my friends Awad, Harisankar, Hashir, Jayaraj, Nishant, Sujith and Tony for encouraging me throughout of my life, and standing beside me during the bad and good times, even after 12 years.

Last but not the least, I would like to thank my family, my daughter, wife and mother who are the driving force of my life, my brother, sisters and the god almighty.

Abstract

Multi-objective problems arise in many real world scenarios where one has to find an optimal solution considering the trade-off between different competing objectives. Typical examples of multi-objective problems arise in classification, information retrieval, dictionary learning, online learning etc. In this thesis, we study and propose algorithms for multi-objective machine learning problems.

We give many interesting examples of multi-objective learning problems which are actively persuaded by the research community to motivate our work. Majority of the state of the art algorithms proposed for multi-objective learning comes under what is called "scalarization method", an efficient algorithm for solving multi-objective optimization problems.

Having motivated our work, we study two multi-objective learning tasks in detail. In the first task, we study the problem of finding the optimal classifier for multivariate performance measures. The problem is studied very actively and recent papers have proposed many algorithms in different classification settings. We study the problem as finding an optimal trade-off between different classification errors, and propose an algorithm based on cost-sensitive classification. In the second task, we study the problem of diverse ranking in information retrieval tasks, in particular recommender systems. We propose an algorithm for diverse ranking making use of the domain specific information, and formulating the problem as a submodular maximization problem for coverage maximization in a weighted similarity graph.

Finally, we conclude that scalarization based algorithms works well for multi-objective learning problems. But when considering algorithms for multi-objective learning problems, scalarization need not be the "to go" approach. It is very important to consider the domain specific information and objective functions. We end this thesis by proposing some of the immediate future work, which are currently being experimented, and some of the short term future work which we plan to carry out.

Contents

List of Figures

List of Tables

Introduction

1

Contents

This chapter introduces the theory behind multi-objective optimization which is at the heart of multi-objective learning. Unlike scalar optimization problems, a multi-objective optimization problem has a vector valued objective function and often, different components of the objective function are "competing". Hence, many notions of scalar optimization problems like minima and maxima do not hold in multi-objective optimization. Here, we introduce the concepts of minimal/maximal elements and describe the scalarization principle; an efficient and commonly employed technique to solve multi-objective optimization problems. We start the chapter with the context of this work, and proceed to motivate our work by giving many interesting examples of multi-objective machine learning problems actively studied by both academia and industry.

1.1 Context

For the past couple of years "machine learning" is a buzzword in the science community. The advent of powerful, cheaper computation facilities, storage methods, and the availability of large scale data helped scientists to solve complex tasks using machine learning based approaches. Now, machine learning algorithms are used in almost all scientific fields: high energy physics, astronomical physics, behavioural studies, economics and medical studies to name a few; to learn predictive statistical models with well bounded generalization performance. Such models, however, are limiting in the sense that they do not consider the interplay between different objectives, often competing with each other, of the problem in the hand.

Consider the problem of learning a ranking function in the context of web search. Here we are interested in building a ranking function which returns a list of web pages related to a given query. From the supervised learning perspective, one learn a ranking function from the given training data by minimizing a loss function (or maximizing an utility function) such that the most relevant web pages to the given query appear at the top ranked positions. In such framework, performance of the learned function depends on the loss function we optimize, which is often designed with a single objective in mind. In web retrieval, the loss function is based only on the relevance aspect of the web page to the given query. Hence, the results of such a ranking function might contain very relevant but redundant pages at the top ranked positions. However, in practice a diverse list is much preferred covering many aspects of the query as advocated by Spärck-Jones et al. (2007). To design such a ranking function, one need to consider different conflicting objectives of the problem in hand, like relevance and diversity here.

In many applications of practical importance, we should design learning algorithms taking into account the different objectives at stake. Hence the study and development of algorithms for multi-objective learning is a very important task with practical importance. In this work we study multi-objective learning from both the theoretical and the application point of view. The application domains where the multi-objective learning plays an important role is prohibitively large. So we limit our focus on two multi-objective problems; *(i)* multi-variate performance metric optimization in classification problems and *(ii)* diverse ranking in recommender systems. We study the state of the art algorithms for the above two problems in detail, and propose new algorithms which take into account the trade-off between different competing objectives of the selected problem.

In this chapter, we motivate our work by giving examples of many multi-objective learning problems of both theoretical and practical interest. We briefly describe these problems in Section 1.2. The heart of

any multi-objective learning algorithm is the multi-objective optimization techniques. In Section 1.3, we give a brief introduction to the theory of multi-objective optimization and the scalarization method; a very popular method for solving multi-objective optimization problems. We conclude the chapter in Section1.4.

1.2 Motivations

Multi-objective learning fits very naturally in many real world application systems due to the inherent trade-off between different variables defining the system. We can view the structured risk minimization (SRM) paradigm employed in supervised learning tasks as a multi-objective (bi-objective) learning problem. The SRM based learning algorithm selects a learner with the optimal trade-off between the approximation-estimation error or equivalently the bias-variance terms. In this section, we exemplify the motivation for our work with many interesting multi-objective machine learning problems.

Many examples of multi-objective learning problems can be found in scientific literature, though sometimes the problems are stated as scalar-objective. Some examples include multi-variate loss minimization in classification problems, relevance-diversity trade-off in recommender systems and information retrieval in general and choosing a learner with a lower error bound with respect to few experts at the expense of higher error bound with respect to the rest of the experts in online learning settings. In this section, we briefly explain some of these problems which serve as the motivation for a detailed study of multi-objective learning algorithms.

Binary Classification

Binary classification is the quintessential classification problem extensively studied by the machine learning community. The problem essentially is a bi-criterion problem, where one tries to find a classifier with optimal trade-off between true positive rate and true negative rate. Equivalently we can frame the problem as choosing a classifier with optimal trade-off between different errors associated with the binary classification like false positive rate and false negative rate or any combination of error rates and true predictive rates.

Given a set of independent and identically distributed training set $(\mathcal{X}, \mathcal{Y})$ with $\mathcal{Y} = \{+1, -1\}$ and a probability measure $\mathbb{P}$ over the joint distribution $(\mathcal{X} \times \mathcal{Y})$, a binary classifier returns a hypothesis of the form $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ from the given hypothesis class $h \in \mathcal{H}$ (for simplicity we restrict ourselves to linear classifiers only) such that the expected error rate on the unseen data is minimal. The error rate can be written as

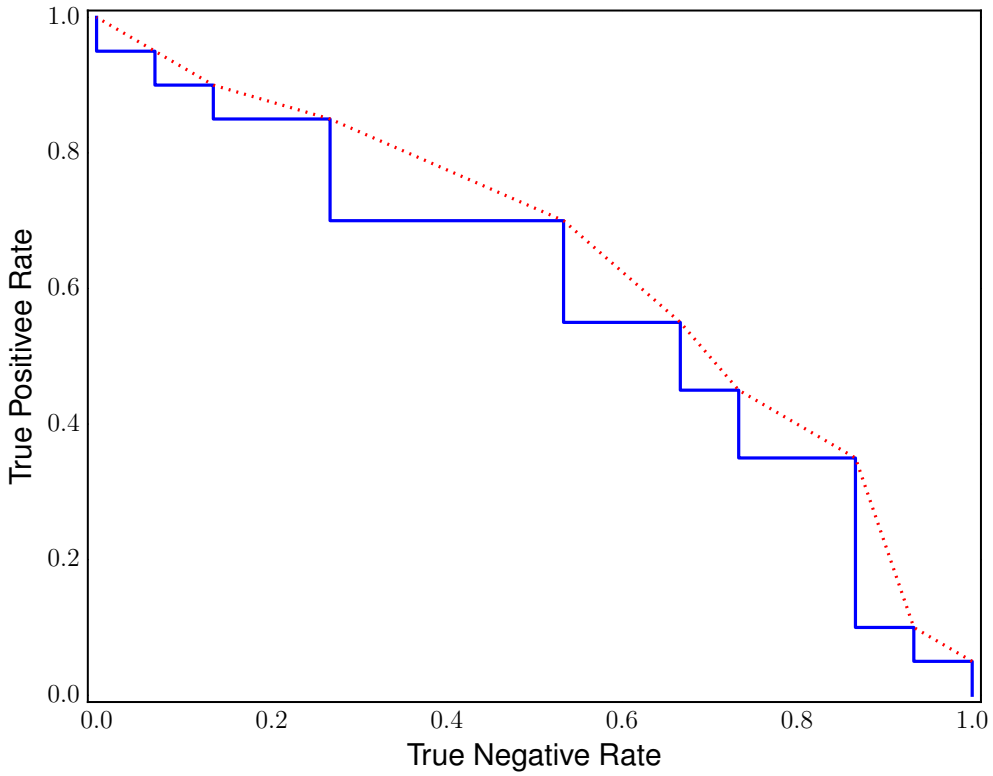


Figure 1.1 – *Optimal Trade-off curve for binary classification problem. The blue line represents the trade-off curve for true positive rate and true negative rate by varying the bias term(b) and the red line represents the optimal trade-off curve which can be obtained using the linear combination of two classifiers (Bach et al. 2006)*

.

$\mathbb{E}_{y|\mathbf{x}}\big[\mathbb{P}(y=+1|\mathbf{x})\mathbb{1}(h(\mathbf{x})\neq+1)+\mathbb{P}(y=-1|\mathbf{x})\mathbb{1}(h(\mathbf{x})\neq-1)\big]$, where $\mathbb{E}$ is the expectation and $\mathbb{1}$ is the indicator function. The first term in the sum corresponds to the false negative error rate and the second term corresponds to false positive error rate. In practice there exists trade-off between these two error rates and the problem can be considered as a bi-criterion optimization problem.

Kim et al. (2006) studied the problem of selecting a Pareto optimal linear classifier for a given true positive rate or a given negative rate for Gaussian and mixture of Gaussian class conditional probabilities by solving a convex objective function at each step. The trade-off curve between true positive rate and true negative rate for binary classification on an artificial data is given in Figure 1.1

Similarly, Bach et al. (2006) studied the problem of generating optimal-classifiers when the costs associated with misclassification rates, false positive rate and false negative rate, are different. They proposed an algorithm to select the optimal classifier by generating the full Receiver Operating Characteristic (ROC) curve by varying both the slope ($\mathbf{w}$) and bias (b) terms of a linear classifier.

Optimizing Multi-variate Performance Metrics for classification

In general classification schemes, one has to consider different optimization criterion depending on the application settings. A large class of optimization criterion used in classification tasks comes under the label "multi-variate performance measures" (Parambath et al. 2014, Koyejo et al. 2014, Narasimhan et al. 2014; 2015, Koyejo et al. 2015, Narasimhan et al. 2015). These measures are defined over the classification outcomes of the entire set of test data, and can not be decomposed into the sum of the classification outcomes of individual examples. Moreover, such measures are non-linear functions of classification outcomes. Examples for such metrics include F_β-measure in binary, multiclass and multilabel classifications, Jaccard index and many others (Koyejo et al. 2014, Narasimhan et al. 2014). These metrics trade-off false positive rate and false negative rate of the classifier, and in many application settings like imbalanced data classification, it is required to find classifiers which results in optimal performance with respect to the chosen multi-variate performance measure.

Diverse Ranking in Information Retrieval

In many information retrieval tasks like web search and recommender systems, it is very important to rank the items such that top-k listings contain diverse items. The need for diversity is usually derived from the uncertainty in the information need of the user or the inherent limitations of the information system to represent and capture complex user requirements.

In a typical web search settings, the user provides a short query to the search engine which often does not represent the exact user intent. For example, consider the classical example of the ambiguous query "jaguar". The query might indicate the animal Jaguar, the aircraft engine Jaguar, the fictional novel Jaguar, the movie Jaguar or the Jaguar cars. When such a query is issued to the search engine, it should return results which cover all the aspects of the associated query and diversification is a means to achieve it. Similarly in recommender systems, top-k recommendations should contain diverse items to increase user satisfaction and reduce the effect of popularity bias.

Diversity is often achieved at the expense of relevance. To induce diversity, one often trades-off the relevance of an item with the dissimilarity of the other items already added to the recommendation set. Such a system will help in reducing the redundancy of the results, and by promoting more dissimilar results, it may result in diverse recommendations. Clearly, here the problem of diversification is again a bi-objective optimization problem, where the task is to rank the items with the optimal trade-off between relevance and diversity.

Subset Selection Problem

In general, subset selection problem refers to the class of problems where one aims to select the best set of representative set from a given ground set. Given a ground set of variables, subset selection problem is defined as selecting a subset of variables from the ground set such that an objective function is optimized. The subset selection problem arises in many application like feature selection, dictionary learning, etc.

Formally, given a set of ground variables, $\mathcal{X} = \{x_1, \cdots x_m\}$, and an objective function $f : 2^{\mathcal{X}} \rightarrow \mathbb{R}$, subset selection algorithm outputs a set $\mathcal{S}$ such that $\text{argmin}_{\mathcal{S} \subseteq \mathcal{X}} f(\mathcal{S})$ such that $|\mathcal{S}| \leq k$. The equation have two conflicting objectives, *(i)* to minimize the objective function f with *(ii)* the cardinality constraint on the set $\mathcal{S}$. Here the trade-off is such that, a set with larger number of variables (higher k values) will give optimal value for the objective function f. The sparse regression problem is a classical representative example of subset selection problem. In sparse regression, we aim to estimate the response variable by linear regression using only a subset of the original predictor or feature vectors, and the quality of the estimation is measured using mean squared error or equivalently squared multiple correlation (Das and Kempe 2011, Qian et al. 2015).

Online Learning with experts

Consider the online learning settings with expert advice. In this task, a prediction algorithm is given access to a set of experts and it needs to make a sequence of decisions (number of sequence is not known in general, and we simply term it as horizon) with the objective of performing as good as the best expert in hindsight. Given a finite set of expert set ($|\mathcal{E}| = k$), our task is to make predictions such that the cumulative regret $\hat{L}_i - L_i^j$ with respect to each expert j is minimized. Here $\hat{L}_i$ is the cumulative loss of the predictor with respect to a given loss function ℓ and L_i^j is the cumulative loss of the expert j with respect to the same loss function. The predictor should be independent of the sequence of the outcomes i.e. the regret of the predictor is minimum for all sequences of outcomes (Cesa-Bianchi and Lugosi 2006).

State of the art algorithms, like exponentially weighted prediction or hedge algorithm achieve a regret bound of $\sqrt{\frac{n}{2log(k)}}$ over a sequence of size n . The bound is uniform over the choice of experts i.e. the regret bound holds with respect to every given expert. However, in practice one would expect a trade-off between the performance of the different experts. In such practical situations, one would prefer an algorithm where the regret with respect to some "good" experts is very low at the expense of increased overhead with respect to "bad" experts.

1.2.1 Objectives

Given the above examples, it is evident that multi-objective learning is of vital importance in many application settings. In this work, we aim to study and analyze the multi-objective machine learning problems and propose new algorithms for such problem. In this study, we limit our attention to bi-objective optimization problems. The class of problems comes under bi-objective optimization is prohibitively large to study within a course of three years. Hence we concentrate on two problems of practical importance from the list of problems given above. We study in detail the problem of *(i)* Optimizing multi-variate performance measures for classification and *(ii)* Diverse ranking in information retrieval.

1.3 Multi-Objective Learning

Multi-objective learning is a natural extension to the single-objective learning problem. In case of single-objective learning problems, our goal is to develop a learning algorithm which returns a function (from a given restricted function class) which has optimal expected value with respect to the given single-objective loss or utility function on the future unknown data. To achieve this goal, in single-objective learning, we make use of scalar optimization techniques. The multi-objective learning problem consists of multiple objectives i.e. the objective function consists of multiple components and each component corresponds to a single objective. Often, the component objectives of a multi-objective learning problems are competing in the sense that an increase in one component objective may result in the decrease of another objective. In such cases, we say that there is a trade-off between multiple objectives.

Given a set of training data, the goal of multi-objective learning is to find a function (from a given restricted function class) which jointly optimizes the different components of the multi-objective function. Similar to the case of single objective learning problems, in case of multi-objective learning problems we make use of multi-objective optimization techniques.

In this section, we give answers to the following questions

- What is a multi-objective optimization problem?
- What is the meaning of optimal solution in case of multi-objective optimization?
- How can we solve multi-objective optimization problem in practice?

We assume that our optimization problem is a minimization task and our discussion is from the minimization task point of view. The discussion

applies to maximization problem also. Any maximization problem can be converted to an equivalent minimization problem by changing the sign of the objective function.

1.3.1 Multi-Objective Optimization

A single objective optimization problem can be formally defined as,

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0,\ i = 1, \ldots, p \\ & h_i(\mathbf{x}) = 0,\ i = 1, \ldots, q. \end{aligned}$$

Here, our goal is to find the value (assuming the solution is unique otherwise values) of the independent variable $\mathbf{x} \in \mathbb{R}^n$ which results in the minimum value of the function $f(\mathbf{x})$, and in addition also satisfy the conditions $g_i(\mathbf{x}) \leq 0,\ i = 1, \ldots, p$, and $h_i(\mathbf{x}) = 0,\ i = 1, \ldots, q$. The function $f(\mathbf{x})$ is called the objective function and it is scalar valued, i.e. $f : \mathbb{R}^n \to \mathbb{R}$. Similarly the functions $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$ are also scalar valued, i.e. $g_i : \mathbb{R}^n \to \mathbb{R}$ and $h_i : \mathbb{R}^n \to \mathbb{R}$. The function g_i is called the inequality constraint function and h_i is called equality constraint function. There is a total of p inequality constraints and q equality constraints. The constraint functions define the acceptable values of $\mathbf{x}$ for the problem in hand. The set of acceptable values of $\mathbf{x}$ which satisfies the constraint functions is called feasible set.

Now, as stated earlier, in case of multi-objective optimization problem, the objective function is vector valued i.e. the objective function can be considered as having multiple components. The output vector corresponds to the values of different components (objectives) for a given value of $\mathbf{x}$. For example, in case of the binary classification problem given in Section 1.2, the output vector consists of two components, one corresponds to the first objective; misclassification rate with respect to the true class 1; and the second corresponds to the misclassification rate with respect to the true class 2. In general for k-objective optimization problem, we have $f : \mathbb{R}^n \to \mathbb{R}^k$, and the goal is to find the value of $\mathbf{x}$ such that the vector returned by f is minimum which also satisfies the constraints. In case of k-objective optimization problem, we can consider that the objective function as composed of k components $f_1, f_2, \cdots f_k$ where each component corresponds to a scalar objective.

In fact, the above mentioned minimization problem implies minimization over vectors, and it is necessary to specify an ordering on $\mathbb{R}^k$ to define the meaning of minima (similarly maxima). We define a partial order on $\mathbb{R}^k$ with respect to a proper cone in $\mathbb{R}^k$. A cone $\mathbb{K}$ is called proper, if

- $\mathbb{K}$ is convex

- $\mathbb{K}$ is pointed
- $\mathbb{K}$ has non-empty interior
- $\mathbb{K}$ is closed

A set $\mathcal{S}$ is convex, if $\forall s_1, s_2 \in \mathcal{S}$, the linear combination $\theta s_1 + (1 - \theta)s_2 \in S$, $0 \leq \theta \leq 1$. A set is pointed, if it contains no lines, i.e. it contains only rays. In a nutshell, it implies that the set corresponding to a proper cone contains an origin vector (zero vector) and does not contain any additive inverse vectors. A set with non-empty interior contains elements other than the boundary elements. A set is closed, if it contains all the limit points , equivalently if all the sequence of rays in the cone converges to the limit ray. We encourage the readers to refer to Rudin (1991), Boyd and Vandenberghe (2004) for more details about the concepts related to proper cones.

Given two vectors $\mathbf{s}, \mathbf{r} \in \mathcal{S}$, we define a generalized inequality consisting of the partial ordering in $\mathbb{R}^k$ with respect to the proper cone $\mathbb{K}$ as,

$$\begin{aligned} \mathbf{s} \succeq \mathbf{r} &\iff \mathbf{s} - \mathbf{r} \in \mathbb{K}, \text{ and} \\ \mathbf{s} \succ \mathbf{r} &\iff \mathbf{s} - \mathbf{r} \in \boldsymbol{int}(\mathbb{K}). \end{aligned}$$

Here, $\boldsymbol{int}(\mathbb{K})$ denotes the interior of the cone $\mathbb{K}$. Similarly, we use the notation $\mathbf{r} \prec \mathbf{s}$ for $\mathbf{s} \succ \mathbf{r}$ and $\mathbf{r} \preceq \mathbf{s}$ for $\mathbf{s} \succeq \mathbf{r}$. Boyd and Vandenberghe (2004) define a multi-criterion optimization problem as a multi-objective optimization where the proper cone associated with the generalized inequality is always the non-negative orthant of $\mathbb{R}^k$ denoted as $\mathbb{R}^k_+$. In our discussion we always assume that the proper cone associated with the generalized inequality is the non-negative orthant $\mathbb{R}^k_+$.

Formally, we define the multi-objective optimization problem as given below, where we follow the same notation as in the case of the scalar optimization problem

$$\begin{aligned} \text{minimize} \quad & \mathbf{f}(\mathbf{x}) \text{ with respect to the proper cone } \mathbb{K} \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0, \; i = 1, \ldots, p \\ & h_i(\mathbf{x}) = 0, \; i = 1, \ldots, q. \end{aligned} \tag{1.1}$$

The input space ($\mathbb{R}^n$) is called the decision space and the output space ($\mathbb{R}^k$) is called the objective space. The set of feasible solutions for Eq 1.1 forms the feasible decision space and the corresponding output values form the set of feasible objective space. Figure 1.2 depicts the feasible decision and objective spaces of a bi-objective optimization problem.

Pareto Optimal Solution

The ordering associated with the generalized inequality defined with respect to the proper cone $\mathbb{K}$ is partial. The concepts of minimum (similarly infimum) and maximum (similarly supremum) is different in case

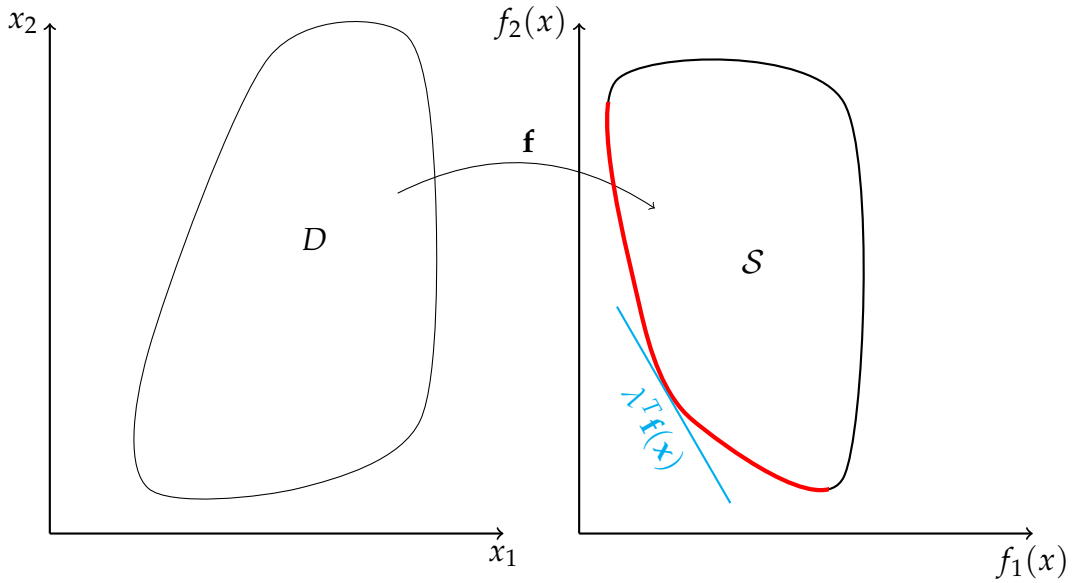


Figure 1.2 – *The feasible decision and objective spaces of a bi-objective optimization problem*

of partial ordering. Here we define the concepts of minimum and minimal elements in case of partial ordering defined over the proper cone $\mathbb{K}$.

Definition 1 (Minimum Element) *Given a set $\mathcal{S}$, an element $\mathbf{s} \in \mathcal{S}$ is the minimum element of $\mathcal{S}$ with respect to the generalized inequality defined over the proper cone $\mathbb{K}$, if*

$$\mathbf{s} \preceq \mathbf{r}, \ \forall \mathbf{r} \in \mathcal{S}$$

In simpler terms, an element is a minimum element if the element can be compared with all the other elements of the set, and it has lower value. The minimum element of a set $\mathcal{S}$ is depicted in Figure 1.3. In the plot, the element $\mathbf{e}$ can be compared with all other elements of the set $\mathcal{S}$, as all the other elements lie on the upper right side of $\mathbf{e}$, and it has lowest value according to the partial ordering defined by the proper cone $\mathbb{R}^2_+$. If a minimum element exists for a set $\mathcal{S}$, it should be unique. Unfortunately a minimum element exists only in the cases where the objectives are noncompeting i.e. the cases where the function does not have to make any compromise between different components of the objective. In other terms, an increase in one component of the objective does not cause a decrease in another component of the objective. In case of competing objectives, we define the minimal element of $\mathcal{S}$ with respect to the generalized inequality defined over the proper cone $\mathbb{K}$.

Definition 2 (Minimal Element) *Given a set $\mathcal{S}$, an element $\mathbf{s} \in \mathcal{S}$ is the minimal element of $\mathcal{S}$ with respect to the generalized inequality defined over the proper cone $\mathbb{K}$, if* $\forall \mathbf{r} \in \mathcal{S}, \quad \mathbf{r} \preceq \mathbf{s}$ *only if* $\mathbf{r} = \mathbf{s}$.

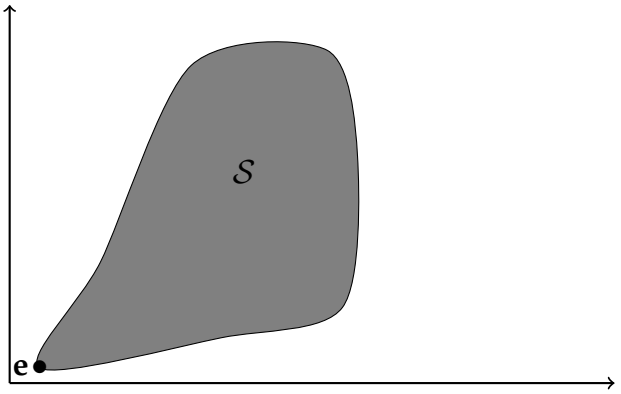


Figure 1.3 – *The point* **e** *is the minimum element of the objective space* $\mathcal{S}$

A minimal element is a feasible element that has the minimum value with respect to the proper cone $\mathbb{K}$ among all the vectors comparable to it. A multi-objective optimization can have multiple minimal elements. A minimal element is called Pareto optimal and the set of minimal elements define a Pareto front. In two dimensional real space $(\mathbb{R}^2)$ the Pareto front is a Pareto curve. The Pareto curve for the bi-objective optimization problem in Figure 1.2 is marked in red. The Pareto front gives the trade-off values between different objectives of the multi-objective function.

1.3.2 Scalarization

The scalarization principle is one of the most popular, effective and efficient method to solve multi-objective optimization problems. The idea behind scalarization is to transform the given multi-objective problem into a single-objective problem. The new single objective problem will have parameters called weights which are not in the original problem formulation. The scalarization guarantees optimality i.e. a solution to the scalarized objective function will be a Pareto optimal solution for the original multi-objective problem (under some constraints on the weight parameters) and for different values of the scalarization parameter we obtain (possibly) different Pareto optimal solutions. The scalarization method is also called by the name weighted-sum approach (Ehrgott and Gandibleux 2002).

Formally, the original multi-objective problem given in Eq 1.1 is transformed to the below scalar objective function

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0,\ i = 1, \ldots, p \\ & h_i(\mathbf{x}) = 0,\ i = 1, \ldots, q \\ & \boldsymbol{\lambda} > 0 \end{aligned} \tag{1.2}$$

Here $\boldsymbol{\lambda}$ is the scalarization parameter (weights) called weight vector.

It should be noted that even though for any $\boldsymbol{\lambda} > 0$, scalarization returns a Pareto optimal value, not all Pareto optimal solutions can be obtained using the scalarization method. Convexity of the objective space plays an important role here. The only Pareto optimal solutions which can be obtained using scalarization are the one which are at the boundary of the convex hull of the objective space. Moreover, the solution obtained using scalarized objective function defines a supporting hyperplane at the point $\mathbf{x}$ i.e. $\boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x})$ is a supporting hyperplane to the objective space at the point $\mathbf{x}$. A Pareto optimal solution and corresponding hyperplane defined by the scalarized objective function is shown as cyan line in Figure 1.2.

1.4 Conclusion

We introduced the problem of multi-objective learning in this chapter. Many of the practical problems studied by the machine learning community under many subfields like online learning, recommender systems, etc are inherently multi-objective. We gave many examples of such problems to motivate our study. We also gave a brief introduction to the multi-objective optimization problem which is at the heart of the multi-objective learning algorithms. The most popular and efficient method for solving multi-objective optimization problem is the scalarization method. We gave a brief introduction to the scalarization method in this chapter.

2 Multivariate Performance Measure Optimization

Contents

State of the art classification algorithms are designed to minimize the misclassification error on the test set, which is a linear function of the per-class false negatives and false positives. Nonetheless, non-linear performance measures are widely used for the evaluation of learning algorithms. For example, F-measure is a commonly used non-linear performance measure in classification problems. We study the theoretical

properties of a subset of non-linear performance measures called pseudo-linear performance measures which includes F-measure and Jaccard index. We establish that many notions of F-measures and Jaccard index are pseudo-linear functions of the per-class false negatives and false positives for binary, multiclass and multilabel classification schemes. Based on this observation, we present a general reduction of such performance measure optimization problem to cost-sensitive classification problem with unknown costs. We then propose an algorithm with provable guarantees to obtain an approximately optimal classifier for the F-measure by solving a series of cost-sensitive classification problems. The strength of our analysis is to be valid on any dataset and any class of classifiers, extending the existing theoretical results on binary F-score, which are asymptotic in nature. Our analysis shows that thresholding cost-insensitive scores, a common technique employed to optimize F-measure, yields sub-optimal results. We also establish the multi-objective nature of the F-measure maximization problem by linking the algorithm with the weighted-sum approach used in multi-objective optimization. We present numerical experiments to illustrate the relative importance of cost asymmetry and thresholding when learning linear classifiers on various F-measure optimization tasks.

2.1 Introduction

Different performance measures exist to assess the efficiency of learning algorithms in different practical settings. For example, the misclassification rate is one of the most commonly used performance measure in classification problems of balanced dataset. Like many other measures, which we will investigate in this paper, it is defined over the set of classification outcomes. The four possible outcomes of a classifier are the true positive (TP), true negative (TN), false Negative (FN) and false positive (FP) (See Section 2.3 for the formal definitions). The misclassification rate is a linear function of these quantities, defined as the sum of FP and FN. Conceptually, classification algorithms solve an optimization problem where the loss function corresponds to the performance measure is minimized or equivalently a utility function is maximized (see Devroye et al. 1996, Anthony and Bartlett 2009). A loss function maps the success or failure of an event to a real value (mostly non-negative). It measures how well the prediction for an event is closer to the actual event. For example, the loss function that corresponds to misclassfication rate is *0-1 loss* (Devroye et al. 1996).

As mentioned, misclassification rate is a commonly used performance measure, albeit unsuitable for specific categories of problems. For example, consider the binary classification of an imbalanced dataset of size 100 with 95 being samples of one specific class (let us say negative) and 5 being other class (say positive). A trivial classifier of the form 'always predict negative' results in a high accuracy albeit useless classifier. In this specific example, F_β (Rijsbergen 1979) can be considered as a more meaningful performance measure than misclassification rate. It is to be noted that F_β is a utility function whereas misclassfication rate is a loss function. In general, performance measures like F_β, are extensively used in practical problems (Cheng et al. 2012, Kim et al. 2013). One of the striking characteristics of these performance measures is the non-linearity with respect to the false negatives and false positives, whereas misclassification rate is a linear function of false negatives and false positives. Moreover, there is no convex surrogate loss function (or equivalently no concave surrogate utility function) that exists for non-linear measures like F_β-measure. Another interesting property of F-measure is: it is a sample level measure and does not decompose over individual examples. These three aspects make the optimization problem a difficult and interesting one.

In the current chapter, we study the theoretical and algorithmic aspects pertaining to the optimization of the pseudo-linear performance measures. The commonly used performance measure F_1 is an example of pseudo-linear performance measure. Less commonly used measures like Jaccard index also come under this title, among many others. Here, we focus primarily on pseudo-linear notions of F-measures. We consider

the setting in which a dataset is to be classified such that the F-measure (restricted to pseudo-linear versions only) of the resulting classification is (approximately) optimal. In the literature, F-measures are also often called F-scores. Here, we will stick to the first terminology, which refers to the measurement of performance, in order to avoid any confusion with classification scores, that is, the real-valued scores that may be provided by classifiers and that are thresholded to produce decisions. Unless otherwise explicitly stated, all the discussion in this chapter refers to F-measure optimization. At a later point, we generalize the results to other pseudo-linear measures.

Our principal goal is to study the algorithms for optimality of pseudo-linear F-measures on the sample level. Given a training set, our analysis proves that optimal F classifier for pseudo-linear F-measures can be found by minimizing the total misclassification cost of a cost-sensitive classification (Elkan 2001). Since the costs are not known *a priori*, approximately optimal F classifier can be obtained by searching over a discretized cost space and solving corresponding cost-sensitive classification problem. Optimality in the state of the art algorithms for pseudo-linear F-measures are asymptotic whereas our results are valid in the non-asymptotic regime without any assumption on the underlying data distribution. It can also be showed that our proposed method is in fact an instantiation of the weighted-sum approach used in the multi-objective optimization. Our experiments reveal the importance of thresholding classification scores to optimize F-measures which has been proposed recently to obtain optimal F_β classifier when using proper losses (Narasimhan et al. 2014, Koyejo et al. 2014; 2015, Narasimhan et al. 2015).

This chapter is an extended version of an already published conference paper (Parambath et al. 2014). The chapter is organized as follows. Section 2.2 introduces basic definitions and notations used throughout this chapter. We also present a brief study of the state of the art algorithms for F-measure optimization. Section 2.3 presents the theoretical analysis, where we establish the pseudo-linearity of different practical F-measures, and prove that optimal F classifier can be found by minimizing the total misclassification cost of a cost-sensitive classification. Since the cost values are not known *a priori*, we also derive the values for the approximate costs for many pseudo-linear F-measures. We establish the multi-objective view of the F-measure optimization problem and link our proposed approach to the popular weighted-sum approach for solving multi-objective optimization problems. Section 2.5 presents the experimental results. Also, we empirically show that thresholding is important for finding optimal solutions. We conclude the paper in Section 2.6.

2.2 Background and Related Work

Here, we introduce the notations and give a brief review of the state of the art methods for F-measure maximization. We start by introducing the notations used throughout in the chapter; we also formally define the F_β-measure in binary, multiclass and multilabel classification schemes.

2.2.1 Notation and Basic Definitions

We are given *(i)* a measurable space $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the feature space and $\mathcal{Y}$ is the (finite) prediction set, *(ii)* a probability measure μ over $\mathcal{X} \times \mathcal{Y}$, and *(iii)* a set of (measurable) classifiers $\mathcal{H}$ from the feature space $\mathcal{X}$ to $\mathcal{Y}$. We distinguish here the prediction set $\mathcal{Y}$ from the label space $\mathcal{L} = \{1, ..., L\}$: in binary or single-label multiclass classification, the prediction set $\mathcal{Y}$ is the label set $\mathcal{L}$, but in multilabel classification, $\mathcal{Y} = 2^{\mathcal{L}}$ is the powerset of the set of possible labels. In that framework, we assume that we have an i.i.d. sample drawn from an underlying data distribution $\mathbb{P}$ on $\mathcal{X} \times \mathcal{Y}$. The empirical distribution of this finite training (or test) sample will be denoted by $\hat{\mathbb{P}}$. Then, we may take $\mathbb{P}$ as measure μ to get results at the population level (concerning expected errors), or we may take $\mu = \hat{\mathbb{P}}$ to get results on a finite sample. Likewise, the set of classifiers $\mathcal{H}$ can be a restricted set of functions such as linear classifiers if $\mathcal{X}$ is a finite-dimensional vector space, or may be the set of all measurable classifiers from $\mathcal{X}$ to $\mathcal{Y}$ to get results in terms of *Bayes-optimal classifiers*. Finally, when required, we will use bold characters for vectors and normal font with subscript for indexing.

Most of the previous work on pseudo-linear measure is centered around the F_β-measure in binary settings. The F_β-measure is defined as the weighted harmonic mean of precision and recall. Precision is defined as the fraction of predicted positive instances that are indeed positive and recall is defined as the fraction of positive instances that are correctly predicted as positive. Formally, we can define these metrics using classifier outcomes. Given a binary dataset and classifier, `TP` corresponds to the correct prediction of a positive label, `TN` corresponds to the correct prediction of a negative label, `FN` corresponds to the incorrect prediction of a positive label as a negative label, and `FP` corresponds to the incorrect prediction of the negative label as positive. In general, these outcomes are depicted using a confusion matrix, also called contingency table (See Table 2.1). The confusion matrix of a multiclass and multilabel will be a $|\mathcal{L}| \times |\mathcal{L}|$ matrix. In terms of the confusion matrix entries (`TP`, `TN`, `FN`, `FP`), we formally define precision, recall and F_β associated with a binary classifier $h \in \mathcal{H}$ as:

$$(precision) \qquad \texttt{Precision}(h) = \frac{\texttt{TP}(h)}{\texttt{TP}(h) + \texttt{FP}(h)}$$

$$(recall) \qquad \texttt{Recall}(h) = \frac{\texttt{TP}(h)}{\texttt{TP}(h) + \texttt{FN}(h)}$$

$$(binary\text{-}F_\beta) \qquad F_\beta(h) = \frac{(1+\beta^2)\texttt{TP}(h)}{(1+\beta^2)\texttt{TP}(h) + \beta^2\texttt{FN}(h) + \texttt{FP}(h)}$$

In the F_β definition, the parameter β weights precision and recall in F_β: F_0 corresponds to precision, F_∞ corresponds to recall, and F_1, the most widely used, corresponds to equal weighting to both precision and recall. In case of the binary classification example mentioned in the introduction, classifying a sample of 100 instances, the precision, recall and F_1 values for the trivial classifier is zero, but the misclassification error rate is 0.95. But precision does not consider the effect of false negatives, and recall does not consider the effect of false positives i.e. precision does not account for classifying a correct label as incorrect and recall does not account for labelling an incorrect label as correct. So in practical problems F_1 (or in general F_β) is preferred. One important property to note here is unlike misclassification rate, F-measure is not invariant under label switching i.e. if we change the positive label to negative, we get a different value of F-measure. Hence it is used in problems where correct classification of minority label is of vital importance. In multilabel and multiclass settings, three different definitions of F-measure can be found; namely instance-wise, macro and micro F-measures. We will give formal definition of these in Section 2.3 in connection with our theoretical framework.

2.2.2 Related Work

Before the recent surge in the study of F-measure optimization, the problem was studied very limitedly (Musicant et al. 2003, Jansche 2005, Joachims 2005, Jansche 2007, Fujino et al. 2008). The last couple of years witnessed an increasing interest in this domain (Dembczynski et al. 2011, Nan et al. 2012, Pillai et al. 2012, Dembczynski et al. 2013, Cheng et al. 2012, Lipton et al. 2014, Koyejo et al. 2014, Narasimhan et al. 2014, Waegeman et al. 2014). The majority of the work was confined to F-measure maximization in binary classification settings, whereas very little work was done on multilabel and multiclass F-measure maximization tasks (Pillai et al. 2012, Dembczynski et al. 2011). Jansche (2005) suggested an algorithm for learning a non-deterministic classifier with locally optimal F_1-measure for binary classification problems by approximating the classification outcomes using logistic models. Since the objective function used is non-convex, the resulting classifier does not guarantee global optimality. A workaround; running the procedure multiple times with

different seeds and select the best classifier from the set of classifiers is also proposed by the author. The orthogonal problem of inferring the hypothesis with optimal F_1 from a probabilistic model is discussed by Jansche (2007). In the scientific literature, the two problem formulation has been referred to as empirical utility maximization (EUM) and decision-theoretic approach (DTA) respectively (Nan et al. 2012).

The two formulations differ with respect to the definition of the expected F-measure. In case of the EUM based approach, the population F-measure is defined as the F-measure of the expected TP,FP and FN. Formally, in EUM, the expected F-measure is defined as

$$F_\beta^{\text{EUM}}(h) = \frac{(1+\beta^2)\mathbb{E}[\texttt{TP}(h)]}{(1+\beta^2)\mathbb{E}[\texttt{TP}(h)] + \beta^2\mathbb{E}[\texttt{FN}(h)] + \mathbb{E}[\texttt{FP}(h)]}$$

An optimal EUM classifier can be defined as

$$h^* = \operatorname*{argmax}_{h\in\mathcal{H}} F_\beta^{\text{EUM}}(h)$$

A general strategy for EUM based algorithm is to estimate the classification score or the class conditional probability from the training data and select a classifier from the set of thresholded classifiers obtained by setting a threshold on the classification score or class conditional probability *a posteriori*.

In DTA, the expected F-measure is formally defined as

$$F_\beta^{\text{DTA}}(h) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathbb{P}}[F_\beta(h)]$$

An optimal DTA classifier is of the form

$$h^* = \operatorname*{argmax}_{h\in\mathcal{H}} F_\beta^{\text{DTA}}(h)$$

A general strategy for DTA based algorithm is to build a probabilistic model for the classifier using the training set, and infer the optimal classifier in an inference step. The inference step requires exponentially many classifier evaluations.

From an algorithmic point of view, DTA based algorithms are computationally more expensive than EUM based algorithms. DTA based algorithms require an efficient method to estimate the class probabilities and iterate over exponentially many combinations of hypothesis h and labels $\boldsymbol{y}$; and the problem of estimating exact probabilities is harder than the original problem. Moreover, DTA is a set classifier in the sense that in case of DTA, expectation is taken over the set of fixed size. Hence, an optimal classifier in DTA is the one with maximal expected F-measure among all the classifiers for a fixed size of training examples. The algorithm given by Jansche (2007) runs in $O(n^4)$, where n is the number of

test examples. The proposed algorithm makes use of a reduction strategy and instead of searching over exponentially many hypotheses, it searches over $n+1$ "best" hypotheses (for a test set size of n). Similarly, assuming i.i.d samples and considering the functional properties of F-measure (it can be written as a function of integer counts), the expectation over the label space $\mathcal{Y}$ can be carried out in $O(n^3)$. Nan et al. (2012) improved the efficiency of this algorithm by reducing the complexity to $O(n^3)$, using dynamic programming to solve the expectation over the label space $\mathcal{Y}$. They also remark that in case of EUM based algorithms, the optimal classifier for binary F_1 is of the form $sign(p(y=1|x)-\delta^*)$, where δ^* is a threshold score dependent on the underlying distribution. Dembczynski et al. (2011) followed a similar approach, and extended the algorithm given by Jansche (2007). They proposed a method to calculate optimal F_β classifier with $O(n^3)$ complexity in time, given n^2+1 parameters of the joint distribution $p(\boldsymbol{y})$. This algorithm was used in a multilabel setting for instance-wise F-measure (see Remark 3). In addition to the high computational footprint, there is no optimality guarantee on finite samples. In general, optimality in DTA algorithms are asymptotic in nature (Nan et al. 2012).

On the other hand, EUM based approaches are computationally less demanding, and are based on the structured risk minimization (SRM) principle. Here, we minimize an approximate surrogate loss function, and select the hypothesis with minimal error on the validation set. The most commonly employed EUM approach is to threshold the score obtained using linear classifiers like logistic regression or support vector machines (SVM) such that F_1 is maximized. An approximate surrogate function based approach named SVM$^{\text{perf}}$ is given by Joachims (2005), based on the structured SVM method for dependent output Tsochantaridis et al. (2005). In the suggested method, the discriminant function is defined over the linear combination of the feature vectors, where the scalar multiplier is the label associated with each feature vector in the training sample. Even though the reported experimental results were promising, the method does not offer any theoretical optimality guarantee. Our experiments confirm that SVM$^{\text{perf}}$ is a sub-optimal method. Musicant et al. (2003) also advocated for SVMs with asymmetric costs i.e. different costs for false negatives and false positives for F_1-measure optimization in binary classification. However, their argument, specific to SVMs, is not methodological but technical (relaxation of the SVM objective function).

In case of multilabel classification, Pillai et al. (2012) argued that the multilabel-micro-F-measure can be optimized by thresholding the classification scores (they used the term class confidence score), one label at a time. Pillai et al. (2012) used k-nearest neighbors and SVM to generate the class confidence scores. In general, thresholding cost-insensitive SVM scores does not guarantee empirical optimality since the hinge loss is not

a proper loss (Reid and Williamson 2010), and the paper does not address the issue of hyperparameter selection of the classification algorithm (k of k-nearest neighbor and regularization coefficient of SVM).

Fujino et al. (2008) proposed a framework for designing a classifier for optimal F_1-measure based on the linear combination of multiple classifier models. The weights of the classifier combination is estimated such that F_1-measure is maximized in cross-validation, and the model parameters for the individual classifier models are estimated independent of each other on a validation set. They combined two logistic models, *(i)* maximum likelihood logistic regression with a fixed threshold value and *(ii)* a concave approximation for F_1-measure where the parameters are estimated using logistic regression by running multidimensional optimization techniques (see Jansche 2005) to maximize multilabel micro, macro and instance-wise F-measure. This line of work comes under *multiple classifier systems*. *Multiple classifier systems* are not widely used for F-measure maximization. In our knowledge, no proper statistical study regarding the optimality of the *multiple classifier systems* for F-measure maximization has been done so far.

Apart from F-measure, some of the most recent work discusses nonlinear performance measures like Jaccard index (Koyejo et al. 2014, Narasimhan et al. 2014, Waegeman et al. 2014). Following the footsteps of Nan et al. (2012), Koyejo et al. (2014) and Narasimhan et al. (2014) proposed algorithms to maximize many non-decomposable performance measures including linear-fractional measures like F_β-measure by thresholding the conditional class probability independently. The algorithm returns a F_β optimal classifier by running a two-phase procedure. In the first phase a class conditional probability estimator is learned on a training set, and in the second phase a threshold is chosen such that F_β-measure is maximal on the validation set. The proposed algorithms are consistent if the empirical conditional probabilities converge to the true class conditional probabilities. Reid and Williamson (2010) studied the loss functions for conditional probability estimation, and proved that a conditional probability estimator is consistent only when the loss function is proper i.e. the proposed algorithms by Koyejo et al. (2014) and Narasimhan et al. (2014) give a consistent F_β classifier if the classification loss function is a proper loss. (Koyejo et al. 2015) extended the algorithm for micro$F$-measure in multilabel classification, where the optimal micro-F_β classifier is obtained in a similar fashion. A conditional probability estimator for each class label is obtained in the first step, and a global threshold on the conditional probability is chosen that maximizes micro-F_β in the second step.

An algorithm that returns an optimal microF_β-measure for multiclass classification is proposed by Narasimhan et al. (2015). They iteratively build a classifier and corresponding contingency table on training and validation data. At each iteration a new classifier is build by optimizing

a loss defined over the contingency table. Like in the above mentioned algorithms, the classifier is based on the conditional probability estimator and for the consistency results to hold the loss function should be a proper loss.

In this work, we aim to perform empirical risk minimization-type learning, that is, to find a classifier with highest population-level F-measure by maximizing its empirical counterpart. In that sense, we follow the EUM framework. Nonetheless, regardless of how we define the generalization performance, our results can be used to maximize the empirical value of the F_β-measure. Our theoretical results are more general in the sense that there is no assumption regarding the underlying probability distribution nor any particular properties of the loss function.

2.3 Theoretical Framework and Analysis

In this section, we present the theoretical framework which is at the heart of this work. Our results are mainly motivated by the maximization of F-measures for binary, multiclass, and multilabel classification. They rely on a general property of these performance measures, namely their pseudo-linearity with respect to the false negative and false positive probabilities.

For binary classification, we prove that, in order to optimize the F-measure, it is sufficient to solve a binary classification problem with different costs allocated to false positive and false negative errors (Proposition 4). However, these costs are not known *a priori*, so in practice we propose to learn several classifiers with different costs, and to select the best one according to the F-measure in a second step. Propositions 5 and 6 provide approximation guarantees on the F-measure which can be obtained by following this principle.

We first establish the results for the F_β-measures in binary classification, and then extend to other cases of F-measures with similar functional forms that are used in multiclass and multilabel classification. We also briefly describe a pseudo-linear notion of Jaccard index, which can also be solved using our framework (Propositions 5 and 6) . We present the results and proofs for the binary case, succeeded by multiclass and multilabel F-measures.

2.3.1 Error Profiles and Pseudo-Linearity

Error Profiles

The performance of a classifier h on distribution μ can be summarized by the elements of the contingency table (See Table 2.1) which contains

		Predicted Label P	Predicted Label N	Cost Matrix	
Actual Label	P	True Positive (TP)	False Negative (FN)	0	$1+\beta^2-t$
	N	False Positive (FP)	True Negative (TN)	t	0

Table 2.1 – *Contingency and cost table for binary classification*

the summary of errors. For all classification tasks (binary, multiclass and multilabel), the F-measures we consider here are functions of the non-diagonal elements of this contingency table, which themselves are defined in terms of the marginal probabilities of classes and the per-class false negative/false positive probabilities. The marginal probabilities of label k will be denoted by P_k. The per-class false negative/false positive probabilities of a classifier h are denoted by $\texttt{FN}_k(h)$ and $\texttt{FP}_k(h)$, and the per-class true positive/true positive probabilities are denoted by $\texttt{TP}_k(h)$ and $\texttt{TN}_k(h)$ respectively. Their definitions are given below:

$$\begin{aligned}
\textit{(binary/multiclass)}\quad P_k &= \mu(\{(x,y)|y=k\}), & \texttt{FN}_k(h) &= \mu(\{(x,y)|y=k \text{ and } h(x)\neq k\})\\
& & \texttt{FP}_k(h) &= \mu(\{(x,y)|y\neq k \text{ and } h(x)=k\})\\
& & \texttt{TP}_k(h) &= \mu(\{(x,y)|y=k \text{ and } h(x)=k\})\\
& & \texttt{TN}_k(h) &= \mu(\{(x,y)|y\neq k \text{ and } h(x)\neq k\})\\
\textit{(multilabel)}\quad P_k &= \mu(\{(x,y)|y\in k\}), & \texttt{FN}_k(h) &= \mu(\{(x,y)|k\in y \text{ and } k\notin h(x)\})\\
& & \texttt{FP}_k(h) &= \mu(\{(x,y)|y\notin k \text{ and } k\in h(x)\})\\
& & \texttt{TP}_k(h) &= \mu(\{(x,y)|y\in k \text{ and } k\in h(x)\})\\
& & \texttt{TN}_k(h) &= \mu(\{(x,y)|y\notin k \text{ and } k\notin h(x)\})
\end{aligned}$$

The error probabilities of a classifier h ($\texttt{FN}$ and $\texttt{FP}$) can then summarized by the *error profile* $\mathbf{E}(h)$:

$$\mathbf{E}(h) = \big(\texttt{FN}_1(h), \texttt{FP}_1(h), ..., \texttt{FN}_L(h), \texttt{FP}_L(h)\big) \in \mathbb{R}^{2L} \ .$$

Pseudo-Linear Functions

Throughout the paper, we rely on the notion of pseudo-linearity of a function, which itself is defined from the notion of pseudo-convexity (See Cambini and Martein 2009, Definition 3.2.1): a differentiable function $F : \mathcal{D} \subset \mathbb{R}^d \to \mathbb{R}$, defined on a convex open subset of $\mathbb{R}^d$, is *pseudo-convex* if

$$\forall \mathbf{r}, \mathbf{e}' \in \mathcal{D} \ , \ F(\mathbf{r}) > F(\mathbf{e}') \ \Rightarrow \ \big\langle \nabla F(\mathbf{r}), \mathbf{e}' - \mathbf{r} \big\rangle < 0 \ ,$$

where $\langle ., . \rangle$ is the canonical dot product on $\mathbb{R}^d$.

Moreover, F is *pseudo-linear* if both F and $-F$ are pseudo-convex. In practice, working with gradients of non-linear functions may be cumbersome, so we will use the following characterization, which is a rephrasing of Cambini and Martein (2009, Theorem 3.3.9), basically stating that level sets of pseudo-linear functions are hyperplanes:

Theorem 1 (Cambini and Martein 2009) *A non-constant function $F : \mathcal{D} \to \mathbb{R}$, defined and differentiable on the open convex set $\mathcal{D} \subseteq \mathbb{R}^d$, is* pseudo-linear *on $\mathcal{D}$ if and only if $\forall \mathbf{r} \in \mathcal{D}$, $\nabla F(\mathbf{r}) \neq \mathbf{0}$, and: $\exists \mathbf{a} : \mathbb{R} \to \mathbb{R}^d$ and $\exists b : \mathbb{R} \to \mathbb{R}$ such that, for any t in the image of F:*

$$F(\mathbf{r}) \geq t \;\Leftrightarrow\; \langle \mathbf{a}(t), \mathbf{r} \rangle + b(t) \leq 0 \quad \textit{and} \quad F(\mathbf{r}) \leq t \;\Leftrightarrow\; \langle \mathbf{a}(t), \mathbf{r} \rangle + b(t) \geq 0 \ .$$

Pseudo-linearity is the main property of linear-fractional functions (ratios of linear functions).

Proposition 2 (Linear-fractional function) *A linear-fractional function $F : \mathcal{D} \subseteq \mathbb{R}^d \to \mathbb{R}$ is the ratio of linear functions, $F(\mathbf{r}) = \frac{\alpha_0 + \langle \boldsymbol{\gamma}, \mathbf{r} \rangle}{\alpha_1 + \langle \boldsymbol{\delta}, \mathbf{r} \rangle}$. A non-constant linear-fractional function is pseudo-linear on the open half-space $\mathcal{D} = \left\{ \mathbf{r} \in \mathbb{R}^d | \alpha_1 + \langle \boldsymbol{\delta}, \mathbf{r} \rangle > 0,\ \alpha_1 \neq 0 \right\}$.*

Proof A linear-fractional function $F : \mathbf{r} \in \mathbb{R}^d \mapsto \frac{\alpha_0 + \langle \boldsymbol{\gamma}, \mathbf{r} \rangle}{\alpha_1 + \langle \boldsymbol{\delta}, \mathbf{r} \rangle}$, $\alpha_1 + \langle \boldsymbol{\delta}, \mathbf{r} \rangle > 0$ is pseudo-linear.

$$\begin{aligned} F(\mathbf{r}) \leq t \Leftrightarrow & \alpha_0 + \langle \boldsymbol{\gamma}, \mathbf{r} \rangle \leq t(\alpha_1 + \langle \boldsymbol{\delta}, \mathbf{r} \rangle) \\ \Rightarrow & (\alpha_0 - t\alpha_1) + \langle \boldsymbol{\gamma} - t\boldsymbol{\delta}, \mathbf{r} \rangle \leq 0 \end{aligned}$$

Now reversing the inequality, we obtain;

$$F(\mathbf{r}) \geq t \Leftrightarrow (\alpha_0 - t\alpha_1) + \langle \boldsymbol{\gamma} - t\boldsymbol{\delta}, \mathbf{r} \rangle \geq 0$$

The above equations represent open hyperplanes.

$$\nabla F(\mathbf{r}) = \frac{(\alpha_1 + \langle \boldsymbol{\delta}, \mathbf{r} \rangle)\boldsymbol{\gamma} - (\alpha_0 + \langle \boldsymbol{\gamma}, \mathbf{r} \rangle)\boldsymbol{\delta}}{(\alpha_1 + \langle \boldsymbol{\delta}, \mathbf{r} \rangle)^2} \neq 0$$

The gradient term is constant if $\boldsymbol{\delta}$ and $\boldsymbol{\gamma}$ are proportional and non-zero otherwise. The above conditions confirm the requirements for the pseudo-linearity given in Theorem 1 and hence the result. □

2.3.2 Pseudo-Linearity of F-measures

Several notions of F-measures used in practical problems are pseudo-linear. Here, we establish that binary F_β and multiclass/multilabel macro/micro F-measures are pseudo-linear functions.

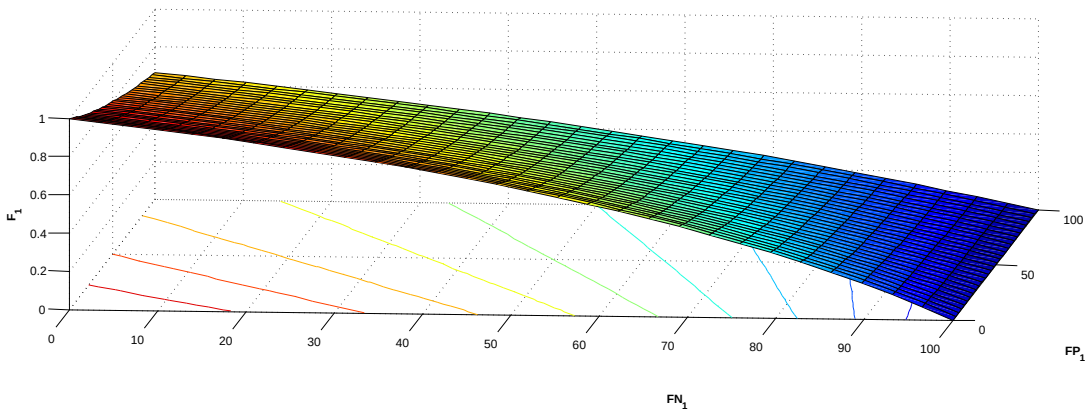


Figure 2.1 – *Surface plot of F_1 as a function of FN_1 and FP_1 with level sets*

Binary Classification

In binary classification, we have $\texttt{FN}_2 = \texttt{FP}_1$ and we can write F-measures only by reference to class 1. Then, for any $\beta > 0$ and any binary classifier h, the F_β-measure is

$$F_\beta(h) = \frac{(1+\beta^2)(P_1 - \texttt{FN}_1(h))}{(1+\beta^2)P_1 + \texttt{FP}_1(h) - \texttt{FN}_1(h)} \ .$$

We can immediately notice that F_β is linear-fractional and hence by Proposition 2 it is pseudo-linear in $\texttt{FN}_1$ and $\texttt{FP}_1$. Thus, with a slight (yet convenient) abuse of notation, we write the F_β-measure for binary classification as a function of vectors in $\mathbb{R}^4 = \mathbb{R}^{2L}$:

$$(\textit{binary}) \qquad \forall \mathbf{r} \in \mathbb{R}^4, F_\beta(\mathbf{r}) = \frac{(1+\beta^2)(P_1 - r_1)}{(1+\beta^2)P_1 + r_2 - r_1}$$

where r_i represents the i^{th} element of the error profile $\mathbf{r}$. A surface plot of F_1 as a function of $\texttt{FN}_1$ and $\texttt{FP}_1$ with level sets is given in Figure 2.1. As stated in Theorem 1, it can be verified from the plot that level sets are hyperplanes.

Multilabel Classification

In multilabel classification, there are several definitions of F-measures. For those based on the error profiles, we first have the macro-F-measure

(denoted by MF_β), which is the average over class labels of the F_β-measure of each binary classification problem associated to the prediction of the presence/absence of a given class:

$$(multilabel\text{–}Macro)MF_\beta(\mathbf{r}) = \frac{1}{L}\sum_{k=1}^{L}\frac{(1+\beta^2)(P_k - r_{2k-1})}{(1+\beta^2)P_k + r_{2k} - r_{2k-1}}$$

MF_β *is not* a pseudo-linear function of an error profile $\mathbf{r}$. However, if the multilabel classification algorithm learns independent binary classifiers for each class (a method known as one-vs-rest or binary relevance, see e.g. Tsoumakas and Katakis 2007), then the k-th binary problem depends only on r_{2k-1} and r_{2k}. The maximization of the macro-F-measure with respect to all binary classifiers is then a separable problem which boils down to independently maximizing the F_β-measure for L binary classification problems. In other words, optimizing MF_β consists in maximizing the pseudo-linear functions in r_{2k-1} and r_{2k} that correspond to each F_β optimization.

There are also micro-F-measures for multilabel classification. They correspond to F_β-measures for a new binary classification problem over $\mathcal{X} \times \mathcal{L}$, in which one maps a multilabel classifier $h\colon \mathcal{X} \to \mathcal{Y}$ ($\mathcal{Y}$ is here the power set of $\mathcal{L}$) to the following binary classifier $\tilde{h}\colon \mathcal{X} \times \mathcal{L} \to \{0,1\}$: we have $\tilde{h}(x,k) = 1$ if $k \in h(x)$, and 0 otherwise. The micro-F_β-measure, written as a function of an error profile $\mathbf{r}$ and denoted by $mF_\beta(\mathbf{r})$, is the F_β-measure of $\tilde{h}$ and can be written as:

$$(multilabel\text{–}micro)mF_\beta(\mathbf{r}) = \frac{(1+\beta^2)\sum_{k=1}^{L}(P_k - r_{2k-1})}{(1+\beta^2)\sum_{k=1}^{L}P_k + \sum_{k=1}^{L}(r_{2k} - r_{2k-1})}$$

This function is also linear-fractional, and thus pseudo-linear in $\mathbf{r}$.

Multiclass Classification

The last example we take is from multiclass classification. It differs from multilabel classification in that a single class must be predicted for each example. This restriction imposes strong global constraints that make the multiclass classification significantly harder. As for the multilabel case, there are many definitions of F-measures for multiclass classification, and in fact several definitions for the micro-F-measure itself. We will focus on the following one, which is used in information extraction (e.g in the BioNLP Challenge Kim et al. 2013). Given L class labels, we will assume that label 1 corresponds to a "default" class, the prediction of which is considered as not important. In information extraction, the default class corresponds to the (majority) case where no information should be extracted. Then, a false negative is an example (x,y) such that $y \neq 1$ and $h(x) \neq y$, while a false positive is an example (x,y) such that $y = 1$ and

$h(x) \neq y$. This micro-F-measure, denoted mcF_β can be written as:

$$^{(multiclass\text{–}micro)}mcF_\beta(\mathbf{r}) = \frac{(1+\beta^2)(1-P_1-\sum_{k=2}^{L} r_{2k-1})}{(1+\beta^2)(1-P_1)-\sum_{k=2}^{L} r_{2k-1}+r_1}$$

Once again, this kind of micro-F_β-measure is linear-fractional and hence pseudo-linear in $\mathbf{r}$.

Remark 3 (Non-pseudo-linear F-measures) *In multilabel settings, the notion of instance-wise F_β has been used in the past (Fujino et al. 2008, Dembczynski et al. 2011, Petterson and Caetano 2010; 2011, Cheng et al. 2012, Dembczynski et al. 2013). It is similar to the micro-F-measure (mF_β) for multilabel case defined above, but defined over samples (instances) instead of labels. It is defined as the average of the per-instance F-measure. Hence, we calculate the F-measures for each instance independently (i.e. estimate mF_β for each individual example by calculating tp,fp,fn for each example in the sample) and take the average (arithmetic mean) over the number of samples. This measure can not be written as a linear-fractional function of "error profile" terms, hence it can not be solved using our framework.*

2.3.3 Optimizing F-Measure by Reduction to Cost-Sensitive Classification

The F_β-measures presented above are non-linear aggregations of false negative/positive proportions that can not be written in the usual expected loss minimization framework; usual learning algorithms are thus, intrinsically, not designed to optimize this kind of performance measures. We show in Proposition 4 that the optimal classifier for a cost-sensitive classification problem with label dependent costs (Elkan 2001, Zhou and Liu 2010) is also an optimal classifier for the pseudo-linear F-measures (within a specific, yet arbitrary classifier set $\mathcal{H}$). In cost-sensitive classification, each entry of the error profile is weighted asymmetrically by a non-negative cost, and the goal is to minimize the weighted average error. Efficient, consistent algorithms exist for such cost-sensitive problems (Abe et al. 2004, Steinwart 2007, Scott 2012). Even though the costs corresponding to the optimal F-measure are not known *a priori*, we show in Proposition 5 that we can approximate the optimal classifier with approximate costs. These costs, explicitly expressed in terms of the optimal F-measure, motivate a practical algorithm. Even though the discussion in this section is more general and applies to any pseudo-linear functions, we start with the discussion in the binary setting. We give the proofs and results for binary F_β and extend the results to multilabel and multiclass F-measures in Section 2.3.4.

Reduction to Cost-Sensitive Classification

Let $F : \mathcal{D} \subset \mathbb{R}^d \to \mathbb{R}$ be a fixed pseudo-linear function. We denote by $\mathbf{a} : \mathbb{R} \to \mathbb{R}^d$ the function mapping values of F to the corresponding level set of Theorem 1. We assume that the distribution μ is fixed, as well as the (arbitrary) set of classifier $\mathcal{H}$. We denote by $\mathcal{E}(\mathcal{H})$ the closure of the image of $\mathcal{H}$ under $\mathbf{E}$, i.e. $\mathcal{E}(\mathcal{H}) = cl(\{\mathbf{E}(h), h \in \mathcal{H}\})$ (the closure ensures that $\mathcal{E}(\mathcal{H})$ is compact and that minima/maxima are well-defined), and we assume $\mathcal{E}(\mathcal{H}) \subseteq \mathcal{D}$. Finally, for the sake of discussion with cost-sensitive classification, we assume that $\mathbf{a}(t) \in \mathbb{R}^d_+$ for any $\mathbf{r} \in \mathcal{E}(\mathcal{H})$, that is, lower values of errors entail higher values of F.

Proposition 4 *Let* $F^\star = \max_{\mathbf{r} \in \mathcal{E}(\mathcal{H})} F(\mathbf{r})$. *We have:* $\mathbf{r}^\star \in \operatorname{argmin}_{\mathbf{r} \in \mathcal{E}(\mathcal{H})} \left\langle \mathbf{a}\left(F^\star\right), \mathbf{r} \right\rangle \;\Leftrightarrow\; F(\mathbf{r}^\star) = F^\star$.

Proof Let $\mathbf{r}^\star \in \operatorname{argmax}_{\mathbf{e}' \in \mathcal{E}(\mathcal{H})} F(\mathbf{e}')$, and let $\mathbf{a}^\star = \mathbf{a}(F(\mathbf{r}^\star)) = \mathbf{a}\left(F^\star\right)$. We first notice that pseudo-linearity implies that the set of $\mathbf{r} \in \mathcal{D}$ such that $\langle \mathbf{a}^\star, \mathbf{r} \rangle = \langle \mathbf{a}^\star, \mathbf{r}^\star \rangle$ corresponds to the level set $\{\mathbf{r} \in \mathcal{D} | F(\mathbf{r}) = F(\mathbf{r}^\star) = F^\star\}$. Thus, we only need to show that $\mathbf{r}^\star$ is a minimizer of $\mathbf{e}' \mapsto \langle \mathbf{a}^\star, \mathbf{e}' \rangle$ in $\mathcal{E}(\mathcal{H})$. To see this, we notice that pseudo-linearity of F (see Theorem 1) implies

$$\forall \mathbf{e}' \in \mathcal{D},\; F(\mathbf{r}^\star) \geq F\left(\mathbf{e}'\right) \Rightarrow \langle \mathbf{a}^\star, \mathbf{r}^\star \rangle \leq \left\langle \mathbf{a}^\star, \mathbf{e}' \right\rangle \;\;,$$

and since $\mathbf{r}^\star$ maximizes F in $\mathcal{E}(\mathcal{H})$, we get $\mathbf{r}^\star \in \operatorname{argmin}_{\mathbf{e}' \in \mathcal{E}(\mathcal{H})} \langle \mathbf{a}^\star, \mathbf{e}' \rangle$. □

This proposition shows that $\mathbf{a}\left(F^\star\right)$ are the cost vectors, which are orthogonal to the level set of F at $F^\star$ and may not need to be unique, that should be assigned to the error profile in order to find the optimal classifier in $\mathcal{H}$ with respect to the measure F. Hence maximizing F amounts to minimizing $\left\langle \mathbf{a}\left(F^\star\right), \mathbf{E}(h) \right\rangle$ with respect to h, that is, amounts to solving a cost-sensitive classification problem. This observation suggests that the optimization of pseudo-linear measures could be a wrapper of cost-sensitive classification algorithms. The costs $\mathbf{a}\left(F^\star\right)$ are, however, not known *a priori*. The following result shows that having only approximate costs is sufficient to have an approximately optimal solution, which gives us the main step towards a practical solution.

Proposition 5 *Let* $\varepsilon_0 \geq 0$ *and* $\varepsilon_1 \geq 0$, *and assume that there exists* $\Phi > 0$ *such that for all* $\mathbf{r}, \mathbf{e}' \in \mathcal{E}(\mathcal{H})$ *satisfying* $F(\mathbf{e}') > F(\mathbf{r})$, *we have:*

$$F\left(\mathbf{e}'\right) - F(\mathbf{r}) \leq \Phi \left\langle \mathbf{a}\left(F(\mathbf{e}')\right), \mathbf{r} - \mathbf{e}' \right\rangle \;\;. \tag{2.1}$$

Then, let us take $\mathbf{r}^\star \in \operatorname{argmax}_{\mathbf{e}' \in \mathcal{E}(\mathcal{H})} F(\mathbf{e}')$, *and denote* $\mathbf{a}^\star = \mathbf{a}(F(\mathbf{r}^\star))$. *Let furthermore* $\hat{\mathbf{a}} \in \mathbb{R}^d_+$ *and* $h \in \mathcal{H}$ *satisfying the following conditions:*

(i) $\|\hat{\mathbf{a}} - \mathbf{a}^\star\|_2 \leq \varepsilon_0$, *(ii)* $\langle \hat{\mathbf{a}}, \mathbf{r} \rangle \leq \min_{\mathbf{e}' \in \mathcal{E}(\mathcal{H})} \left\langle \hat{\mathbf{a}}, \mathbf{e}' \right\rangle + \varepsilon_1$.

We have: $\forall \mathbf{r} \in \mathcal{E}(\mathcal{H}),\ F(\mathbf{r}) \geq F(\mathbf{r}^\star) - \Phi \cdot (2\varepsilon_0 M + \varepsilon_1)$, *where* $M = \max_{\mathbf{e}' \in \mathcal{E}(\mathcal{H})} \|\mathbf{e}'\|_2$.

Proof Let $\mathbf{e}' \in \mathcal{E}(\mathcal{H})$, we can write $\langle \hat{\mathbf{a}}, \mathbf{e}' \rangle = \langle \mathbf{a}^\star, \mathbf{e}' \rangle + \langle \hat{\mathbf{a}} - \mathbf{a}^\star, \mathbf{e}' \rangle$.

Applying Cauchy-Schwarz inequality and condition *(i)*, we get

$$\begin{aligned} \langle \hat{\mathbf{a}}, \mathbf{e}' \rangle &\leq \langle \mathbf{a}^\star, \mathbf{e}' \rangle + \|\hat{\mathbf{a}} - \mathbf{a}^\star\|_2 \|\mathbf{e}'\|_2 \\ &\leq \langle \mathbf{a}^\star, \mathbf{e}' \rangle + \varepsilon_0 M \ . \end{aligned}$$

In particular, we have:

$$\begin{aligned} \min_{\mathbf{e}' \in \mathcal{E}(\mathcal{H})} \langle \hat{\mathbf{a}}, \mathbf{e}' \rangle &\leq \min_{\mathbf{e}' \in \mathcal{E}(\mathcal{H})} \langle \mathbf{a}^\star, \mathbf{e}' \rangle + \varepsilon_0 M \\ &\leq \langle \mathbf{a}^\star, \mathbf{r}^\star \rangle + \varepsilon_0 M \ , \end{aligned} \tag{2.2}$$

since $\mathbf{r}^\star \in \operatorname{argmin}_{\mathbf{e}' \in \mathcal{E}(\mathcal{H})} \langle \mathbf{a}^\star, \mathbf{e}' \rangle$ as shown in Proposition 4.

Similarly, we have $\langle \mathbf{a}^\star, \mathbf{r} \rangle = \langle \hat{\mathbf{a}}, \mathbf{r} \rangle + \langle \mathbf{a}^\star - \hat{\mathbf{a}}, \mathbf{r} \rangle$; applying Cauchy-Schwarz and conditions *(i)* and *(ii)*, we have:

$$\begin{aligned} \forall \mathbf{r} \in \mathcal{E}(\mathcal{H}),\ \langle \mathbf{a}^\star, \mathbf{r} \rangle &\leq \langle \hat{\mathbf{a}}, \mathbf{r} \rangle + \|\mathbf{a}^\star - \hat{\mathbf{a}}\|_2 \|\mathbf{r}\|_2 \\ &\leq \langle \hat{\mathbf{a}}, \mathbf{r} \rangle + \varepsilon_0 M \\ &\leq \min_{\mathbf{e}' \in \mathcal{E}(\mathcal{H})} \langle \hat{\mathbf{a}}, \mathbf{e}' \rangle + \varepsilon_1 + \varepsilon_0 M \ . \end{aligned} \tag{2.3}$$

Combining Inequalities (2.2) and (2.3), we get

$$\begin{aligned} &\forall \mathbf{r} \in \mathcal{E}(\mathcal{H}),\ \langle \mathbf{a}^\star, \mathbf{r} \rangle \leq \langle \mathbf{a}^\star, \mathbf{r}^\star \rangle + \varepsilon_1 + 2\varepsilon_0 M \\ &\forall \mathbf{r} \in \mathcal{E}(\mathcal{H}),\ \langle \mathbf{a}^\star, \mathbf{r} - \mathbf{r}^\star \rangle \leq \varepsilon_1 + 2\varepsilon_0 M \ , \end{aligned}$$

and the final result follows from Assumption (2.1). □

The above proposition suggests that pseudo-linear measures could be optimized by wrapping cost-sensitive classification, in an inner loop, with an outer loop setting the appropriate costs. This proposition also gives an upper bound on the achievable optimal F-score. This value depends on the size of the maximum error associated with the given hypothesis space M, measured in ℓ_2 sense and the constant Φ. The value of M depends on the selected hypothesis class ($\mathcal{E}(\mathcal{H})$). We call Φ a discretization factor as it defines the granularity of the approximation. It depends on the specific form of F-measure and training sample. We can find an approximately optimal classifier using a procedure, where we search for an approximately optimal cost and associated error profile by iterating through the pre-selected cost interval in small steps. Thus

searching for a cost such that ε_0 is close to zero, we can find an approximately *optimal F classifier*. The ε_1 can be regarded as the approximation guarantee provided by the underlying cost-sensitive classification algorithm. Practical implementations use convex surrogate loss instead of the non-convex *o-1* loss. A discussion on convex approximation of *o-1 loss* can be found in (Rosasco et al. 2004). The discretization factor, Φ gives the magnitude of the step size. A larger value of Φ indicates more fine-grained discretization (very small step size), and a smaller value of Φ indicates coarse- grained discretization. Later, we will derive the exact values of Φ and the cost interval for specific F-measures.

Discretization Factor and Cost Interval for F_β

Here, we derive the values of the discretization factor (Φ) and the range of the cost interval ($\mathbf{a}$) for binary F_β-measure.

Proposition 6 *F_β defined in Section 2.3.2 satisfy the conditions of Proposition 5 with:*

$$(\textit{binary})\ \ F_\beta\text{:} \qquad \Phi = \frac{1}{\beta^2 P_1} \qquad \textit{and}\ \ \mathbf{a} : t \in [0,1] \mapsto (1+\beta^2 - t, t, 0, 0)$$

Proof Since F_β is linear-fractional as a function of the error profile, it is pseudo-linear on the open convex set $\{\mathbf{r} \in \mathbb{R}^d | (1+\beta^2)P_1 - e_1 + e_2 > 0\}$ (i.e. when the denominator is strictly positive). Moreover, for every set of classifiers $\mathcal{H}$, we have $\mathcal{E}(\mathcal{H}) \subseteq \mathcal{D}_0 = [O, P_1] \times [0, 1-P_1] \times [1-P_1] \times [1, P_1]$.

Now, by the definition of F_β, we have

$$\forall \mathbf{r} \in \mathcal{D}_0, F_\beta(\mathbf{r}) \leq t \quad \Leftrightarrow \quad (1+\beta^2 - t)r_1 + tr_2 + (1+\beta^2)P_1(t-1) \geq 0$$

and the equation still holds by reversing the inequalities. We thus have that $\mathbf{a}(t) = (1+\beta^2 - t, t, 0, 0)$ satisfy the condition of Theorem 1 (with $b(t) = (1+\beta^2)P_1(t-1)$).

We now show that the condition of Equation 2.1 is satisfied for $\mathbf{a}(t) = (1+\beta^2 - t, t, 0, 0)$ and all $\mathbf{r}, \mathbf{r}' \in \mathcal{D}_0$ by taking $\Phi = \frac{1}{\beta^2 P_1}$. To that end, let $\mathbf{r}$ and $\mathbf{e}'$ in $\mathcal{E}(\mathcal{H})$ and t and t' in $\mathbb{R}$ such that $t' = F_\beta(\mathbf{e}') > F_\beta(\mathbf{r}) = t$. Denote by ε the quantity $\langle \mathbf{a}(t'), \mathbf{r} - \mathbf{e}' \rangle$. Note that $\varepsilon > 0$ and that:

$$0 = \quad \langle \mathbf{a}(t), \mathbf{r} \rangle \quad + b(t) \ = (1+\beta^2 - t)r_1 \ + \ tr_2 \ + (1+\beta^2)P_1(t-1)$$

$$0 = \quad \langle \mathbf{a}(t'), \mathbf{e}' \rangle \quad + b(t') = (1+\beta^2 - t')e'_1 + t'e'_2 + (1+\beta^2)P_1(t'-1)$$

$$\varepsilon = \langle \mathbf{a}(t'), \mathbf{r} - \mathbf{e}' \rangle \qquad\qquad = (1+\beta^2 - t')r_1 + t'r_2 + (1+\beta^2)P_1(t'-1)$$

where the first two equalities are given by the definition of hyperplane corresponds to $F_\beta(\mathbf{r}) = t$ and $F_\beta(\mathbf{e}') = t'$, and the last one is obtained from the definition of $\langle \mathbf{a}(t'), \mathbf{r} - \mathbf{e}' \rangle$. Taking the difference of the third and first equality, we obtain:

$$\varepsilon = (t - t')r_1 + (t' - t)r_2 + (1 + \beta^2)P_1(t' - t)$$

From which we get, since $(1 + \beta^2)P_1 - r_1 + r_2 > 0$ for $\mathbf{r} \in \mathcal{D}_0$:

$$F_\beta(\mathbf{e}') - F_\beta(\mathbf{r}) = t' - t = \varepsilon\big((1 + \beta^2)P_1 - r_1 + r_2\big)^{-1} \leq \frac{\varepsilon}{\beta^2 P_1} \ ,$$

because $\beta^2 P_1$ the minimum of $(1 + \beta^2)P_1 - r_1 + r_2$ on $\mathcal{D}_0$ (taking $r_1 = P_1$ and $r_2 = 0$). We obtain the result since $\varepsilon = \langle \mathbf{a}(t'), \mathbf{r} - \mathbf{e}' \rangle$ by definition. □

This proposition gives the exact values of Φ and the range for $\mathbf{a}$ in binary settings. Here, the discretization factor depends on the marginal probability of the positive class (assuming label 1 represents positive class). A larger value of the discretization factor demands smaller step size in the cost interval. Looking at the approximation guarantee in Proposition 5, with a larger value of Φ, reasonable approximation can be obtained by taking ε_0 close to zero. Intuitively, we can think of this as follows, higher values of Φ indicates a highly imbalanced data with very few positive examples, hence to eliminate the influence of class-imbalance, we need to discretize in smaller step through cost interval. Given the error profile (in the form of contingency table) and associated costs as a matrix, as shown in in Figure 2.1, corresponding F_β-measure is the sum of the elements of the Hadamard product of the two matrices.

Corollary 7 *For the F_1-measure, the optimal classifier is the solution to the cost-sensitive binary classifier with costs $\left(1 - \frac{F^\star}{2}, \frac{F^\star}{2}\right)$*

Proof From Proposition 4, by putting $\beta = 1$, we have

$$(2 - F^\star)r_1 + r_2 F^\star + 2P_1(F^\star - 1) \geq 0$$

dividing by 2, we get

$$(1 - \frac{F^\star}{2})r_1 + r_2\frac{F^\star}{2} + P_1(F^\star - 1) \geq 0$$

Cost vector, $\mathbf{a}(t)$, according to Theorem 1 is $(1 - \frac{F^\star}{2}, \frac{F^\star}{2})$. □

This proposition extends the result obtained by Lipton et al. (2014) to the non-asymptotic regime. If we take $\mathcal{H}$ as the set of all measurable functions, the Bayes-optimal classifier for this cost is to predict class 1 when $\mu(y = 1|x) \geq \frac{F^\star}{2}$ (see Lipton et al. 2014, Steinwart 2007).

Algorithm 1: Optimization of the F_β-measure

Input : Training Data D, β

$F^* = 0$;

Split Training Data into two D_{tra}, D_{val};

for $t = (0 \ldots 1 + \beta^2)$; // actual cost

do

  h, δ, F = F_Cost_Sensitive_Learner($D_{tra}, D_{val}, t, \beta$) ; // learn cost-sensitive model. It returns the model h, an optimal threshold δ and corresponding F_β score F

  if $F > F^*$ **then**

    $h^* = h$, $\delta^* = \delta$, $F^* = F$;

Output: h^*

Algorithm for F_β Maximization

Based on the above results, we give a practical algorithm to find optimal F_β. In case of F_β, the cost function $\mathbf{a} : [0,1] \rightarrow \mathbb{R}^d$, which assigns costs to probabilities of error, is Lipschitz-continuous with Lipschitz constant equal to $\max(1, \beta^2)$. Hence it is sufficient to discretize the interval $[0,1]$ to have a set of evenly spaced values $\{t_1, ..., t_C\}$ (say, $t_{j+1} - t_j = \varepsilon_0/2$) to obtain an ε_0-cover $\{\mathbf{a}(t_1), ..., \mathbf{a}(t_C)\}$ of the possible costs. Using the approximate guarantee of Proposition 5, learning a cost-sensitive classifier (h_i) for each $\mathbf{a}(t_i)$ and selecting the one with minimum total misclassification cost($\langle \mathbf{a}(t_i), h_i(\mathbf{r}) \rangle$) on a validation set is sufficient to obtain a $\Phi(2\varepsilon_0 M + \varepsilon_1)$-optimal solution where ε_1 is the approximation guarantee of the cost-sensitive classification algorithm. Our proposed algorithm is presented in Algorithm 1.

The cost-sensitive classification algorithm that is used in the inner loop of Algorithm 2 returns a cost sensitive classification model on the training set with cost t'. The *get_total_cost* method in Algorithm 2 retruns the total misclassification cost on the validation set w.r.to the actual cost t. The *computeFmeasure* method returns the optimal threshold and corresponding F_β-measure on the validation set. Even though our theoretical results do not suggest thresholding the scores *a posteriori*, experimental results indicate the need for a posterior thresholding of the scores. We will elaborate on this point in Section 2.5. This meta-algorithm can be instantiated with any cost-sensitive learning algorithm (cost_sensitive_learner in Algorithm 2). The actual algorithm may simply consist of adjusting the hyper-parameters of a cost-insensitive classifier so as to optimize cost-sensitive classification, as in many practical implementation of cost-sensitive algorithm. This rudimentary approach results in considerable savings in computational time compared to methods where one has to re-train the algorithm for every parameter settings.

Algorithm 2: F_Cost_Sensitive_Learner

Input : D_{tra} = Training Set, D_{val} = Validation Set, **a**=cost, β

$\mathbf{c}^* = +\infty$;

for $t' = (0 \dots 2(1+\beta^2))$; // surrogate cost

do

  $\hat{h}$ = cost_sensitive_learner(D_{tra}, t') ; // generic cost-sensitive learner

  c= get_total_cost($\hat{h}, D_{val}, t$) ; // get total misclassification cost w.r.to t

  if $\mathbf{c}^* > \mathbf{c}$ **then**

    $\mathbf{c}^* = \mathbf{c}$;

    $h = \hat{h}$;

δ, F= computeFmeasure(h, D_{val}, β) ; // get optimal threshold and corresponding F_β-measure

Output: h, δ, F

2.3.4 Beyond Binary F-measure

As mentioned earlier, many notions of F-measures in multiclass and multilabel problems are pseudo-linear and can be solved using our framework. Here, we derive the values for cost vector and discretization factor, and propose optimal F-measure algorithm for pseudo-linear F-measures described in Sections 2.3.2 and 2.3.2.

Multilabel micro-F-measure

Proposition 8 *multilabel micro-F(mF_β) defined in Section 2.3.2 satisfies the conditions of Proposition 5 with:*

$$(multilabel\text{–}micro)\ \ mF_\beta\text{:}\ \ \Phi = \frac{1}{\beta^2 \sum_{k=1}^{L} P_k}\ \text{ and } a_i(t) = \begin{cases} 1+\beta^2 - t & \text{if } i \text{ is odd} \\ t & \text{if } i \text{ is even} \end{cases}$$

Proof

$$mF_\beta(\mathbf{r}) \leq t \implies \frac{(1+\beta^2)\sum_{k=1}^{L}(P_k - r_{2k-1})}{(1+\beta^2)\sum_{k=1}^{L} P_k + \sum_{k=1}^{L}(r_{2k} - r_{2k-1})} \leq t$$

$$\implies (1+\beta^2 - t)\sum_{k=1}^{L} r_{2k-1} + t\sum_{k=1}^{L} r_{2k} + (1+\beta^2)(t-1)\sum_{k=1}^{L} P_k \geq 0$$

Thus, we have that

$$a_i(t) = \begin{cases} 1+\beta^2 - t & \text{if } i \text{ is odd} \\ t & \text{if } i \text{ is even} \end{cases}$$

Following the same arguments as in Proposition 6, we get

$$mF_\beta(\mathbf{e}') - mF_\beta(\mathbf{r}) = t' - t = \varepsilon \left[(1+\beta^2) \sum_{k=1}^{L} P_k - \sum_{k=1}^{L} r_{2k-1} + \sum_{k=1}^{L} r_{2k} \right]^{-1} \leq \frac{\varepsilon}{\beta^2 \sum_{k=1}^{L} P_k}$$

because $\beta^2 \sum_{k=1}^{L} P_k$ the minimum of $(1+\beta^2) \sum_{k=1}^{L} P_k - \sum_{k=1}^{L} r_{2k-1} + \sum_{k=1}^{L} r_{2k}$ in the respective domain (taking $r_{2k-1} = P_k$ and $r_{2k} = 0$). We obtain the result since $\varepsilon = \langle \mathbf{a}(t'), \mathbf{r} - \mathbf{e}' \rangle$ by definition. □

Here, the discretization factor depends on the sum of marginal probabilities of each label. A large value of Φ indicates that majority of the labels are rare, and smaller value of Φ indicates that few labels are rare. Since the impact of misclassification of rare labels does not influence the micro-F-measure to a greater extend (F-score is independent of true negatives), we have to discretize in a smaller step only if the majority of the classes are rare. Given the above result on cost vector $\mathbf{a}$ and discretization factor Φ, and following the arguments given for F_β (here also the cost function $\mathbf{a}$ is Lipschitz-continuous with Lipschitz constant taking value $max(1, \beta^2)$), we can develop an algorithm for finding optimal classifier for mF_β. Like in binary case, here we run cost-sensitive learner with discretized cost values to find the classifier with lowest total misclassification cost($\langle \mathbf{a}(t_i), h_i(\mathbf{r}) \rangle$). Our proposed algorithm is given in Algorithm 3. The algorithm is similar to the F_β algorithm given in Algorithm 1. The most important thing to note is that the threshold is chosen with respect to all the labels such that it maximizes the mF_β-measure. This observation is theoretically confirmed by Koyejo et al. (2015). We also need the cardinality of the label space as an additional input parameter to estimate the actual and surrogate cost values.

Multiclass micro-F-measure

Proposition 9 *multiclass micro-F(mcF_β) defined in Section 2.3.2 satisfies the conditions of Proposition 5 with:*

$$(multiclass\text{–}micro) \quad mcF_\beta: \quad \Phi = \frac{1}{\beta^2(1-P_1)} \quad and$$

$$a_i(t) = \begin{cases} 1+\beta^2 - t & if\ i\ is\ odd\ and\ i \neq 1 \\ t & if\ i = 1 \\ 0 & otherwise \end{cases}$$

Algorithm 3: Optimization of the mF_β-measure

Input : D = Data, L = $|\mathcal{L}|$, β

$mF^* = 0$;

Split Training Data into two D_{tra}, D_{val} ;

for $t = (0 \ldots 1 + \beta^2)$; `// Actual Cost`

do

  a = gen_mF_β_cost_vector(L, t, β) ; `// get the cost values as given in Proposition 8`

  h, δ, mF = mF_Cost_Sensitive_Learner(D_{tra}, D_{val}, **a**, β) ; `// learn cost-sensitive model which returns the model, the optimal threshold and corresponding` mF_β`-measure`

  if $(mF > mF^*)$ **then**

    $h^* = h$;

    $\delta^* = \delta$;

    $mF^* = mF$;

Output: h^*

Proof

$$mcF_\beta(\mathbf{r}) \leq t \implies \frac{(1+\beta^2)(1-P_1-\sum_{k=2}^{L} r_{2k-1})}{(1+\beta^2)(1-P_1)-\sum_{k=2}^{L} r_{2k-1}+r_1} \leq t$$

$$\implies (1+\beta^2-t)\sum_{k=2}^{L} r_{2k-1} + t r_1 + (1+\beta^2)(t-1)(1-P_1) \geq 0$$

Thus, we have that

$$a_i(t) = \begin{cases} 1+\beta^2-t & \text{if } i \text{ is odd and } i \neq 1 \\ t & \text{if } i = 1 \\ 0 & \text{otherwise} \end{cases}$$

Following the same arguments as in Proposition 6 , we get

$$mcF_\beta(\mathbf{e}') - mcF_\beta(\mathbf{r}) = t' - t = \varepsilon \left[(1+\beta^2)(1-P_1) - \sum_{k=2}^{L} r_{2k-1} + r_1 \right]^{-1} \leq \frac{\varepsilon}{\beta^2(1-P_1)}$$

because $\beta^2(1-P_1)$ the minimum of $(1+\beta^2)(1-P_1)-\sum_{k=2}^{L} r_{2k-1}+r_1$ in the respective domain (taking $\sum_{k=2}^{L} r_{2k-1} = 1-P_1$ and $r_1 = 0$). We obtain the result since $\varepsilon = \langle \mathbf{a}(t'), \mathbf{r} - \mathbf{e}' \rangle$ by definition. □

Following the arguments given for multilabel micro-F-measure, we can use the Algorithm 3 for finding optimal mcF_β with a small modification to the *gen_mF_β_cost_vector* method. The new cost generation method for multiclass micro-F-measure follows result of proposition 9.

Algorithm 4: mF_Cost_Sensitive_Learner

Input : D_{tra} = Training Set, D_{val} = Validation Set, $\mathbf{a}$ = Actual Cost Vector, β

$c^* = +\infty$;
for $\mathbf{a}' = (0 \dots 2a)$; // surrogate cost
do
  $\hat{h}$ = cost_sensitive_learner($D_{tra}, \mathbf{a}'$) ; // generic cost-sensitive learner with surrogate cost
  θ, c= get_total_cost($\hat{h}, D_{val}, \mathbf{a}$) ; // get total misclassification cost w.r.to actual cost
  if $(c^* > c)$ **then**
    $c^* = c$;
    $h = \hat{h}$;
δ, mF= computemFmeasure(h, D_{val}, β) ; // get optimal threshold and corresponding mF_β-measure

Output: h, δ, mF

Remark 10 (Beyond F-Measures) *The Jaccard index is a set-based similarity measure. Given two sets, the Jaccard index is defined as the ratio of intersection to union. Like F_1-measure, it ranges from 0 to 1, where 0 indicates distinct sets and 1 indicates identical sets (Kaufman and Rousseeuw 2009). It is used in cluster analysis and co-citation analysis to name a few applications. Some recent work (Waegeman et al. 2014, Koyejo et al. 2014) examined the use of Jaccard index as a performance measure in classification problems. The Jaccard index is a pseudo-linear performance function of per-class false negatives and false positives. We can define Jaccard indexes for binary, multiclass and multilabel problems in terms of the* error profile *entries,*

$$(binary) \qquad \forall \mathbf{r} \in \mathbb{R}^4, \quad Jac(\mathbf{r}) = \frac{P_1 - r_1}{P_1 + r_2}$$

$$(multilabel\text{–}micro) \qquad \forall \mathbf{r} \in \mathbb{R}^{2L}, \quad mJac(\mathbf{r}) = \frac{\sum_{k=1}^{L}(P_k - r_{2k-1})}{\sum_{k=1}^{L} P_k + \sum_{k=1}^{L} r_{2k}}$$

$$(multiclass\text{–}micro) \qquad \forall \mathbf{r} \in \mathbb{R}^{2L}, \quad mcJac(\mathbf{r}) = \frac{1 - P_1 - \sum_{k=2}^{L} r_{2k-1}}{(1 - P_1) + r_1}$$

As we can see from the above equations, these quantities are pseudo-linear and hence, we can use the methodology developed in Section 2.3.3, thresholding cost-sensitive classifiers, to find the optimal Jaccard index classifier. Our analysis confirms the remark of Waegeman et al. (2014) "We also see that algorithms maximizing the F-measure perform the best for Jaccard index".

2.4 Relationship to Multi-Objective Optimization

Finding "good" classifiers amounts to finding good trade-offs between the different types of errors. In any case, it is a natural requirement that the chosen classifier has an error profile that is a minimal element of $\mathcal{E}(\mathcal{H})$ according to the partial order of Pareto dominance, which is denoted by $\preceq$ and is defined as:

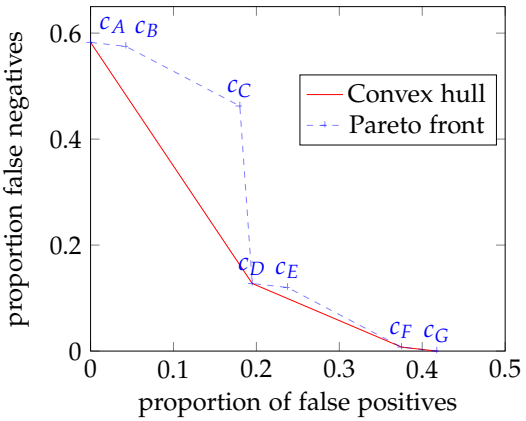
$$\forall \mathbf{r}, \mathbf{e}' \in \mathbb{R}^d \ , \ \mathbf{r} \preceq \mathbf{e}' \ \Leftrightarrow \ \forall k \in \{1, ..., d\} \ , \ r_k \leq e'_k \ .$$

The set of optimal solutions defines the Pareto front (see Chapter 1).

Multi-objective optimization defines methods for finding the Pareto front, or approximations of it (Ehrgott and Gandibleux 2002), and one of the motivations is to find (approximately) optimal solutions of a vector function that is hard to optimize. The process is to generate candidate points in the Pareto front, and take that candidate with optimal value of the vector function. The advantage is generating candidate points is faster than the direct optimization of the vector function. In our case, the goal is to find $h \in \mathcal{E}(\mathcal{H})$ that achieves small values of $\langle \mathbf{a}, \mathbf{r}(h) \rangle$ for a predefined cost vector $\mathbf{a}$.

The reduction from pseudo-linear functions to cost-sensitive classification exactly corresponds to this Pareto front method. In fact, a general way of finding Pareto-optimal solutions of a multi-objective problem is called the weighted-sum method (see e.g. Ehrgott and Gandibleux 2002, Boyd and Vandenberghe 2004). Applied to error profiles, the weighted-sum method would minimize positive weighted combinations of the elements of the error profiles, which corresponds to solving a cost-sensitive classification problem. In usual multi-objective optimization settings, such a Pareto set method is not useful for pseudo-linear aggregation functions, because most such functions are linear-fractional, and single-objective problems with a linear-fractional objective function can be rewritten in terms of a linear objective with linear constraints (see e.g. Boyd and Vandenberghe 2004). In our context however, the linearization would not help because it introduces constraints involving values of the error profiles, which are not linear in general. What we gain with the reduction to cost-sensitive classification (or, equivalently, with the weighted-sum method), is that efficient algorithms for cost-sensitive classification, which are known to work in practice and are asymptotically optimal, are already known. In addition, weighted-sum method requires the user to know the relative preferences of the objectives in advance, which is not known in general. Hence the weight components are unbounded. Our reduction gives approximate values for the possible weight vector $\mathbf{a}(t)$.

The relationship between the cost-sensitive classification and the weighted-sum method allows us to discuss pseudo-linear F-measures in



	x_0	x_1	x_2	
$\mu(x)$	0.65	0.30	0.05	
$\mu(y=1\|x)$	0.70	0.40	0.15	
classifier	x_0	x_1	x_2	F_1^μ (%)
$h_A(x)$	2	2	2	2.22
$h_B(x)$	2	2	1	2.37
$h_C(x)$	2	1	2	27.22
$h_D(x)$	1	2	2	73.83
$h_E(x)$	1	2	1	72.12
$h_F(x)$	1	1	2	75.24
$h_G(x)$	1	1	1	73.62

Figure 2.2 – *Pareto front for a binary classification problem ($\mathcal{Y} = \{1, 2\}$, the positive class is 1), where the input space contains three points x_1, x_2, x_3. The table on the right describes the data distribution, and defines the 8 possible classifiers and gives their F_1^μ-measure.*

terms of Pareto-optimal solutions. It is well-known that in general, not all Pareto-optimal solutions can be found by the weighted-sum method; in fact, only those that are on the boundary of the *convex hull* of the feasible set can be reached. In general however, many classification problems have Pareto-optimal solutions that do not lie on this boundary, especially if the input space is finite (as is the case on any finite dataset). Figure 2.2 gives the example of the Pareto front of a binary classification problem with 3 examples. The Pareto front can be depicted on a 2D plane where the axis are false positives and false negatives; up to a change of basis, this Pareto front is the ROC curve (Bach et al. 2006, Clémençon and Vayatis 2009) for the problem. In the figure, the blue points on the left plot correspond to Pareto-optimal classifiers (none of them can be improved both in terms of proportion of false positives and false negatives), while the red curve is the Pareto set of the convex hull of the error profiles of the 8 classifiers. Our result of reduction to cost-sensitive classification proves that only the classifiers whose error profile is both Pareto-optimal and on the boundary of the convex hull are candidates as optimal classifiers for any pseudo-linear aggregation function (here, the candidates are c_A, c_D, c_F), even though all classifiers are optimal for some trade-off rule. For instance, c_B is the optimal classifier for the rule "minimize the proportion of false negatives under the constraint that the proportion of false positives is smaller than 0.1".

2.5 Experiments

This section assesses the accuracy of the algorithms suggested by our theoretical framework, using the F_1-measure, in binary and multilabel classification. Our experimental results for binary and multilabel-macro F-measure (using binary relevance) shows that *(i)* choosing a classifier by minimizing $\langle \mathbf{a}, \mathbf{r} \rangle$ results in classifier with optimal F-measure (*ii*) thresholding the class conditional probabilities or the classification scores of the cost-sensitive classification often results in classifier with optimal F-measure.

We compare thresholded cost-sensitive classification, as implemented by SVMs and logistic regression (LR), with asymmetric costs, to thresholded linear classifiers (SVMs and logistic regression, with a decision threshold set *a posteriori* by maximizing the F_1-score on the validation set). Besides, the structured SVM approach to F_1-measure maximization of Joachims (2005), SVM$^{\text{perf}}$, provides another baseline. For completeness, we also report results for non-thresholded cost-sensitive SVMs, non-thresholded cost-sensitive logistic regression, and for the thresholded versions of SVM$^{\text{perf}}$.

Since the practical cost-sensitive algorithms are based on convex surrogate loss optimization (Scott 2012), the approximate cost approximation we presented in Proposition 5 will not hold in general. We call the cost given in Proposition 5 as actual cost (test cost) and cost used in the practical surrogate loss based algorithm as surrogate cost (training cost) (Bach et al. 2006). Since there is no one-to-one mapping between actual cost and surrogate cost, in practical implementations we have to iterate over the convex surrogate cost for each value of the actual cost.

SVM and LR differ in the loss they optimize (weighted hinge loss for SVMs, weighted log-loss for LR), and even though both losses are calibrated in the cost-sensitive setting (that is, converging toward a Bayes-optimal classifier as the number of examples and the capacity of the class of function grow to infinity) (Steinwart 2007), they behave differently on finite datasets or with restricted classes of functions. We may also note that asymptotically, the Bayes-classifier for a cost-sensitive binary classification problem is a classifier which thresholds the posterior probability of being class 1. Thus, all methods but SVM$^{\text{perf}}$ are asymptotically equivalent, and our goal here is to analyze their non-asymptotic behavior on a restricted class of functions.

For each experiment, the training set was split at random, keeping $1/3$ for the validation set used to select all hyper-parameters, based on the maximization of the F_1-measure on this set. For datasets that do not come with a separate test set, the data was first split to keep $1/4$ for test. All results are averaged over five random splits i.e. hold-out validation with five random splits. The algorithms have from one to four hyper-parameters: *(i)* all algorithms are run with L_2 regularization, with a regu-

larization parameter $C \in \{2^{-6}, 2^{-5}, ..., 2^{6}\}$; *(ii)* for the cost-sensitive algorithms, the cost for false negatives is chosen in $\{\frac{2-t}{t}, t \in \{0.1, 0.2, ..., 1.9\}\}$ of Proposition 4 [1]; *(iii)* for the thresholded algorithms, the threshold is chosen among all the scores of the validation examples; *(iv)* for kernel based SVM, we used radial basis function (RBF) kernel with γ (measure of influence of a single training example) value $\gamma \in \{2^{-6}, 2^{-5}, ..., 2^{6}\}$.

The library *LIBLINEAR* (Fan et al. 2008) was used to implement non-kernel SVMs[2] and logistic regression. *LIBSVM* (Chang and Lin 2011) library was used for the kernel SVM. A constant feature with value 100 (to simulate an unregularized offset) was added to each dataset.

2.5.1 Importance of Thresholding

Although our theoretical developments do not indicate any need to threshold the scores of classifiers, the practical benefits of a post-hoc adjustment of these scores can be important in terms of F_1-measure maximization, as already noted in cost-sensitive learning scenarios (Grandvalet et al. 2005, Bach et al. 2006). Recent study also indicated the importance of thresholding when proper losses are used for binary and multilabel classifications (Koyejo et al. 2014, Narasimhan et al. 2014, Koyejo et al. 2015). We study the importance thresholding classification scores *a posteriori* using a didactic data called "Galaxy". The data can be visualized as given in Figure 2.3. The data distribution consist in four clusters of 2D-examples, indexed by $z \in \{1, 2, 3, 4\}$, with prior probability $\mu(z = 1) = 0.01$, $\mu(z = 2) = 0.1$, $\mu(z = 3) = 0.001$, and $\mu(z = 4) = 0.889$, with respective class prior probabilities $\mu(y = 1|z = 1) = 0.9$, $\mu(y = 1|z = 2) = 0.09$, $\mu(y = 1|z = 3) = 0.9$, and $\mu(y = 1|z = 4) = 0$. "Galaxy" is an example of highly imbalanced dataset.

We drew a very large sample (100,000 examples) from the distribution, whose optimal F_1-measure is 67.5%. Without thresholding the scores of the classifiers, the best F_1-measure among the classifiers is 58.0%, obtained by cost-sensitive SVM, whereas tuning thresholds enables to reach the optimal F_1-measure for SVM$^{\text{perf}}$ and cost-sensitive SVM. On the other hand, LR is severely affected by the non-linearity of the level sets of the posterior probability distribution, and does not reach this limit (best F_1-measure of 56.5%). Note also that, even with this very large sample size, the SVM and LR classifiers are very different. This result suggests that thresholding the classification scores *a posteriori* may improve the optimal F-scores, especially thresholding the cost-sensitive classifier scores.

[1] We take t greater than 1 in case the training asymmetry would be different from the true asymmetry (Bach et al. 2006).

[2] The maximum number of iteration for SVMs was set to 50,000 instead of the default 1,000.

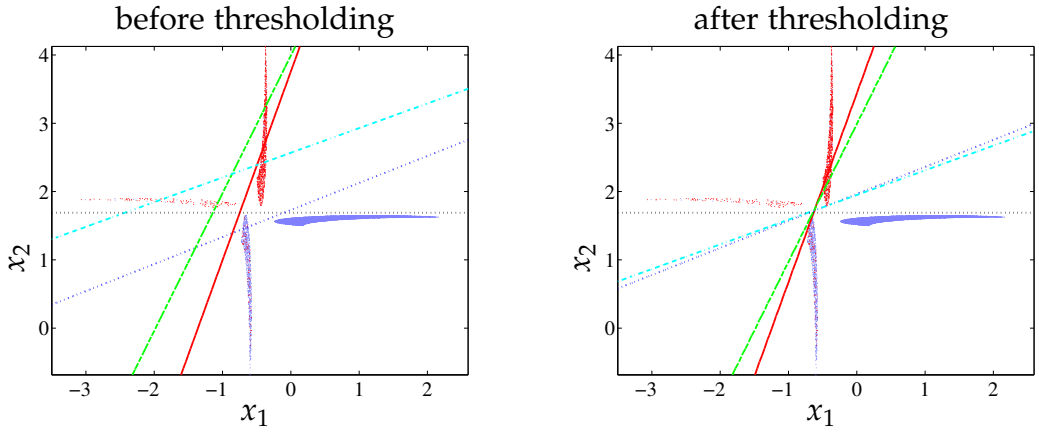


Figure 2.3 – *Decision boundaries for the galaxy dataset before and after thresholding the classifier scores of SVM^{perf} (dotted, blue), weighted SVM (dot-dashed, cyan), unweighted logistic regression (solid, red), and weighted logistic regression (dashed, green). The horizontal black dotted line is an optimal decision boundary.*

Name	Type	Labels	Train	Test	Features	Label Freq. (%) (min/max)
`Adult`	binary	2	32,561	16,281	123	0.32
`Galaxy`	binary	2	18,000	7,000	2	0.02
`RCV1`	multilabel	101	23,149	10,000	47,236	0.008/46.6
`Scene`	multilabel	6	1,211	1,196	294	13.6/22.8
`Siam`	multilabel	22	21,519	7,077	30,438	1.4/59.8
`Yeast`	multilabel	14	1,500	917	103	25.2/43.0

Table 2.2 – *Attributes of the Dataset*

2.5.2 Binary F_β and Multilabel MF_β

The other datasets we use are `Adult`, `RCV1`, `Scene`, `Siam` and `Yeast`. In addition, we used a subsample from the `Galaxy` data to demonstrate the empirical validity of the algorithm. `Adult`, `RCV1` and `Yeast` are obtained from the UCI repository[3], and `Scene` and `Siam` from the *Libsvm* repository[4]. The attributes of the data used in our empirical study are given in Table 2.2.

The results for binary-F_β and multilabel-macro-F (MF_β) are reported in Table 2.3 and 2.4 respectively. As it is evident from the experimental results, cost-sensitive learning and thresholded cost-sensitive learning give optimal results, whereas other methods performs suboptimally. But the difference between methods is less extreme than on the artificial Galaxy dataset. The `Adult` dataset is an example where all methods perform nearly identical; the surrogate loss used in practice seems unimportant. On the other datasets, we observe that thresholding has a relatively large

[3] `https://archive.ics.uci.edu/ml/datasets.html`

[4] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multilabel.html`

Table 2.3 – *F_1-measures (in %) for baseline algorithms with their usual settings (–) and different options: T for thresholded classification scores, CS for cost-sensitive training, CS&T for cost-sensitive training and thresholded classification scores*

Baseline	SVM^{perf}		SVM				LR			
Options	–	T	–	T	CS	CS&T	–	T	CS	CS&T
Adult	67.3	67.3	66.9	67.5	**67.9**	67.8	65.0	67.7	67.7	**67.9**
Galaxy	48.4	61.7	43.1	61.4	58.0	**62.0**	35.4	51.9	41.8	56.5

Table 2.4 – *Macro-F_1-measures MF_1 (in %) for baseline algorithms with their usual settings (–) and different options: T for thresholded classification scores, CS for cost-sensitive training, CS&T for cost-sensitive training and thresholded classification scores*

Baseline	SVM^{perf}		SVM				LR			
Options	–	T	–	T	CS	CS&T	–	T	CS	CS&T
RCV1	44.0	52.8	46.6	54.2	50.9	**54.5**	40.9	52.9	48.5	53.3
Scene	68.3	69.6	66.2	69.6	69.6	69.6	67.0	69.9	69.8	**70.1**
Siam	48.2	52.8	48.1	52.4	52.7	**53.4**	44.7	51.9	51.7	52.2
Yeast	46.4	46.4	39.1	46.2	47.2	46.3	38.8	**47.4**	**47.4**	47.2

impact, especially for SVM^{perf} and cost-insensitive classifiers. The unthresholded and cost-insensitive SVM and LR results are very poor compared to thresholded and cost-sensitive versions. The cost-sensitive classifiers (thresholded and unthresholded) outperforms all other methods, as suggested by the theory. Te cost-sensitive SVM is probably the method of choice to optimize binary-F_β or multilabel-macro-F(MF_β) when predictive performance is a must. On these datasets, thresholded LR still performs reasonably well considering its relatively low computational cost. In general, on the computational cost front, LR converges faster than SVM or SVM^{perf}.

Table 2.5 presents the optimal MF_β-measure with kernel SVM. We used Radial Basis Function (RBF) as the kernel function and trained RBF SVM without a bias term. Our experiments exemplify our theoretical findings in kernel settings. In case of Scene, thresholding the cost-sensitive scores marginally improves the MF_1-score whereas in case of Yeast data, cost-sensitive kernel SVM outperforms other methods. In both cases, thresholding the cost-insensitive scores deteriorates the MF_1-scores.

2.5.3 Multilabel mF_β

In case of multilabel-micro-F-measure, we compare our algorithm with a commonly used method to find best mF_β-score suggested by Fan and Lin (2007). In the proposed method, one assumes that an optimal classifier for

Table 2.5 – *Macro-F_1-measures MF_1 (in %) for SVM with RBF kernel with their usual settings (–) and different options: T for thresholded classification scores, CS for cost-sensitive training, CS&T for cost-sensitive training and thresholded classification scores*

Options	–	T	CS	CS&T
Scene	68.9	68.3	70.5	**70.9**
Yeast	48.6	48.5	**48.8**	47.9

Table 2.6 – *Micro-F_1-measures mF_1 (in %) for for baseline algorithms with their usual settings (–) and different options: T for thresholded classification scores, CS for cost-sensitive training, CS&T for cost-sensitive training and thresholded classification scores. Two optimization strategies are compared: C_{min} for mF_1 by proposed algorithm and F_{max} for mF_1 corresponding to optimal MF_1*

Baseline		SVM^{perf}		SVM				LR			
Options		–	T	–	T	CS	CS&T	–	T	CS	CS&T
RCV1	C_{min}	48.2	49.6	47.6	49.7	49.9	**50.2**	46.3	49.8	49.9	49.9
	F_{max}	42.8	44.7	47.6	44.1	49.2	44.2	46.4	44.3	49.3	44.5
Scene	C_{min}	66.7	68.5	65.4	68.7	68.8	68.6	66.6	69.2	68.6	**69.4**
	F_{max}	66.6	68.3	65.2	68.3	68.3	68.3	66.4	69.2	68.6	68.8
Siam	C_{min}	59.2	62.5	60.3	62.2	**62.6**	62.5	60.2	62.4	62.0	62.3
	F_{max}	59.2	62.0	60.1	62.0	62.3	62.2	59.0	61.8	61.9	62.0
Yeast	C_{min}	61.8	65.1	64.1	64.8	**65.6**	65.2	63.3	64.9	65.3	64.9
	F_{max}	60.2	60.2	60.6	59.3	60.7	61.2	63.2	59.8	61.0	60.9

macro-F-measure is an optimal classifier for micro-F-measure. Hence, the micro-F-score corresponds to optimal macro-F-score is deemed as the optimal micro-F-score. We compare our algorithm for micro-F-score against the micro-F-score corresponds to the optimal macro-F-score obtained by running binary relevance as explained in section 2.3.2.

Table 2.6 contains the multilabel-micro-F (mcF_β) results for the multilabel datasets. The results clearly demonstrates that choosing the optimal classifier for macro-F measure (corresponds to F_{max}in table) for maximizing micro-F-measure always return suboptimal results. So in practice, algorithms based on per-label macro-F optimization should be avoided for micro-F optimization. In case of micro-F, effect due to thresholding is not very significant, except for RCV1 data. The unthresholded classifiers performs nearly as good as the thresholded versions. This is true for SVM^{perf} also. As suggested by theory, cost-sensitive classification is the preferred method to optimize multilabel-micro-F. Here also, thresholded LR can be considered as an alternate option considering the computational cost.

Table 2.7 presents the optimal mcF_β-measure with RBF kernel SVM. Similar to the MF_β results, thresholding the cost-sensitive score gives better *mFbeta* results for kernel SVM.

Table 2.7 – *Micro-F_1 for SVM with RBF kernel with their usual settings (–) and different options: T for thresholded classification scores, CS for cost-sensitive training, CS&T for cost-sensitive training and thresholded classification scores. C_{min} for mF_1 by proposed algorithm and F_{max} for mF_1 corresponding to optimal MF_1*

Options		–	T	CS	CS&T
Scene	$C_{\min}$	67.2	67.1	**67.5**	67.1
	$F_{\max}$	67.0	67.0	67.2	67.4
Yeast	$C_{\min}$	65.9	66.3	66.3	**66.6**
	$F_{\max}$	59.4	62.9	59.9	63.5

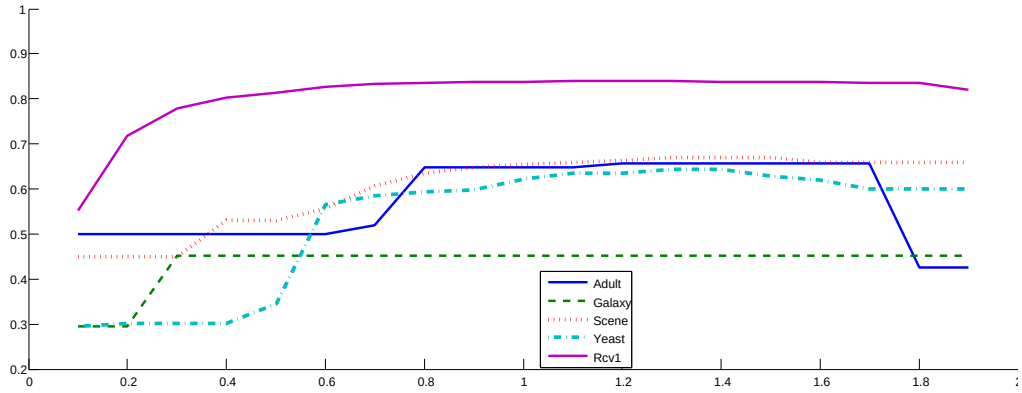


Figure 2.4 – *Plot of micro-F-measure against false negative cost*

2.5.4 Cost Space Search Overhead

Since the actual cost associated with misclassification differs from the cost associated with surrogate loss, it introduces an extra loop in our algorithm. Hence searching for optimal cost vector in the discretized cost interval might not be a feasible approach, especially when the value of Φ is large. A simple workaround is to disregard the difference between the classifier performance with actual cost and the classifier performance with surrogate cost. But this will result in choosing suboptimal classifier as pointed out by Bach et al. (2006).

The unimodularity of the F_β-measure (a pseudo-linear function is pseudo-convex) with respect to the costs suggests a way to limit the search over the cost space using bracketing based approaches. By using The bracketing based approach, we limit the search over specific interval of cost space by keeping track of the F_β-score obtained at each iteration. Figure 2.4 contains the plot of micro-F-measure against varying false negative cost. An idea similar to this, based on bisection, is used by Narasimhan et al. (2015) in case of multi-class micro-F score.

Bracketing methods (Press et al. 2007) are extensively used to find global maxima of unimodal functions like quasi-concave function. We will not be able to use the exact bracketing algorithm to find the optimal cost, since it requires the knowledge of *error profile* associated with each

value of F-measure). But we can use the idea of bracketing to limit the discretization interval.

Here, we find three cost vectors $(\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r})$, such that $F(\boldsymbol{p}) < F(\boldsymbol{q}) > F(\boldsymbol{r})$, then instead of discretizing the whole interval, we can limit the discretization only to the sub-interval $(\boldsymbol{p}, \boldsymbol{r})$. We start with two intervals defined by the three points: start of the interval (0), median of the interval $((1 + \beta^2))$ and the end of the interval $(2(1 + \beta^2))$. Then we search for the triplets $(\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r})$ within the two subintervals recursively. We could use binary search to search for the subinterval containing the approximately optimal F-measure. Depending up on the F-measure values obtained at each subinterval, we limit the discretization only to the corresponding cost interval. In the best case, we requires exponentially fewer cost values.

2.6 Conclusion

We presented an analysis of F-measures, leveraging the property of pseudo-linearity of specific notions of F-measures to obtain a strong non-asymptotic reduction to cost-sensitive classification. The results hold on any dataset, for any class of function and on any data distribution. We suggested algorithms for F-measure optimization based on minimizing the total misclassification cost of the cost-sensitive classification. We demonstrated experiments on linear classifiers, showing the theoretical interest of using cost-sensitive classification algorithms rather than probability thresholding. It is also shown that for F-measure maximization, thresholding even the cost-sensitive algorithms helps to achieve good performances.

Empirically and algorithmically, we only explored the simplest case of our result (F_β-measure in binary classification and macro-F_β-measure and micro-F_β-measure in multilabel classification), but much more remains to be done. Algorithms for the optimization of the non-pseudo-linear notions of F-measures like instance-wise-F_β-measure in multilabel classification received interest recently as well (Dembczynski et al. 2011, Cheng et al. 2012), but are for now limited. We also believe that our result can lead to progresses towards optimizing the micro-F_β measure in multiclass classification.

3 Relevance-Diversity Trade-off in Information Retrieval Problems

Contents

Here, we study the problem of diverse ranking in information retrieval tasks. The problem was first studied in the context of document retrieval as a way to increase end user satisfaction and reduce the query abandonment rate (Carbonell and Goldstein 1998, Zhai et al. 2003, Zhang et al. 2005). Now, diversification algorithms are used in many information

retrieval tasks including web search, recommender systems and summarization. It has been established that submodular functions can be used to promote the notion of diversity using the 'diminishing return' property of the submodular function. State of the art diversification algorithms make use of this property, but achieve it by explicitly trading-off the linear combination of a relevance objective function and a diversity objective function, thus resulting in a two-step procedure. In this chapter, we propose a diversification algorithm based on a submodular objective function which does not trade-off relevance and diversity explicitly, and compare the performance with the state of the art diversification algorithms on benchmark datasets.

3.1 Introduction

Most information retrieval systems are designed with the assumption that the relevance of the answers to the query are independent of each others, commonly referred as "Probability Ranking Principle" (Rijsbergen 1979) in the scientific literature. The 'Probability Ranking Principle' states that "If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data." However, in many real world applications, like web search and recommender systems, the usefulness of results depend on each other. For example, if a web search engine provides a user with 10 most relevant results to a given query, which are near duplicates but having highest probability of relevance to the query, the overall effectiveness of the system is zero if the result does not satisfy the user. Moreover, often information retrieval system results in imprecise responses due to the inherent limitations to represent and capture the complex and time-varying user requirements (Spärck-Jones et al. 2007). The above two factors demand a system to respond with diverse results.

The above argument regarding the shortcoming of the "Probability Ranking Principle" suggests that a good information retrieval system design should consider aspects other than relevance when retrieving items. In general, this notion of "other aspects" aim to make the system responses more diverse. In scientific literature, the problem of diversification is studied from different points of view like decreasing redundancy, increasing novelty, increasing serendipity, increasing freshness etc. But we argue that intrinsically all the above capture the idea of diversification. By decreasing redundancy, one aim to limit the number of duplicate relevant items by adding dissimilar items, thus making the items diverse with respect to each other. Similarly, other notions like novelty, freshness and serendipity can be increased by including more diverse items. Hence, in abstract sense all these notions imply each other.

Often in practice, the relevance of an item is indicated using a non-negative numeric score. For example, in web search relevance of a web page to a query is expressed using an ordinal scale between one and five where five indicates that the page is very relevant to the query and one indicates that the page is irrelevant to the query. Similarly, in movie recommender systems, preference of a user to a movie is indicated using similar ordinal scale, where five indicates the most preferred movie and one indicates the least preferred movie. Thus, we could associate non-negative utility scores to each item in a given set. In this regard, by diversification, we aim to rank the items such that items with higher

utility values but different from one another appears in the top rankings. Formally, we require the rankings to have the "diminishing return" property i.e. "if an item with high utility values is added to the list, the marginal utility (increase in utility score by adding a new item to the list) of adding a similar item should be less than adding a dissimilar item". Submodular function can be used to model this notion of diminishing return. Submodular functions are extensively studied (Fujishige 2005) and found applications in many machine learning problems including diversification , extractive summarization, structured sparse norm etc (Bach 2013, Krause and Golovin 2012).

Our approach is grounded on the idea of submodular function maximization. We view items as nodes in a similarity graph, and we define the coverage of a set of items by another set of items from the similarities between pairs of nodes. The objective is then to generate a set of unrated items that covers the set of items that were positively rated by the user. In this approach, diversity is obtained by defining the coverage as a submodular function: there is little gain in improving the coverage of a rated item that is already covered, whereas there can be a large gain in covering a new positively rated item.

As in the case of other diversification algorithms, which we describe in Section 3.4, the submodularity of the objective function provides approximation guarantees to the greedy algorithm iteratively building the set of recommended items. We also experiment with a slightly more involved inference algorithm based on a convex relaxation of the problem, but with limited success. We conclude that the greedy algorithm gives satisfactory results in practice.

In contrast to existing approaches that rely on two separate objectives for relevance and diversity, coverage accounts for both relevance and diversity: relevance is captured through the set to be covered, defined by positively rated items, and diversity through the preference towards lightly covering many items instead of covering heavily a few items. We compare our approach to existing baselines for the diversity/relevance trade-off in recommender system settings where our approach is grounded on the item-based collaborative filtering setup (Sarwar et al. 2001), and web search diversification where our approach is grounded on the transductive semi-supervised learning (Chapelle et al. 2006) settings on benchmark datasets, and show that our algorithm compares favorably in terms of various relevance and diversity metrics.

The remainder of this chapter is structured as follows. We give a brief introduction to submodular function in 3.2 before discussing our algorithm and its applications in many diversification task. Section 3.3 describes our framework, the optimization problem, greedy algorithm and the convex relaxation,and its relationship with other well-known problems. We carry out large scale experiments on benchmark data in the

context of movie recommendation systems in section 3.4.We conclude the chapter in Section 3.5 with some future directions.

3.2 Background & Preliminaries

We briefly describe many interesting properties of the submodular functions which are useful in our context. The results given in this section are stated here for the sake of completeness and improved readability. We use calligraphic letters for sets, bold capital letters for matrices, bold small letters for vectors, and indexed small letters for individual components of vectors wherever applicable.

3.2.1 Submodular Functions

Submodular functions are a special class of real valued functions defined over lattices Fujishige (2005). Here, we limit ourself to the set lattice with set intersection and set union as the meet and join operation. Given the ground set of objects $\mathcal{E} = \{e_1, e_2, \cdots, e_n\}$, we define a submodular function having the property,

Definition 3 (Submodular Function) *A set function $F : 2^{\mathcal{E}} \to \mathbb{R}$ is submodular, if, $\forall \mathcal{A}, \mathcal{B} \subseteq \mathcal{E}$,*

$$F(\mathcal{A}) + F(\mathcal{B}) \geq F(\mathcal{A} \cup \mathcal{B}) + F(\mathcal{A} \cap \mathcal{B}) \tag{3.1}$$

One of the most important defining characteristic of a submodular function is the "diminishing return property". Submodular function can be defined in terms of the diminishing return property.

Definition 4 (Submodular Function) *A set function $F : 2^{\mathcal{E}} \to \mathbb{R}$ is submodular, if, $\forall \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{E}$ and $e \in \mathcal{E} \setminus \mathcal{B}$,*

$$F(\mathcal{A} \cup \{e\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{e\}) - F(\mathcal{B}) \tag{3.2}$$

Lemma 11 *Definition.4 is equivalent to Definition.3*

Proof Let, $\mathcal{C} = \{c_1, c_2, \cdots c_k\}$ be the set of elements not in $\mathcal{B}$ i.e. $\mathcal{B} \cap \mathcal{C} = \emptyset$ and $\mathcal{B} \cup \mathcal{C} = \mathcal{E}$.

By Equation 3.2,

$$F(\mathcal{A} \cup \{c_1\}) - F(A) \geq F(\mathcal{B} \cup \{c_1\}) - F(B)$$

Now iteratively adding the elements, we get

$$F(\mathcal{A} \cup \{c_1, \cdots, c_{i-1}\} \cup \{c_i\}) - F(\mathcal{A} \cup \{c_1, \cdots, c_{i-1}\}) \geq F(\mathcal{B} \cup \{c_1, \cdots, c_{i-1}\} \cup \{c_i\}) - F(\mathcal{B} \cup \{c_1, \cdots, c_{i-1}\})$$

Adding the k equations for $i = 1 \cdots k$, we get

$$F(\mathcal{A} \cup \{c_1, \cdots, c_k\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{c_1, \cdots, c_k\}) - F(\mathcal{B})$$

We define $\mathcal{D} = \mathcal{A} \cup \mathcal{C}$, and $\mathcal{G} = \mathcal{B}$, then $\mathcal{A} = \mathcal{D} \cap \mathcal{G}$ and $\mathcal{B} \cup \mathcal{C} = \mathcal{D} \cup \mathcal{G}$. Putting these values in the above equation, we get

$$F(\mathcal{D}) - F(\mathcal{D} \cap \mathcal{F}) \geq F(\mathcal{D} \cup \mathcal{G}) - F(\mathcal{G})$$

Rearranging, we get $F(\mathcal{D}) + F(\mathcal{G}) \geq F(\mathcal{D} \cup \mathcal{G}) + F(\mathcal{D} \cap \mathcal{G})$

Now, to prove the reverse,

Assume $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{E}$, $c \notin \mathcal{B}$, and by Equation 3.1, $\forall \mathcal{C}, \mathcal{D} \subseteq \mathcal{E}$

$$F(\mathcal{C}) + F(\mathcal{D}) \geq F(\mathcal{C} \cup \mathcal{D}) + F(\mathcal{C} \cap \mathcal{D})$$

Define, $\mathcal{C} = \mathcal{A} \cup \{e\}$ and $\mathcal{D} = \mathcal{B}$, then $\mathcal{C} \cup \mathcal{D} = \mathcal{B} \cup \{e\}$ and $\mathcal{C} \cap \mathcal{D} = \mathcal{A}$. Putting the values in the above equation

We get,

$$F(\mathcal{A} \cup \{e\}) + F(\mathcal{B}) \geq F(\mathcal{B} \cup \{e\}) + F(\mathcal{A}))$$

rearranging, we get

$$F(\mathcal{A} \cup \{e\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{e\}) - F(\mathcal{B})$$

□

We use one more equivalent definition of submodularity which is useful in the forthcoming sections.

Definition 5 (Submodular Function) *A set function $F : 2^{\mathcal{E}} \to \mathbb{R}$ is submodular, if, $\forall \mathcal{A} \subseteq \mathcal{E}$, and $e_1, e_2 \in \mathcal{E} \setminus \mathcal{A}$,*

$$F(\mathcal{A} \cup \{e_1\}) - F(\mathcal{A}) \geq F(\mathcal{A} \cup \{e_1, e_2\}) - F(\mathcal{A} \cup \{e_2\}) \tag{3.3}$$

Lemma 12 *Definition.5 is equivalent to Definition.4*

Proof This can be verified, by putting $\mathcal{B} = \mathcal{A} \cup \{e_2\}$ in (3.2). □

Definition 6 (Monotonic Submodular Function) *A submodualr function F is monotonic non-decreasing, if* $\forall \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{E}$, $F(\mathcal{A}) \leq F(\mathcal{B})$ *and monotonic non-increasing, if* $\forall \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{E}$, $F(\mathcal{A}) \geq F(\mathcal{B})$.

Definition 7 (Polymatroid) *A normalized* ($F(\emptyset) = 0$) *monotonic non-decreasing submodular function is called polymatroid .*

Definition 8 (Supermodular Function) *A set function* $F : 2^{\mathcal{E}} \rightarrow \mathbb{R}$ *is supermodular, if* $-F$ *is submodular,*

Definition 9 (Modular Function) *A set function* $F : 2^{\mathcal{E}} \rightarrow \mathbb{R}$ *which is both supermodular and submodular is called modular, i.e,* $\forall \mathcal{A}, \mathcal{B} \subseteq \mathcal{E}$,

$$F(\mathcal{A}) + F(\mathcal{B}) = F(\mathcal{A} \cup \mathcal{B}) + F(\mathcal{A} \cap \mathcal{B}) \tag{3.4}$$

Lemma 13 *if* $F(\mathcal{A})$ *is a modular function, then* $F(\mathcal{A}) = F(\emptyset) + \sum_{e \in \mathcal{A}} (F(\{e\} - F(\emptyset_)))$

Proof Let $\mathcal{A} = \{e_1, \cdots, e_k\}$, by definition of modular function,

$$\begin{aligned} F(\{e_1, e_2\}) &= F(\{e_1\}) + F(\{e_2\}) - F(\emptyset) \\ F(\{e_1, e_2, e_3\}) &= F(\{e_1\}) + F(\{e_2\}) + F(\{e_3\}) - 2F(\emptyset) \end{aligned}$$

continuing for the entire set $\mathcal{A}$, we get

$$\begin{aligned} F(\{e_1, \cdots, e_k\} &= F(\{e_1\}) + \cdots F(\{e_k\}) - (k-1)F(\emptyset) \\ &= F(\emptyset) + \sum_{e \in \mathcal{A}} (F(\{e\} - F(\emptyset)) \end{aligned}$$

□

Lemma 14 *Let* g *be a concave function and* $F : 2^{\mathcal{E}} \rightarrow \mathbb{R}_+$ *be a non-negative modular function, then* $\forall \mathcal{A} \subseteq \mathcal{E}$, $g(F(\mathcal{A}))$ *is a submodular function. If* g *is monotonic then* $g(F(\mathcal{A}))$ *is also monotonic.*

Proof Due to the non-negativity of the modular function and by Lemma13 $F(A) = \sum_{e \in A} F(e) \geq 0$. For any $e_1, e_2 \in \mathcal{E} \setminus \mathcal{A}$, such that $0 < F(e_1) < F(e_2)$, due to the fact that a concave function has monotonically non-increasing differential quotient, we have

$$g(F(A) + F(e_1)) - g(F(A)) \geq g(F(A) + F(e_1) + F(e_2)) - g(F(e_2)$$

It is equivalent to the Definition 5 of submodularity. □

3.2.2 Submodular Function Maximization

An interesting and practically important problem associated with the submodular function is the constrained maximization problem. Consider

the problem of submodular function maximization with cardinality constraints. Here, the problem is to select a subset objects of given cardinality (k) from a given ground set of objects such that the submodular function defined over the set is maximal. Formally, the problem can be stated as

$$\mathcal{A}^* = \underset{\substack{\mathcal{A}\subseteq\mathcal{E}\\ |\mathcal{A}|\leq k}}{\operatorname{argmax}} F(\mathcal{A}) \tag{3.5}$$

A trivial algorithm requires us to evaluate the function on exponentially many instances ($|2^{\mathcal{E}}|$). The problem can be reduced to the maximum set coverage problem, a well known NP-Hard problem.There does not exist any exact algorithm to solve the submodular function maximization problem. The general strategy to solve the submodular maximization problem is based on the greedy heuristic as given by Nemhauser et al. (1978). The algorithm iteratively selects an element from the ground set such that it gives the maximum value for the incremental update value $(F(\mathcal{A}\cup\{e\}) - F(\mathcal{A}) : \mathcal{A}\subseteq\mathcal{E}, e\in\mathcal{E}\setminus\mathcal{A})$ at each iteration, where the ties are broken arbitrarily. In case of polymatroids with cardinality constraints, Nemhauser et al. (1978) gives a worst case lower bound on the optimality gap between the optimal solution and greedy solution as given in Theorem 15. In fact, the bound holds for any polymatroids with matroid constraints and cardinality is a special kind of matroid constraint.

Theorem 15 (Nemhauser et al. 1978) *For a non-decreasing submodular function $F : 2^{\mathcal{E}} \rightarrow \mathbb{R}_+$, let $\mathcal{A}^*$ be the optimizer of* (3.5) *and $\hat{\mathcal{A}}$ be the set returned by the greedy heuristic outlined above, then*

$$F(\hat{\mathcal{A}}) \geq (1-(1-\tfrac{1}{k})^k)\, F(\mathcal{A}^*) \geq (1-\tfrac{1}{e})\, F(\mathcal{A}^*)$$

Interchange heuristic is another familiar method to approximately solve a non-decreasing submodular maximization problem with cardinality constraints. Here, we start with an arbitrary solution set matching the cardinality constraint, and at each iteration another set with same cardinality is selected which shares a predefined number of elements with the solution set in the previous iteration. The sets are updated only if there is an improvement in the objective value. As pointed out by Nemhauser et al. (1978), the worst case performance of interchange heuristic, in terms of the optimality gap obtained, is par below the greedy heuristic. Moreover, interchange heuristic performance heavily depends on the intermediate element selection procedure to find improved solutions. Hence, we do not consider the interchange heuristic in our study.

3.3 Submodular Diversity Function

We consider a general information retrieval system framework wherein we are given a set of objects and few of the objects are already rated by the

user. For example, in a personalized recommender system, each object corresponds to an item (e.g. movie/book), and the rating corresponds to the preference score given by the user to the object. In case of web search, each object corresponds to a webpage and the rating corresponds to the relevance score of the page to a query, as given by the editors. We define a general submodular diversity function based on the coverage of the set of rated objects by the set of objects which are not yet rated by the user. This section forms the crux of this chapter.

3.3.1 Utility-Weighted Coverage for Relevant Diverse Sets

We are given a set of n objects $\mathcal{X} = \{x_1, \ldots, x_n\}$, together with a similarity measure defined over the set of objects.We do not assume the similarity function to be symmetric or transitive. We use the similarity matrix $\mathbf{W} = (W_{ij})_{i,j=1\ldots n}$ to represent the similarity values of n objects in $\mathcal{X}$. We can view $(\mathcal{X}, \mathbf{W})$ as a weighted graph, where W_{ij}, which weights the edge between objects x_i and x_j, should be interpreted as how much item x_i is *similar* to item x_j. Our goal is to return diverse relevant items, and we formalize it as a property of the returned solution set $\mathcal{S}$, based on the coverage of the subset of nodes in the graph $\mathcal{X}$ representing already rated objects $\mathcal{R}$.

3.3.2 Coverage of a Node

From now on, to simplify the notation, we identify the set of nodes $\mathcal{X}$ with $\{1, \ldots, n\}$. For a subset $\mathcal{S}$ of $\mathcal{X}$, given a node $i \notin \mathcal{S}$, we define the coverage score of i by $\mathcal{S}$ as:

$$\mathrm{cov}(i, \mathcal{S}) = f\big(\sum_{j \in \mathcal{S}} f^{-1}(W_{ij})\big) \ . \tag{3.6}$$

where $f : \mathbb{R}_+ \to \mathbb{R}_+$ is a non-decreasing invertible concave function, so as to ensure that $\mathcal{S} \to \mathrm{cov}(i, \mathcal{S})$ is non-decreasing with respect to inclusion and submodular. We call the function f in (3.6) the *saturation function*, because its main usage is to make the coverage of a single node of the graph saturate as we enlarge $\mathcal{S}$.

3.3.3 Utility-Weighted Coverage of a Set of Nodes

We now extend the definition of the coverage of a node to a set of nodes through the utilities attached to the nodes $\{v_1, \ldots, v_n\}$. Utility is a degree of liking for an item given by a user, such as the rating given to this item. We assume that $v_j \geq 0$. Given a set of m items $\mathcal{R} = \{\kappa_1, \ldots, \kappa_m\} \subset \mathcal{X}$ with their corresponding observed utility values $\mathrm{Y} = \{v_1, \ldots, v_m\}$, we define the *profile* of the user as $\mathcal{P} = \{(\kappa_j, v_j)\}_{j=1,\ldots,m}$. In the recommendation

example, the profile is the set of pairs (item, rating) known for a user and in the web search the profile is the set of pairs (webpage, rating) for a given query.

Now, given a profile $\mathcal{P}$ and a set of items $\mathcal{S}$ such that $\mathcal{S} \subset \mathcal{X} \setminus \mathcal{R}$, the coverage of profile $\mathcal{P}$ by $\mathcal{S}$ is defined as:

$$\text{cov}(\mathcal{P}, \mathcal{S}) = \sum_{(\kappa, v) \in \mathcal{P}} v \, \text{cov}(\kappa, \mathcal{S}) \ . \tag{3.7}$$

Here, we use a slight abuse of notation for cov, which can take as first argument either a profile or an item, but we assume that the context is clear considering the use of calligraphic notation for sets. From now on, we use the terminology of *point-wise* coverage for (3.6) and *profile* coverage for (3.7).

3.3.4 Optimal Utility-Diversity Trade-Off

The *profile* of a user is a representation of his/her different interests. Given a fixed *saturation function*, a set $\mathcal{S}$ with cardinality *k*having higher value of *profile-coverage* compared to other sets of cardinality *k*indicates that the set $\mathcal{S}$ covers a larger spectrum of users interest compared to other sets of same cardinality. We define diversity as a measure of users interest coverage. In that sense, a set covering larger spectrum of users interest will be the one with the best utility-diversity trade-off.

Formally, a set $\mathcal{S}^*$ realizes the optimal utility-diversity trade-off if it solves:

$$\max_{\substack{\mathcal{S} \subset \mathcal{X} \setminus \mathcal{R} \\ |\mathcal{S}| \leq k}} \text{cov}(\mathcal{P}, \mathcal{S}) \ . \tag{3.8}$$

When f in (3.6) is non-decreasing concave, the objective function of problem (3.8) is submodular as per Lemma 14. The problem is equivalent to one given in 3.5. As a result, the greedy approximation heuristic by Edmonds (1971) can be used to approximately solve (3.8), with the approximation guarantees stated in Theorem 15. The greedy algorithm for the maximal *profile coverage* is given in Algorithm 5. We call our proposed greedy algorithm "Submodular Diverse Ranking" (SDR) algorithm.

The computational complexity of the greedy algorithm depends on the computational complexity of the evaluation oracle for the cov function. At each step of the greedy algorithm, we need to call the evaluation oracle k times with a maximum over the set. Hence the computational complexity of the greedy algorithm becomes $\mathcal{O}(kp).\mathcal{O}(\text{cov})$ where $p = n - m$. The cov evaluation oracle has the time complexity $\mathcal{O}(mk)$, thus the greedy algorithm has time complexity $\mathcal{O}(mpk^2)$. Minoux (1978) proposed an "accelerated" version of the greedy algorithm which returns the greedy solution in the fewest possible running time. The algorithm

Algorithm 5: Submodular Diverse Ranking (SDR) Algorithm

Input : set of items $\mathcal{X}$, profile $\mathcal{P}$, similarity matrix $\mathbf{W}$, # of recommendations k

$\mathcal{X} = \mathcal{X} \setminus \mathcal{R}, \mathcal{S} = \emptyset$;
repeat
  $i^* = \text{argmax}_{i \in \mathcal{X}} \text{cov}(\mathcal{P}, \mathcal{S} \cup \{i\})$;
  $\mathcal{S} = \mathcal{S} \cup \{i^*\}$;
  $\mathcal{X} = \mathcal{X} \setminus \{i^*\}$;
until $|\mathcal{S}| = k$;
Output: set of diverse items $\mathcal{S}$

Algorithm 6: Submodular Diverse Ranking (SDR) Accelerated GreedyAlgorithm

Input : set of items $\mathcal{X}$, profile $\mathcal{P}$, similarity matrix $\mathbf{W}$, # of recommendations k

$\mathcal{X} = \mathcal{X} \setminus \mathcal{R}, \mathcal{S} = \emptyset$;
for $i \in \mathcal{X}$ **do**
  $\Delta(i) = \text{cov}(\mathcal{P}, \{i\})$; `// compute and store the marginal gain for each item in a priority queue`
repeat
  $i^* = \text{argmax}_{i \in \mathcal{X}} \Delta(i)$;
  $\delta = \text{cov}(\mathcal{P}, \mathcal{S} \cup \{i^*\}) - \text{cov}(\mathcal{P}, \mathcal{S})$;
  $\Delta(i^*) = \delta$;
  if $\delta < \max_{i \in \mathcal{X} \setminus \{i^*\}} \Delta(i^*)$ **then**
    goto 5
  $\mathcal{S} = \mathcal{S} \cup \{i^*\}$;
  $\mathcal{X} = \mathcal{X} \setminus \{i^*\}$;
until $|\mathcal{S}| = k$;
Output: set of diverse items $\mathcal{S}$

makes use of priority queues (Cormen et al. 2001) for faster retrieval. The accelerated greedy algorithm is given in Algorithm 6.

The priority queue implementation enables constant retrieval time ($\mathcal{O}(1)$) of argmax and $\mathcal{O}(log)$ priority queue updation time. Previous experimental results on large scale datasets showed that accelerated greedy algorithm gives substantial performance boost Leskovec et al. (2007).

3.3.5 Convex Relaxation for Inference

Instead of solving Problem (3.8) approximately using the greedy Algorithm 5, another approach is to solve exactly a convex relaxation of Prob-

Algorithm 7: Algorithm Based on Frank-Wolfe

Input : Set of items $\mathcal{X}$, profile $\mathcal{P}$, similarity matrix $\mathbf{W}$, # of recommendations k

$i = 1$, $tol = 10^{-6}$, $maxiter = 10^6$;

$\mathcal{S} = \mathcal{X} \setminus \mathcal{R}$, $\boldsymbol{\alpha}^i = \mathbf{0}$;

repeat

  $\mathbf{L}^i = -\nabla_{\boldsymbol{\alpha}} \Big(\sum_{(\kappa,v)\in\mathcal{P}} v f\big(\sum_{j=1}^{n} f^{-1}(W_{\kappa j})\alpha_j\big)\Big)\Big|_{\boldsymbol{\alpha}^i}$;

  $\boldsymbol{\alpha}^{i+1} = \text{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^n} \boldsymbol{\alpha}^T \mathbf{L}^i$; // subject to the constraints in (3.9)

  $\delta = \boldsymbol{\alpha}^{i+1} - \boldsymbol{\alpha}^i$;

  $\boldsymbol{\alpha}^{i+1} = \boldsymbol{\alpha}^i + \frac{2}{2+i}\delta$;

  $i = i + 1$;

until $\big(abs(\delta) \le tol\big) \vee \big(i \ge maxiter\big)$;

Output: Indexes of k largest elements in $\boldsymbol{\alpha}$

lem (3.8). As f is concave, the constrained problem:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \sum_{(\kappa,v)\in\mathcal{P}} v f\left(\sum_{j=1}^{n} f^{-1}(W_{\kappa j})\alpha_j\right) \\ &\text{such that } \forall j \in \mathcal{R},\ \alpha_j = 0 \\ &\qquad\quad \forall j \in \mathcal{X}\setminus\mathcal{R},\ 0 \le \alpha_j \le 1 \\ &\qquad\quad \textstyle\sum_{j=1}^{n} \alpha_j = k\ , \end{aligned} \tag{3.9}$$

has a concave objective with simple linear constraints, which can be solved efficiently using the Frank-Wolfe algorithm (Frank and Wolfe 1956) given in Algorithm 7. The inequality constraints are expected to have a sparsifying effect, leading to a vast majority of the α_j to be zero. In any case, a solution of cardinality $k$can always be returned by taking the $k$items with largest values of α_j.

3.3.6 A Graphical Intuition

Here, we give an intuitive explanation of the algorithm using a small set of artificial data. Consider the partially labelled data in Figure 3.1(a). The data can be considered as clusters of movies or clusters of web pages for a learning type query. The four clusters correspond to the groups of objects which are similar in some aspects, like thematic content. For example, in case of movie recommendation, each cluster can be regarded as similar movies, like genre similarity or cast similarity etc. The blue circles in the clusters represent relevant items, the cyan triangles represent less relevant items with respect to a user ratings, and the black triangle down symbol represents the items with no relevance labels (actual unlabelled

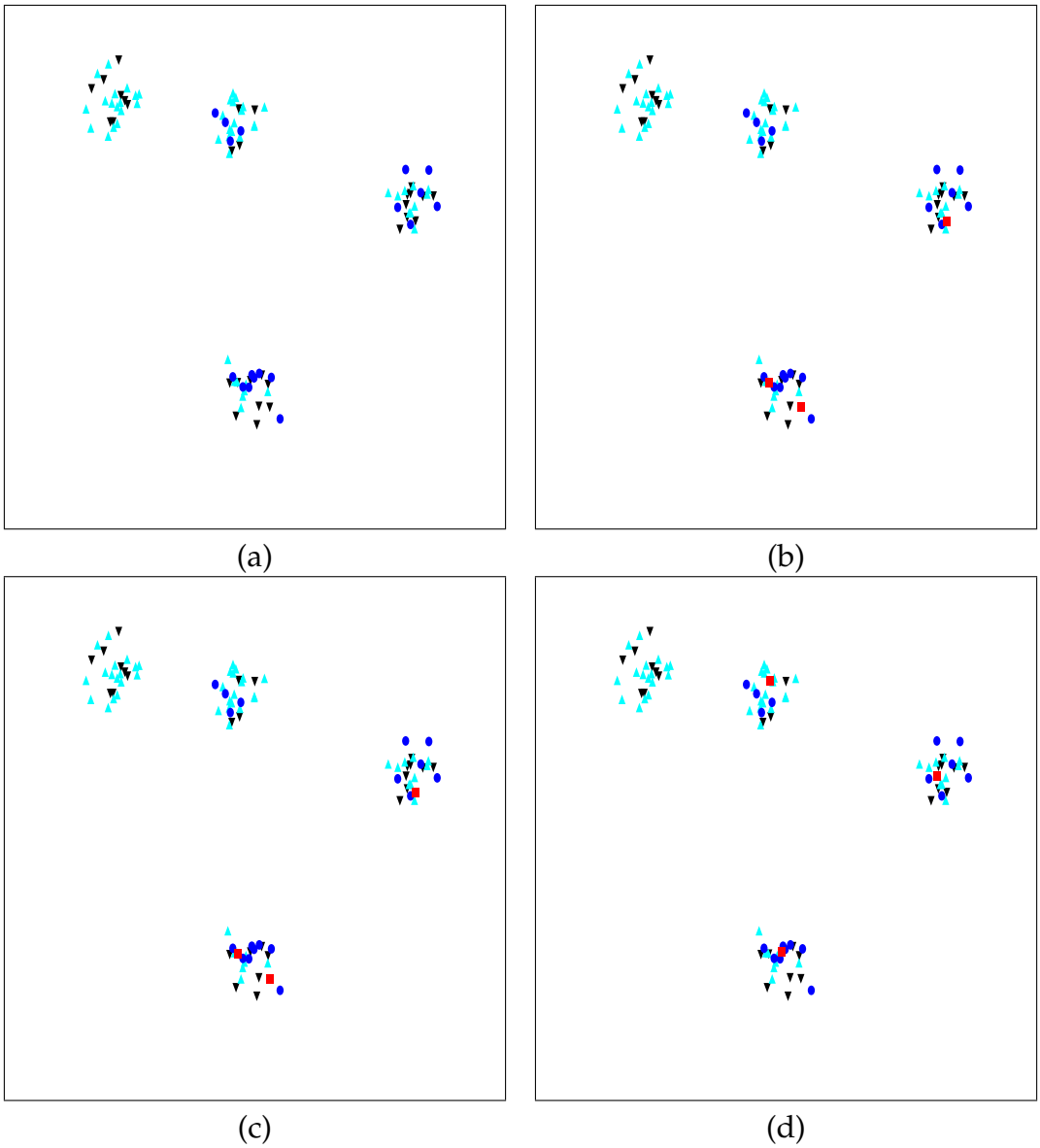


Figure 3.1 – *(1) (a) contains the clusters corresponding to an artificial data. The blue circles represent the already available relevant items for a user, the black down triangle symbol represents unrated items and the cyan triangle represents irrelevant items. (2) Top 3 ranking on the artificial data. The red rectangle represents the predictions made by the algorithms (b) MSD (c) MMR (d) SDR (with saturation function* $f(t) = t^{0.1}$

.

data contains a mix of relevant and irrelevant objects but we randomly removed the relevance information). Out of the four clusters, one cluster contain no relevant items and one cluster contains very few relevant items compared to the irrelevant items. A good diversification algorithm should be able to retrieve one relevant item per the relevant clusters and should avoid the non-relevant cluster.

The results of state of the art diversification algorithms on this artificial data for top three rankings are plotted in Figure: 3.1(b)&(c) and the result obtained using our coverage based greedy algorithm is given in

Figure: 3.1(d). As it is very evident, our algorithm retrieves items covering all the relevant clusters. More details about the compared algorithms and experiments is described in Section 3.4.

3.3.7 Special Cases

The non-linearities of the saturation function, if any, are the critical features that will allow us to make the trade-off between cumulated utility and diversity in profile coverage. We already established that for any non-decreasing concave saturation function, profile coverage is a submodular function. Here, we investigate other functional formulations for the saturation function which are of practical importance.

0/1 Saturation Function and Covering Problems

We first consider the limiting case where $f(t) = \lim_{\epsilon \to 0} t^{\epsilon}$. For the sake of clarity, we assume that $v = 1$ for every $(k, v) \in \mathcal{P}$, and that items are embedded in a metric space.

Covering $\mathcal{R}$ with balls: Let us assume that W_{ij} is 1 if i lies in the ball of radius ρ centered on j and 0 otherwise, for some fixed radius $\rho > 0$. Then, $\mathrm{cov}(\mathcal{P}, \mathcal{S})$ counts the number of items of $\mathcal{R}$ that are covered by the balls of radius ρ centered on the items of $\mathcal{S}$. Maximizing $\mathrm{cov}(\mathcal{P}, \mathcal{S})$ with a cardinality constraint on $\mathcal{S}$ corresponds to finding a maximum subset of $\mathcal{R}$ that is covered with $k$balls of radius ρ centered in points of $\mathcal{X} \setminus \mathcal{R}$. Problem (3.8) is then a maximal coverage problem.

k-nearest neighbors and clusters: If $\mathbf{W}$ is the adjacency matrix of a k-nearest neighbor graph, then $\mathrm{cov}(\mathcal{P}, \mathcal{S})$ counts the number of items in $\mathcal{R}$ that are in the k-nearest neighbors of items in $\mathcal{S}$. Likewise, assume that the items are clustered and that the similarity W_{ij} is 1 if i and j belong to the same cluster and 0 otherwise. Then, $\mathrm{cov}(\mathcal{P}, \mathcal{S})$ is the number of items in $\mathcal{S}$ that are in the same cluster as at least one item in $\mathcal{R}$.

Linear Saturation Function

Here, we consider the case where the *saturation function* is a linear. If f is, say the identity function, then maximizing coverage boils down to choosing the set of items $j \in \mathcal{S}$ such that $\sum_{(\kappa, v) \in \mathcal{P}} v W_{\kappa j}$ is maximal. In recommender system settings, this corresponds to one way of performing item-based collaborative filtering (Sarwar et al. 2001).

3.4 Diversity in Recommender Systems

Here, we demonstrate the applicability of our proposed algorithm in the recommender system framework. We consider the diversity/relevance trade-off in the context of item based collaborative filtering methods. In that context, personalized recommendations propose items that are similar to items that are known to be of interest to the user. Collaborative filtering based recommender systems have proved effective in practice (e.g. Sarwar et al. 2001); they are also particularly relevant in online recommendation settings since recommendations can be generated on-the-fly.

In recommender systems, a user may have eclectic tastes, and diversification is a mean to cover items from all relevant types. The most usual way of inducing diversity is to perform a two-step approach, in which a ranked list of top-$k$results is first retrieved, and a re-ranking algorithm is then run on the list such that diverse results appear at the top ranking positions. The re-ranking algorithms optimize an objective function that explicitly trades-off a relevance term and a diversity term. There are many variants of this scheme and we broadly refer to them as re-ranking algorithms for diversification. We do a brief literature review about the diversification algorithms in the recommender systems, followed by the experimental settings and results.

3.4.1 Related Work

Considerable work on recommender system diversification stem from the diversification work in web retrieval. The recommendation list diversification problem was studied from different points of view in the past (Vargas and Castells 2011, Vargas et al. 2014, Ashkan et al. 2015, Hurley 2013, Su et al. 2013, Oh et al. 2011, Wu et al. 2016). Here, we do a brief literature review about the diversification algorithms confined to the recommender system research.

The majority of the work on diversification is based on the Maximal Marginal Relevance (MMR) algorithm suggested by Carbonell and Goldstein (1998), originally proposed for web search diversification. MMR is based on the scalarization principle in multi-objective optimization techniques (Ehrgott and Gandibleux 2002). In MMR, two objectives, one corresponding to relevance aspect and the other one corresponding to dissimilarity (a measure of diversity) are linearly combined. The resulting objective is submodular, and diverse items are re-ranked using a greedy approach. Vargas and Castells (2011) propose a unified view of the state of the art metrics used in recommender system diversification. Vargas et al. (2014) discuss a diversity metric based on probabilistic models for genre coverage, and propose a re-ranking algorithm to diversify the recommendation list obtained using a baseline recommender system. Ashkan et al. (2015) maximize a modular objective function with

submodular constraints to maximize the genre coverage of movies and demonstrate the effectiveness of the approach on benchmark datasets. The proposed method maximizes the genre coverage of a list by choosing the most relevant item with largest number of genres. They used a collaborative filtering algorithm based on matrix factorization as their baseline recommender system. Hurley (2013) proposes a diversification method which does not require a ranked list beforehand, by weighting the pairwise rank difference with the dissimilarity score within the framework of RankALS (Takács and Tikk 2012). However, the theoretical properties of the objective function do not give a clear insight about the relevance-diversity trade-off. A similar approach, proposed by Su et al. (2013), creates a user profile based on relevant and irrelevant items, and diverse items are generated by optimizing a set-oriented AUC (area under the curve) objective function. Interestingly, Su et al. (2013) integrate the relevance and diversity estimation in a single objective function, where the objective is based on latent factor models augmented with a diversity inducing part. However the diversity term is defined over the item categories and in practice category information might not be readily available. Oh et al. (2011) proposed an algorithm for novel movie recommendation by accounting for popularity bias. The proposed algorithm predicts a set of recommendations such that it is matching with the individual personal popularity tendency, a popularity measure defined over the gross market collection of a movie, and at the same time aligns with the average popularity tendency of the dataset. Like many other diversification algorithm, the algorithm require a base recommender system to be run in the first stage. A closely related approach to our proposed algorithm is suggested by Wu et al. (2016). The proposed algorithm is based on maximizing coverage of a user based on the user neighborhood. Given a target user, a set of neighborhood users are selected such that selected users are similar to the target user in their movie preferences and coverage is defined over the common movies with respect to the neighborhood users. The final objective is a monotone non-decreasing submodular function similar to the one of proposed by Carbonell and Goldstein (1998).

Since majority of the work in recommender system is based on search diversification in web retrieval. We discuss some of the diversification work in the web retrieval. Diverse ranking in web search has attracted significant interest from the research community in the last decade, and has been studied extensively in the past (Carbonell and Goldstein 1998, Zhai et al. 2003, Zhang et al. 2005, Zhu et al. 2007, Yue and Joachims 2008, Radlinski et al. 2008; 2009, Agrawal et al. 2009, Gollapudi and Sharma 2009, Santos et al. 2010, Chapelle et al. 2011, He et al. 2012, Borodin et al. 2012, Raman et al. 2012, He et al. 2012). Zhai et al. (2003) follows an approach very similar to Carbonell and Goldstein (1998), where they select a new item conditioned on the relevance and novelty of the items already selected. Here, relevance and novelty measures are defined over the language models for information retrieval. Zhang et al. (2005) solves

the same problem using a random walk based formulation. They starts by selecting the node with the highest PageRank score and at subsequent steps, scores of the unselected nodes are updated with respect to the previously selected item. Similarly, Zhu et al. (2007) propose a random walk based approach, where they select the item with the highest PageRank score as the first item, and in each subsequent iteration selected node is converted to an absorbing state. The remaining items are selected based on the expected visit of the transient nodes in the absorbing Markovian chain based on the intuition that expected visit to the diverse items are more in an absorbing markovian chain. Supervised learning of diverse ranking using structured SVM is studied by Yue and Joachims (2008). The learning algorithm requires the training data to be associated with a set of topics, which is seldom available in practical scenarios like web search. Radlinski and Dumais (2006) propose an algorithm by generating set of 'related queries' corresponding to the user specified query. Agrawal et al. (2009) proposed a re-ranking algorithm by modeling the user intents through the publicly available taxonomies. Here, the queries and documents are categorized according to the taxonomy and the objective is to minimize the query abandonment by explicitly trading off the relevance and diversity (covering many taxonomical categories) aspects. Gollapudi and Sharma (2009) study the theoretical properties of the dispersion based objective functions for the diversification task. Similar to Radlinski and Dumais (2006), Santos et al. (2010) proposed a method based on sub-queries by query reformulation techniques. Mei et al. (2010) used time-variant random walk process to model the relevance-diversity trade-off. Dubey et al. (2011) suggest a method for diversification by finding topical centers of the transition graph similar to Zhu et al. (2007), but the teleportation probabilities are estimated using an inference algorithm. Tong et al. (2011), He et al. (2012) use the greedy algorithm for set cover on top of the PageRank based algorithm to select diverse items by forming a submodular objective function which also explicitly trades-off relevance and diversity. All the above work make use of the editorially judged training data. Radlinski et al. (2008) proposed a multi-armed bandit based online algorithm to learn diverse ranking from click-through data.

Another interesting problem is about quantifying the diversity of a given ranked list, and many diversity metrics are proposed in the past. Ziegler et al. (2005) propose intra-list similarity (ILS), which measures the distance between the items in the feature space. This measure does not take into account the ranking of the items. Clarke et al. (2008) propose $\alpha-$DCG as an extension of DCG for the diversification task. Agrawal et al. (2009) extended the commonly used IR metrics like NDCG, MAP etc to diversification task. Chapelle et al. (2011) studied the theoretical properties such metrics and proposed a submodular diversity measure intent-aware expected reciprocal rank. But calculation of such metric requires taxonomical and topical information.

Our approach differs from the previous work in the sense that we propose a single criterion to account for both diversity and relevance like in Hurley (2013) and Su et al. (2013). The submodular structure of our criterion gives theoretical basis in terms of the "diminishing return" property for the diversity unlike in Hurley (2013) and Su et al. (2013). The trade-off between relevance and diversity is dealt with by the exact definition of coverage we use.

3.4.2 Experiments

We compare our proposed algorithm against state of the art algorithms for recommender system diversification on baseline datasets. We carried out the experiment with saturation function of the form $f(t) = t^{\gamma}$ with $0 \leq \gamma < 1$.

Lemma 16 *For saturation function of the form $f(t) = t^{\gamma}$ with $0 \leq \gamma < 1$,* cov *defined in equation 3.6 is submodular.*

Proof . For $0 \leq \gamma < 1$, $f(t)$ is monotonic increasing concave function, and by Lemma 14, cov is submodular. □.

Baselines

We chose two baselines: Maximal Marginal Relevance (MMR) (Carbonell and Goldstein 1998) and Max-Sum Diversification (MSD) (Borodin et al. 2012).

Maximal Marginal Relevance MMR selects a set $\mathcal{S}$ solving the maximizing problem

$$\max_{\mathcal{S} \subseteq \mathcal{X} \setminus \mathcal{R}} \sum_{i \in \mathcal{S}} \Big(\lambda * sim_1(u, i) - (1 - \lambda) \max_{j \in \mathcal{S} - \{i\}} sim_2(i, j) \Big) \ \textit{such that} \quad |\mathcal{S}| \leq k$$

where sim_1 and sim_2 are similarities, and u is related to the user profile.

The objective function explicitly trades-off the similarity of a user to an item (measures the relevance aspect) with the dissimilarity of the item to the already selected items (measures the diversity aspect).

Given the set of already selected items $\mathcal{S}$ (initialized to the empty set), the MMR algorithm greedily selects an item i^* such that

$$i^* \in \operatorname*{argmax}_{i \in \mathcal{X} \setminus (\mathcal{R} \cup \mathcal{S})} \lambda \, sim_1(u, i) - (1 - \lambda) \max_{j \in \mathcal{S}} sim_2(i, j) \ ,$$

In our settings, this reads:

$$i^* \in \operatorname*{argmax}_{i \in \mathcal{X} \setminus (\mathcal{R} \cup \mathcal{S})} \lambda \sum_{(\kappa, v) \in \mathcal{P}} v W_{\kappa i} - (1 - \lambda) \max_{j \in \mathcal{S}} W_{ij} \ .$$

As the trade-off parameter $\lambda \in [0, 1]$ is decreased, more emphasis is put on the diversity of the resulting set. MMR can be interpreted as a greedy scheme for maximizing a non-monotone submodular objective function, for which the approximation guarantees of Theorem 15 do not apply (Lin and Bilmes 2011).

Max-Sum Diversification The Max-Sum diversification (MSD) algorithm is based on the facility dispersion problem, where one aims to find a subset of optimal locations such that the distance between the selected locations is maximized. Like in MMR, the objective function comprises two terms, a modular relevance term and a supermodular sum of distance diversity term. Formally, MSD returns the set $\mathcal{S}$ of cardinality k, that solves: [1]

$$\max_{\mathcal{S} \subseteq \mathcal{X} \setminus \mathcal{R}} \lambda\, g(\mathcal{S}) + (1 - \lambda) \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S} - \{i\}} dist(i, j) \;\; s.t \quad |\mathcal{S}| \leq k$$

where λ is the trade-off parameter, $g(\mathcal{S})$ is the utility function and $dist(i, j)$ is the distance function between item i and j. The problem is NP-Hard, but efficient greedy algorithm with provable approximation guarantees exist (Borodin et al. 2012). Starting from the empty set, at each step the greedy algorithm selects the optimal item i^* such that

$$i^* \in \operatorname*{argmax}_{i \in \mathcal{X} \setminus (\mathcal{R} \cup \mathcal{S})} \lambda \sum_{(\kappa, v) \in \mathcal{P}} v W_{\kappa i} - (1 - \lambda) \sum_{j \in \mathcal{S}} (1 - W_{ij}) \;\; .$$

Performance Metrics

Our experiments aim to assess the diversity of the recommended set. Even though there has been some work on defining the performance metrics for diversity, there is no clear consensus, especially in recommendation tasks. So we measure several features of the different solutions in the movie recommender settings, involving relevance, coverage, and popularity bias (Pradel et al. 2012). We describe the different metrics used in our experiment setup.

Genre Coverage A diversifying algorithm should produce results that cover different relevant interest groups. One way to measure the coverage of the user interests is to count the number of relevant genres recommended to the user. We define the *Genre Coverage* for the set $\mathcal{U}$ of users

[1] The original formulation of Borodin et al. (2012) is slightly different but equivalent.

as the average ratio of relevant genres recommended to each user.

$$\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{\Big|\bigcup_{i\in\mathcal{S}_u} genres(i)\ \bigcap\ \bigcup_{i\in\mathcal{R}_u^+} genres(i)\Big|}{\Big|\bigcup_{i\in\mathcal{R}_u^+} genres(i)\Big|}\ ,$$

where, for user u, $\mathcal{R}_u^+$ is the set of relevant rated movies, $\mathcal{S}_u$ is the set of recommended items, and $genres(i)$ returns the genres associated with item i.

Catalog Coverage Catalog Coverage is defined as the fraction of the relevant items that are recommended at least once, across all users. Higher values of catalog coverage indicate that the algorithm counterbalances the popularity bias by covering a large portion of the overall set of items.

Formally, it is defined as:

$$\frac{\Big|\bigcup_{u\in\mathcal{U}} \mathcal{S}_u^+\Big|}{|\mathcal{X}|}\ .$$

where $\mathcal{S}_u^+$ is the set of recommended items that are known to be relevant for user u (among the top-$k$recommended items).

Popularity Stratified Recall@k This metric is suggested by Steck (2011) to measure the ability of a recommender system to compensate for the popularity bias. As argued earlier, a diversity inducing recommendation system should cover diverse interests and may in turn cover items from the tail of the item-popularity distribution. *Popularity Stratified Recall@k* is defined as:

$$\frac{\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{S}_u^+}\left(\frac{1}{N_i^+}\right)^\beta}{\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{T}_u}\left(\frac{1}{N_i^+}\right)^\beta}\ ,$$

where $\mathcal{S}_u^+$ is the set of recommended items that are known to be relevant for user u (among the k recommended items), $\mathcal{T}_u$ is the set of items in the test set that are known to be truly relevant for user u, N_i^+ is the number of relevant ratings for item i in the test set and β is a hyperparameter which adjusts for the popularity bias. Higher values of *Popularity Stratified Recall@k* indicate that more relevant movies from the tail distribution are recommended. In our experiments, we used $\beta = 0.5$ and $k$was set to 5, 10, 20 or 50.

Intra-List Distance (ILD) Proposed by (Zhang and Hurley 2008), it measures the diversity of the set of recommended items by the mean distance between all pairs of items in this set. In our experiments, we used the Hamming distance:

$$\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{k(k-1)}\sum_{(i,j)\in\mathcal{S}_u}|genres(i)-genres(j)| \quad .$$

Discounted Cumulative Gain (DCG) It is a commonly used metric in ranking problems. It measures the relevance of a ranked list by the sum of the graded relevance discounted by the rank of the item. In our experiments, we used:

$$\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{S}_u}\frac{2^{r_i}-1}{log(i+1)} \quad ,$$

where r_i is the *graded* relevance score of the ith item. In our experiments, the ith item is either the ith item entering $\mathcal{S}_u$ for the greedy algorithm, or the one with ith largest α_i in the convex relaxation formulation (3.9).

Precision@*k* It is the fraction of relevant items in the recommended list of $k$items.

$$\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{k}\left|\mathcal{S}_u^+\cap\mathcal{T}_u\right| \quad .$$

Genre Coverage and *ILD* measure diversity, *Catalog Coverage* and *Stratified Recall@k* mix diversity and relevance, and DCG and *Precision@k* measure the relevance. Higher values of the aforementioned metrics indicate a better recommendation list.

Experimental Protocol

We used two benchmark datasets *(i)* MovieLens and *(ii)* Yahoo! Movies to evaluate the proposed algorithm. Following Cremonesi et al. (2010), we carried out holdout validation by splitting the data randomly into training and test set such that 3% of the original data goes into testing and remaining goes into training. To reduce the variability in the result, split is carried out five times and the reported results are the average values over the five splits. The rating values and the corresponding movies in the training set are used to create the profile $\mathcal{P}$ and the unrated movies in the training set are used as $\mathcal{S}$. For the purpose of evaluation, whenever necessary, we discretized rating scores to binary values such that rating scores of 4 and 5 are deemed as relevant and otherwise irrelevant.

We estimated the unobserved rating values for MMR and MSD in the training set using item-based collaborative filtering (Sarwar et al. 2001) (matrix-factorization based collaborative filtering methods gave inferior

Table 3.1 – *Experimental Results on MovieLens (top 10 recommendations)*

	MMR λ=0.1	MMR λ=0.5	MMR λ=0.8	MSD λ=0.1	MSD λ=0.5	MSD λ=0.8
Genre Coverage	67.12	66.34	66.26	66.84	65.35	65.50
ILD	19.73	19.75	19.75	19.22	19.44	19.55
Catalog Coverage	6.55	6.20	6.19	7.92	6.37	6.51
Stratified Recall@k	7.78	7.57	7.56	8.00	7.36	7.60
DCG	31.83	31.38	31.35	25.09	25.33	29.93
Precision@k	4.06	3.98	3.98	4.02	3.86	3.97

Table 3.2 – *Experimental Results on MovieLens (top 10 recommendations)*

	MOD	SDR γ=0.8	SDR γ=0.5	SDR γ=0.1	SDR $\gamma\to0$	SDR conv γ=0.5
Genre Coverage	66.23	66.31	66.56	70.04	63.64	66.27
ILD	19.75	19.78	19.88	19.77	18.78	19.70
Catalog Coverage	6.18	6.26	6.57	10.49	5.37	6.76
Stratified Recall@k	7.56	7.63	7.90	10.27	6.23	7.89
DCG	31.33	31.51	32.10	34.86	24.42	31.95
Precision@k	3.98	4.01	4.10	4.60	3.19	4.08

results). We used the observed movie ratings to create the user profiles $\mathcal{P}$ and the similarity matrix $\mathbf{W}$, which is computed by a cosine similarity. For evaluation purposes, we used the original observed rating values or whenever applicable, their binarized version, in the test set. We used $f(t) = t^\gamma$ with $\gamma = \{0.1, 0.5, 0.8\}$ and the limiting case where $\gamma \to 0$, which corresponds to the ℓ_∞-norm, for the *saturation function* in the submodular (SUB) setting. It should be noted that in the limiting case ($\gamma \to 0$), the greedy algorithm selects the item with maximum *profile coverage* score at each iteration, and the *max* function is a non-decreasing submodular function for both positive and negative values Bach (2013). For the modular setting (MOD), we used saturation function $f(t) = t$.

Results

MovieLens MovieLens[2]*1M* dataset contains ratings from 6040 users on 3706 movies (excluding movies with no rating values). Each movie is associated with a set of genres, among 18 distinct categories. The performances of the different algorithms on the MovieLens dataset for k=10 are given in Table 3.1 and 3.2, and for k=20 in Table 3.3 and 3.4 (all values are multiplied by 100). The relevance-diversity values as the function of recommendation size (k) are given in Figure 3.2.

[2] http://grouplens.org/datasets/movielens/

Table 3.3 – *Experimental Results on MovieLens (top 20 recommendations)*

	MMR λ=0.1	MMR λ=0.5	MMR λ=0.8	MSD λ=0.1	MSD λ=0.5	MSD λ=0.8
Genre Coverage	79.96	79.53	79.48	79.41	79.25	79.22
ILD	20.06	20.26	20.27	20.18	20.01	19.93
Catalog Coverage	8.66	8.23	8.19	8.49	8.73	8.76
Stratified Recall@k	11.79	11.53	11.50	11.60	11.64	11.60
DCG	38.39	37.87	37.82	37.84	34.54	32.89
Precision@k	5.86	5.77	5.76	5.78	5.78	5.77

Table 3.4 – *Experimental Results on MovieLens (top 20 recommendations)*

	MOD	SDR γ=0.8	SDR γ=0.5	SDR γ=0.1	SDR $\gamma\to0$	SDR conv γ=0.5
Genre Coverage	79.46	79.48	79.50	82.28	78.02	79.58
ILD	20.28	20.28	20.28	19.20	19.48	20.13
Catalog Coverage	8.17	8.36	8.86	14.75	7.65	8.99
Stratified Recall@k	11.49	11.63	12.06	16.35	10.07	11.97
DCG	37.80	38.03	38.83	43.18	30.57	38.56
Precision@k	5.76	5.80	5.96	6.89	4.89	5.90

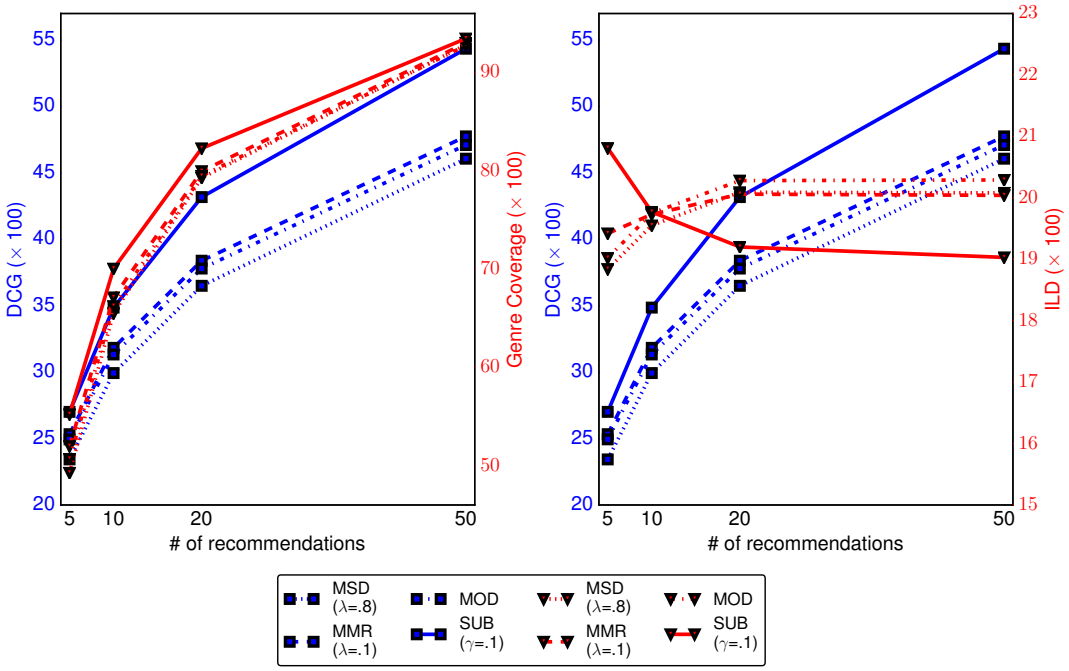


Figure 3.2 – *Relevance-Diversity values for the MovieLens data as the function of recommendation size k*

Yahoo! Movies Yahoo! Movies[3] dataset contains separate training and test set, but we used only the training set due to the unavailability of

[3] https://webscope.sandbox.yahoo.com/

Table 3.5 – *Experimental Results on Yahoo! Movies (top 10 recommendations)*

	MMR λ=0.1	MMR λ=0.5	MMR λ=0.8	MSD λ=0.1	MSD λ=0.5	MSD λ=0.8
Genre Coverage	72.61	70.42	69.96	72.80	71.67	70.78
ILD	16.03	15.75	15.67	14.99	15.93	15.84
Catalog Coverage	2.96	2.61	2.55	3.15	2.78	2.63
Stratified Recall@k	16.44	15.08	14.83	15.36	15.51	14.85
DCG	13.59	13.37	13.27	9.64	10.30	10.61
Precision@k	1.62	1.55	1.53	1.40	1.55	1.51

Table 3.6 – *Experimental Results on Yahoo! Movies (top 10 recommendations)*

	MOD	SDR γ=0.8	SDR γ=0.5	SDR γ=0.1	SDR $\gamma\to0$	SDR conv γ=0.5
Genre Coverage	69.77	70.44	71.75	73.61	66.59	69.58
ILD	15.63	15.72	15.82	15.52	14.87	14.84
Catalog Coverage	2.54	2.64	2.87	3.03	1.91	1.71
Stratified Recall@k	14.75	15.33	16.48	16.00	11.36	14.45
DCG	13.24	13.47	13.88	12.92	9.75	13.02
Precision@k	1.53	1.56	1.63	1.50	1.21	1.53

genre information on the test set. The training data contain 211,231 rating values for 7,642 users and 11,915 movies. We removed the movies with missing genres, being left with 187,435 ratings spanning 7,636 users and 8,647 movies. Yahoo! Movies span a total of 25 distinct genres. Table 3.5 and 3.6 contains the result for Yahoo! Movies for recommendation size k=10 and Table 3.7 and 3.8 (all values multiplied by 100) and Figure 3.3 contains the relevance-diversity values as the function of recommendation size k.

Table 3.7 – *Experimental Results on Yahoo! Movies (top 20 recommendations)*

	MMR λ=0.1	MMR λ=0.5	MMR λ=0.8	MSD λ=0.1	MSD λ=0.5	MSD λ=0.8
Genre Coverage	84.00	82.14	81.75	82.14	82.82	83.06
ILD	16.24	16.24	16.19	16.21	16.19	16.14
Catalog Coverage	3.61	3.15	3.12	3.19	3.43	3.58
Stratified Recall@k	22.09	20.38	20.18	20.53	21.41	22.05
DCG	15.23	15.00	14.92	13.58	12.14	12.25
Precision@k	2.07	1.99	1.98	2.00	2.04	2.08

Table 3.8 – *Experimental Results on Yahoo! Movies (top 20 recommendations)*

	MOD	SDR γ=0.8	SDR γ=0.5	SDR γ=0.1	SDR $\gamma\to0$	SDR conv γ=0.5
Genre Coverage	81.62	82.06	83.09	84.99	79.44	82.26
ILD	16.16	16.18	16.13	15.62	15.50	16.01
Catalog Coverage	3.10	3.25	3.54	3.84	2.52	3.40
Stratified Recall@k	20.08	20.83	22.14	22.06	15.78	21.34
DCG	14.88	15.14	15.58	14.69	11.22	14.73
Precision@k	1.98	2.02	2.09	1.98	1.62	2.09

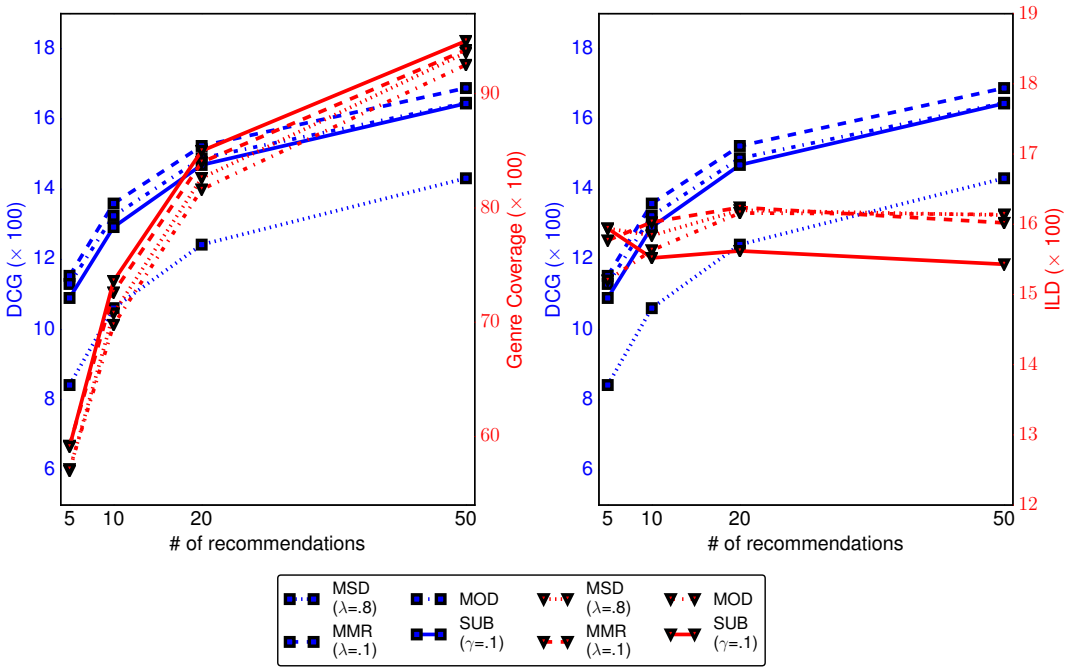


Figure 3.3 – *Relevance-Diversity values for the Yahoo! Movies data as the function of recommendation size k*

Discussion

It should be noted that the modular version of our algorithm is equivalent to MMR and MSD with $\lambda = 1$. Surprisingly, MMR does not exhibit any trade-off between relevance and diversity metrics as λ is varied. As the λ value is increased from 0.1 to 1, values corresponding to DCG, Precision@k, Catalog Coverage and Genre Coverage remain the same or decrease only marginally. In effect, MMR does not recommend very relevant and less diverse movies by weighting the relevance term highly. The same trend can be noted for Yahoo! Movies as well. On the other hand, MMR compensates for popularity bias by recommending less popular movies covering a larger spectrum of the set as the diversity term is weighted high, thus increasing Stratified Recall@k and Catalog Coverage. But for MSD, on MovieLens, as the λ value is increased, the recom-

mendation list becomes more relevant but nothing can be inferred about diversity. But on Yahoo! Movies, trade-off between relevance (DCG) and diversity (Genre Coverage) is clearer. As the λ value is increased, recommended list becomes more relevant and less diverse. However, there is no clear indication that MSD compensates for popularity bias, even though it recommends movies spanning large spectrum of movies from the set. The modular (equivalent to item-based collaborative filtering) version perform as good as the MMR and MSD versions.

For our algorithm, as the γ value is decreased, both the relevance and the diversity values increase and the increase is more significant. The best in-class relevance-diversity values are obtained for $\gamma = 0.1$. For MovieLens, we see a 3% increase for both DCG and Genre Coverage metrics compared to the second best algorithm. For Yahoo! Movies, $\gamma = 0.1$, gives the best diversity value (Genre Coverage) for a marginally smaller value of relevance (DCG). On MovieLens, the submodular algorithm returns greater number of movies from the tail distribution which are collectively distinct, i.e. covering large spectrum of movies, as indicated by the larger values of Stratified Recall@k and Catalog Coverage. But this effect is not very evident on Yahoo! Movies. As the γ value approaches zero the quality of the recommendation list deteriorates. Convex relaxation based algorithm performance is on par with other algorithms, but we found it computationally more expensive. The performance of relevance-diversity metrics for varying recommendation size is given in Figure 3.2 and 3.3. On MovieLens, submodular (solid blue line) algorithm returns greater number of relevant movies (square markers) which are diverse (red solid line with triangle markers on the left plot), whereas MMR and MSD returns diverse movies which are less relevant (red non-solid lines with triangle markers on the left plot). It can also be noted that as the *k*value increases, submodular algorithm recommends the most diverse and relevant movies (higher DCG & Genre Coverage) whereas MMR and MSD recommend the most diverse, but irrelevant movies (higher ILD & lower DCG; see right plot in Figure 3.2). But in Yahoo! Movies, even though the diversity metrics are superior for submodular algorithm, the relevance values are close to other algorithms.

Effect on Eclectic Users

The problem of diverse recommendation is more critical for users with eclectic interests. Here, we study the effect of diversification on eclectic users by sampling an arbitrary number of eclectic users from the MovieLens and Yahoo! Movies. We define eclectic users based on their affinity towards many diverse items, as measured by their number of positive ratings and a mean similarity between rated items W_{ij} below some threshold. We selected 209 users from MovieLens by setting the minimum number of relevant ratings to 100 and the mean similarity between

Table 3.9 – *Experimental Results on MovieLens (top 10 recommendations for eclectic users)*

	MMR λ=0.1	MMR λ=0.5	MMR λ=0.8	MSD λ=0.1	MSD λ=0.5	MSD λ=0.8
Genre Coverage	59.54	59.41	59.39	58.02	59.03	59.40
ILD	20.69	20.69	20.69	20.23	20.45	20.69
Catalog Coverage	3.12	3.12	3.12	2.97	3.19	3.12
Stratified Recall	4.62	4.62	4.62	4.32	4.67	4.62
DCG	87.41	87.45	87.45	67.43	86.91	89.23
Precision@k	11.74	11.76	11.76	11.01	11.77	11.76

Table 3.10 – *Experimental Results on MovieLens (top 10 recommendations for eclectic users)*

	MOD	SDR γ=0.8	SDR γ=0.5	SDR γ=0.1	SDR $\gamma\to0$	SDR conv γ=0.5
Genre Coverage	59.39	59.46	59.51	64.54	59.87	59.01
ILD	20.69	20.72	20.82	21.03	20.55	20.34
Catalog Coverage	3.12	3.16	3.29	4.13	2.61	3.36
Stratified Recall	4.62	4.67	4.87	6.33	4.06	4.89
DCG	87.44	87.97	90.02	95.32	71.05	90.22
Precision@k	11.76	11.86	12.21	13.21	9.64	12.13

movies below 0.2, and 109 users from Yahoo! Movies by setting the minimum number of relevant ratings to 50 and the mean similarity between movies below 0.1. The experimental results for recommendation size k=10 is given in Tables 3.9 and 3.10 for MovieLens data and Tables 3.13 and 3.14 for Yahoo! Movies, and for recommendation size k=20 is given in Tables 3.11 and 3.12 for MovieLens and in Tabel 3.15 and 3.16 for Yahoo! Movies respectively. The submodular algorithm significantly improves the DCG and Genre Coverage values compared to the second best diversification algorithm. The relevance and diversity metric values for different recommendation size for eclectic users is given in Figures 3.4 and 3.5. As we can see from the figures, the blue solid line with square markers and red solid line with triangle markers dominate the DCG-Genre Coverage graph for varying sizes of k. On eclectic user set, as the recommendation size grows, MSD and MMR return movies which are diverse with respect to each other (higher ILD values) but less relevant to the users (smaller DCG and Genre Coverage) compared to submodular algorithm.

Table 3.11 – *Experimental Results on MovieLens (top 20 recommendations for eclectic users)*

	MMR λ=0.1	MMR λ=0.5	MMR λ=0.8	MSD λ=0.1	MSD λ=0.5	MSD λ=0.8
Genre Coverage	72.72	72.63	72.62	72.04	72.63	72.63
ILD	20.14	20.17	20.17	19.76	20.06	20.17
Catalog Coverage	4.69	4.66	4.66	4.84	4.76	4.66
Stratified Recall	7.74	7.69	7.69	7.96	7.81	7.69
DCG	111.09	110.68	110.67	94.64	110.64	112.46
Precision@*k*	18.29	18.18	18.18	18.52	18.33	18.18

Table 3.12 – *Experimental Results on MovieLens (top 20 recommendations for eclectic users)*

	MOD	SUB γ=0.8	SDR γ=0.5	SDR γ=0.1	SDR $\gamma\rightarrow$0	SDR conv γ=0.5
Genre Coverage	72.61	72.57	72.81	77.70	74.05	72.69
ILD	20.17	20.20	20.30	19.63	20.38	19.88
Catalog Coverage	4.66	4.69	4.85	6.39	4.15	4.99
Stratified Recall	7.68	7.75	8.05	10.83	6.85	8.13
DCG	110.64	111.23	113.72	123.39	90.41	114.31
Precision@*k*	18.17	18.28	18.75	20.98	15.00	18.75

Table 3.13 – *Experimental Results on Yahoo! Movies (top 10 recommendations for eclectic users)*

	MMR λ=0.1	MMR λ=0.5	MMR λ=0.8	MSD λ=0.1	MSD λ=0.5	MSD λ=0.8
Genre Coverage	55.88	54.12	54.03	56.66	56.46	55.74
ILD	14.57	14.03	13.98	14.58	14.43	14.32
Catalog Coverage	0.28	0.25	0.25	0.31	0.26	0.25
Stratified Recall@k	4.46	3.97	3.88	5.75	4.30	4.00
DCG	19.92	18.40	18.15	18.13	15.87	18.89
Precision@*k*	2.59	2.39	2.35	2.99	2.42	2.44

Significance Testing

Our experimental results shows that we get significant improvement for many of the relevance-diversity metrics. In particular, we get the best in-class results for MovieLens with complete set of users and Yahoo! Movies with eclectic users using SDR algorithm. Here, we do a statistical study regarding the consistency of the results we obtained.

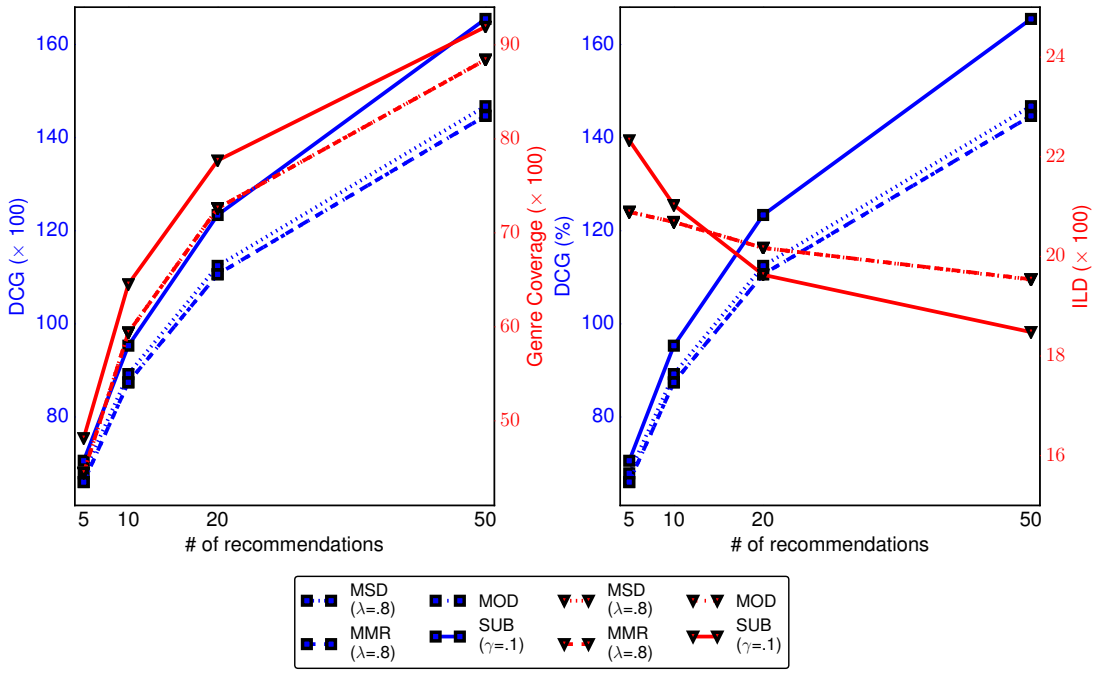


Figure 3.4 – *Relevance-Diversity values for the MovieLens (eclectic users) as the function of recommendation size k*

Table 3.14 – *Experimental Results on Yahoo! Movies (top 10 recommendations for eclectic users)*

	MOD	SDR γ=0.8	SDR γ=0.5	SDR γ=0.1	SDR $\gamma\to0$	SDR conv γ=0.5
Genre Coverage	53.91	55.47	57.22	60.03	45.93	56.32
ILD	13.96	14.26	14.68	14.00	11.78	14.49
Catalog Coverage	0.24	0.27	0.29	0.27	0.19	0.28
Stratified Recall@k	4.46	3.97	3.88	5.75	4.30	4.00
DCG	19.92	18.40	18.15	18.13	15.87	18.89
Precision@*k*	2.59	2.39	2.35	2.99	2.42	2.44

Demšar (2006) proposes the use of Friedman test (Hollander and Wolfe 1973) for statistical significance testing of multiple algorithms on multiple datasets. It is a non-parametric test where the algorithms are ranked for each dataset separately and average rank of for each algorithms are computed. The Friedman statistic value is computed over the average rank. The null hypothesis states that all the algorithms are equivalent and average ranks for each of the algorithms over different datasets should be same indicating that the difference in the values of the performance measure is random.

If the null hypothesis is rejected, we carry out Friedman post-hoc test to compare the pairwise comparison of different algorithms. In our settings, If the null hypothesis is rejected i.e. if there is a significant dif-

Table 3.15 – *Experimental Results on Yahoo! Movies (top 20 recommendations for eclectic users)*

	MMR λ=0.1	MMR λ=0.5	MMR λ=0.8	MSD λ=0.1	MSD λ=0.5	MSD λ=0.8
Genre Coverage	65.43	63.55	63.36	70.48	66.44	64.66
ILD	14.32	14.00	13.96	14.34	14.21	14.10
Catalog Coverage	0.38	0.36	0.35	0.53	0.38	0.36
Stratified Recall@k	7.00	6.49	6.40	10.59	7.13	6.60
DCG	24.06	22.66	22.42	26.00	20.79	23.31
Precision@k	3.72	3.56	3.52	5.17	3.76	3.65

Table 3.16 – *Experimental Results on Yahoo! Movies (top 20 recommendations for eclectic users)*

	MOD	SDR γ=0.8	SDR γ=0.5	SDR γ=0.1	SDR $\gamma\to0$	SDR conv γ=0.5
Genre Coverage	63.37	64.39	67.24	74.42	61.74	65.43
ILD	13.97	13.99	13.95	13.22	12.44	14.16
Catalog Coverage	0.35	0.38	0.44	0.42	0.25	0.39
Stratified Recall@k	6.26	7.09	8.38	9.42	4.79	7.50
DCG	22.16	24.17	26.88	24.73	14.91	24.86
Precision@k	3.49	3.87	4.44	4.02	2.29	3.96

ference between the algorithms (we set the critical value to p = 0.05 for Friedman test), we do a pairwise comparison using Nemenyi post-hoc test (Demšar 2006).

We excluded Yahoo! Movies results from the significance testing as the difference between different performance metric values for different algorithms on different datasets are very marginal. We carried out Friedman test on MovieLens with full users and MovieLens and Yahoo! Movies with eclectic users for the case of top 10 recommendations, on four metrics, DCG, Genre Coverage, Catalog Coverage and Stratified Recall over the five random split values. For MMR and MSD algorithms, we selected the results for the best performing trade-off parameter (λ) in terms of DCG and Genre Coverage for significance testing. For SDR, we used the results for $\gamma = 0.1$. The Friedman test p-values for different algorithms on the four above mentioned performance metrics are given in Table 3.17.

The p-values in Table 3.17 indicates that in case of MovieLens data, both on complete and eclectic users, the algorithm significantly differs from each other i.e. the average rank of the algorithms for respective performance metrics is not the same. But in case of Yahoo! Movies, except

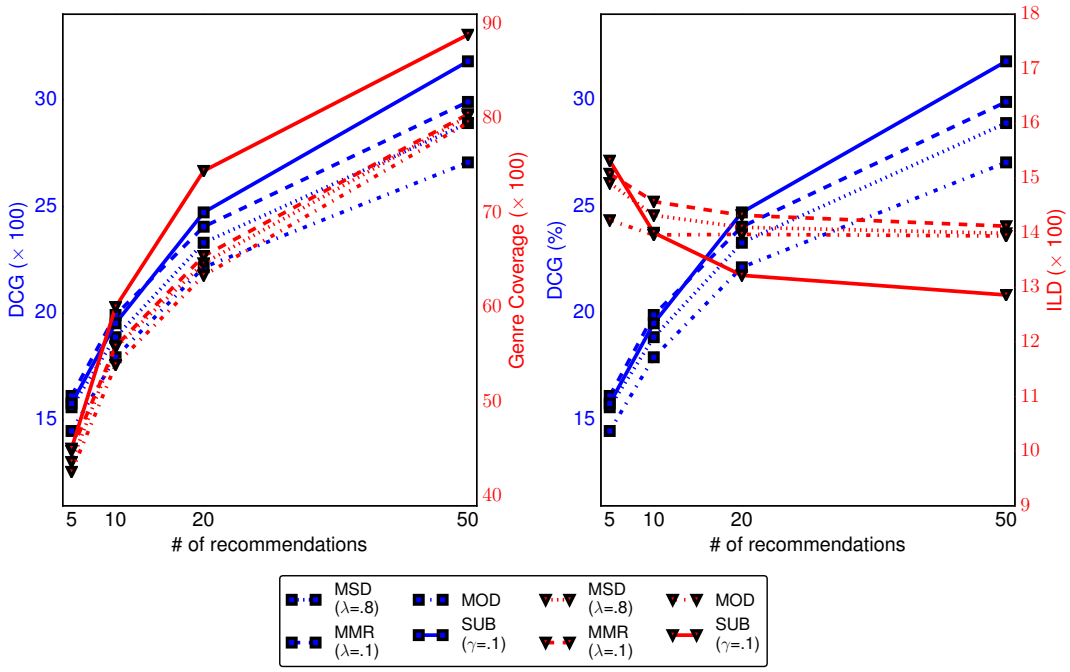


Figure 3.5 – *Relevance-Diversity values for the Yahoo! Movies (eclectic users) as the function of recommendation size k*

Table 3.17 – *Friedman test p-values*

	DCG	Genre Coverage	Catalog Coverage	Stratified Recall
MovieLens	0.00182	0.0018	0.0043	0.0029
MovieLens (ecl. users)	0.0029	0.0099	0.0018	0.0030
Yahoo! Movies (ecl. users)	0.0624	0.0036	0.1503	0.1777

for the Genre Coverage, the algorithm performance is indistinguishable which indicates that the average rank of the algorithms over the five random splits is same.

As mentioned earlier, for further analysis, we carry pairwise comparisons between the algorithms using Friedman post-hoc nemenyi test (Hollander and Wolfe 1973).

Nemenyi test p-values for MovieLens complete users, eclectic users and Yahoo! Movies eclectic users, for the four performance metrics are given in Table 3.18, 3.19 and 3.20 respectively. Our proposed algorithm SDR performs significantly better than the non-diversification baseline modular algorithm, whereas other diversification algorithms, MMR and MSD performance is statistically inconsistent to the modular algorithm (higher p-values). Between SDR, MMR and MSD, SDR results are statistically significant to MSD but insignificant to MMR. But in case of eclectic users, SDR results is statistically significant compared to MMR but in-

Table 3.18 – *Nemenyi test p-values for MovieLens on top 10 recommendations*

	MSD	MMR	MOD
MMR	0.068	-	-
MOD	0.611	0.611	-
SDR	0.001	0.611	0.068
(DCG)			

	MSD	MMR	MOD
MMR	0.068	-	-
MOD	0.611	0.611	-
SDR	0.001	0.611	0.068
(Genre Coverage)			

	MSD	MMR	MOD
MMR	0.999	-	-
MOD	0.383	0.316	-
SDR	0.204	0.256	0.002
(Catalog Coverage)			

	MSD	MMR	MOD
MMR	0.122	-	-
MOD	0.883	0.456	-
SDR	0.003	0.611	0.036
(Stratified Recall)			

Table 3.19 – *Nemenyi test p-values for MovieLens on top 10 recommendations for eclectic users*

	MSD	MMR	MOD
MMR	0.16	-	-
MOD	0.16	1.0	-
SDR	0.88	0.02	0.02
(DCG)			

	MSD	MMR	MOD
MMR	0.83	-	-
MOD	0.93	0.99	-
SDR	0.20	0.02	0.05
(Genre Coverage)			

	MSD	MMR	MOD
MMR	0.99	-	-
MOD	0.38	0.32	-
SDR	0.20	0.25	0.002
(Catalog Coverage)			

	MSD	MMR	MOD
MMR	0.98	-	-
MOD	0.98	1.00	-
SDR	0.12	0.05	0.05
(Stratified Recall)			

Table 3.20 – *Nemenyi test p-values for Yahoo Movies on top 10 recommendations for eclectic users*

	MSD	MMR	MOD
MMR	0.883	-	-
MOD	0.316	0.068	-
SDR	0.961	0.995	0.122
(DCG)			

	MSD	MMR	MOD
MMR	0.9948	-	-
MOD	0.2035	0.3159	-
SDR	0.3159	0.2035	0.0014
(Genre Coverage)			

	MSD	MMR	MOD
MMR	0.53	-	-
MOD	0.83	0.12	-
SDR	0.96	0.83	0.53
(Catalog Coverage)			

	MSD	MMR	MOD
MMR	0.76	-	-
MOD	0.99	0.61	-
SDR	0.32	0.88	0.20
(Stratified Recall)			

significant to MSD. As expected, in case of Yahoo! Movies, SDR gives significant results for Genre Coverage compared to the modular version of our algorithm.

Latent Factor Based Models

In addition to the above experimental setup, we also experimented by generating latent user and movie factors using matrix factorization based methods. But the results we obtained are inferior to the one detailed above. For the sake of completeness of the work and further investigation we present the result here.

In the latent factor based models, we learn a user factor matrix (one vector per user) and an item factor matrix (one vector per movie) from the observed rating data. Formally,

$$\mathbf{R} \sim \mathbf{PQ}$$

where $\mathbf{R}$ is the $k \times n$ rating matrix; k is the total number of users and n is the total number of items, and $\mathbf{P}$ is the $k \times z$ user factor matrix (each row vector corresponds to a user) and $\mathbf{Q}$ is the $z \times n$ item factor matrix (each column corresponds to a movie). We set the rank z of the factor matrices $\mathbf{P}$ and $\mathbf{Q}$ to be such that $z \ll k, n$.

We follow (Hu et al. 2008, Steck 2010; 2013), and use alternative least square based method to estimate the factor matrices $\mathbf{P}$ and $\mathbf{Q}$. To reduce the problem with overfitting, we use ℓ_2 regularization. Our final objective function is as given below

$$\min_{\mathbf{P},\mathbf{Q}} \mathbf{J} = \mathbf{C}\Big(||\mathbf{R} - \mathbf{PQ}||^2 + \lambda(||\mathbf{P}||^2 + ||\mathbf{Q}||^2)\Big) \tag{3.10}$$

We use the training weight matrix $\mathbf{C}$ following (Steck 2013). The training weight matrix is defined as follows,

$$\mathbf{C}_{ij} = \begin{cases} 1, & \text{if } \mathbf{R}_{ij} \text{ is observed} \\ 0, & \text{otherwise} \end{cases}$$

The equation 3.10 is non-convex, but convex if one of the variable is fixed. Morever, in typical settings the the number of users and movies can be very large, and hence direct optimization of $k \times n$ variables might not be feasible. Here, we use alternating least squares(ALS) based approach. At each iteration, we fix one of the variables $\mathbf{P}$ and $\mathbf{Q}$, and use stochastic gradient descent to solve the resulting convex optimization problem.

The gradient with respect to the user and item factor vectors becomes,

$$\nabla \mathbf{J}_{\mathbf{P}_i} = -2\mathbf{Q}^T\mathbf{C}^i\mathbf{R}_i + 2\mathbf{Q}^T\mathbf{C}^i\mathbf{Q}\mathbf{P}_i + 2\lambda\mathbf{P}_i$$
$$\nabla \mathbf{J}_{\mathbf{Q}_j} = -2\mathbf{P}^T\mathbf{C}^j\mathbf{R}_j + 2\mathbf{P}^T\mathbf{C}^j\mathbf{P}\mathbf{Q}_j + 2\lambda\mathbf{Q}_j$$

The final update formula becomes,

$$\mathbf{P}_i = \left(\mathbf{Q}^T\mathbf{C}^i\mathbf{Q} + \lambda\mathbf{I}\right)^{-1}\mathbf{Q}^T\mathbf{C}^i\mathbf{R}_i$$
$$\mathbf{Q}_i = \left(\mathbf{P}^T\mathbf{C}^j\mathbf{P} + \lambda\mathbf{I}\right)^{-1}\mathbf{P}^T\mathbf{C}^j\mathbf{R}_j$$

Table 3.21 – *Experimental Results on MovieLens using ALS (top 10 recommendations)*

	MMR λ=0.1	MMR λ=0.5	MMR λ=0.8	MSD λ=0.1	MSD λ=0.5	MSD λ=0.8
Genre Coverage	53.33	51.65	51.40	51.48	52.09	52.46
ILD	14.60	14.12	14.10	14.01	14.17	14.30
Catalog Coverage	2.42	4.79	4.99	4.47	3.83	3.38
Stratified Recall	0.71	1.26	1.31	1.15	1.04	0.94
DCG	1.19	2.76	2.97	2.25	1.85	1.64
Precision@k	0.19	0.41	0.45	0.39	0.31	0.27

Table 3.22 – *Experimental Results on MovieLens using ALS (top 10 recommendations)*

	MOD	SDR γ=0.8	SDR γ=0.5	SDR γ=0.1	SDR $\gamma\to0$	SDR conv γ=0.5
Genre Coverage	50.60	51.03	51.11	53.74	61.45	50.58
ILD	14.47	14.52	14.52	14.91	17.92	14.31
Catalog Coverage	1.64	1.61	1.63	2.54	4.64	1.57
Stratified Recall	2.07	2.05	2.07	2.72	4.93	1.96
DCG	8.99	8.37	8.43	10.03	19.13	7.98
Precision@k	1.15	1.08	1.09	1.39	2.39	1.03

where $\mathbf{I}$ is the identity matrix, and $\mathbf{C}^i$ is the diagonal matrix corresponding to the entry for $\mathbf{C}_i$ i.e. $\mathbf{C}^i = diag(\mathbf{C}_i)$.

Ideally, regularization parameter λ is tuned using hold-out or cross-validation. But in our experiments, we set it to the value 0.01. We used cosine function as the similarity measure but the similarity is defined over the item factors obtained using the ALS algorithm unlike the aforementioned setup. In all other aspects, we followed the same settings as in the earlier case. The results of our experiments are given in Table 3.21 and 3.22 for MovieLens and Table 3.23 and 3.24 for Yahoo! Movies.

As it can be noted, the results are inferior to the previous experimental setup. In case of MovieLens, as in the previous case SDR algorithm gives better result compared to MMR and MSD, but the relative magnitude is less compared to the previous setup. In case of Yahoo! Movies, SDR performance degrades drastically compared to MMR and MSD algorithms on all performance metrics. We did not further explore the ALS based approach on eclectic users. We reserve to carry out further analysis on latent factor based approach for diversification in the future.

Table 3.23 – *Experimental Results on Yahoo! Movies using ALS (top 10 recommendations)*

	MMR λ=0.1	MMR λ=0.5	MMR λ=0.8	MSD λ=0.1	MSD λ=0.5	MSD λ=0.8
Genre Coverage	73.28	75.74	75.25	75.11	75.75	75.85
ILD	16.37	17.08	16.95	16.74	16.93	16.96
Catalog Coverage	0.67	0.87	0.85	0.80	0.80	0.77
Stratified Recall	1.75	2.05	1.86	1.80	2.13	2.14
DCG	1.18	1.56	1.43	1.18	1.42	1.43
Precision@k	0.20	0.24	0.21	0.21	0.25	0.25

Table 3.24 – *Experimental Results on Yahoo! Movies using ALS (top 10 recommendations)*

	MOD	SDR γ=0.8	SDR γ=0.5	SDR γ=0.1	SDR $\gamma\to0$	SDR conv γ=0.5
Genre Coverage	53.97	53.98	54.11	55.81	62.30	54.08
ILD	10.72	10.72	10.73	10.96	12.68	10.73
Catalog Coverage	0.34	0.34	0.34	0.44	0.64	0.33
Stratified Recall	1.33	1.34	1.36	1.91	3.05	1.31
DCG	0.44	0.45	0.46	0.69	2.22	0.43
Precision@k	0.05	0.06	0.06	0.10	0.21	0.06

3.5 Conclusion

We presented a new criterion that captures both relevance and diversity for ranking applications. The criterion can be approximately optimized with an efficient greedy algorithm; the algorithm can be applied in any ranking scenario where we have access to similarities between items and a subset of items which are known to be of interest to the user or relevant to the query. Experiments on benchmark datasets for recommender systems showed that the algorithm performs well, both in terms of relevance and diversity compared to a strong baseline. But in case of web search, experiments on the benchmark datasets showed that the algorithm has clear performance advantage over the state of the art 'Learning to Rank' algorithm but inferior to the state of the art re-ranking algorithms. But considering the fact that our algorithm works in a transductive setting and thus bypassing the relevance score estimation step, it generates results cheaply and reasonably fast.

Conclusion

4

In this dissertation, we studied algorithms for two practically important problems which comes under the general class of machine learning problems called multi-objective learning problems. In Chapter 1, we gave some examples of multi-objective learning problems. In fact, many of the well studied problems in machine learning can be classified under multi-objective learning problems, which includes the problem of finding classifiers for optimal multi-variate performance measures, subset selection problems, ranking items in recommender systems and information retrieval problems in general etc.

The scalarization method is one of the most popular and efficient method for finding solutions for multi-objective learning problems. We introduced the concept of scalarization in chapter 1. Many of the state of the art algorithms for the aforementioned problems are in fact instantiations of the scalarization method.

We studied the problem of finding the optimal classifier for multivariate performance measures like F_β-measure and Jaccard Index in chapter 2. Our analysis established that the optimal classifier for F_β-measure and Jaccard Index can be obtained using cost-sensitive classification with the proper cost vectors in binary, multiclass and multilabel classification schemes. Moreover, we have established the fact that cost-sensitive classification is an instantiation of the scalarization method.

When considering algorithms for multi-objective learning problems, scalarization need not be the "to go" approach. It is very important to consider the domain specific information and objective functions. In chapter 3, we have demonstrated that optimizing the application specific objective function will give superior results compared to scalarization based methods. We proposed a new objective function which captures both relevance and diversity in a single criterion and experimentally vali-

dated that the proposed method outperforms state of the art scalarization based approaches for the task of diverse ranking.

4.1 Future Work

We now discuss some avenues for future work in the area of multi-objective learning. Some of these works are an extension of the work we presented in the earlier chapters.

4.1.1 Group Recommendation

Group recommendation is a personalized recommendation task where one has to deal with many competing objectives. Given a group of users and a set of items whose preferences to the users in the group is not known, the group recommendation task can be defined as selecting a subset of items with fixed cardinality such that the selected set is universally acceptable by the members of the group. One has to consider the agreements and the disagreements for the items to be recommended between different users of the group. In the state of the art methods, the problem is solved using the weighted sum approach, where the algorithm explicitly trades-off the agreement and disagreement aspects (Boratto and Carta 2010). Based on the approach we proposed in chapter 3, the problem of Group Recommendation can be done very efficiently by optimizing a domain specific objective function based on the user interest coverage. Here, we propose an algorithm for Group Recommendation based on submodular maximization.

A typical scenario arises in online deal marketplaces like Groupon[1] and LivingSocial[2]. In such settings, one has to choose a fixed set of deal coupons for product discounts to recommend such that it maximizes the user participation. The deal aggregation and recommendation algorithm has to select a fixed number of deals on a daily, weekly or monthly basis for the given city demography. Here, the algorithm has to take into account many competing individual user preferences within the demography for different set of deals, and the customers within the city can be considered as a group. In addition to the individual user preferences, the algorithm has to deal with customers participation behaviour. Some users might be be more loyal than others. Based on the above observation, we propose a new group recommendation algorithm which is a generalization of our algorithm proposed in chapter 3.

Here, we consider a more general setup. We are given a set of n items $\mathcal{X}$ and set of m users $\mathcal{U}$. The subset of users form the set of groups $\mathcal{G}$. The group demographics can evolve over time , and a user may or may

[1] https://www.groupon.com/
[2] https://www.livingsocial.com/

not be a member of a group, and a user can be part of multiple groups. We also assume the existence of an affinity function over the item space $h : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$. We does not require the affinity function to be symmetric or transitive. The user preference for the items are indicated using an ordinal number where we assume higher values indicates stronger preference.

Given $\mathcal{X}$ and the corresponding matrix of affinity values represented as $\mathbf{W}$, we could view the pair $(\mathcal{X}, \mathbf{W})$ as a complete graph where the edges are weighted according to the values in $\mathbf{W}$. Given the group $\mathcal{G}$ of user set $\mathcal{U}$ and their corresponding past preferences for item set $\mathcal{I} \subset \mathcal{X}$, the item set $\mathcal{I}$ defines a subgraph of $(\mathcal{X}, \mathbf{W})$.

We define the group consensus score with respect to the set $\mathcal{S} = \mathcal{X} \setminus \mathcal{I}$ as given below

$$GScore(\mathcal{G}, \mathcal{I}, \mathcal{S}) = \sum_{u \in \mathcal{G}} C_u \sum_{i \in \mathcal{I}} f\Big(\sum_{j \in \mathcal{S}} f^{-1}(\mathbf{W}_{ij}) \Big) \tag{4.1}$$

where C_u is a user specific value which can be used to adjust the varying user dynamics.

For any concave function f, the objective function *GScore* in 4.1 becomes a submodular function, and the group recommendation problem reduces to submodular function maximization with cardinality constraint.

4.1.2 Online F-measure Optimization

Recently Busa-Fekete et al. (2015) proposed an algorithm to select optimal threshold in case of threshold based algorithms for optimizing F_β-measure. Online learning is becoming increasing popular in machine learning, where the aim is to develop learning algorithms which learns from stream data (online data) to minimize the cumulative regret over the number of examples whereas in the traditional learning algorithms learn from batch data Cesa-Bianchi and Lugosi (2006). There is a growing interest in developing algorithms in online settings. An interesting future work is to develop algorithms for optimizing multivariate performance metrics discussed in chapter 2 in online settings. Majority of the online algorithms for binary, multiclass and multilabel consider only the cumulative error rate and so far there is no work related to developing algorithms for complex performance metrics. In Busa-Fekete et al. (2015), the authors argued that thresholding the class probability scores result in optimal F_β-measure in online settings. But the problem of estimating class probability scores in online fashion is not well studied.

4.1.3 Online Submodular Maximization

In many real world applications, particularly in online world, data comes in the form streams and with the current size of internet, it is almost impossible to store the entire data in a disk. Typical examples of applications where one has to deal with such data includes mining access logs of internet servers like web, mail etc, summarization for news-wire services, classifying video streams etc. In applications like summarization and exemplar clustering, the objective is to maximize a submodular function over this data stream. The greedy algorithm and the accelerated greedy algorithm for submodular maximization given in chapter 3 requires full access to the data. So in practice one has to devise an "online greedy algorithm" for submodular function maximization over data streams.

Krause and Gomes (2010) proposed an online version of the greedy algorithm discussed in chapter 3 for the task of cardinality constrained submodular maximization. The algorithm is based on keeping the most prominent k elements seen so far in the memory. When a new data point comes, the algorithm checks whether swapping it with any of the k currently stored data points results in the value of the utility function, and swaps accordingly. A recent algorithm by Badanidiyuru et al. (2014) makes use of heuristic approach. They use the minimum and maximum bound on the optimal value based on the current element wise maximum and greedily select the data points based on thresholding over a discretized interval. The algorithm is in fact multi-pass (have to go over the data streams multiple times), but they propose to run each pass parallely.

An interesting line of future work is to propose a true single-pass online greedy algorithm for submodular maximization. We are working on an algorithm based on the idea proposed by Krause and Gomes (2010). In addition to the above proposed work, there is a plethora of work in the area of multi-objective learning which is currently under investigation, related to bi-objective matching, online matrix completion etc.

Bibliography

Naoki Abe, Bianca Zadrozny, and John Langford. An iterative method for multi-class cost-sensitive learning. In *KDD*, pages 3–11. ACM, 2004. ISBN 1-58113-888-1. (Cité page 29.)

Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *WSDM*. ACM, 2009. (Cité pages 64 and 65.)

M. Anthony and P.L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 2009. ISBN 9780521118620. (Cité page 17.)

Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. Optimal greedy diversity for recommendation. In *IJCAI*, 2015. (Cité page 63.)

F. Bach. *Learning with Submodular Functions: A Convex Optimization Perspective*. Foundations and trends in machine learning. Now Publishers, 2013. ISBN 9781601987570. (Cité pages 52 and 70.)

Francis R. Bach, David Heckerman, and Eric Horvitz. Considering cost asymmetry in learning classifiers. *J. Mach. Learn. Res.*, 7:1713–1741, December 2006. ISSN 1532-4435. (Cité pages 5, 40, 41, 42, and 46.)

Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 671–680. ACM, 2014. (Cité page 88.)

Ludovico Boratto and Salvatore Carta. State-of-the-art in group recommendation and new approaches for automatic identification of groups. In *Information retrieval and mining in distributed environments*, pages 1–20. Springer, 2010. (Cité page 86.)

Allan Borodin, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *PODS*, pages 155–166. ACM, 2012. (Cité pages 64, 66, and 67.)

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787. (Cité pages 10 and 39.)

Róbert Busa-Fekete, Balázs Szörényi, Krzysztof Dembczynski, and Eyke Hüllermeier. Online f-measure optimization. In *Advances in Neural Information Processing Systems*, pages 595–603, 2015. (Cité page 87.)

Alberto Cambini and Laura Martein. *Generalized Convexity and Optimization*, volume 616 of *Lecture Notes in Economics and Mathematical Systems*. Springer, 2009. (Cité pages 25 and 26.)

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336. ACM, 1998. (Cité pages 49, 63, 64, and 66.)

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006. (Cité pages 7 and 87.)

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. (Cité page 42.)

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised Learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN 9780262033589. URL `https://books.google.fr/books?id=kfqvQgAACAAJ`. (Cité page 52.)

Olivier Chapelle, Shihao Ji, Ciya Liao, Emre Velipasaoglu, Larry Lai, and Su-Lin Wu. Intent-based diversification of web search results: metrics and algorithms. *Information Retrieval*, 14(6):572–592, 2011. (Cité pages 64 and 65.)

Weiwei Cheng, Krzysztof Dembczynski, Eyke Hüllermeier, Adrian Jaroszewicz, and Willem Waegeman. F-measure maximization in topical classification. In *RSCTC*, volume 7413 of *LNCS*, pages 439–446. Springer, 2012. (Cité pages 17, 20, 29, and 47.)

Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM, 2008. (Cité page 65.)

Stéphan Clémençon and Nicolas Vayatis. Adaptive estimation of the optimal roc curve and a bipartite ranking algorithm. In *Algorithmic Learning Theory*, pages 216–231. Springer, 2009. (Cité page 40.)

T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction To Algorithms*. MIT Press, 2001. ISBN 9780262032933. (Cité page 59.)

Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys*, pages 39–46. ACM, 2010. (Cité page 69.)

Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1057–1064, New York, NY, USA, 2011. ACM. (Cité page 7.)

Krzysztof Dembczynski, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. An exact algorithm for F-measure maximization. In *NIPS*, pages 1404–1412, 2011. (Cité pages 20, 22, 29, and 47.)

Krzysztof Dembczynski, Arkadiusz Jachnik, Wojciech Kotlowski, Willem Waegeman, and Eyke Hüllermeier. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *ICML*, volume 28, pages 1130–1138. JMLR Workshop and Conference Proceedings, May 2013. (Cité pages 20 and 29.)

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006. (Cité pages 76 and 78.)

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Applications of mathematics : stochastic modelling and applied probability. Springer, 1996. ISBN 9780387946184. (Cité page 17.)

Avinava Dubey, Soumen Chakrabarti, and Chiranjib Bhattacharyya. Diversity in ranking via resistive graph centers. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 78–86. ACM, 2011. (Cité page 65.)

Jack Edmonds. Matroids and the greedy algorithm. *Mathematical programming*, 1(1):127–136, 1971. (Cité page 58.)

M. Ehrgott and X. Gandibleux. *Multiple Criteria Optimization. State of the art annotated bibliographic surveys*. Kluwer Academic, Dordrecht, 2002. (Cité pages 12, 39, and 63.)

Charles Elkan. The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 973–978, 2001. (Cité pages 18 and 29.)

Rong E. Fan and C. J. Lin. A study on threshold selection for multi-label classification. Technical report, National Taiwan University, 2007. (Cité page 44.)

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008. (Cité page 42.)

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956. (Cité page 60.)

Akinori Fujino, Hideki Isozaki, and Jun Suzuki. Multi-label text categorization with model combination based on f1score maximization. In *Proceedings of IJCNLP*, pages 823–828, 2008. (Cité pages 20, 23, and 29.)

Satoru Fujishige. *Submodular functions and optimization*, volume 58. Elsevier, 2005. (Cité pages 52 and 53.)

Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *WWW*, pages 381–390. ACM, 2009. (Cité pages 64 and 65.)

Yves Grandvalet, Johnny Mariéthoz, and Samy Bengio. A probabilistic interpretation of SVMs with an application to unbalanced classification. In *NIPS*, 2005. (Cité page 42.)

Jingrui He, Hanghang Tong, Qiaozhu Mei, and Boleslaw Szymanski. Gender: A generic diversified ranking algorithm. In *NIPS*, pages 1142–1150, 2012. (Cité pages 64 and 65.)

M. Hollander and D.A. Wolfe. *Nonparametric statistical methods*. Wiley Series in Probability and Statistics - Applied Probability and Statistics Section. Wiley, 1973. ISBN 9780471406358. URL `https://books.google.fr/books?id=ajxMAAAAMAAJ`. (Cité pages 77 and 79.)

Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. Ieee, 2008. (Cité page 81.)

Neil J Hurley. Personalised ranking with diversity. In *RecSys*, pages 379–382. ACM, 2013. (Cité pages 63, 64, and 66.)

Martin Jansche. Maximum expected F-measure training of logistic regression models. In *HLT/EMNLP*. The Association for Computational Linguistics, 2005. (Cité pages 20 and 23.)

Martin Jansche. A maximum expected utility framework for binary sequence labeling. In *ACL*. The Association for Computational Linguistics, 2007. (Cité pages 20, 21, and 22.)

Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 377–384. ACM Press, 2005. (Cité pages 20, 22, and 41.)

Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009. (Cité page 38.)

Jin-Dong Kim, Yue Wang, and Yamamoto Yasunori. The genia event extraction shared task, 2013 edition - overview. In *Proceedings of the*

BioNLP Shared Task 2013 Workshop, pages 8–15, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. (Cité pages 17 and 28.)

Seung-Jean Kim, Alessandro Magnani, Sikandar Samar, Stephen Boyd, and Johan Lim. Pareto optimal linear classification. In *Proceedings of the 23rd international conference on Machine learning*, pages 473–480. ACM, 2006. (Cité page 5.)

Oluwasanmi O Koyejo, Nagarajan Natarajan, Pradeep K Ravikumar, and Inderjit S Dhillon. Consistent binary classification with generalized performance metrics. In *Advances in Neural Information Processing Systems 27*, pages 2744–2752. Curran Associates, Inc., 2014. (Cité pages 6, 18, 20, 23, 38, and 42.)

Oluwasanmi O Koyejo, Nagarajan Natarajan, Pradeep K Ravikumar, and Inderjit S Dhillon. Consistent multilabel classification. In *Advances in Neural Information Processing Systems*, pages 3321–3329, 2015. (Cité pages 6, 18, 23, 36, and 42.)

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3(19):8, 2012. (Cité page 52.)

Andreas Krause and Ryan G Gomes. Budgeted nonparametric learning from data streams. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 391–398, 2010. (Cité page 88.)

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007. (Cité page 59.)

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *ACL*, pages 510–520, 2011. (Cité page 67.)

Zachary C. Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize F1 measure. In *Machine Learning and Knowledge Discovery in Databases*, volume 8725 of *Lecture Notes in Computer Science*, pages 225–239. Springer, 2014. (Cité pages 20 and 33.)

Qiaozhu Mei, Jian Guo, and Dragomir Radev. Divrank: the interplay of prestige and diversity in information networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1009–1018. Acm, 2010. (Cité page 65.)

Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pages 234–243. Springer, 1978. (Cité page 58.)

David R. Musicant, Vipin Kumar, and Aysel Ozgur. Optimizing F-measure with support vector machines. In *Proceedings of the FLAIRS Conference*, pages 356–360, 2003. (Cité pages 20 and 22.)

Ye Nan, Kian Ming Adam Chai, Wee Sun Lee, and Hai Leong Chieu. Optimizing F-measure: A tale of two approaches. In *ICML*. icml.cc / Omnipress, 2012. (Cité pages 20, 21, 22, and 23.)

Harikrishna Narasimhan, Rohit Vaish, and Shivani Agarwal. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *Advances in Neural Information Processing Systems 27*, pages 1493–1501. Curran Associates, Inc., 2014. (Cité pages 6, 18, 20, 23, and 42.)

Harikrishna Narasimhan, Harish Ramaswamy, Aadirupa Saha, and Shivani Agarwal. Consistent multiclass algorithms for complex performance measures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2398–2407, 2015. (Cité pages 6, 18, 23, and 46.)

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1), 1978. (Cité page 56.)

Jinoh Oh, Sun Park, Hwanjo Yu, Min Song, and Seung-Taek Park. Novel recommendation based on personal popularity tendency. In *ICDM*, pages 507–516. IEEE, 2011. (Cité pages 63 and 64.)

Shameem Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. Optimizing F-measures by cost-sensitive classification. In *Advances in Neural Information Processing Systems 27*, pages 2123–2131. Curran Associates, Inc., 2014. (Cité pages 6 and 18.)

James Petterson and Tibério S Caetano. Reverse multi-label learning. In *NIPS*, volume 1, pages 1912–1920, 2010. (Cité page 29.)

James Petterson and Tibério S Caetano. Submodular multi-label learning. In *NIPS*, pages 1512–1520, 2011. (Cité page 29.)

Ignazio Pillai, Giorgio Fumera, and Fabio Roli. F-measure optimisation in multi-label classifiers. In *ICPR*, pages 2424–2427. IEEE, 2012. ISBN 978-1-4673-2216-4. (Cité pages 20 and 22.)

Bruno Pradel, Nicolas Usunier, and Patrick Gallinari. Ranking with non-random missing ratings: influence of popularity and positivity on evaluation metrics. In *RecSys*, pages 147–154. ACM, 2012. (Cité page 67.)

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007. ISBN 0521880688, 9780521880688. (Cité page 46.)

Chao Qian, Yang Yu, and Zhi-Hua Zhou. Subset selection by pareto optimization. In *Advances in Neural Information Processing Systems*, pages 1774–1782, 2015. (Cité page 7.)

Filip Radlinski and Susan Dumais. Improving personalized web search using result diversification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 691–692. ACM, 2006. (Cité page 65.)

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *ICML*, pages 784–791. ACM, 2008. (Cité pages 64 and 65.)

Filip Radlinski, Paul N Bennett, Ben Carterette, and Thorsten Joachims. Redundancy, diversity and interdependent document relevance. In *SIGIR Forum*, volume 43, pages 46–52. ACM, 2009. (Cité page 64.)

Karthik Raman, Pannaga Shivaswamy, and Thorsten Joachims. Online learning to diversify from implicit feedback. In *SIGKDD*, pages 705–713. ACM, 2012. (Cité page 64.)

Mark D Reid and Robert C Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11(Sep):2387–2422, 2010. (Cité page 23.)

C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN 0408709294. (Cité pages 17 and 51.)

L. Rosasco, E. De, Vito A. Caponnetto, M. Piana, and A. Verri. Are loss functions all the same. *Neural Computation*, 15, 2004. (Cité page 32.)

Walter Rudin. *Functional analysis*. International Series in Pure and Applied Mathematics. McGraw-Hill Inc., New York, second edition, 1991. (Cité page 10.)

Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. Exploiting query reformulations for web search result diversification. In *WWW*. ACM, 2010. (Cité pages 64 and 65.)

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*. ACM, 2001. (Cité pages 52, 62, 63, and 69.)

Clayton Scott. Calibrated asymmetric surrogate losses. *Electronic Journal of Statistics*, 6:958–992, 2012. (Cité pages 29 and 41.)

Karen Spärck-Jones, Stephen E Robertson, and Mark Sanderson. Ambiguous requests: implications for retrieval tests, systems and theories. In *SIGIR Forum*. ACM, 2007. (Cité pages 3 and 51.)

Harald Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 713–722. ACM, 2010. (Cité page 81.)

Harald Steck. Item popularity and recommendation accuracy. In *RecSys*, pages 125–132. ACM, 2011. (Cité page 68.)

Harald Steck. Evaluation of recommendations: rating-prediction and ranking. In *RecSys*, pages 213–220. ACM, 2013. (Cité page 81.)

Ingo Steinwart. How to compare different loss functions and their risks. *Constructive Approximation*, 26(2):225–287, 2007. (Cité pages 29, 33, and 41.)

Ruilong Su, Li'Ang Yin, Kailong Chen, and Yong Yu. Set-oriented personalized ranking for diversified top-n recommendation. In *RecSys*, pages 415–418. ACM, 2013. (Cité pages 63, 64, and 66.)

Gábor Takács and Domonkos Tikk. Alternating least squares for personalized ranking. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 83–90. ACM, 2012. (Cité page 64.)

Hanghang Tong, Jingrui He, Zhen Wen, Ravi Konuru, and Ching-Yung Lin. Diversified ranking on large graphs: an optimization viewpoint. In *SIGKDD*, pages 1028–1036. ACM, 2011. (Cité page 65.)

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(Sep): 1453–1484, 2005. (Cité page 22.)

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007. (Cité page 28.)

S. Vargas and P. Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *RecSys*, pages 109–116. ACM, 2011. (Cité page 63.)

Saúl Vargas, Linas Baltrunas, Alexandros Karatzoglou, and Pablo Castells. Coverage, redundancy and size-awareness in genre diversity for recommender systems. In *RecSys*, pages 209–216. ACM, 2014. (Cité page 63.)

Willem Waegeman, Krzysztof Dembczyński, Arkadiusz Jachnik, Weiwei Cheng, and Eyke Hüllermeier. On the bayes-optimality of F-measure maximizers. *Journal of Machine Learning Research*, 15:3333–3388, 2014. (Cité pages 20, 23, and 38.)

Le Wu, Qi Liu, Enhong Chen, Nicholas Jing Yuan, Guangming Guo, and Xing Xie. Relevance meets coverage: A unified framework to generate diversified recommendations. *ACM Trans. Intell. Syst. Technol.*, 7(3), 2016. (Cité pages 63 and 64.)

Yisong Yue and Thorsten Joachims. Predicting diverse subsets using structural svms. In *ICML*, pages 1224–1231. ACM, 2008. (Cité pages 64 and 65.)

Cheng Xiang Zhai, William W Cohen, and John Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR*, pages 10–17. ACM, 2003. (Cité pages 49 and 64.)

Benyu Zhang, Hua Li, Yi Liu, Lei Ji, Wensi Xi, Weiguo Fan, Zheng Chen, and Wei-Ying Ma. Improving web search results using affinity graph. In *SIGIR*, pages 504–511. ACM, 2005. (Cité pages 49 and 64.)

Mi Zhang and Neil Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys*, pages 123–130. ACM, 2008. (Cité page 69.)

Zhi-Hua Zhou and Xu-Ying Liu. On multi-class cost-sensitive learning. *Computational Intelligence*, 26(3):232–257, 2010. (Cité page 29.)

Xiaojin Zhu, Andrew B Goldberg, Jurgen Van Gael, and David Andrzejewski. Improving diversity in ranking using absorbing random walks. In *HLT-NAACL*, pages 97–104, 2007. (Cité pages 64 and 65.)

Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM, 2005. (Cité page 65.)