



## UvA-DARE (Digital Academic Repository)

### Combining strategies efficiently: high-quality decisions from conflicting advice

Koolen, W.M.

**Publication date**  
2011

[Link to publication](#)

#### **Citation for published version (APA):**

Koolen, W. M. (2011). *Combining strategies efficiently: high-quality decisions from conflicting advice*. [Thesis, fully internal, Universiteit van Amsterdam].

#### **General rights**

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

#### **Disclaimer/Complaints regulations**

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

---

## Abstract

In this dissertation we study *machine learning*: the automated discovery and exploitation of regularities in data. We may use regularities identified in *objects* to explain the past (e.g. archaeology, justice), as well as regularities found in *processes* to predict the future (e.g. weather, stock market) and guide our actions.

With ubiquitous computational resources, machine learning algorithms have become pervasive. For example, they manage financial portfolios and power-saving policy, provide personalised movie recommendations as well as advertisements, and form the core of state-of-the-art data compression software.

This dissertation develops the theory of *online learning*, a branch of machine learning that investigates sequential decision problems with immediate feedback. In particular, we study the setting called *prediction with expert advice*. Our task is to predict a sequence of data. Each trial, we may first consult a given set of experts. We then combine their advice and issue our prediction of the next outcome. Finally, the next outcome is revealed, and we incur loss based on the discrepancy between our prediction and it.

The goal is to build efficient algorithms with small *regret*, i.e. the difference between the incurred cumulative loss and the loss of the best strategy in hindsight from a fixed reference class. In this sense, the strategies in the reference class are the patterns, and achieving small regret means *learning* which reference strategy best models the data. The main difference between the learning problems we consider is the complexity of the reference set. Algorithms for prediction with expert advice have many applications including classification, regression, hypothesis testing, model selection, data compression, gambling and

investing in the stock market.

In Chapter 2 we give a game-theoretic analysis of the simplest on-line learning problem, the prediction of a sequence of binary outcomes under 0/1 loss with the help of two experts. For this simple problem, we compute the minimax, i.e. game-theoretically optimal, regret, and show how to implement the optimal strategy efficiently. We then give special attention to the case that one of the experts is good. We conclude with a new result: the optimal algorithm for competing with the set of meta-experts that switch between the two basic experts.

In Chapter 3 we show how models for prediction with expert advice can be defined concisely and clearly using hidden Markov models (HMMs); standard algorithms can then be used to efficiently calculate how the expert predictions should be weighted. We focus on algorithms for *tracking the best expert*. Here the strategies in the reference set follow the advice of a single expert, but this expert may change between trials. We cast existing models as HMMs, starting from the fixed share algorithm, recover the running times and regret bounds for each algorithm, and discuss how they are related. We also describe three new models for switching between experts.

In Chapter 4 we extend the setting to tracking the best *learning* expert. Whereas vanilla experts can be tapped for advice about the current trial, learning experts may be queried for advice *given each possible subset* of the past data. This additional power is available to both the algorithm and the reference strategies. Achieving small regret thus means learning how to partition the trials, and which learning expert to train and follow within each partition cell. We give efficient algorithms with small regret for tracking learning experts that can themselves be formalised using the expert HMMs of Chapter 3.

In Chapter 5 we consider reference strategies that switch between two experts based on their *cumulative loss* instead of on *time*. This chapter is formulated in financial terms to make the presentation more intuitive. We present a simple online two-way trading algorithm that exploits fluctuations in the unit price of an asset. Rather than analysing worst-case performance under some assumptions, we prove a novel, unconditional performance bound that is parameterised either by the actual dynamics of the price of the asset, or by a simplifying model thereof. We discuss application of the results to prediction with expert advice, data compression and hypothesis testing.

In Chapter 6 we consider prediction with *structured concepts*. Each round we select a concept, which is composed of components. The loss of a concept is the sum of the losses of its components. Whereas the losses of different components are independent, the losses of different *concepts* are highly related. We develop an online algorithm, called *Component Hedge* that exploits this dependence, and thereby avoids the so called *range factor* that arises when the dependences are ignored. We show that Component Hedge has optimal regret bounds for a large variety of structured concept classes.