

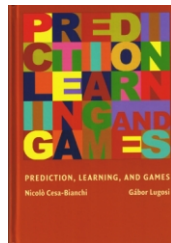
Ensembles and Multiple Classifiers: A Game-Theoretic View

Nicolò Cesa-Bianchi

Università degli Studi di Milano

A nonstatistical look at machine learning

- The statistical approach is at the basis of the most successful applications of machine learning in the past twenty years
- As the range of machine learning applications widens, new paradigms are needed



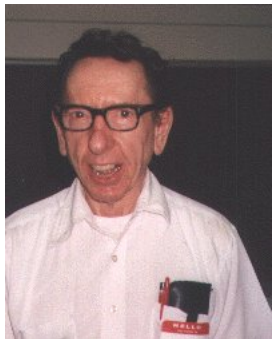
Some hard cases for statistical modelling

- Data source is highly nonstationary
- Environment reacts to the learner (e.g., spam)

On a more philosophical level

Is statistics the only language for describing the phenomenon of learning in machines?

Theory of repeated games



James Hannan



David Blackwell

Learning to play a game (1956)

Play a game repeatedly against a possibly suboptimal opponent

Zero-sum 2-person games played more than once

	1	2	...	M
1	$\ell(1,1)$	$\ell(1,2)$...	
2	$\ell(2,1)$	$\ell(2,2)$...	
\vdots	\vdots	\vdots	\ddots	
N				

$N \times M$ known loss matrix

- Row player (**player**) has N actions
- Column player (**opponent**) has M actions

For each game round $t = 1, 2, \dots$

- Player chooses action i_t and opponent chooses action y_t
- The player suffers loss $\ell(i_t, y_t)$ (= gain of opponent)

Player can learn from opponent's history of past choices y_1, \dots, y_{t-1}

Prediction with expert advice



Volodya Vovk



Manfred Warmuth

Opponent's moves y_1, y_2, \dots define a **sequential prediction problem** with loss function ℓ

- 1 Play action I_t from $1, \dots, N$
- 2 Observe next value y_t
- 3 Incur loss $\ell(I_t, y_t)$

Exponentially weighted forecaster

At time t pick action i with probability proportional to

$$\exp(-\eta \text{Loss}_{i,t})$$

where $\text{Loss}_{i,t}$ is **total loss** of action i up to now

How to use expert advice

The average per-round expected loss of the forecaster converges to that of the **best action for the observed sequence** at optimal rate

$$\sqrt{\frac{\ln N}{T}}$$

where N is number of actions and T is the number of time steps

Note: no dependence on number of opponent's actions

Sequential Aggregation of Experts

- N experts (oracles —no assumptions) suggest actions to play
- The forecaster mixes over experts rather than over actions

Exploiting structure of action space

- Convex action space \mathcal{X}
- Outcome space \mathcal{Y}
- Convex loss $\ell : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$

For $t = 1, 2, \dots$

- 1 Get expert advice $\xi_{1,t}, \dots, \xi_{N,t} \in \mathcal{X}$
- 2 Compute aggregated prediction $x_t = F_t(\xi_{1,t}, \dots, \xi_{N,t}) \in \mathcal{X}$
- 3 Observe next value $y_t \in \mathcal{Y}$
- 4 Incur loss $\ell(x_t, y_t)$

Exploiting geometry of loss function

Mixing the advice of N experts

$$x_t = \sum_{i=1}^N \xi_{i,t} p_{i,t} \quad \text{where} \quad p_{i,t} \propto \exp(-\eta \text{Loss}_{i,t})$$

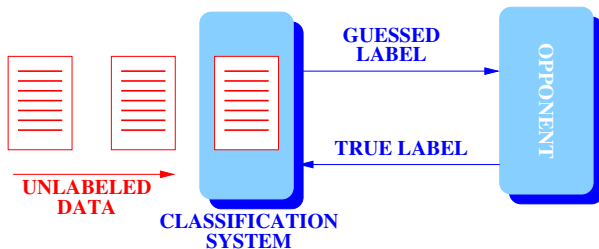
$\text{Loss}_{i,t}$ is total loss of expert i up to now

Optimal rates of aggregation

The average per-round loss of the forecaster converges to that of the **best expert for the observed sequence** at rate

- Convex losses: $\sqrt{\frac{\ln N}{T}}$
- Exp-concave losses: $\frac{\ln N}{T}$ (relative entropy loss, square loss)

From game theory to machine learning



- Opponent's moves y_t are viewed as **values or labels** assigned to observations $x_t \in \mathbb{R}^d$ (e.g., categories of documents)
- A repeated game between the player choosing a **model** f_t and the opponent choosing a label y_t for x_t
- Convergence to performance of **best model** in a given class (e.g., linear predictors with bounded norm)

Online learning algorithms

- **Simple:** easy to implement
- **Scalable:** local optimization vs. global optimization
- **Robust:** game-theoretic performance guarantees
- **Versatile:** classification, regression, ranking, structured prediction

Example: Ridge regression with square loss in \mathbb{R}^d

$$\sum_{t=1}^T \ell_t(f_t) \leq \inf_{\text{lin. models } g} \left(\sum_{t=1}^T \ell_t(g) + \|g\|^2 \right) + d \ln \left(1 + \frac{T}{d} \right)$$

where $\ell_t(f_t) = (f_t(\mathbf{x}_t) - y_t)^2$

Relating statistical to online learning

Statistical learning

- Examples (\mathbf{x}_t, y_t) are realizations of (\mathbf{X}_t, Y_t) drawn i.i.d. from fixed but unknown distribution
- Goal is to find a model $f : \mathbb{R}^d \rightarrow \mathbb{R}$ with small statistical risk
$$\text{risk}(f) = \mathbb{E} \left[\ell(f(\mathbf{X}), Y) \right]$$

Loss rate

When run on a sample $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)$ an online algorithm generates an **ensemble** f_1, \dots, f_T of models

Online analysis bounds the **loss rate** of the ensemble

$$\frac{1}{T} \sum_{t=1}^T \ell_t(f_t)$$

where $\ell_t(f_t)$ is loss of f_t on next example (\mathbf{x}_t, y_t)

Online to batch conversion

How does the ensemble's loss rate relate to risk?

- Model f_t is determined by past examples $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_{t-1}, y_{t-1})$
- If the data sequence is i.i.d., then the loss process defines a **martingale difference sequence** $\mathbb{E}_t[\ell_t(f_t) - \text{risk}(f_t)] = 0$
- Via martingale concentration inequalities,

$$\underbrace{\frac{1}{T} \sum_{t=1}^T \ell_t(f_t)}_{\text{loss rate}} - \underbrace{\frac{1}{T} \sum_{t=1}^T \text{risk}(f_t)}_{\text{average risk}} \rightarrow 0 \quad \text{a.s.}$$

- For linear models with convex loss,

$$\underbrace{\text{risk} \left(\frac{1}{T} \sum_{t=1}^T f_t \right)}_{\text{risk of average model}} \leq \underbrace{\frac{1}{T} \sum_{t=1}^T \text{risk}(f_t)}_{\text{average risk}}$$

Interacting online learners

Two basic scenarios

1 Multikernel learning:

- each online algorithm trains a model in its own RKHS
- want to improve over taking a flat average of kernels

2 Multitask learning:

- each online algorithm is solving a different prediction task
- want to improve over running each algorithm independently

Basic algorithm

Perceptron with update modulated by the gradient of an arbitrary norm

Online multikernel classification

- N kernels K_1, \dots, K_N
- Baselines: best kernel and flat kernel average

$$\frac{1}{T} \sum_i K_i(\mathbf{x}, \cdot)$$

Multikernel group-norm Perceptron

It learns a **linear classifier** $\text{sgn}(f)$ of the form

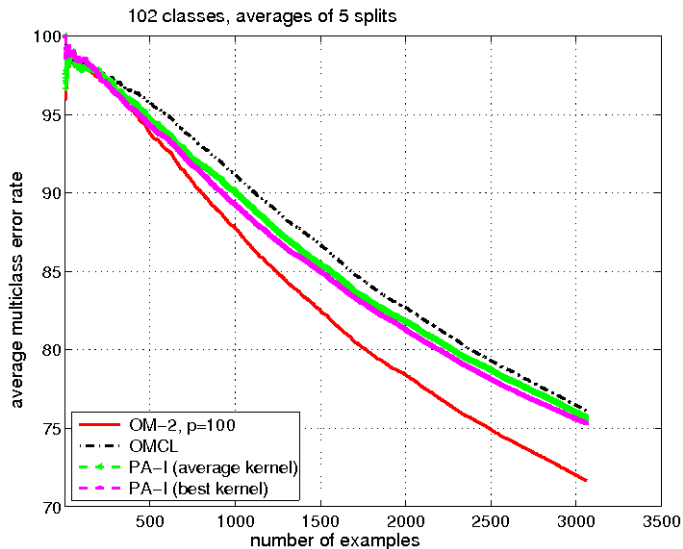
$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i \langle f_i, K_i(\mathbf{x}, \cdot) \rangle \quad f_i \in \mathcal{H}_i \quad \alpha_i \in \mathbb{R}$$

Performance is controlled by $(2, p)$ **group norm** of best classifiers

$$\underbrace{\| (f_1^*, \dots, f_N^*) \|_{2,p}}_{\text{best classifiers}} = \| (\|f_1^*\|_2, \dots, \|f_N^*\|_2) \|_p$$

For p large, performance improves when $(\|f_1^*\|_2, \dots, \|f_N^*\|_2)$ is **sparse**

Experiments on Caltech101 dataset (48 kernels)



Online multitask classification

- N classification tasks, where N can be large
- One stream of instances for each task (e.g., mail spam)
- Improve over N independent learners when tasks are **similar**

Multitask p -norm Perceptron

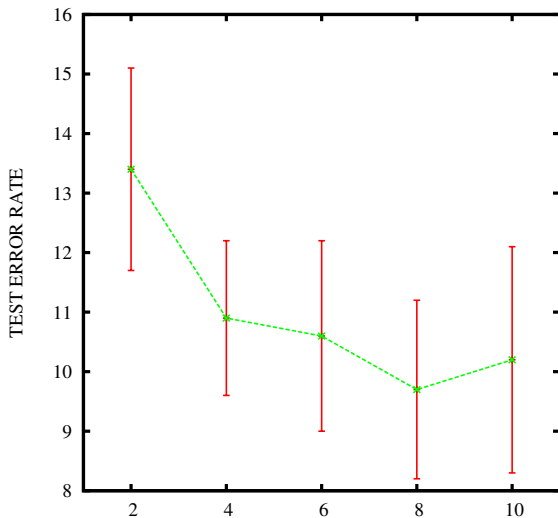
Performance is controlled by p -norm of SVD vector of matrix

$$U = \underbrace{[u_1^*, \dots, u_N^*]}_{\text{best classifiers}}$$

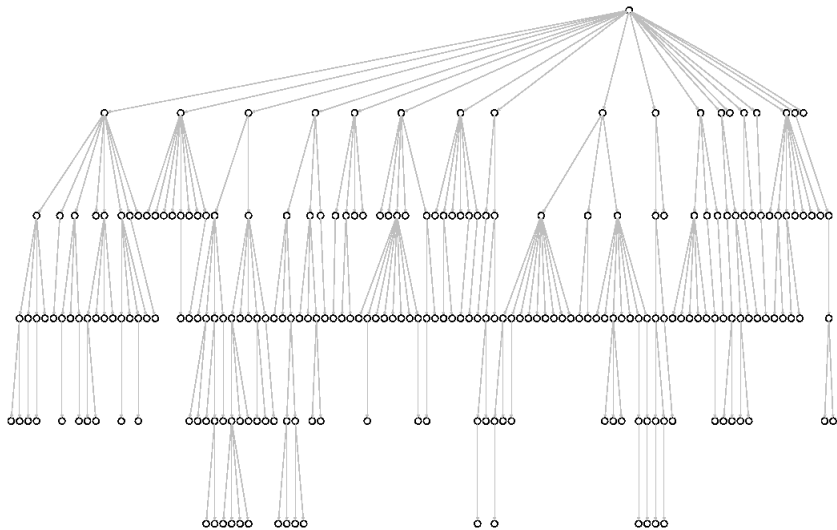
For $p = 2$, algorithm reduces to running N independent Perceptrons

For large p , performance improves when U has **small rank**

Experiments on the Spam dataset (15 tasks)



Online hierarchical classification



- Multiple and partial paths
- Loss exploits structure: a mistake in a node does not cause further mistakes in the subtree rooted at that node
- Probabilistic data model exploits structure: children are asked to predict on examples that are predicted positive by parents
- Online learning of Bayes optimal predictor for loss and data model

- Game theoretic approach: aggregation, interaction
- Efficiency allows large scale problems: data, tasks, classes, views
- In progress: aggregation of strategic experts (e.g., crowdsourcing)