On-Line Learning

Nicolò Cesa-Bianchi

Università degli Studi di Milano

November 3, 2005
Background

- Theory of repeated games
  (Hannan, 1956; Blackwell, 1956)
- Compression of individual sequences
  (Lempel and Ziv, 1976)
- Gambling and portfolio selection
  (Cover, 1965 and 1991)
- Pattern classification
  (Novikov, 1962; Littlestone, 1989)

Unifying framework
Prediction with expert advice
1. Prediction with expert advice
2. Connections with game theory
3. Learning with linear experts
4. The Perceptron algorithm and its extensions
5. Mistake bounds
6. Online learning with kernels
7. From mistake bounds to risk bounds
A forecaster predicts a binary sequence one bit at the time.

At each step $t = 1, 2, \ldots$ the forecaster predicts the $t$-th bit knowing the previous $t - 1$ bits:

0100010110 ? \ldots

After the prediction is made, the $t$-th bit is observed and the forecaster finds out whether a mistake was made.

Goal

Bound the number of prediction mistakes without making any statistical assumptions on the way the data sequence is generated.
The role of experts

- Want a nonstatistical framework where **good** forecasters can be distinguished from **bad** forecasters
- Any forecaster must use some map of the form

\[
\text{past observations} \rightarrow \text{predictions}
\]

- For each forecaster, there exists a bit sequence on which a mistake is made at each step

**Competitive analysis**

Compare the performance of the forecaster to that of a set of reference forecasters (**experts**).
A simple example

Forecaster competes against three experts on sequence 1101

<table>
<thead>
<tr>
<th></th>
<th>$t = 1$</th>
<th>$t = 2$</th>
<th>$t = 3$</th>
<th>$t = 4$</th>
<th>Mistakes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$M_1 = 1$</td>
</tr>
<tr>
<td>Expert 2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$M_2 = 3$</td>
</tr>
<tr>
<td>Expert 3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$M_3 = 3$</td>
</tr>
<tr>
<td>Forecaster</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$M = 2$</td>
</tr>
<tr>
<td>Bit sequence</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Goal (refined)

Predict each sequence almost as well as the best expert for that sequence
A more general prediction model

- Predict an unknown sequence \( y_1, y_2, \ldots \in y \) (outcome space)
- Predictions \( \hat{p} \) are chosen from \( X \) (decision space)
- Forecasters are scored with their cumulative loss

\[
\ell(\hat{p}_1, y_1) + \ell(\hat{p}_2, y_2) + \ldots
\]

where \( \ell : X \times Y \to \mathbb{R} \) is a loss function

Example

- Zero-one loss: \( X = Y = \{0, 1\} \) and \( \ell(\hat{p}, y) = \mathbb{I}_{\{\hat{p} \neq y\}} \)
- Quadratic loss: \( X = Y = [0, 1] \) and \( \ell(\hat{p}, y) = (\hat{p} - y)^2 \)
- Absolute loss: \( X = [0, 1], Y = \{0, 1\} \) and \( \ell(\hat{p}, y) = |\hat{p} - y| \)
On-line prediction with expert advice

Measure performance relatively to a set of $N$ experts

At each step $t = 1, 2, \ldots$

1. Get predictions (advice) $f_{1,t}, \ldots, f_{N,t} \in X$ of the experts
2. Compute prediction $\hat{p}_t \in X$
3. Outcome $y_t \in Y$ is revealed
4. Forecaster incurs loss $\ell(\hat{p}_t, y_t)$ and each expert $i$ incurs loss $\ell(f_{i,t}, y_t)$

Note

Experts are viewed as abstract entities, generating predictions in an unspecified way
Regret

\[ r_{i,t} = \ell(\hat{p}_{t}, y_t) - \ell(f_{i,t}, y_t) \]

\[ \hat{L}_n = \sum_{t=1}^{n} \ell(\hat{p}_{t}, y_t) \quad L_{i,n} = \sum_{t=1}^{n} \ell(f_{i,t}, y_t) \]

\[ R_{i,n} = \sum_{t=1}^{n} r_{i,t} = \hat{L}_n - L_{i,n} \]

\[ \max_{i=1,\ldots,N} R_{i,n} = \hat{L}_n - \min_{i=1,\ldots,N} L_{i,n} \]

We want to design \textit{consistent} forecasters, i.e. such that

\[ \lim_{n \to \infty} \frac{1}{n} \left( \max_{i=1,\ldots,N} R_{i,n} \right) = 0 \]

for any sequence of outcomes and all choices of expert advice.
Weighted average forecasters

- Assume decision space $\mathcal{X}$ is a **convex subset** of a linear space

  $$x, x' \in \mathcal{X} \implies \alpha x + (1 - \alpha)x' \in \mathcal{X} \quad \text{for all } 0 \leq \alpha \leq 1$$

- If $R_{i,t-1}$ is big, then we should predict more like expert $i$

  $$\hat{p}_t = \frac{\sum_{i=1}^{N} \mu(R_{i,t-1}) f_{i,t}}{\sum_{j=1}^{N} \mu(R_{j,t-1})}$$

  where $\mu$ is some positive monotone increasing function

- This is the **weighted average** forecaster
Convex loss functions

- Assume loss function $\ell(x, y)$ is convex in its first argument
- Then, the prediction $\hat{p}_t$ satisfies

$$
\ell(\hat{p}_t, y_t) = \ell\left(\frac{\sum_{i=1}^{N} \mu(R_{i,t-1}) f_{i,t}}{\sum_{j=1}^{N} \mu(R_{j,t-1})}, y_t\right) \leq \frac{\sum_{i=1}^{N} \mu(R_{i,t-1}) \ell(f_{i,t}, y_t)}{\sum_{j=1}^{N} \mu(R_{j,t-1})}
$$

- Using $r_{i,t} = \ell(\hat{p}_t, y_t) - \ell(f_{i,t}, y_t)$ and rearranging we obtain that, irrespective to $y_t$,

$$
\sum_{i=1}^{N} r_{i,t} \mu(R_{i,t-1}) \leq 0
$$
Choose $\mu = \phi'$

where $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is s.t. $\phi, \phi' \geq 0$ and $\phi''$ exists

Weighted average forecaster is then

$$\hat{p}_t = \frac{\sum_{i=1}^{N} \phi'(R_{i,t-1})f_{i,t}}{\sum_{j=1}^{N} \phi'(R_{j,t-1})}$$

Definition

Potential function $\Phi : \mathbb{R}^N \rightarrow \mathbb{R}$

$$\Phi(R) = \psi \left( \sum_{i=1}^{N} \phi(R_i) \right)$$

where $\psi : \mathbb{R} \rightarrow \mathbb{R}$ is such that $\psi \geq 0, \psi' > 0, \psi'' \leq 0$
Then the prediction at time $t$ becomes

$$ \hat{p}_t = \frac{\sum_{i=1}^{N} \nabla \Phi(R_{i,t-1})_i f_{i,t}}{\sum_{j=1}^{N} \nabla \Phi(R_{j,t-1})_j} $$

And the condition

$$ \sum_{i=1}^{N} r_{i,t} \mu(R_{i,t-1}) \leq 0 $$

gets rewritten as

$$ \nabla \Phi(R_{t-1})^\top r_t \leq 0 \quad (\text{Blackwell condition}) $$
Gradient descent interpretation

\[ \nabla \Phi(R) \]

Expert 1
R+r

R
Since $\phi$ is monotone increasing

$$
\psi \left( \phi \left( \max_{i=1,\ldots,N} R_{i,n} \right) \right) = \psi \left( \max_{i=1,\ldots,N} \phi(R_{i,n}) \right)
\leq \psi \left( \sum_{i=1}^{N} \phi(R_{i,n}) \right) = \Phi(R_n)
$$

Assuming $\psi$ is also invertible,

$$
\max_{i=1,\ldots,N} R_{i,n} \leq \phi^{-1}\psi^{-1}\left( \Phi(R_n) \right)
$$

So, a bound on $\Phi(R_n)$ implies a bound on $\max_{i=1,\ldots,N} R_{i,n}$
Proof of the potential bound

We bound the increment $\Phi(\mathbf{R}_t) - \Phi(\mathbf{R}_{t-1})$ of the potential by taking a linear approximation of $\Phi(\mathbf{R}_t)$ around $\Phi(\mathbf{R}_{t-1})$

$$\Phi(\mathbf{R}_t) = \Phi(\mathbf{R}_{t-1} + \mathbf{r}_t)$$

$$= \Phi(\mathbf{R}_{t-1}) + \nabla \Phi(\mathbf{R}_{t-1})^\top \mathbf{r}_t$$

$$+ \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial^2 \Phi(\mathbf{u})}{\partial u_i \partial u_j} \bigg|_{\mathbf{u} = \xi} \mathbf{r}_{i,t} \mathbf{r}_{j,t} \quad \text{(for some } \xi \in \mathbb{R}^N)$$

$$\leq \Phi(\mathbf{R}_{t-1}) + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial^2 \Phi(\mathbf{u})}{\partial u_i \partial u_j} \bigg|_{\mathbf{u} = \xi} \mathbf{r}_{i,t} \mathbf{r}_{j,t}$$
As $\Phi(u) = \psi \left( \sum_{k=1}^{N} \phi(u_k) \right) = \psi(\Sigma(u))$, we have that

$$
\frac{\partial \Phi(u)}{\partial u_i} = \psi'(\Sigma(u)) \phi'(u_i)
$$

For $i \neq j$,

$$
\frac{\partial^2 \Phi(u)}{\partial u_i \partial u_j} = \psi''(\Sigma(u)) \phi'(u_i) \phi'(u_j)
$$

For $i = j$,

$$
\frac{\partial^2 \Phi(u)}{\partial u_i^2} = \psi''(\Sigma(u)) \phi'(u_i)^2 + \psi'(\Sigma(u)) \phi''(u_i)
$$
Hence,

\[
\sum_{i=1}^{N} \sum_{j=1}^{N} \left. \frac{\partial^2 \Phi(u)}{\partial u_i \partial u_j} \right|_{u=\xi} r_{i,t} r_{j,t} \\
= \psi''(\Sigma(\xi)) \sum_{i=1}^{N} \sum_{j=1}^{N} \phi'(\xi_i) \phi'(\xi_j) r_{i,t} r_{j,t} \\
+ \psi'(\Sigma(\xi)) \sum_{i=1}^{N} \phi''(\xi_i) r_{i,t}^2 \\
= \psi''(\Sigma(\xi)) \left( \sum_{i=1}^{N} \phi'(\xi_i) r_{i,t} \right)^2 + \psi'(\Sigma(\xi)) \sum_{i=1}^{N} \phi''(\xi_i) r_{i,t}^2 \\
\leq \psi'(\Sigma(\xi)) \sum_{i=1}^{N} \phi''(\xi_i) r_{i,t}^2
\]
Proof (cont.)

- We have proven that

\[
\Phi(R_t) \leq \Phi(R_{t-1}) + \frac{1}{2} \max_u \psi' \left( \sum_{k=1}^{N} \phi(u_k) \right) \sum_{i=1}^{N} \phi''(u_i) r_{i,t}^2
\]

- \(C(r_t)\) bounds Taylor error

- By iterating, we get \(\Phi(R_n) \leq \Phi(0) + \frac{1}{2} \sum_{t=1}^{n} C(r_t)\)

- This holds for any forecaster satisfying \(\nabla \Phi(R_{t-1})^\top r_t \leq 0\) (e.g., weighted average forecaster for convex losses)
Assume: Loss $\ell$ is convex and takes values in $[0, 1]$

- Potential function

$$\Phi_p(R) = \left(\sum_{i=1}^{N} (R_i)_+^p\right)^{2/p} = \| (R)_+ \|_p^2 \quad \text{for } p \geq 2$$

- Prediction

$$\hat{p}_t = \frac{\sum_{i=1}^{N} \phi'(R_{i,t-1}) f_{i,t}}{\sum_{j=1}^{N} \phi'(R_{j,t-1})} = \frac{\sum_{i=1}^{N} (R_{i,t-1})_+^{p-1} f_{i,t}}{\sum_{j=1}^{N} (R_{j,t-1})_+^{p-1}}$$

- Taylor error bound: $C(r_t) = (p - 1) \| r_t \|_p^2 \leq (p - 1)N^2/p$

- Bound on regret: $\max_{i=1,...,N} R_{i,n} \leq \sqrt{n(p - 1)N^2/p}$
The polynomial forecaster is consistent with rate $O(1/ \sqrt{n})$

There is a trade-off for the choice of $p$ in the bound

$$\max_{i=1,\ldots,N} R_{i,N} \leq \sqrt{n(p - 1)N^2/p}$$

Choosing $p = 2 \ln N$ yields

$$\max_{i=1,\ldots,N} R_{i,N} \leq \sqrt{(2e)n \ln N}$$

Is this the best possible bound in term of $n$ and $N$?
Exponential potential

**Assume:** Loss $\ell$ is convex and takes values in $[0, 1]$

- **Potential function**

  $$\Phi_\eta(R) = \frac{1}{\eta} \ln \left( \sum_{i=1}^{N} e^{\eta R_i} \right) \quad \text{for } \eta > 0$$

- **Prediction:**

  $$\hat{p}_t = \frac{\sum_{i=1}^{N} e^{\eta (\hat{L}_{t-1} - L_{i,t-1})} f_{i,t}}{\sum_{j=1}^{N} e^{\eta (\hat{L}_{t-1} - L_{j,t-1})}} = \frac{\sum_{i=1}^{N} e^{-\eta L_{i,t-1}} f_{i,t}}{\sum_{j=1}^{N} e^{-\eta L_{j,t-1}}}$$

- **Taylor error bound:** $C(r_t) = \eta \max_{i=1,\ldots,N} r_{i,t}^2 \leq \eta$

- **Bound on regret:**

  $$\max_{i=1,\ldots,N} R_{i,n} \leq \frac{\ln N}{\eta} + \frac{\eta}{2} n$$
Choosing $\eta = \sqrt{2(\ln N)/n}$ gets

$$\max_{i=1,\ldots,N} R_{i,n} \leq \sqrt{2n \ln N}$$

Better constant than polynomial potential but not consistent (horizon-dependent tuning)

Disregarding consistency, is the constant $\sqrt{2}$ optimal?
Best lower bound is for the absolute loss \( \ell(p, y) = |p - y| \) (\( p \in [0, 1] \) and \( y \in \{0, 1\} \))

\[
\max_{y_1, \ldots, y_n} \frac{1}{n} \left( \max_{i=1, \ldots, N} R_{i,n} \right) = (1 - o(1)) \sqrt{\frac{n}{2 \ln N}}
\]

for any forecasting strategy \( o(1) \to 0 \) for \( N, n \to \infty \)

Refining the analysis of the exponential potential we get the optimal constant

\[
\max_{i=1, \ldots, N} R_{i,N} \leq \sqrt{\frac{n}{2 \ln N}}
\]

But tuning of \( \eta \) is still horizon-dependent
More sophisticated forecasters

- Using a **time-dependent** tuning $\eta_t = \sqrt{8(\ln N)/t}$ we get

  $$\max_{i=1,\ldots,N} R_{i,N} \leq \sqrt{2n \ln N} + \sqrt{\frac{\ln N}{8}}$$

- Consistent, but suboptimal leading constant

- **Loss-dependent** tuning $\eta_t = \sqrt{\frac{\ln N}{C \ln L_{t-1}^*}}$ gives

  $$\max_{i=1,\ldots,N} R_{i,N} \leq 2 \sqrt{2L_n^* \ln N} + O(\ln N)$$

  where

  $$L_t^* = \min_{i=1,\ldots,N} L_{i,t}$$
Summary

- Bound for forecasters satisfying Blackwell condition

\[ \Phi(R_n) \leq \Phi(0) + \frac{1}{2} \sum_{t=1}^{n} C(r_t) \]

- Polynomial potential with \( p = 2 \ln N \)

\[ \max_{i=1,...,N} R_{i,n} \leq \sqrt{(2e)n \ln N} \]

- Exponential potential with time-varying parameter

\[ \max_{i=1,...,N} R_{i,n} \leq \sqrt{2n \ln N} + \sqrt{\frac{\ln N}{8}} \]
The greedy forecaster for the exponential potential

- Predicts by minimizing the increase of regret

$$
\hat{p}_t = \arg\min_{\hat{p} \in X} \sup_{y_t \in Y} \Phi (R_{t-1} + r_t)
$$

$$
= \arg\min_{\hat{p} \in X} \sup_{y_t \in Y} \left( \ell(\hat{p}, y_t) + \frac{1}{\eta} \ln \sum_{i=1}^{N} e^{-\eta L_{i,t}} \right)
$$

- Not necessarily a weighted average forecaster

**Definition**

A loss is **mixable** if for some $\eta^*$ and for all $t = 1, 2, \ldots$ the greedy forecasters satisfies

$$
\ell(\hat{p}_t, y_t) \leq -\frac{1}{\eta} \ln \frac{\sum_{i=1}^{N} e^{-\eta^* L_{i,t}}}{\sum_{j=1}^{N} e^{-\eta^* L_{j,t-1}}}
$$
Properties of mixable losses

- If a loss is mixable for $\eta^*$, then $\Phi_{\eta^*}(R_n) \leq \Phi_{\eta^*}(0)$

- In fact, we have

$$\hat{L}_n \leq -\frac{1}{\eta^*} \ln \frac{\sum_{j=1}^{N} e^{-\eta^* L_{j,n}}}{N} \leq L_{i,n} + \frac{\ln N}{\eta^*} \quad \text{for any } i$$

- This immediately implies

$$\max_{i=1,...,N} R_{i,n} \leq \frac{1}{\eta^*} \ln N$$

- A constant regret bound!
Which losses are mixable?

- Assume $0 \leq \hat{p}, y \leq 1$
- Square loss $\ell(\hat{p}, y) = (\hat{p} - y)^2$ is mixable for $\eta^* = 2$ (greedy ≠ weighted average)
- Relative entropy loss $\ell(\hat{p}, y) = y \ln \frac{y}{\hat{p}} + (1 - y) \ln \frac{1 - y}{1 - \hat{p}}$ is mixable for $\eta^* = 1$ (greedy = weighted average)
- Absolute loss $\ell(\hat{p}, y) = |\hat{p} - y|$ is not mixable for any $\eta^* > 0$
Zero-sum games

$N \times M$ known loss matrix with entries $0 \leq \ell(i, y) \leq 1$

$$\begin{array}{ccc}
\ell(1,1) & \ell(1,2) & \ldots \\
\ell(2,1) & \ell(2,2) & \ldots \\
\vdots & \vdots & \ddots
\end{array}$$

Row player has $N$ actions
Column player has $M$ actions

- Players independently draw actions $I$ (with law $P$) and $Y$ (with law $Q$)
- Row player suffers loss $\ell(I, Y)$ (= gain of column player)
- Value of the game

$$\min_P \max_Q \sum_{i, y} \ell(i, y) P(i) Q(y) = \max_Q \min_P \sum_{i, y} \ell(i, y) P(i) Q(y)$$
Repeated zero-sum games

- Game is played repeatedly
- At round $t = 1, 2, \ldots$ players draw actions $(I_t, Y_t)$ which may depend on past realizations of $(I_s, Y_s)$, $s = 1, \ldots, t - 1$
- Regret after $n$ plays

$$ R_{i,n} = \sum_{t=1}^{n} \ell(I_t, Y_t) - \sum_{t=1}^{n} \ell(i, Y_t) $$

- Regret is small when empirical distribution of row player’s actions performs not much worse than any fixed action
Hannan consistency

Definition

A row player is **Hannan consistent** if

\[ \limsup_{n \to \infty} \max_{i=1, \ldots, N} \frac{R_{i,n}}{n} = 0 \quad \text{with probability 1} \]

irrespective to what column player does.

- H.C. implies \( \limsup_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} \ell(I_t, Y_t) \) is at most the game value.
- **Note:** Row player can beat the game value if column player is suboptimal.
Randomized forecasting

At each round $t$:
1. **forecaster** announces distribution $P_t$ over $\{1, \ldots, N\}$
2. **adversary** picks $y_t \in \mathcal{Y}$ and **forecaster** draws $I_t \sim P_t$
3. $I_t$ and $y_t$ are both revealed

- We may apply the exp. weighted average forecaster

\[
P_t(i) \propto \exp \left( -\eta_t \sum_{s=1}^{t-1} \ell(i, y_s) \right) \quad \eta_t = \sqrt{\frac{8 \ln N}{t}}
\]

- This achieves Hannan consistency

\[
\max_{i=1, \ldots, N} \frac{R_{i,n}}{n} \leq \left( \sqrt{2 \frac{\ln N}{n}} + o(1) \right) \quad \text{w.h.p.}
\]
What if forecaster queries only $m$ out of $n$ adversary’s actions?

For any fixed $n$, we get

$$
\max_{i=1,...,N} \frac{R_{i,n}}{n} \leq c \sqrt{\frac{\ln N}{m}} \quad \text{w.h.p.}
$$

Optimal to within constants

A query rate slightly faster than

$$
\frac{(\ln n)(\ln \ln n)}{n}
$$

is sufficient for Hannan consistency
What if forecaster observes signal $h(I_t, y_t)$ instead of $y_t$?

**Feedback matrix $H$**

**Example**

- Forecaster’s action $I_t \in \{1, 2, \ldots, N\}$ is the price at which a product sold online is offered to $t$-th customer.
- Adversary’s action $y_t \in \{1, 2, \ldots, N\}$ is maximum price at which $t$-th customer is willing to buy the product.
- Feedback matrix is:

$$h(I_t, y_t) = \begin{cases} 
\text{SOLD} & \text{if } I_t \leq y_t \\
\text{NOT SOLD} & \text{otherwise}
\end{cases}$$
Partial monitoring and Hannan consistency

- Assume $\mathcal{Y}$ is finite
- $L$ is the loss matrix, $H$ is any (legal) encoding of the feedback matrix
- Sufficient (and almost necessary) condition:
  \[ L = KH \quad \text{for some matrix } K \]

- Then per-round regret vanishes at rate $n^{-1/3}$
- Optimal, but worse than $n^{-1/2}$ of the full monitoring scenario
- Can still get $n^{-1/2}$ in special cases (e.g., nonstochastic bandits)
Binary pattern classification
Pattern classification model

- Data instances encoded as vectors $\mathbf{x}_t \in [0, 1]^d$
- A binary label $y_t \in \{-1, 1\}$ expresses some property of $\mathbf{x}_t$

**Example**

Given today’s closing prices $\mathbf{x}_t$, predict whether tomorrow market index will increase ($y_t = 1$)
Linear classifiers

Linear classification

\[ \hat{p}_t = \text{sgn}(\mathbf{w}_{t-1}^\top \mathbf{x}_t) \quad \mathbf{w}_{t-1} \in \mathbb{R}^d \]
If $\hat{p}_t \neq y_t$ then mistake at step $t$

**Goal**

On any arbitrary sequence $(x_1, y_1), (x_2, y_2), \ldots$ perform not much worse than the **best fixed linear classifier**

Experts = all linear classifiers
Consider the class $\mathcal{F}$ of all linear classifiers $\hat{p}_t = \text{sgn}(u^\top x_t)$ for $u \in \mathbb{R}^d$ with $\|u\|$ bounded.

A covering of $\mathcal{F}$ has size exponential in $d$.

Running the weighted average forecaster on the covering requires managing an exponential number of weights.
Allocate $d$ experts $F_1, \ldots, F_d$

On instance $x_t = (x_{t,1}, \ldots, x_{t,d})$ expert $F_j$ predicts $x_{t,j}$

**Regret**

$$r_t = y_t x_t \mathbb{I}_{\{\hat{p}_t \neq y_t\}}$$

We write $m_t$ to denote $\mathbb{I}_{\{\hat{p}_t \neq y_t\}}$
A reduction (cont.)

- Unnormalized weighted average forecaster for binary classification

\[ \mathbf{w}_{t-1} = \nabla \Phi(\mathbf{R}_{t-1}) \quad \hat{p}_t = \text{sgn}(\mathbf{w}_{t-1}^\top \mathbf{x}_t) \]

- To apply previous results we need **Blackwell condition** \( \mathbf{w}_{t-1}^\top \mathbf{r}_t \leq 0 \) to hold

  Indeed,

\[ \mathbf{w}_{t-1}^\top \mathbf{r}_t = y_t \mathbf{w}_{t-1}^\top \mathbf{x}_t \mathbb{I}_{\{\hat{p}_t \neq y_t\}} = \begin{cases} 
0 & \text{if } \mathbb{I}_{\{\hat{p}_t \neq y_t\}} = 0 \\
< 0 & \text{otherwise}
\end{cases} \]

since \( \mathbb{I}_{\{\hat{p}_t \neq y_t\}} = 1 \) iff \( \text{sgn}(\mathbf{w}_{t-1}^\top \mathbf{x}_t) \neq y_t \)
Polynomial potential

- Recall

\[ \Phi_p(R) = \left( \sum_{i=1}^{N} R_i^p \right)^{2/p} = \| R \|_p^2 \quad \text{for } p \geq 2 \]

Before we had \( \Phi_p(R) = \| (R)_+ \|_p^2 \)

Now \((\cdot)_+\) dropped for technical reasons

- Applying the main result

\[ \| R_n \|_p^2 \leq \frac{p - 1}{2} \sum_{t=1}^{n} \| r_t \|_p^2 \leq \frac{p - 1}{2} \left( \max_t \| x_t \|_p \right)^2 \sum_{t=1}^{n} m_t \]

- Next we lower bound \( \| R_n \|_p = \sqrt{\Phi_p(R_n)} \)
For any linear classifier of parameter $u$

$$\| R_n \|_p \geq R_n^\top \frac{u}{\|u\|_q} \quad \text{(by Hölder’s inequality)}$$

$$= R_{n-1}^\top \frac{u}{\|u\|_q} + m_n \frac{y_n u^\top x_n}{\|u\|_q}$$

$$\geq R_{n-1}^\top \frac{u}{\|u\|_q} + m_n \frac{1 - d_n(u)}{\|u\|_q}$$

where $d_n(u) = (1 - y_n u^\top x_n)_+$ is the hinge loss of $u$

Iterating we get

$$\| R_n \|_p \geq \sum_{t=1}^{n} m_t - d_t(u) \frac{\|u\|_q}{\|u\|_q}$$
A general mistake bound

Piecing together upper and lower bounds:

$$\sum_{t=1}^{n} \frac{m_t - d_t(u)}{\|u\|_q} \leq \|R_n\|_p \leq \sqrt{\frac{(p-1)\chi_p^2}{2}} \sum_{t=1}^{n} m_t$$

Solving for $M_n = \sum_{t=1}^{n} m_t$ and overapproximating

$$M_n - D_n(u) \leq \frac{p-1}{2} \left( \chi_p \|u\|_q \right)^2 + \left( \chi_p \|u\|_q \right) \sqrt{\frac{p-1}{2} D_n(u)}$$

where $D_n(u) = \sum_{t=1}^{n} d_t(u)$

This holds for all $u \in \mathbb{R}^d$ and all sequences $(x_1, y_1), (x_2, y_2), \ldots \in \mathbb{R}^d \times \{-1, +1\}$
The hinge loss upper bounds the mistake indicator function

$$d_t = \left(1 - y_t u^\top x_t\right)_+ \geq \text{sgn}(u^\top x_t)$$

This “regret” is a bit unfair

$$M_n - \inf_u D_n(u) \leq M_n - \inf_u M_n(u)$$
Formulation as an incremental algorithm

We want to express \( \mathbf{w}_t = \nabla \Phi (\mathbf{R}_t) \) recursively as \( \mathbf{w}_t = F(\mathbf{w}_{t-1}) \)

**Definition**

A potential \( \Phi : \mathbb{R}^d \rightarrow \mathbb{R} \) is **Legendre** if \( \Phi \) is strictly convex, differentiable, and has a convex domain (plus some additional technical requirements)

If a potential is Legendre, then \( \nabla \Phi \) is invertible

\[
\mathbf{w}_t = \nabla \Phi (\mathbf{R}_t) = \nabla \Phi (\mathbf{R}_{t-1} + \mathbf{r}_t) = \nabla \Phi \left( (\nabla \Phi)^{-1} (\mathbf{w}_{t-1}) + \mathbf{r}_t \right)
\]

**Incremental formulation**

\[
\mathbf{w}_t = \nabla \Phi \left( (\nabla \Phi)^{-1} (\mathbf{w}_{t-1}) + y_t \mathbf{x}_t \mathbb{I}_{\{\hat{p}_t \neq y_t\}} \right) \quad \text{update rule}
\]
\[ \mathbf{w}_t = \nabla \Phi \left( (\nabla \Phi)^{-1} (\mathbf{w}_{t-1}) + y_t x_t \mathbb{I}_{\{\hat{y}_t \neq y_t\}} \right) \]
Polynomial potential $\Phi_p(\cdot) = \|\cdot\|_p^2$ is Legendre

$$
\left( \nabla_{\frac{1}{2}} \| \mathbf{u} \|_p^2 \right)_i = \frac{\text{sgn}(u_i) |u_i|^{p-1}}{\| \mathbf{u} \|_p^{p-2}} \quad \left( \nabla_{\frac{1}{2}} \| \mathbf{u} \|_p^2 \right)^{-1} = \nabla_{\frac{1}{2}} \| \mathbf{u} \|_q^2
$$

where $q$ is such that $1/p + 1/q = 1$

- When $p = 2$ we have $\nabla \Phi_2(\mathbf{R}) = \mathbf{R}$
- The incremental formulation then is simply

$$
\mathbf{w}_t = \mathbf{w}_{t-1} + y_t \mathbf{x}_t \mathbb{I}_{\{\hat{p}_t \neq y_t\}}
$$

the **Perceptron algorithm** (Rosenblatt, 1952)
The Perceptron algorithm

\[ w_t = w_{t-1} + y_t x_t \mathbb{I}_{\{\hat{y}_t \neq y_t\}} \]
Mistake bounds

We can specialize our previous bound by setting $p = 2$

$$M_n - D_n(u) \leq (X_2 \|u\|_2)^2 + (X_2 \|u\|_2) \sqrt{D_n(u)}$$

Definition

A sequence $(x_1, y_1), \ldots, (x_n, y_n)$ is **linearly separable** with margin $\gamma > 0$ if there exists $u \in \mathbb{R}^d$ with $\|u\|_2 = 1$ such that

$$\min_{t=1,\ldots,n} y_t \ u^\top x_t \geq \gamma$$

In this case we recover the **Perceptron convergence theorem**

$$M_n \leq \left( \frac{X_2}{\gamma} \right)^2$$
Linear separability
Recall

\[ \Phi_\eta(R) = \frac{1}{\eta} \ln \left( \sum_{i=1}^{d} e^{\eta R_i} \right) \]

The weights have form

\[ w_t = \nabla \Phi_\eta(R_t) = \frac{e^{\eta R_{i,n}}}{\sum_{k=1}^{d} e^{\eta R_{k,n}}} \]

Note that \( w_t \) belongs to the simplex in \( \mathbb{R}^d \)
Incremental formulation

- Problem: $\Phi_\eta$ is not strictly convex because $\nabla \Phi_\eta(R)$ is constant along the line $R_1 = R_2 = \cdots = R_d$
- Thus, $\Phi$ is not Legendre and $(\nabla \Phi)^{-1}$ is not defined
- However, the potential $\Phi_{\exp}(R) = e^{R_1} + \cdots + e^{R_d}$ is Legendre with
  \[ \nabla \Phi_{\exp}(R) = (e^{R_1}, \ldots, e^{R_d}) \]
  \[ (\nabla \Phi_{\exp})^{-1}(\mathbf{w}') = (\ln w'_1, \ldots, \ln w'_d) \]

- We get the following update rule
  \[ w'_{i,t} = \left[ \nabla \Phi_{\exp} \left( (\nabla \Phi_{\exp})^{-1}(\mathbf{w}'_{t-1}) + \eta r_{t-1} \right) \right]_i \]
  \[ = \exp(\ln w'_{i,t-1} + \eta r_{i,t-1}) \]
  \[ = w'_{i,t-1} e^{\eta r_{i,t-1}} \]
Incremental formulation (cont.)

- Summarizing, we have

\[ w'_t = \nabla \Phi_{\exp} \left( (\nabla \Phi_{\exp})^{-1} (w'_{t-1}) + \eta r_{t-1} \right) \]

\[ w_{i,t} = \left[ \nabla \Phi_{\eta} (R_t) \right]_i = \frac{e^{\eta R_{i,t}}}{\sum_{k=1}^{d} e^{\eta R_{k,t}}} = \frac{w'_{i,t}}{\sum_{k=1}^{d} w'_{k,t}} \]

- This is the **Winnow algorithm** (Littlestone, 1988)
Applying the main result

\[ \ln \Phi_\eta(R_n) \leq \frac{\ln d}{\eta} + \frac{\eta}{2} \sum_{t=1}^{n} \left( \max_{i=1,...,d} r_{i,t}^2 \right) \]
\[ \leq \frac{\ln d}{\eta} + \frac{\eta}{2} (\chi_\infty)^2 M_n \]

As before, we lower bound \( \ln \Phi_\eta(R_n) \)
Recalling that \( \mathbf{w} \) belong to the simplex, we apply the log-sum inequality

\[
\ln \sum_{i=1}^{d} v_i \geq \sum_{i=1}^{d} u_i \ln v_i + H(u) \quad u_i, v_i \geq 0 \quad \sum_{i=1}^{d} u_i = 1
\]

for any linear classifier \( \mathbf{u} \) in the simplex

Applying this to the exponential potential we get

\[
\ln \Phi_{\eta}(\mathbf{R}_n) = \frac{1}{\eta} \ln \sum_{i=1}^{d} e^{\eta R_{i,n}} \geq \mathbf{R}_n^\top \mathbf{u} + H(\mathbf{u})
\]

Dropping \( H(\mathbf{u}) \) and proceeding as for the polynomial potential we get

\[
\ln \Phi_{\eta}(\mathbf{R}_n) \geq \sum_{t=1}^{n} (m_t - d_t(\mathbf{u})) = M_n - D_n(\mathbf{u})
\]
Piecing together upper and lower bound

\[ M_n - D_n(u) \leq \ln \Phi_\eta(R_n) \leq \frac{\ln d}{\eta} + \frac{\eta}{2}(X_\infty)^2 M_n \]

Solving for \( M_n \)

\[ M_n \leq \frac{1}{1 - \eta(X_\infty)^2/2} \left( D_n(u) + \frac{\ln d}{\eta} \right) \]

for any sequence \((x_1, y_1), (x_2, y_2), \ldots \in \mathbb{R}^d \times \{-1, +1\}\) and for any \( u \) in the simplex.

Same tuning problem as in prediction with expert advice

Proper choice of \( \eta = \eta(D_n, X_\infty, d) \) gives

\[ M_n - D_n(u) \leq X_\infty \sqrt{2D_n(u) \ln d} + (1 + o(1))(X_\infty)^2 \ln d \]

where \( o(1) \to 0 \) for \( D(u) \to \infty \)
Recall that, for the exponential potential, the difference $M_n - D_n(u)$ is bounded for all $u$ in the simplex.

If we consider $u$ in the scaled simplex \[ \{ u \in \mathbb{R}^d : u_i \geq 0, u_1 + \cdots + u_d = U \} \] we get

\[ M_n - D_n(u) \leq (X_\infty U) \sqrt{2D_n(u)} \ln d + (1+o(1)) (X_\infty U)^2 \ln d \]

To remove the constraint $u_i \geq 0$ we can transform the instances

\[ x = (x_1, \ldots, x_d) \mapsto (x_1, -x_1, \ldots, x_d, -x_d) \]

at the price of replacing $d$ by $2d$. 
Comparison between poly. and exp. potential

The two bounds on $M_n - D_n(u)$

$$\frac{p-1}{2} \left( X_p \left\| u \right\|_q \right)^2 + \left( X_p \left\| u \right\|_q \right) \sqrt{\frac{p-1}{2} D_n(u)}$$

$$(1 + o(1)) \ln(2d) \left( X_\infty \left\| u \right\|_1 \right)^2 + \left( X_\infty \left\| u \right\|_1 \right) \sqrt{2D_n(u) \ln(2d)}$$

- Bound for exp. pot. assumes tuning (previous knowledge of $X_\infty$ and choice of $\left\| u \right\|_1$)
- Both bounds depend on pairs of dual norms: $\left\| x \right\|_p \left\| u \right\|_q$ vs. $\left\| x \right\|_\infty \left\| u \right\|_1$
- For $p \approx 2 \ln d$ the bounds are essentially equal
Consider a sequence \((x_1, y_1), (x_2, y_2) \ldots\) such that \(x_t \in \{-1, 1\}^d\) and \(y_t = \text{sgn}(x_1, t)\)

Then \(u = (1, 0, \ldots, 0)\) is an optimal classifier (no loss)

Moreover,

\[
\left(\|u\|_2 X_2\right)^2 = d \quad \text{and} \quad \left(\|u\|_1 X_\infty\right)^2 = 1
\]

Then

\[
M_n \leq d \quad \text{(polynomial potential, } p = 2) \\
M_n \leq 4 \ln(2d) \quad \text{(exponential potential)}
\]

an exponential advantage (verified by experiments)

Opposite situation when instances \(x_t\) are sparse and best expert \(u\) is dense
On-line learning with kernels

- Feature map \( \phi : \mathbb{R}^d \rightarrow \text{RKHS} \)
- Kernel \( K(x, x') = \langle \phi(x), \phi(x') \rangle \)
- Assume a linear algorithm learns \( w \) such that

\[
w = \sum_i \alpha_i x_{ti}
\]

Then we can learn \( w = \sum_i \alpha_i \phi(x_{ti}) \) in the RKHS because

\[
\text{SGN}(\langle w, \phi(x) \rangle) = \text{SGN}\left( \sum_i y_{ti} \langle \phi(x_{ti}), \phi(x) \rangle \right) = \text{SGN}\left( \sum_i y_{ti} K(x_{ti}, x) \right)
\]
Checking applicability of kernels

Let $R_t = \sum_t y_t x_t \mathbb{I}_{\{\hat{y}_t \neq y_t\}}$

- **Winnow** $\omega_{i,t} = \frac{e^{\eta R_{i,t}}}{\sum_{k=1}^{d} e^{\eta R_{k,t}}}$

- **p-norm Perceptron** $\omega_{i,t} = \frac{\text{sgn}(R_{i,t}) |R_{i,t}|^{p-1}}{\|R_t\|_p^{p-2}}$

- **Perceptron** $\omega_t = R_t$

Perceptron’s potential is spherical $\rightarrow$ rotational invariance
Kernel Perceptron

Initialize $\mathcal{L} = \emptyset$

For $t = 1, 2, \ldots$

1. Read next example $(x_t, y_t)$

2. Compute prediction $\hat{p} = \text{sgn}\left(\sum_{k \in \mathcal{L}} y_k K(x_k, x_t)\right)$

3. If $\hat{p}_t \neq y_t$ then $\mathcal{L} \leftarrow \mathcal{L} \cup \{t\}$

Now mistake bounds are extended to the whole RKHS

$$M_n \leq D_n(f) + \left(\max_t K(x_t, x_t)\right) \|f\|^2 + \ldots$$

for any $f$ in the RKHS
- Linear classifiers $H(x) = \text{sgn}(w^\top x)$
- Examples $(x_t, y_t)$ are i.i.d. according to a fixed and unknown probability distribution on $\mathbb{R}^d \times \{-1, +1\}$
- $\text{risk}(H) = P(H(x) \neq y)$
- Learning algorithm

$$(x_1, y_1), \ldots, (x_n, y_n) \rightarrow \begin{array}{c} \text{A} \\ \end{array} \rightarrow \hat{H} : \mathbb{R}^d \rightarrow \{-1, +1\}$$

$\hat{H}$ is (random) hypothesis output by learner
Data-dependent VC theory

- $\mathcal{H}$ is set of classifiers from which $\hat{\mathcal{H}}$ is selected
- For all $H \in \mathcal{H}$

\[
\text{risk}(H) - \text{risk}_{\text{emp}}(H) \leq c_1 \sqrt{\text{risk}_{\text{emp}}(H) \frac{V_{\mathcal{H}} \ln n}{n}} + c_2 \frac{V_{\mathcal{H}} \ln n}{n} \quad \text{w.h.p.}
\]

- VC theory of generalization studies properties of $\mathcal{H}$
- We use on-line learning to study a small subclass of $\mathcal{H}$ generated by the interaction with the training data
The ensemble of hypotheses

- Run an incremental learner on the training set
- Everytime $H(x_t) \neq y_t$, $H$ is changed by the update rule
- This process generates an ensemble of classifiers

$H_0, H_1, \ldots, H_n$

Goals

1. Bound the average risk of the ensemble in terms of the size of the ensemble
2. Find an element of the ensemble whose risk is close to the ensemble average
Step 1: bound the average risk

The difference

\[ \text{risk}(H_{t-1}) - \mathbb{I}_{\{H_{t-1}(x_t) \neq y_t\}} \]

is a \textbf{martingale difference sequence} because

\[ \mathbb{E} \left[ \text{risk}(H_{t-1}) - \mathbb{I}_{\{H_{t-1}(x_t) \neq y_t\}} \mid (x_1, y_1), \ldots, (x_{t-1}, y_{t-1}) \right] = 0 \]

The associated martingale is

\[ \sum_{t=1}^{n} \left( \text{risk}(H_{t-1}) - \mathbb{I}_{\{H_{t-1}(x_t) \neq y_t\}} \right) \]

\[ \iff \quad \frac{1}{n} \sum_{t=1}^{n} \text{risk}(H_{t-1}) - \frac{1}{n} \sum_{t=1}^{n} \mathbb{I}_{\{H_{t-1}(x_t) \neq y_t\}} \]

\[ \begin{array}{ll}
\text{average risk} & \text{fraction of mistakes}
\end{array} \]
Bernstein’s bound

If $Z_1, Z_2, \ldots$ is a martingale difference sequence with increments bounded by 1 and

$$V_n = \sum_{t=1}^{n} \mathbb{E} \left[ Z_t^2 \mid Z_1, \ldots, Z_{t-1} \right]$$

then for all $S, K > 0$

$$\mathbb{P} \left( \sum_{t=1}^{n} Z_n \geq S, \ V_n \leq K \right) \leq \exp \left( -\frac{S^2}{2(S/3 + K)} \right)$$
Application of Bernstein’s bound

Since $0 \leq \mathbb{I}_{\{H(x) \neq y\}} \leq 1$,

$$\text{VAR} \left[ \mathbb{I}_{\{H_{t-1}(x_t), y_t\}} \left\| (x_1, y_1), \ldots, (x_{t-1}, y_{t-1}) \right\| \right] \leq \mathbb{E} \left[ \text{risk}(H_{t-1}) \right\| (x_1, y_1), \ldots, (x_{t-1}, y_{t-1}) \right]$$

Applying Bernstein’s gives

$$\frac{1}{n} \sum_{t=1}^{n} \text{risk}(H_{t-1}) \leq \frac{M_n}{n} + \frac{c}{n} \left( \ln M_n + \sqrt{M_n \ln M_n} \right) \quad \text{w.h.p.}$$

Recall

$$\frac{M_n}{n} = \frac{1}{n} \sum_{t=1}^{n} \mathbb{I}_{\{H_{t-1}(x_t) \neq Y_t\}} \quad \text{is the fraction of mistakes}$$
Step 2: pick a good classifier in the ensemble

- Start from the ensemble $H_0, H_1, \ldots, H_n$
- Do the following:
  1. test each $H_t$ on $(x_{t+1}, y_{t+1}), \ldots, (x_n, y_n)$
  2. pick $\hat{H} = H_{t^*}$ minimizing a penalized risk estimate

Guaranteed bound

$$\text{risk}(\hat{H}) \leq \frac{M_n}{n} + \frac{c}{n} \left( (\ln n)^2 + \sqrt{M_n \ln n} \right) \quad \text{w.h.p.}$$
Conclusions!

- We made a reduction from pattern classification to prediction with experts.
- Potential-based forecasters have on-line classifiers as natural counterparts.
- We get a different viewpoint on kernel-based learning.
- A simple large deviation inequality is enough to get data-dependent tail risk bounds.