Vovk's aggregating algorithm
Mixable and unmixable loss functions

Yoav Freund

January 13, 2025

Section 3.5 in "Prediction, Learning and Games"

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

What we are going to investigate

- Different loss functions and their regret bounds.

What we are going to investigate

- Different loss functions and their regret bounds.
- Vovk's Meta Algorithm and it's analysis

What we are going to investigate

- Different loss functions and their regret bounds.
- Vovk's Meta Algorithm and it's analysis
- Instead of associating a loss with an action, we have experts that make predictions, and use the following protocol

What we are going to investigate

- Different loss functions and their regret bounds.
- Vovk's Meta Algorithm and it's analysis
- Instead of associating a loss with an action, we have experts that make predictions, and use the following protocol
  - Each expert makes a prediction

What we are going to investigate

- Different loss functions and their regret bounds.
- Vovk's Meta Algorithm and it's analysis
- Instead of associating a loss with an action, we have experts that make predictions, and use the following protocol
  - Each expert makes a prediction
  - The algorithm makes a prediction

What we are going to investigate

- Different loss functions and their regret bounds.
- Vovk's Meta Algorithm and it's analysis
- Instead of associating a loss with an action, we have experts that make predictions, and use the following protocol
  - Each expert makes a prediction
  - The algorithm makes a prediction
  - nature chooses an outcome

What we are going to investigate

- Different loss functions and their regret bounds.
- Vovk's Meta Algorithm and it's analysis
- Instead of associating a loss with an action, we have experts that make predictions, and use the following protocol
  - Each expert makes a prediction
  - The algorithm makes a prediction
  - nature chooses an outcome
  - each expert, and the algorithm suffer a loss that is a function of the prediction and the outcome.

What we are going to investigate

- Different loss functions and their regret bounds.
- Vovk's Meta Algorithm and it's analysis
- Instead of associating a loss with an action, we have experts that make predictions, and use the following protocol
  - Each expert makes a prediction
  - The algorithm makes a prediction
  - nature chooses an outcome
  - each expert, and the algorithm suffer a loss that is a function of the prediction and the outcome.
- The goal is to minimize the regret.

Absolute loss

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$

Absolute loss

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$
- Loss: $\lambda(p, x) = |x - p|$

Absolute loss

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$
- Loss: $\lambda(p, x) = |x - p|$
- N experts, expert i at time t outputs $q_i^t \in [0, 1]$

Absolute loss

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$
- Loss: $\lambda(p, x) = |x - p|$
- N experts, expert i at time t outputs $q_i^t \in [0, 1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$

Absolute loss

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss: $\lambda(p,x) = |x - p|$
- N experts, expert i at time t outputs $q_i^t \in [0,1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
- Experts algorithm (Hedge):

Absolute loss

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$
- Loss: $\lambda(p, x) = |x - p|$
- N experts, expert i at time t outputs $q_i^t \in [0, 1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
- Experts algorithm (Hedge):
  - Assign weights: $w_i^t = \frac{1}{N} \exp(-\eta L_i^{t-1})$

Absolute loss

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss: $\lambda(p,x) = |x-p|$
- N experts, expert i at time t outputs $q_i^t \in [0,1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
- Experts algorithm (Hedge):
  - Assign weights: $w_i^t = \frac{1}{N}\exp(-\eta L_i^{t-1})$
  - Master prediction: $q_M^t = \frac{\sum_{i=1}^{N} q_i^t w_i^t}{\sum_{i=1}^{N} w_i^t}$

Absolute loss

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$
- Loss: $\lambda(p, x) = |x - p|$
- N experts, expert i at time t outputs $q_i^t \in [0, 1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
- Experts algorithm (Hedge):
  - Assign weights: $w_i^t = \frac{1}{N} \exp(-\eta L_i^{t-1})$
  - Master prediction: $q_M^t = \frac{\sum_{i=1}^{N} q_i^t w_i^t}{\sum_{i=1}^{N} w_i^t}$
- Regret Bound for known horizon.

Absolute loss

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss: $\lambda(p,x) = |x-p|$
- N experts, expert i at time t outputs $q_i^t \in [0,1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
- Experts algorithm (Hedge):
  - Assign weights: $w_i^t = \frac{1}{N}\exp(-\eta L_i^{t-1})$
  - Master prediction: $q_M^t = \frac{\sum_{i=1}^{N} q_i^t w_i^t}{\sum_{i=1}^{N} w_i^t}$
- Regret Bound for known horizon.
  - Set η according to T: $\eta \approx \sqrt{\frac{2\ln N}{T}}$

Absolute loss

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss: $\lambda(p,x) = |x-p|$
- N experts, expert i at time t outputs $q_i^t \in [0,1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
- Experts algorithm (Hedge):
  - Assign weights: $w_i^t = \frac{1}{N}\exp(-\eta L_i^{t-1})$
  - Master prediction: $q_M^t = \frac{\sum_{i=1}^{N} q_i^t w_i^t}{\sum_{i=1}^{N} w_i^t}$
- Regret Bound for known horizon.
  - Set η according to T: $\eta \approx \sqrt{\frac{2\ln N}{T}}$
  - Regret bound:

$$L_A \leq \min_i L_i + \sqrt{2T\ln N} + \ln N$$

Absolute loss

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss: $\lambda(p,x) = |x - p|$
- N experts, expert i at time t outputs $q_i^t \in [0,1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
- Experts algorithm (Hedge):
  - Assign weights: $w_i^t = \frac{1}{N} \exp(-\eta L_i^{t-1})$
  - Master prediction: $q_M^t = \frac{\sum_{i=1}^N q_i^t w_i^t}{\sum_{i=1}^N w_i^t}$
- Regret Bound for known horizon.
  - Set η according to T: $\eta \approx \sqrt{\frac{2 \ln N}{T}}$
  - Regret bound:

$$L_A \leq \min_i L_i + \sqrt{2T \ln N} + \ln N$$

  - Is there an advantage to the algorithm relative to DTOL?

Binary log-loss

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$

Binary log-loss

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$
- Loss: $\lambda(p, x) = -x \log p - (1 - x) \log(1 - p)$

Binary log-loss

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss: $\lambda(p,x) = -x \log p - (1-x)\log(1-p)$
- N experts, expert i at time t outputs $q_i^t \in [0,1]$

Binary log-loss

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$
- Loss: $\lambda(p, x) = -x \log p - (1 - x) \log(1 - p)$
- N experts, expert i at time t outputs $q_i^t \in [0, 1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$

Binary log-loss

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$
- Loss: $\lambda(p, x) = -x \log p - (1 - x) \log(1 - p)$
- N experts, expert i at time t outputs $q_i^t \in [0, 1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
- Experts algorithm:

Binary log-loss

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss: $\lambda(p,x) = -x \log p - (1-x)\log(1-p)$
- N experts, expert i at time t outputs $q_i^t \in [0,1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
- Experts algorithm:
  - Assign weights: $w_i^t = \frac{1}{N}\exp(-L_i^{t-1})$

Binary log-loss

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss: $\lambda(p,x) = -x\log p - (1-x)\log(1-p)$
- N experts, expert i at time t outputs $q_i^t \in [0,1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
- Experts algorithm:
  - Assign weights: $w_i^t = \frac{1}{N}\exp(-L_i^{t-1})$
  - Master prediction: $q_M^t = \frac{\sum_{i=1}^N q_i^t w_i^t}{\sum_{i=1}^N w_i^t}$

Binary log-loss

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss: $\lambda(p,x) = -x \log p - (1-x)\log(1-p)$
- N experts, expert i at time t outputs $q_i^t \in [0,1]$
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
- Experts algorithm:
  - Assign weights: $w_i^t = \frac{1}{N}\exp(-L_i^{t-1})$
  - Master prediction: $q_M^t = \frac{\sum_{i=1}^{N} q_i^t w_i^t}{\sum_{i=1}^{N} w_i^t}$
- Regret Bound:

$$L_A^T \leq \min_i L_i^T + \ln N$$

Exponential Weights algorithm for log loss = Bayes Algorithm

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$

Exponential Weights algorithm for log loss = Bayes Algorithm

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$
- Loss = $\lambda(p, x) = -x \log p - (1 - x) \log(1 - p)$

Exponential Weights algorithm for log loss = Bayes Algorithm

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$
- Loss = $\lambda(p, x) = -x \log p - (1 - x) \log(1 - p)$
- N distribution models, model i at time t assigns probability $q_i^t(x_i^t) \in [0, 1]$ to outcome x^s

Exponential Weights algorithm for log loss = Bayes Algorithm

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss = $\lambda(p,x) = -x \log p - (1-x)\log(1-p)$
- N distribution models, model i at time t assigns probability $q_i^t(x_i^t) \in [0,1]$ to outcome x^s
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
  Same as minus log likelihood of model i: $L_i^t = -\log \prod_{s=1}^{t-1} q_i^s(x^s)$

Exponential Weights algorithm for log loss = Bayes Algorithm

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss = $\lambda(p,x) = -x \log p - (1-x) \log(1-p)$
- N distribution models, model i at time t assigns probability $q_i^t(x_i^t) \in [0,1]$ to outcome x^s
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
  Same as minus log likelihood of model i: $L_i^t = -\log \prod_{s=1}^{t-1} q_i^s(x^s)$
- algorithm:

Exponential Weights algorithm for log loss = Bayes Algorithm

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$
- Loss = $\lambda(p, x) = -x \log p - (1 - x) \log(1 - p)$
- N distribution models, model i at time t assigns probability $q_i^t(x_i^t) \in [0, 1]$ to outcome x^s
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
  Same as minus log likelihood of model i: $L_i^t = -\log \prod_{s=1}^{t-1} q_i^s(x^s)$
- algorithm:
  - Assign weights to experts: $w_i^t = \frac{1}{N} \exp(-L_i^{t-1})$

Exponential Weights algorithm for log loss = Bayes Algorithm

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss = $\lambda(p,x) = -x \log p - (1-x)\log(1-p)$
- N distribution models, model i at time t assigns probability $q_i^t(x_i^t) \in [0,1]$ to outcome x^s
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
  Same as minus log likelihood of model i: $L_i^t = -\log \prod_{s=1}^{t-1} q_i^s(x^s)$
- algorithm:
  - Assign weights to experts: $w_i^t = \frac{1}{N}\exp(-L_i^{t-1})$
  - Assign posterior over models= $w_i^t = \frac{1}{N}\prod_{s=1}^{t-1} q_i^s(x_i^s)$

Exponential Weights algorithm for log loss = Bayes Algorithm

- Prediction: $p \in [0, 1]$ outcome $x \in \{0, 1\}$
- Loss = $\lambda(p, x) = -x \log p - (1 - x) \log(1 - p)$
- N distribution models, model i at time t assigns probability $q_i^t(x_i^t) \in [0, 1]$ to outcome x^s
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^t \lambda(q_i^s, x^s)$
  Same as minus log likelihood of model i: $L_i^t = -\log \prod_{s=1}^{t-1} q_i^s(x^s)$
- algorithm:
  - Assign weights to experts: $w_i^t = \frac{1}{N} \exp(-L_i^{t-1})$
  - Assign posterior over models= $w_i^t = \frac{1}{N} \prod_{s=1}^{t-1} q_i^s(x_i^s)$
  - prediction: $q_M^t = \frac{\sum_{i=1}^N q_i^t w_i^t}{\sum_{i=1}^N w_i^t}$

Exponential Weights algorithm for log loss = Bayes Algorithm

- Prediction: $p \in [0,1]$ outcome $x \in \{0,1\}$
- Loss = $\lambda(p,x) = -x \log p - (1-x)\log(1-p)$
- N distribution models, model i at time t assigns probability $q_i^t(x_i^t) \in [0,1]$ to outcome x^s
- Cumulative loss of expert i at time t: $L_i^t = \sum_{s=1}^{t} \lambda(q_i^s, x^s)$
  Same as minus log likelihood of model i: $L_i^t = -\log \prod_{s=1}^{t-1} q_i^s(x^s)$
- algorithm:
  - Assign weights to experts: $w_i^t = \frac{1}{N}\exp(-L_i^{t-1})$
  - Assign posterior over models= $w_i^t = \frac{1}{N}\prod_{s=1}^{t-1} q_i^s(x_i^s)$
  - prediction: $q_M^t = \frac{\sum_{i=1}^{N} q_i^t w_i^t}{\sum_{i=1}^{N} w_i^t}$
  - same as posterior average

Vovk's general prediction game

Γ - prediction space. Ω - outcome space.

Vovk's general prediction game

Γ - prediction space. Ω - outcome space.
On each trial $t = 1, 2, \ldots$

Vovk's general prediction game

Γ - prediction space. Ω - outcome space.
On each trial $t = 1, 2, \ldots$

1. Each expert $i \in \{1 \ldots N\}$ makes a prediction $\gamma_i^t \in \Gamma$

Vovk's general prediction game

Γ - prediction space. Ω - outcome space.
On each trial $t = 1, 2, \ldots$

1. Each expert $i \in \{1 \ldots N\}$ makes a prediction $\gamma_i^t \in \Gamma$
2. The learner, after observing $\langle \gamma_1^t \ldots \gamma_N^t \rangle$, makes its own prediction γ^t

Vovk's general prediction game

Γ - prediction space. Ω - outcome space.
On each trial $t = 1, 2, \ldots$

1. Each expert $i \in \{1 \ldots N\}$ makes a prediction $\gamma_i^t \in \Gamma$
2. The learner, after observing $\langle \gamma_1^t \ldots \gamma_N^t \rangle$, makes its own prediction γ^t
3. Nature chooses an outcome $\omega^t \in \Omega$

Vovk's general prediction game

Γ - prediction space. Ω - outcome space.
On each trial $t = 1, 2, \ldots$

1. Each expert $i \in \{1 \ldots N\}$ makes a prediction $\gamma_i^t \in \Gamma$
2. The learner, after observing $\langle \gamma_1^t \ldots \gamma_N^t \rangle$, makes its own prediction γ^t
3. Nature chooses an outcome $\omega^t \in \Omega$
4. Each expert incurs loss $\ell_t(i) = \lambda(\omega^t, \gamma_i^t)$
   The learner incurs loss $\ell_t(A) = \lambda(\omega^t, \gamma^t)$

Achievable loss bounds

- $L_A \doteq \sum_{t=1}^{T} \ell_t(A)$ - total loss of algorithm

Achievable loss bounds

- $L_A \doteq \sum_{t=1}^{T} \ell_t(A)$ - total loss of algorithm
- $L_i \doteq \sum_{t=1}^{T} \ell_t(i)$ - total loss of expert i

Achievable loss bounds

- $L_A \doteq \sum_{t=1}^{T} \ell_t(A)$ - total loss of algorithm
- $L_i \doteq \sum_{t=1}^{T} \ell_t(i)$ - total loss of expert i
- **Goal:** find an algorithm which guarantees that

$$(a, c) \in [0, \infty), \quad L_A \leq aL_{\min} + c \ln N$$

For any sequence of events.

Achievable loss bounds

- $L_A \doteq \sum_{t=1}^{T} \ell_t(A)$ - total loss of algorithm
- $L_i \doteq \sum_{t=1}^{T} \ell_t(i)$ - total loss of expert i
- **Goal:** find an algorithm which guarantees that

$$(a, c) \in [0, \infty), \quad L_A \leq aL_{\min} + c \ln N$$

  For any sequence of events.
- We say that the pair (a, c) is achievable.

The set of achievable bounds

- Fix loss function $\lambda : \Omega \times \Gamma \to [0, \infty)$

The set of achievable bounds

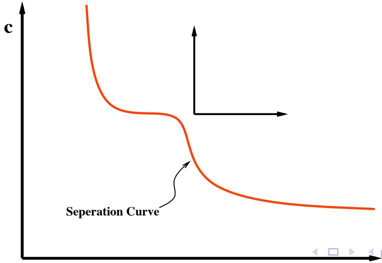
- Fix loss function $\lambda : \Omega \times \Gamma \to [0, \infty)$
- The pair (a, c) is *achievable* if there exists *some* prediction algorithm such that for *any* $N > 0$, *any* set of N prediction sequences and *any* sequence of outcomes

$$L_A \leq aL_{\min} + c \ln N$$

The set of achievable bounds

- Fix loss function $\lambda : \Omega \times \Gamma \to [0, \infty)$
- The pair (a, c) is *achievable* if there exists *some* prediction algorithm such that for *any* $N > 0$, *any* set of N prediction sequences and *any* sequence of outcomes

$$L_A \leq aL_{\min} + c \ln N$$



Analysis for specific loss functions

- Outcomes: $\omega^1, \omega_2, \ldots \omega^t \in [0, 1]$

Analysis for specific loss functions

- Outcomes: $\omega^1, \omega_2, \ldots \omega^t \in [0, 1]$
- Predictions: $\gamma^1, \gamma^2, \ldots \gamma^t \in [0, 1]$

Log loss (Entropy loss)

- $$\lambda_{\text{ent}}(\omega, \gamma) = \omega \ln \frac{\omega}{\gamma} + (1 - \omega) \ln \frac{1 - \omega}{1 - \gamma}$$

Log loss (Entropy loss)

- $$\lambda_{\text{ent}}(\omega, \gamma) = \omega \ln \frac{\omega}{\gamma} + (1-\omega) \ln \frac{1-\omega}{1-\gamma}$$
- When $q_t \in \{0, 1\}$ Cumulative log loss $=$ coding length ± 1

Log loss (Entropy loss)

- $$\lambda_{\text{ent}}(\omega,\gamma) = \omega \ln \frac{\omega}{\gamma} + (1-\omega) \ln \frac{1-\omega}{1-\gamma}$$
- When $q_t \in \{0,1\}$ Cumulative log loss $=$ coding length ± 1
- If $P[\omega_t = 1] = q$, optimal prediction $\gamma^t = q$

Log loss (Entropy loss)

- $$\lambda_{\text{ent}}(\omega, \gamma) = \omega \ln \frac{\omega}{\gamma} + (1 - \omega) \ln \frac{1 - \omega}{1 - \gamma}$$
- When $q_t \in \{0, 1\}$ Cumulative log loss $=$ coding length ± 1
- If $P[\omega_t = 1] = q$, optimal prediction $\gamma^t = q$
- Unbounded loss.

Log loss (Entropy loss)

- $$\lambda_{\text{ent}}(\omega, \gamma) = \omega \ln \frac{\omega}{\gamma} + (1 - \omega) \ln \frac{1 - \omega}{1 - \gamma}$$
- When $q_t \in \{0, 1\}$ Cumulative log loss $=$ coding length ± 1
- If $P[\omega_t = 1] = q$, optimal prediction $\gamma^t = q$
- Unbounded loss.
- Not symmetric $\exists p, q \;\; \lambda(p, q) \neq \lambda(q, p)$.

Log loss (Entropy loss)

- $$\lambda_{\text{ent}}(\omega,\gamma) = \omega \ln \frac{\omega}{\gamma} + (1-\omega) \ln \frac{1-\omega}{1-\gamma}$$
- When $q_t \in \{0,1\}$ Cumulative log loss $=$ coding length ± 1
- If $P[\omega_t = 1] = q$, optimal prediction $\gamma^t = q$
- Unbounded loss.
- Not symmetric $\exists p,q \;\; \lambda(p,q) \neq \lambda(q,p)$.
- No triangle inequality $\exists p_1,p_2,p_3 \;\; \lambda(p_1,p_3) > \lambda(p_1,p_2) + \lambda(p_2,p_3)$

Log loss (Entropy loss)

- $$\lambda_{\text{ent}}(\omega,\gamma) = \omega \ln \frac{\omega}{\gamma} + (1-\omega) \ln \frac{1-\omega}{1-\gamma}$$
- When $q_t \in \{0,1\}$ Cumulative log loss $=$ coding length ± 1
- If $P[\omega_t = 1] = q$, optimal prediction $\gamma^t = q$
- Unbounded loss.
- Not symmetric $\exists p, q \;\; \lambda(p,q) \neq \lambda(q,p)$.
- No triangle inequality $\exists p_1, p_2, p_3 \;\; \lambda(p_1,p_3) > \lambda(p_1,p_2) + \lambda(p_2,p_3)$
- $$L_A^T \leq \min_i L_i^T + \ln N$$

Square loss (Breier Loss)

- $$\lambda_{\mathrm{sq}}(\omega, \gamma) = (\omega - \gamma)^2$$

Square loss (Breier Loss)

- $$\lambda_{\text{sq}}(\omega,\gamma) = (\omega-\gamma)^2$$
- $P[\omega^t = 1] = q, \;\; P[\omega^t = 0] = 1 - q$,
  optimal prediction $\gamma^t = q$

Square loss (Breier Loss)

- $$\lambda_{\mathsf{sq}}(\omega,\gamma) = (\omega - \gamma)^2$$
- $P[\omega^t = 1] = q, \;\; P[\omega^t = 0] = 1 - q$,
  optimal prediction $\gamma^t = q$
- Bounded loss.

Square loss (Breier Loss)

- $$\lambda_{\mathsf{sq}}(\omega,\gamma) = (\omega-\gamma)^2$$
- $P[\omega^t = 1] = q, \;\; P[\omega^t = 0] = 1-q$,
  optimal prediction $\gamma^t = q$
- Bounded loss.
- Defines a metric (symmetric and triangle ineq.)

Square loss (Breier Loss)

- $$\lambda_{\mathsf{sq}}(\omega, \gamma) = (\omega - \gamma)^2$$
- $P[\omega^t = 1] = q, \quad P[\omega^t = 0] = 1 - q$,
  optimal prediction $\gamma^t = q$
- Bounded loss.
- Defines a metric (symmetric and triangle ineq.)
- Corresponds to regression.

Square loss (Breier Loss)

- $$\lambda_{\mathrm{sq}}(\omega, \gamma) = (\omega - \gamma)^2$$
- $P[\omega^t = 1] = q, \quad P[\omega^t = 0] = 1 - q$,
  optimal prediction $\gamma^t = q$
- Bounded loss.
- Defines a metric (symmetric and triangle ineq.)
- Corresponds to regression.
- $$L_A^T \leq \min_i L_i^T + \frac{1}{2} \ln N$$

Hellinger Loss

- $$\lambda_{\text{hel}}(\omega,\gamma) = \frac{1}{2}\left(\left(\sqrt{\omega}+\sqrt{\gamma}\right)^2 + \left(\sqrt{1-\omega}+\sqrt{1-\gamma}\right)^2\right)$$

Hellinger Loss

- $$\lambda_{\text{hel}}(\omega,\gamma) = \frac{1}{2}\left(\left(\sqrt{\omega}+\sqrt{\gamma}\right)^2 + \left(\sqrt{1-\omega}+\sqrt{1-\gamma}\right)^2\right)$$
- If $P[\omega^t = 1] = q, \quad P[\omega^t = 0] = 1-q$,
  optimal prediction $\gamma^t = q$

Hellinger Loss

- $$\lambda_{\text{hel}}(\omega,\gamma) = \frac{1}{2}\left(\left(\sqrt{\omega}+\sqrt{\gamma}\right)^2 + \left(\sqrt{1-\omega}+\sqrt{1-\gamma}\right)^2\right)$$
- If $P[\omega^t = 1] = q, \;\; P[\omega^t = 0] = 1 - q$, optimal prediction $\gamma^t = q$
- Loss is bounded.

Hellinger Loss

- $$\lambda_{\text{hel}}(\omega, \gamma) = \frac{1}{2}\left(\left(\sqrt{\omega} + \sqrt{\gamma}\right)^2 + \left(\sqrt{1-\omega} + \sqrt{1-\gamma}\right)^2\right)$$
- If $P[\omega^t = 1] = q, \quad P[\omega^t = 0] = 1 - q$, optimal prediction $\gamma^t = q$
- Loss is bounded.
- Defines a metric.

Hellinger Loss

- $$\lambda_{\text{hel}}(\omega, \gamma) = \frac{1}{2}\left(\left(\sqrt{\omega} + \sqrt{\gamma}\right)^2 + \left(\sqrt{1-\omega} + \sqrt{1-\gamma}\right)^2\right)$$
- If $P[\omega^t = 1] = q, \quad P[\omega^t = 0] = 1 - q$, optimal prediction $\gamma^t = q$
- Loss is bounded.
- Defines a metric.
- $\lambda_{\text{hel}}(p, q) \approx \lambda_{\text{ent}}(p, q)$ when $p \approx q$ and $p, q \in (0, 1)$

Hellinger Loss

- $$\lambda_{\text{hel}}(\omega, \gamma) = \frac{1}{2}\left(\left(\sqrt{\omega} + \sqrt{\gamma}\right)^2 + \left(\sqrt{1-\omega} + \sqrt{1-\gamma}\right)^2\right)$$
- If $P[\omega^t = 1] = q, \quad P[\omega^t = 0] = 1 - q$, optimal prediction $\gamma^t = q$
- Loss is bounded.
- Defines a metric.
- $\lambda_{\text{hel}}(p, q) \approx \lambda_{\text{ent}}(p, q)$ when $p \approx q$ and $p, q \in (0, 1)$
- $$L_A^T \leq \min_i L_i^T + \frac{1}{\sqrt{2}} \ln N$$

Absolute loss

- $$\lambda(\omega, \gamma) = |\omega - \gamma|$$

Absolute loss

- $$\lambda(\omega, \gamma) = |\omega - \gamma|$$
- Probability of making a mistake if predicting 0 or 1 using a biased coin

Absolute loss

- $$\lambda(\omega, \gamma) = |\omega - \gamma|$$
- Probability of making a mistake if predicting 0 or 1 using a biased coin
- If $P[\omega^t = 1] = q, \quad P[\omega^t = 0] = 1 - q$, then the optimal prediction is
$$\gamma^t = \begin{cases} 1 & \text{if } q > 1/2, \\ 0 & \text{otherwise} \end{cases}$$

Structureless bounded loss

- Prediction is a distribution $\gamma = \langle p_1, \ldots, p_N \rangle$, $p_i \geq 0$, $\sum_{i=1}^{N} p_i = 1$

Structureless bounded loss

- Prediction is a distribution $\gamma = \langle p_1, \ldots, p_N \rangle$, $p_i \geq 0$, $\sum_{i=1}^{N} p_i = 1$
- Outcome is a loss vector $\omega = \langle \omega_1, \ldots, \omega_N \rangle$, $0 \leq \omega_i \leq 1$

Structureless bounded loss

- Prediction is a distribution $\gamma = \langle p_1, \ldots, p_N \rangle$, $p_i \geq 0$, $\sum_{i=1}^{N} p_i = 1$
- Outcome is a loss vector $\omega = \langle \omega_1, \ldots, \omega_N \rangle$, $0 \leq \omega_i \leq 1$
- Loss is the dot product: $\lambda_{\text{dot}}(\omega, \gamma) = \gamma \cdot \omega$

Structureless bounded loss

- Prediction is a distribution $\gamma = \langle p_1, \ldots, p_N \rangle$, $p_i \geq 0$, $\sum_{i=1}^{N} p_i = 1$
- Outcome is a loss vector $\omega = \langle \omega_1, \ldots, \omega_N \rangle$, $0 \leq \omega_i \leq 1$
- Loss is the dot product: $\lambda_{\text{dot}}(\omega, \gamma) = \gamma \cdot \omega$
- Corresponds to the hedging game.

Structureless bounded loss

- Prediction is a distribution $\gamma = \langle p_1, \ldots, p_N \rangle$, $p_i \geq 0$, $\sum_{i=1}^{N} p_i = 1$
- Outcome is a loss vector $\omega = \langle \omega_1, \ldots, \omega_N \rangle$, $0 \leq \omega_i \leq 1$
- Loss is the dot product: $\lambda_{\text{dot}}(\omega, \gamma) = \gamma \cdot \omega$
- Corresponds to the hedging game.
- For hedge loss the regret is $\Omega(\sqrt{T \log N})$.

Structureless bounded loss

- Prediction is a distribution $\gamma = \langle p_1, \ldots, p_N \rangle$, $p_i \geq 0$, $\sum_{i=1}^{N} p_i = 1$
- Outcome is a loss vector $\omega = \langle \omega_1, \ldots, \omega_N \rangle$, $0 \leq \omega_i \leq 1$
- Loss is the dot product: $\lambda_{\text{dot}}(\omega, \gamma) = \gamma \cdot \omega$
- Corresponds to the hedging game.
- For hedge loss the regret is $\Omega(\sqrt{T \log N})$.
- For the log loss the regret is $O(\log N)$

Structureless bounded loss

- Prediction is a distribution $\gamma = \langle p_1, \ldots, p_N \rangle$, $p_i \geq 0$, $\sum_{i=1}^{N} p_i = 1$
- Outcome is a loss vector $\omega = \langle \omega_1, \ldots, \omega_N \rangle$, $0 \leq \omega_i \leq 1$
- Loss is the dot product: $\lambda_{\text{dot}}(\omega, \gamma) = \gamma \cdot \omega$
- Corresponds to the hedging game.
- For hedge loss the regret is $\Omega(\sqrt{T \log N})$.
- For the log loss the regret is $O(\log N)$
- **Which losses behave like entropy loss and which behave like hedge loss?**

Some technical requirements

- There should be a topology on the prediction set Γ such that

Some technical requirements

- There should be a topology on the prediction set Γ such that
- Γ is compact.

Some technical requirements

- There should be a topology on the prediction set Γ such that
- Γ is compact.
- $\forall \omega \in \Omega$, the function $\gamma \to \lambda(\omega, \gamma)$ is continuous

Some technical requirements

- There should be a topology on the prediction set Γ such that
- Γ is compact.
- $\forall \omega \in \Omega$, the function $\gamma \to \lambda(\omega, \gamma)$ is continuous
- There is a universally reasonable prediction
  $\exists \gamma \in \Gamma, \forall \omega \in \Omega, \lambda(\omega, \gamma) < \infty$

Some technical requirements

- There should be a topology on the prediction set Γ such that
- Γ is compact.
- $\forall \omega \in \Omega$, the function $\gamma \to \lambda(\omega, \gamma)$ is continuous
- There is a universally reasonable prediction
  $\exists \gamma \in \Gamma, \forall \omega \in \Omega, \lambda(\omega, \gamma) < \infty$
- There is no universally optimal prediction
  $\neg \exists \gamma \in \Gamma, \forall \omega \in \Omega, \lambda(\omega, \gamma) = 0$

Vovk's meta-algorithm

- Fix an achievable pair (a, c) and set $\eta = a/c$

Vovk's meta-algorithm

- Fix an achievable pair (a, c) and set $\eta = a/c$

Vovk's meta-algorithm

- Fix an achievable pair (a, c) and set $\eta = a/c$
- 1.

$$W_i^t = \frac{1}{N}\, e^{-\eta L_i^t}$$

Vovk's meta-algorithm

- Fix an achievable pair (a, c) and set $\eta = a/c$
- 1. $$W_i^t = \frac{1}{N}\, e^{-\eta L_i^t}$$
  2. Choose γ_t so that, for all $\omega^t \in \Omega$:
  $$\lambda(\omega^t, \gamma^t) - c \ln \sum_i W_i^t \leq -c \ln \left(\sum_i W_i^t e^{-\eta \lambda(\omega^t, \gamma_i^t)} \right)$$

Vovk's meta-algorithm

- Fix an achievable pair (a, c) and set $\eta = a/c$
- 1. $$W_i^t = \frac{1}{N}\, e^{-\eta L_i^t}$$
  2. Choose γ_t so that, for all $\omega^t \in \Omega$:
  $$\lambda(\omega^t, \gamma^t) - c \ln \sum_i W_i^t \le -c \ln \left(\sum_i W_i^t e^{-\eta \lambda(\omega^t, \gamma_i^t)} \right)$$
- If choice of γ_t always exists, then the total loss satisfies:
  $$\sum_t \lambda(\omega^t, \gamma^t) \le -c \ln \sum_i W_i^{T+1} \le aL_{\min} + c \ln N$$

Vovk's meta-algorithm

- Fix an achievable pair (a, c) and set $\eta = a/c$
- 1.
  $$W_i^t = \frac{1}{N} e^{-\eta L_i^t}$$
  2. Choose γ_t so that, for all $\omega^t \in \Omega$:
  $$\lambda(\omega^t, \gamma^t) - c \ln \sum_i W_i^t \leq -c \ln \left(\sum_i W_i^t e^{-\eta\lambda(\omega^t, \gamma_i^t)} \right)$$
- If choice of γ_t always exists, then the total loss satisfies:
  $$\sum_t \lambda(\omega^t, \gamma^t) \leq -c \ln \sum_i W_i^{T+1} \leq aL_{\min} + c \ln N$$
- Vovk's result: *yes!* a good choice for γ_t always exists!

Vovk's algorithm is the the highest achiever [Vovk95]

The pair (a, c) is achieved by some algorithm if and only if it is achieved by Vovk's algorithm.

Vovk's algorithm is the the highest achiever [Vovk95]

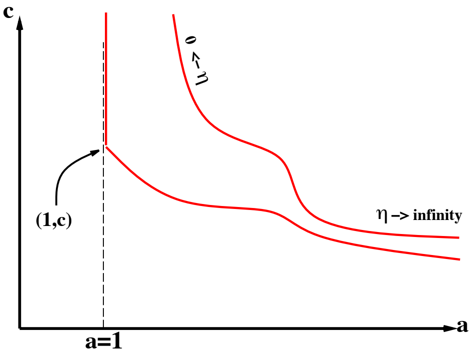
The pair (a, c) is achieved by some algorithm if and only if it is achieved by Vovk's algorithm.

The separation curve is $\left\{ \left(a(\eta), \frac{a(\eta)}{\eta}\right) \middle| \eta \in [0, \infty] \right\}$

Vovk's algorithm is the the highest achiever [Vovk95]

The pair (a, c) is achieved by some algorithm if and only if it is achieved by Vovk's algorithm.

The separation curve is $\left\{ \left(a(\eta), \frac{a(\eta)}{\eta} \right) \middle| \eta \in [0, \infty] \right\}$



Mixable Loss Functions

- A Loss function is mixable if a pair of the form $(1, c),\ c < \infty$ is achievable.

$$L_A \leq L_{\min} + c \ln N$$

Mixable Loss Functions

- A Loss function is mixable if a pair of the form $(1, c),\ c < \infty$ is achievable.

$$L_A \leq L_{\min} + c \ln N$$

- Vovk's algorithm with $\eta = 1/c$ achieves this bound.

Mixable Loss Functions

- A Loss function is mixable if a pair of the form $(1, c),\ c < \infty$ is achievable.

$$L_A \leq L_{\min} + c \ln N$$

- Vovk's algorithm with $\eta = 1/c$ achieves this bound.
- $\lambda_{\text{ent}}, \lambda_{\text{sq}}, \lambda_{\text{hel}}$ are mixable

Mixable Loss Functions

- A Loss function is mixable if a pair of the form $(1, c),\ c < \infty$ is achievable.

$$L_A \leq L_{\min} + c \ln N$$

- Vovk's algorithm with $\eta = 1/c$ achieves this bound.
- $\lambda_{\text{ent}}, \lambda_{\text{sq}}, \lambda_{\text{hel}}$ are mixable
- $\lambda_{\text{abs}}, \lambda_{\text{dot}}$ are not mixable

The convexity condition

- requirement for loss to be $(1, 1/\eta)$ mixable

The convexity condition

- requirement for loss to be $(1, 1/\eta)$ mixable
- $\forall \langle (\gamma_1, W_1), \ldots, (\gamma_N, W_N) \rangle$
  $\exists \gamma \in \Gamma$
  $\forall \omega \in \Omega$:

$$\lambda(\omega, \gamma) - \frac{1}{\eta} \ln \sum_i W_i \leq -\frac{1}{\eta} \ln \left(\sum_i W_i e^{-\eta \lambda(\omega, \gamma_i)} \right)$$

The convexity condition

- requirement for loss to be $(1, 1/\eta)$ mixable
- $\forall\langle(\gamma_1, W_1), \ldots, (\gamma_N, W_N)\rangle$
  $\exists\gamma \in \Gamma$
  $\forall\omega \in \Omega$:

$$\lambda(\omega, \gamma) - \frac{1}{\eta} \ln \sum_i W_i \leq -\frac{1}{\eta} \ln \left(\sum_i W_i e^{-\eta\lambda(\omega,\gamma_i)} \right)$$

- Can be re-written as:

$$e^{-\eta\lambda(\omega,\gamma)} \geq \sum_i \left(\frac{W_i}{\sum_j W_j} \right) e^{-\eta\lambda(\omega,\gamma_i)}$$

The convexity condition

- requirement for loss to be $(1, 1/\eta)$ mixable
- $\forall\langle(\gamma_1, W_1), \ldots, (\gamma_N, W_N)\rangle$
  $\exists\gamma \in \Gamma$
  $\forall\omega \in \Omega$:

$$\lambda(\omega, \gamma) - \frac{1}{\eta}\ln\sum_i W_i \leq -\frac{1}{\eta}\ln\left(\sum_i W_i e^{-\eta\lambda(\omega,\gamma_i)}\right)$$

- Can be re-written as:

$$e^{-\eta\lambda(\omega,\gamma)} \geq \sum_i \left(\frac{W_i}{\sum_j W_j}\right) e^{-\eta\lambda(\omega,\gamma_i)}$$

- Equivalently - the image of the set Γ under the mapping $F(\gamma) = \left\langle e^{-\eta\lambda(\omega,\gamma)}\right\rangle_{\omega\in\Omega}$ is concave.

convexity condition: Pictorially

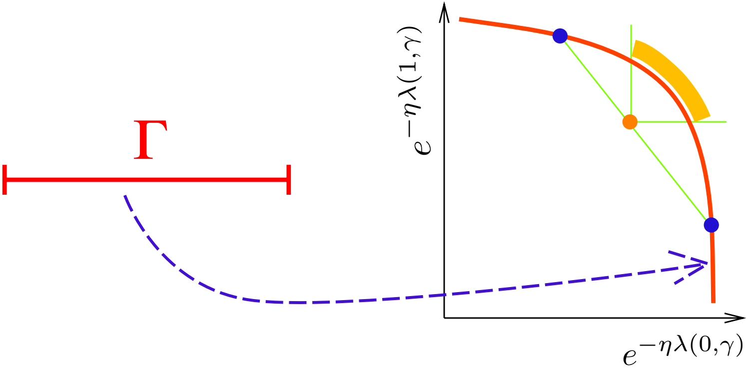
- **Example:** Suppose $\Omega = \{0, 1\}$, $\Gamma = [0, 1]$. then

$$F(\gamma) = \left\langle e^{-\eta\lambda(0,\gamma)}, e^{-\eta\lambda(1,\gamma)} \right\rangle$$

convexity condition: Pictorially

- **Example:** Suppose $\Omega = \{0, 1\}$, $\Gamma = [0, 1]$. then

$$F(\gamma) = \left\langle e^{-\eta\lambda(0,\gamma)}, e^{-\eta\lambda(1,\gamma)} \right\rangle$$



-

Vovk Algorithm for log loss

- The log loss is mixable with $\eta = 1$

Vovk Algorithm for log loss

- The log loss is mixable with $\eta = 1$
- The image of $[0, 1]$ through $F(\gamma) = \left\langle e^{-\eta\lambda(0,\gamma)}, e^{-\eta\lambda(1,\gamma)} \right\rangle$ is a straight line segment.

Vovk Algorithm for log loss

- The log loss is mixable with $\eta = 1$
- The image of $[0, 1]$ through $F(\gamma) = \left\langle e^{-\eta\lambda(0,\gamma)}, e^{-\eta\lambda(1,\gamma)} \right\rangle$ is a straight line segment.
- The only satisfactory prediction is

$$\gamma = \frac{\sum_i W_i \gamma_i}{\sum_i W_i}$$

Vovk Algorithm for log loss

- The log loss is mixable with $\eta = 1$
- The image of $[0, 1]$ through $F(\gamma) = \left\langle e^{-\eta\lambda(0,\gamma)}, e^{-\eta\lambda(1,\gamma)} \right\rangle$ is a straight line segment.
- The only satisfactory prediction is

$$\gamma = \frac{\sum_i W_i \gamma_i}{\sum_i W_i}$$

- We are back to the online Bayes algorithm.

Vovk algorithm for square loss

- The square loss is mixable with $\eta = 2$.

Vovk algorithm for square loss

- The square loss is mixable with $\eta = 2$.
- Prediction must satisfy

$$1 - \sqrt{-\frac{1}{2}\ln\sum_i V_i^t e^{-2(1-p_i^t)^2}} \le p^t \le \sqrt{-\frac{1}{2}\ln\sum_i V_i^t e^{-2(p_i^t)^2}}$$

where $V_i^t = \frac{W_i^t}{\sum_s W_i^s}$.

Vovk algorithm for square loss

- The square loss is mixable with $\eta = 2$.
- Prediction must satisfy

$$1 - \sqrt{-\frac{1}{2} \ln \sum_i V_i^t e^{-2(1-p_i^t)^2}} \leq p^t \leq \sqrt{-\frac{1}{2} \ln \sum_i V_i^t e^{-2(p_i^t)^2}}$$

where $V_i^t = \frac{W_i^t}{\sum_s W_i^s}$.

- $$L_A \leq L_{\min} + \frac{1}{2} \ln N$$

Simple prediction for square loss

- We can use the prediction

$$\gamma = \frac{\sum_i W_i \gamma_i}{\sum_i W_i}$$

Simple prediction for square loss

- We can use the prediction

$$\gamma = \frac{\sum_i W_i \gamma_i}{\sum_i W_i}$$

- But in that case we must use $\eta = 1/2$ when updating the weights.

Simple prediction for square loss

- We can use the prediction

$$\gamma = \frac{\sum_i W_i \gamma_i}{\sum_i W_i}$$

- But in that case we must use $\eta = 1/2$ when updating the weights.
- Which yields the bound

$$L_A \leq L_{\min} + 2 \ln N$$

Summary of bounds for mixable losses

Loss Functions:	c values: $(\eta = 1/c)$	
	$\mathbf{pred}_{\text{wmean}}(v, x)$	$\mathbf{pred}_{\text{Vovk}}(v,$
$L_{\text{sq}}(p, q)$	2	$1/2$
$L_{\text{ent}}(p, q)$	1	1
$L_{\text{hel}}(p, q)$	1	$1/\sqrt{2}$

Figure 2. $(c, 1/c)$-realizability: c values for loss and prediction