Learning Methods for Online Prediction Problems

Peter Bartlett
Statistics and EECS
UC Berkeley

Online Learning

- Repeated game:

$$\text{Decision method plays } a_t$$
$$\text{World reveals } \ell_t \in \mathcal{L}$$

- Aim: minimize $\hat{L}_n = \sum_{t=1}^{n} \ell_t(a_t)$.
- For example, aim to minimize regret, that is, perform well compared to the best (in retrospect) from some class:

$$\begin{aligned}\text{regret} &= \sum_{t=1}^{n} \ell_t(a_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{n} \ell_t(a) \\ &= \hat{L}_n - L_n^*.\end{aligned}$$

- Data can be adversarially chosen.

Online Learning

Minimax regret is the value of the game:

$$\min_{a_1} \max_{\ell_1} \cdots \min_{a_n} \max_{\ell_n} \left(\sum_{t=1}^{n} \ell_t(a_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{n} \ell_t(a) \right).$$

Online Learning: Motivations

1. Adversarial model is appropriate for
   - Computer security.
   - Computational finance.
2. Adversarial model assumes little:
   It is often straightforward to convert a strategy for an adversarial environment to a method for a probabilistic environment.
3. Studying the adversarial model sometimes reveals the *deterministic core* of a statistical problem: there are strong similarities between the performance guarantees in the two cases, and in particular between their dependence on the complexity of the class of prediction rules.
4. There are significant overlaps in the design of methods for the two problems:
   - *Regularization* plays a central role.
   - Many online prediction strategies have a natural interpretation as a *Bayesian method*.

Computer Security: Spam Detection

Computer Security: Spam Email Detection

- Here, the action a_t might be a classification rule, and ℓ_t is the indicator for a particular email being incorrectly classified (e.g., spam allowed through).
- The sender can determine if an email is delivered (or detected as spam), and try to modify it.
- An adversarial model allows an arbitrary sequence.
- We cannot hope for good classification accuracy in an absolute sense; regret is relative to a comparison class.
- Minimizing regret ensures that the spam detection accuracy is close to the best performance in retrospect on the particular spam sequence.

Computer Security: Spam Email Detection

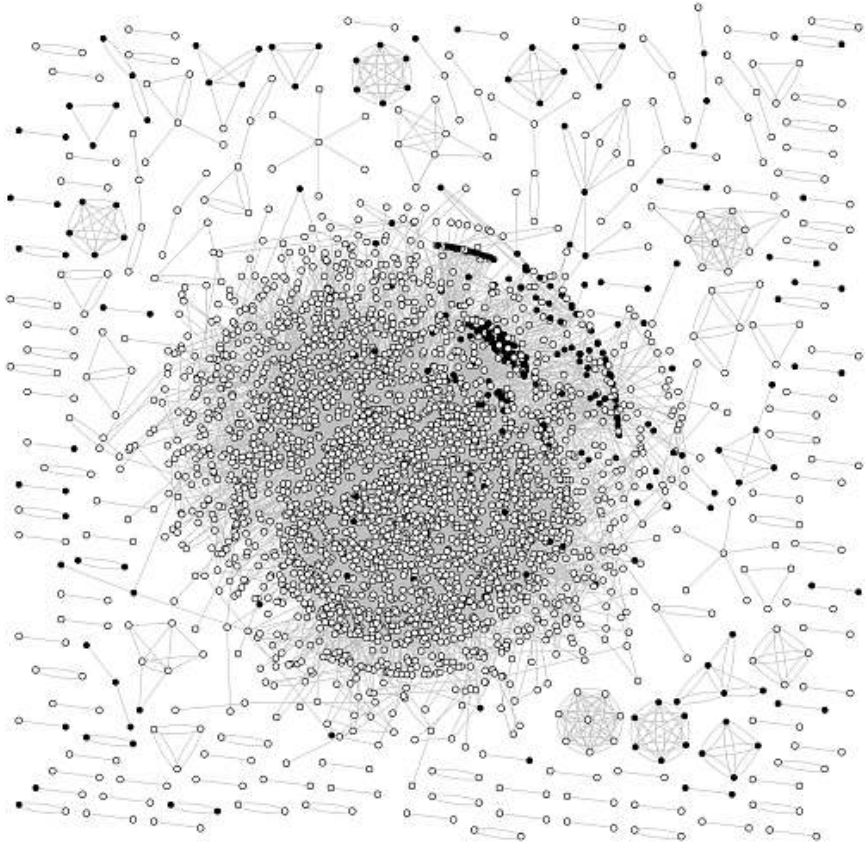
- Suppose we consider features of email messages from some set $\mathcal{X}$ (e.g., information about the header, about words in the message, about attachments).
- The decision method's action a_t is a mapping from $\mathcal{X}$ to $[0, 1]$ (think of the value as an estimated probability that the message is spam).
- At each round, the adversary chooses a feature vector $x_t \in \mathcal{X}$ and a label $y_t \in \{0, 1\}$, and the loss is defined as

$$\ell_t(a_t) = (y_t - a_t(x_t))^2 .$$

- The regret is then the excess squared error, over the best achievable on the data sequence:

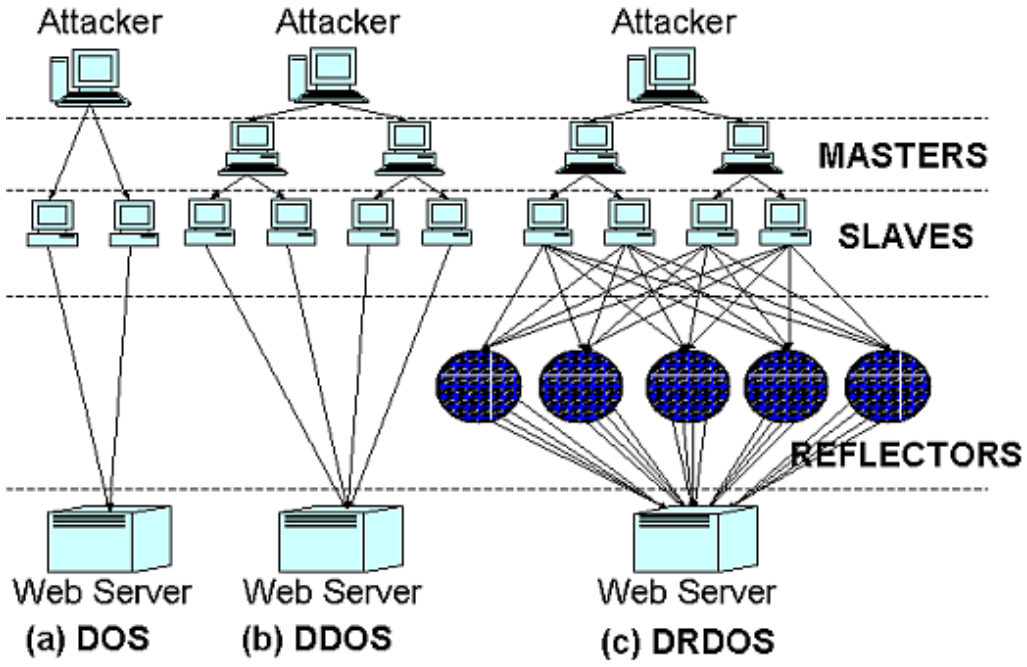
$$\sum_{t=1}^{n} \ell_t(a_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{n} \ell_t(a) = \sum_{t=1}^{n} (y_t - a_t(x_t))^2 - \min_{a \in \mathcal{A}} \sum_{t=1}^{n} (y_t - a(x_t))^2 .$$

Computer Security: Web Spam Detection



Web Spam Challenge (www.iw3c2.org)

Computer Security: Detecting Denial of Service



ACM

Computational Finance: Portfolio Optimization

Computational Finance: Portfolio Optimization

- Aim to choose a portfolio (distribution over financial instruments) to maximize utility.
- Other market players can profit from making our decisions bad ones. For example, if our trades have a market impact, someone can *front-run* (trade ahead of us).
- Here, the action a_t is a distribution on instruments, and ℓ_t might be the negative logarithm of the portfolio's increase, $a_t \cdot r_t$, where r_t is the vector of relative price increases.
- We might compare our performance to the best stock (distribution is a delta function), or a set of indices (distribution corresponds to Dow Jones Industrial Average, etc), or the set of all distributions.

Computational Finance: Portfolio Optimization

- The decision method's action a_t is a distribution on the m instruments, $a_t \in \Delta^m = \{a \in [0,1]^m : \sum_i a_i = 1\}$.
- At each round, the adversary chooses a vector of returns $r_t \in \mathbb{R}_+^m$; the ith component is the ratio of the price of instrument i at time t to its price at the previous time, and the loss is defined as

$$\ell_t(a_t) = -\log(a_t \cdot r_t).$$

- The regret is then the log of the ratio of the maximum value the portfolio would have at the end (for the best mixture choice) to the final portfolio value:

$$\sum_{t=1}^{n} \ell_t(a_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{n} \ell_t(a) = \max_{a \in \mathcal{A}} \sum_{t=1}^{n} \log(a \cdot r_t) - \sum_{t=1}^{n} \log(a_t \cdot r_t).$$

Online Learning: Motivations

2. Online algorithms are also effective in probabilistic settings.

- Easy to convert an online algorithm to a batch algorithm.
- Easy to show that good online performance implies good i.i.d. performance, for example.

Online Learning: Motivations

3. Understanding statistical prediction methods.

- Many statistical methods, based on *probabilistic assumptions*, can be effective in an adversarial setting.
- Analyzing their performance in adversarial settings provides perspective on their robustness.
- We would like violations of the probabilistic assumptions to have a limited impact.

Key Points

- Online Learning:
  - repeated game.
  - aim to minimize *regret*.
  - Data can be *adversarially* chosen.
- Motivations:
  - Often appropriate (security, finance).
  - Algorithms also effective in probabilistic settings.
  - Can provide insight into statistical prediction methods.

Course Synopsis

- A finite comparison class: $\mathcal{A} = \{1, \ldots, m\}$.
- Converting online to batch.
- Online convex optimization.
- Log loss.

Finite Comparison Class

1. "Prediction with expert advice."
2. With perfect predictions: $\log m$ regret.
3. Exponential weights strategy: $\sqrt{n \log m}$ regret.
4. Refinements and extensions:
   - Exponential weights and $L^* = 0$
   - n unknown
   - L^* unknown
   - Bayesian interpretation
   - *Convex* (versus linear) losses
5. Statistical prediction with a finite class.

Prediction with Expert Advice

Suppose we are predicting whether it will rain tomorrow. We have access to a set of m experts, who each make a forecast of 0 or 1. Can we ensure that we predict almost as well as the best expert?

Here, $\mathcal{A} = \{1, \ldots, m\}$. There are m experts, and each has a forecast sequence $f_1^i, f_2^i, \ldots$ from $\{0, 1\}$. At round t, the adversary chooses an outcome $y_t \in \{0, 1\}$, and sets

$$\ell_t(i) = 1[f_t^i \neq y_t] = \begin{cases} 1 & \text{if } f_t^i \neq y_t, \\ 0 & \text{otherwise.} \end{cases}$$

Online Learning

Minimax regret is the value of the game:

$$\min_{a_1} \max_{\ell_1} \cdots \min_{a_n} \max_{\ell_n} \left(\sum_{t=1}^{n} \ell_t(a_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{n} \ell_t(a) \right).$$

$$\hat{L}_n = \sum_{t=1}^{n} \ell_t(a_t), \qquad L_n^* = \min_{a \in \mathcal{A}} \sum_{t=1}^{n} \ell_t(a).$$

Prediction with Expert Advice

An easier game: suppose that the adversary is constrained to choose the sequence y_t so that some expert incurs no loss ($L_n^* = 0$), that is, there is an $i^* \in \{1, \ldots, m\}$ such that for all t, $y_t = f_t^{i^*}$.
How should we predict?

Prediction with Expert Advice: Halving

- Define the set of experts who have been correct so far:

$$C_t = \{i : \ell_1(i) = \cdots = \ell_{t-1}(i) = 0\}.$$

- Choose a_t any element of

$$\left\{i : f_t^i = \text{majority}\left(\{f_t^j : j \in C_t\}\right)\right\}.$$

Theorem

This strategy has regret no more than $\log_2 m$.

Prediction with Expert Advice: Halving

Theorem

The halving strategy has regret no more than $\log_2 m$.

Proof.

If it makes a mistake (that is, $\ell_t(a_t) = 1$), then the minority of $\{f_t^j : j \in C_t\}$ is correct, so at least half of the experts are eliminated:

$$|C_{t+1}| \leq \frac{|C_t|}{2}.$$

And otherwise $|C_{t+1}| \leq |C_t|$ (because $|C_t|$ never increases). Thus,

$$\begin{aligned}\hat{L}_n &= \sum_{t=1}^{n} \ell_t(a_t) \\ &\leq \log_2 \frac{|C_1|}{|C_{n+1}|} = \log_2 m - \log_2 |C_{n+1}| \leq \log_2 m.\end{aligned}$$

Prediction with Expert Advice

The proof follows a pattern we shall see again:
find some measure of progress (here, $|C_t|$) that

- changes monotonically when excess loss is incurred (here, it halves),
- is somehow constrained (here, it cannot fall below 1, because there is an expert who predicts perfectly).

What if there is no perfect expert?
Maintaining C_t makes no sense.

Finite Comparison Class

1. "Prediction with expert advice."
2. With perfect predictions: $\log m$ regret.
3. Exponential weights strategy: $\sqrt{n \log m}$ regret.
4. Refinements and extensions:
   - Exponential weights and $L^* = 0$
   - n unknown
   - L^* unknown
   - Bayesian interpretation
   - *Convex* (versus linear) losses
5. Statistical prediction with a finite class.

Prediction with Expert Advice: Mixed Strategies

- We have m experts.
- Allow a mixed strategy, that is, a_t chosen from the simplex Δ^m—the set of distributions on $\{1, \ldots, m\}$,

$$\Delta^m = \left\{ a \in [0,1]^m : \sum_{i=1}^{m} a^i = 1 \right\}.$$

- We can think of the strategy as choosing an element of $\{1, \ldots, m\}$ randomly, according to a distribution a_t. Or we can think of it as playing an element a_t of Δ^m, and incurring the expected loss,

$$\ell_t(a_t) = \sum_{i=1}^{m} a_t^i \ell_t(e_i),$$

where $\ell_t(e_i) \in [0,1]$ is the *loss* incurred by expert i. (e_i denotes the vector with a single 1 in the ith coordinate, and the rest zeros.)

Prediction with Expert Advice: Exponential Weights

- Maintain a set of (unnormalized) weights over experts:

$$w_0^i = 1,$$
$$w_{t+1}^i = w_t^i \exp\left(-\eta \ell_t(e_i)\right).$$

- Here, $\eta > 0$ is a parameter of the algorithm.
- Choose a_t as the normalized vector,

$$a_t = \frac{1}{\sum_{i=1}^m w_t^i} w_t.$$

Prediction with Expert Advice: Exponential Weights

Theorem

The exponential weights strategy with parameter

$$\eta = \sqrt{\frac{8 \ln m}{n}}$$

has regret satisfying

$$\hat{L}_n - L_n^* \leq \sqrt{\frac{n \ln m}{2}}.$$

Exponential Weights: Proof Idea

We use a measure of progress:

$$W_t = \sum_{i=1}^{m} w_t^i.$$

1. W_n grows at least as

$$\exp\left(-\eta \min_i \sum_{t=1}^{n} \ell_t(e_i)\right).$$

2. W_n grows no faster than

$$\exp\left(-\eta \sum_{t=1}^{n} \ell_t(a_t)\right).$$

Exponential Weights: Proof 1

$$
\begin{aligned}
\ln \frac{W_{n+1}}{W_1} &= \ln \left(\sum_{i=1}^{m} w_{n+1}^i \right) - \ln m \\
&= \ln \left(\sum_{i=1}^{m} \exp \left(-\eta \sum_t \ell_t(e_i) \right) \right) - \ln m \\
&\geq \ln \left(\max_i \exp \left(-\eta \sum_t \ell_t(e_i) \right) \right) - \ln m \\
&= -\eta \min_i \left(\sum_t \ell_t(e_i) \right) - \ln m \\
&= -\eta L_n^* - \ln m.
\end{aligned}
$$

Exponential Weights: Proof 2

$$\begin{aligned}
\ln \frac{W_{t+1}}{W_t} &= \ln \left(\frac{\sum_{i=1}^m \exp(-\eta \ell_t(e_i)) w_t^i}{\sum_i w_t^i} \right) \\
&\leq -\eta \frac{\sum_i \ell_t(e_i) w_t^i}{\sum_i w_t^i} + \frac{\eta^2}{8} \\
&= -\eta \ell_t(a_t) + \frac{\eta^2}{8},
\end{aligned}$$

where we have used Hoeffding's inequality:
for a random variable $X \in [a, b]$ and $\lambda \in \mathbb{R}$,

$$\ln \left(\mathbf{E} e^{\lambda X} \right) \leq \lambda \mathbf{E} X + \frac{\lambda^2 (b-a)^2}{8}.$$

Aside: Proof of Hoeffding's inequality

Define

$$A(\lambda) = \log\left(\mathbf{E}e^{\lambda X}\right)$$
$$= \log\left(\int e^{\lambda x}\, dP(x)\right),$$

where $X \sim P$. Then A is the log normalization of the exponential family random variable X_λ with reference measure P and sufficient statistic x. Since P has bounded support, $A(\lambda) < \infty$ for all λ, and we know that

$$A'(\lambda) = \mathbf{E}(X_\lambda),$$
$$A''(\lambda) = \mathrm{Var}(X_\lambda).$$

Since P has support in $[a, b]$, $\mathrm{Var}(X_\lambda) \leq (b-a)^2/4$. Then a Taylor expansion about $\lambda = 0$ (where X_λ has the same distribution as X) gives

$$A(\lambda) \leq \lambda \mathbf{E}X + \frac{\lambda^2}{8}(b-a)^2.$$

Exponential Weights: Proof

$$-\eta L_n^* - \ln m \leq \ln \frac{W_{n+1}}{W_1} \leq -\eta \hat{L}_n + \frac{n\eta^2}{8}.$$

Thus,

$$\hat{L}_n - L_n^* \leq \frac{\ln m}{\eta} + \frac{\eta n}{8}.$$

Choosing the optimal η gives the result:

Theorem

The exponential weights strategy with parameter $\eta = \sqrt{8 \ln m/n}$ *has regret no more than* $\sqrt{\frac{n \ln m}{2}}$.

Key Points

For a finite set of actions (experts):

- If one is perfect (zero loss), halving algorithm gives per round regret of
$$\frac{\ln m}{n}.$$
- Exponential weights gives per round regret of
$$O\left(\sqrt{\frac{\ln m}{n}}\right).$$

Prediction with Expert Advice: Refinements

1. Does exponential weights strategy give the faster rate if $L^* = 0$?
2. Do we need to know n to set η?

Prediction with Expert Advice: Refinements

1. Does exponential weights strategy give the faster rate if $L^* = 0$?
Replace Hoeffding:

$$\ln \mathbf{E} e^{\lambda X} \leq \lambda \mathbf{E} X + \frac{\lambda^2}{8},$$

with 'Bernstein':

$$\ln \mathbf{E} e^{\lambda X} \leq (e^{\lambda} - 1)\mathbf{E} X.$$

(for $X \in [0, 1]$).

Exponential Weights: Proof 2

$$\ln \frac{W_{t+1}}{W_t} = \ln \left(\frac{\sum_{i=1}^{m} \exp(-\eta \ell_t(e_i)) w_t^i}{\sum_i w_t^i} \right)$$
$$\leq \left(e^{-\eta} - 1\right) \ell_t(a_t).$$

Thus

$$\hat{L}_n \leq \frac{\eta}{1 - e^{-\eta}} L_n^* + \frac{\ln m}{1 - e^{-\eta}}.$$

For example, if $L_n^* = 0$ and η is large, we obtain a regret bound of roughly $\ln m/n$ again. And η large is like the halving algorithm (it puts roughly equal weight on all experts that have zero loss so far).

Prediction with Expert Advice: Refinements

2. Do we need to know n to set η?

   - We used the optimal setting $\eta = \sqrt{8 \ln m/n}$. But can this regret bound be achieved uniformly across time?
   - Yes; using a time-varying $\eta_t = \sqrt{8 \ln m/t}$ gives the same rate (worse constants).
   - It is also possible to set η as a function of L_t^*, the best cumulative loss so far, to give the improved bound for small losses uniformly across time (worse constants).

Prediction with Expert Advice: Refinements

3. We can interpret the exponential weights strategy as computing a Bayesian posterior.
Consider $f_t^i \in [0,1]$, $y_t \in \{0,1\}$, and $\ell_t^i = |f_t^i - y_t|$. Then consider a Bayesian prior that is uniform on m distributions. Given the ith distribution, y_t is a Bernoulli random variable with parameter

$$\frac{e^{-\eta(1-f_t^i)}}{e^{-\eta(1-f_t^i)} + e^{-\eta f_t^i}}.$$

Then exponential weights is computing the posterior distribution over the m distributions.

Prediction with Expert Advice: Refinements

4. We could work with arbitrary convex losses on Δ^m:
We defined loss as linear in a:

$$\ell_t(a) = \sum_i a^i \ell_t(e^i).$$

We could replace this with any bounded convex function on Δ^m. The only change in the proof is an equality becomes an inequality:

$$-\eta \frac{\sum_i \ell_t(e_i) w_t^i}{\sum_i w_t^i} \leq -\eta \ell_t(a_t).$$

Prediction with Expert Advice: Refinements

But note that the exponential weights strategy only competes with the *corners* of the simplex:

Theorem

For convex functions $\ell_t : \Delta^m \to [0,1]$*, the exponential weights strategy, with* $\eta = \sqrt{8 \ln m/n}$*, satisfies*

$$\sum_{t=1}^{n} \ell_t(a_t) \leq \min_i \sum_{t=1}^{n} \ell_t(e^i) + \sqrt{\frac{n \ln m}{2}}.$$

Finite Comparison Class

1. "Prediction with expert advice."
2. With perfect predictions: $\log m$ regret.
3. Exponential weights strategy: $\sqrt{n \log m}$ regret.
4. Refinements and extensions:
   - Exponential weights and $L^* = 0$
   - n unknown
   - L^* unknown
   - Bayesian interpretation
   - *Convex* (versus linear) losses
5. Statistical prediction with a finite class.

Probabilistic Prediction Setting

Let's consider a probabilistic formulation of a prediction problem.

- There is a sample of size n drawn i.i.d. from an unknown probability distribution P on $\mathcal{X} \times \mathcal{Y}$: $(X_1, Y_1), \ldots, (X_n, Y_n)$.
- Some method chooses $\hat{f} : \mathcal{X} \to \mathcal{Y}$.
- It suffers regret

$$\mathbf{E}\ell(\hat{f}(X), Y) - \min_{f \in F} \mathbf{E}\ell(f(X), Y).$$

- Here, F is a class of functions from $\mathcal{X}$ to $\mathcal{Y}$.

Probabilistic Setting: Zero Loss

Theorem

If some $f^* \in F$ *has* $\mathbf{E}\ell(f^*(X), Y) = 0$*, then choosing*

$$\hat{f} \in C_n = \left\{ f \in F : \hat{\mathbf{E}}\ell(f(X), Y) = 0 \right\}$$

leads to regret that is

$$O\left(\frac{\log |F|}{n}\right).$$

Probabilistic Setting: Zero Loss

Proof.

$$\begin{aligned}\Pr(\mathbf{E}\ell(\hat{f}) \geq \epsilon) &\leq \Pr(\exists f \in F : \hat{\mathbf{E}}\ell(f) = 0, \mathbf{E}\ell(\hat{f}) \geq \epsilon) \\ &\leq |F|(1-\epsilon)^n \\ &\leq |F|e^{-n\epsilon}.\end{aligned}$$

Integrating the tail bound $\Pr(\mathbf{E}\ell(\hat{f})n/\ln|F| \geq x) \geq 1 - e^{-x}$ gives $\mathbf{E}\ell(\hat{f}) \leq c\ln|F|/n$. □

Probabilistic Setting

Theorem

Choosing $\hat{f}$ to minimize the empirical risk, $\hat{\mathbf{E}}\ell(f(X), Y)$, leads to regret that is

$$O\left(\sqrt{\frac{\log |F|}{n}}\right).$$

Probabilistic Setting

Proof.

By the triangle inequality and the definition of $\hat{f}$,
$\mathbf{E}\ell_{\hat{f}} - \min_{f\in F} \mathbf{E}\ell_f \leq 2\mathbf{E}\sup_{f\in F} \left|\mathbf{E}\ell_f - \hat{\mathbf{E}}\ell_f\right|$.

$$
\begin{aligned}
\mathbf{E}\sup_{f\in F}\left|\mathbf{E}\ell_f - \hat{\mathbf{E}}\ell_f\right| &= \mathbf{E}\sup_{f\in F}\left|\mathbf{E}\hat{\mathbf{E}}'\ell_f - \hat{\mathbf{E}}\ell_f\right| \\
&\leq \mathbf{E}\sup_{f\in F}\left|\frac{1}{n}\sum_t \epsilon_t\left(\ell_f(X_t', Y_t') - \ell_f(X_t, Y_t)\right)\right| \\
&\leq 2\mathbf{E}\sup_{f\in F}\left|\frac{1}{n}\sum_t \epsilon_t \ell_f(X_t, Y_t)\right| \\
&\leq 2\max_{X_i, Y_i}\sqrt{\sum_t \ell(f(X_i, Y_i))^2}\,\frac{\sqrt{2\log|F|}}{n} \\
&\leq 2\sqrt{\frac{2\log|F|}{n}}.
\end{aligned}
$$

Probabilistic Setting: Key Points

For a finite function class

- If one is perfect (zero loss), choosing $\hat{f}$ to minimize the empirical risk, $\hat{\mathbf{E}}\ell(f(X), Y)$, gives per round regret of

$$\frac{\ln |F|}{n}.$$

- In any case, this $\hat{f}$ has per round regret of

$$O\left(\sqrt{\frac{\ln |F|}{n}}\right).$$

just as in the adversarial setting.

Course Synopsis

- A finite comparison class: $\mathcal{A} = \{1, \ldots, m\}$.
  1. "Prediction with expert advice."
  2. With perfect predictions: $\log m$ regret.
  3. Exponential weights strategy: $\sqrt{n \log m}$ regret.
  4. Refinements and extensions.
  5. Statistical prediction with a finite class.
- Converting online to batch.
- Online convex optimization.
- Log loss.

Online to Batch Conversion

- Suppose we have an online strategy that, given observations $\ell_1, \ldots, \ell_{t-1}$, produces $a_t = A(\ell_1, \ldots, \ell_{t-1})$.
- Can we convert this to a method that is suitable for a probabilistic setting? That is, if the ℓ_t are chosen i.i.d., can we use A's choices a_t to come up with a $\hat{a} \in \mathcal{A}$ so that

$$\mathbf{E}\ell_1(\hat{a}) - \min_{a \in \mathcal{A}} \mathbf{E}\ell_1(a)$$

  is small?
- Consider the following simple randomized method:
  1. Pick T uniformly from $\{0, \ldots, n\}$.
  2. Let $\hat{a} = A(\ell_{T+1}, \ldots, \ell_n)$.

Online to Batch Conversion

Theorem

If A has a regret bound of C_{n+1} for sequences of length $n+1$, then for any stationary process generating the $\ell_1, \ldots, \ell_{n+1}$, this method satisfies

$$\mathbf{E}\ell_{n+1}(\hat{a}) - \min_{a \in \mathcal{A}} \mathbf{E}\ell_n(a) \leq \frac{C_{n+1}}{n+1}.$$

(Notice that the expectation averages also over the randomness of the method.)

Online to Batch Conversion

Proof.

$$
\begin{aligned}
\mathbf{E}\ell_{n+1}(\hat{a}) &= \mathbf{E}\ell_{n+1}(A(\ell_{T+1},\ldots,\ell_n)) \\
&= \mathbf{E}\frac{1}{n+1}\sum_{t=0}^{n}\ell_{n+1}(A(\ell_{t+1},\ldots,\ell_n)) \\
&= \mathbf{E}\frac{1}{n+1}\sum_{t=0}^{n}\ell_{n-t+1}(A(\ell_1,\ldots,\ell_{n-t})) \\
&= \mathbf{E}\frac{1}{n+1}\sum_{t=1}^{n+1}\ell_t(A(\ell_1,\ldots,\ell_{t-1})) \\
&\leq \mathbf{E}\frac{1}{n+1}\left(\min_a\sum_{t=1}^{n+1}\ell_t(a)+C_{n+1}\right) \\
&\leq \min_a \mathbf{E}\ell_t(a)+\frac{C_{n+1}}{n+1}.
\end{aligned}
$$

Online to Batch Conversion

- The theorem is for the expectation over the randomness of the method.
- For a high probability result, we could
  1. Choose $\hat{a} = \frac{1}{n}\sum_{t=1}^{n} a_t$, provided $\mathcal{A}$ is convex and the ℓ_t are all convex.
  2. Choose

$$\hat{a} = \arg\min_{a_t} \left(\frac{1}{n-t} \sum_{s=t+1}^{n} \ell_s(a_t) + c\sqrt{\frac{\log(n/\delta)}{n-t}} \right).$$

In both cases, the analysis involves concentration of martingale sequences.
The second (more general) approach does not recover the C_n/n result: the penalty has the wrong form when $C_n = o(\sqrt{n})$.

Online to Batch Conversion

Key Point:

- An online strategy with regret bound C_n can be converted to a batch method.
  The regret per trial in the probabilistic setting is bounded by the regret per trial in the adversarial setting.

Course Synopsis

- A finite comparison class: $\mathcal{A} = \{1, \ldots, m\}$.
- Converting online to batch.
- Online convex optimization.
  1. Problem formulation
  2. Empirical minimization fails.
  3. Gradient algorithm.
  4. Regularized minimization
  5. Regret bounds
- Log loss.

Online Convex Optimization

1. Problem formulation
2. Empirical minimization fails.
3. Gradient algorithm.
4. Regularized minimization
   - Bregman divergence
   - Regularized minimization equivalent to minimizing latest loss and divergence from previous decision
   - Constrained minimization equivalent to unconstrained plus Bregman projection
   - Linearization
   - Mirror descent
5. Regret bounds
   - Unconstrained minimization
   - Seeing the future
   - Strong convexity
   - Examples (gradient, exponentiated gradient)
   - Extensions

Online Convex Optimization

- $\mathcal{A} =$ convex subset of $\mathbb{R}^d$.
- $\mathcal{L} =$ set of convex real functions on $\mathcal{A}$.

For example,

$$\ell_t(a) = (x_t \cdot a - y_t)^2.$$
$$\ell_t(a) = |x_t \cdot a - y_t|.$$

Online Convex Optimization: Example

Choosing a_t to minimize past losses,
$a_t = \arg\min_{a \in \mathcal{A}} \sum_{s=1}^{t-1} \ell_s(a)$, can fail.
('fictitious play,' 'follow the leader')

- Suppose $\mathcal{A} = [-1, 1]$, $\mathcal{L} = \{a \mapsto v \cdot a : |v| \leq 1\}$.
- Consider the following sequence of losses:

$$
\begin{aligned}
a_1 &= 0, & \ell_1(a) &= \frac{1}{2}a, \\
a_2 &= -1, & \ell_2(a) &= -a, \\
a_3 &= 1, & \ell_3(a) &= a, \\
a_4 &= -1, & \ell_4(a) &= -a, \\
a_5 &= 1, & \ell_5(a) &= a, \\
&\vdots & &\vdots
\end{aligned}
$$

- $a^* = 0$ shows $L_n^* \leq 0$, but $\hat{L}_n = n - 1$.

Online Convex Optimization: Example

- Choosing a_t to minimize past losses can fail.
- The strategy must avoid overfitting, just as in probabilistic settings.
- Similar approaches (regularization; Bayesian inference) are applicable in the online setting.
- First approach: gradient steps.
  Stay close to previous decisions, but move in a direction of improvement.

Online Convex Optimization: Gradient Method

$$
\begin{aligned}
a_1 &\in \mathcal{A}, \\
a_{t+1} &= \Pi_{\mathcal{A}} \left(a_t - \eta \nabla \ell_t(a_t) \right),
\end{aligned}
$$

where $\Pi_{\mathcal{A}}$ is the Euclidean projection on $\mathcal{A}$,

$$\Pi_{\mathcal{A}}(x) = \arg\min_{a \in \mathcal{A}} \|x - a\|.$$

Theorem

For $G = \max_t \|\nabla \ell_t(a_t)\|$ and $D = diam(\mathcal{A})$, the gradient strategy with $\eta = D/(G\sqrt{n})$ has regret satisfying

$$\hat{L}_n - L_n^* \leq GD\sqrt{n}.$$

Online Convex Optimization: Gradient Method

Theorem

For $G = \max_t \|\nabla \ell_t(a_t)\|$ *and* $D = diam(\mathcal{A})$*, the gradient strategy with* $\eta = D/(G\sqrt{n})$ *has regret satisfying*

$$\hat{L}_n - L_n^* \leq GD\sqrt{n}.$$

Example

$\mathcal{A} = \{a \in \mathbb{R}^d : \|a\| \leq 1\}$, $\mathcal{L} = \{a \mapsto v \cdot a : \|v\| \leq 1\}$.
$D = 2$, $G \leq 1$.
Regret is no more than $2\sqrt{n}$.
(And $O(\sqrt{n})$ is optimal.)

Online Convex Optimization: Gradient Method

Theorem

For $G = \max_t \|\nabla \ell_t(a_t)\|$ and $D = diam(\mathcal{A})$, the gradient strategy with $\eta = D/(G\sqrt{n})$ has regret satisfying

$$\hat{L}_n - L_n^* \leq GD\sqrt{n}.$$

Example

$\mathcal{A} = \Delta^m$, $\mathcal{L} = \{a \mapsto v \cdot a : \|v\|_\infty \leq 1\}$.
$D = 2$, $G \leq \sqrt{m}$.
Regret is no more than $2\sqrt{mn}$.

Since competing with the whole simplex is equivalent to competing with the vertices (experts) for linear losses, this is worse than exponential weights ($\sqrt{m}$ versus $\log m$).

Online Convex Optimization: Gradient Method

Proof.

$$\text{Define} \qquad \tilde{a}_{t+1} = a_t - \eta \nabla \ell_t(a_t),$$
$$a_{t+1} = \Pi_{\mathcal{A}}(\tilde{a}_{t+1}).$$

Fix $a \in \mathcal{A}$ and consider the measure of progress $\|a_t - a\|$.

$$\begin{aligned} \|a_{t+1} - a\|^2 &\leq \|\tilde{a}_{t+1} - a\|^2 \\ &= \|a_t - a\|^2 + \eta^2 \|\nabla \ell_t(a_t)\|^2 - 2\eta \nabla_t(a_t) \cdot (a_t - a). \end{aligned}$$

By convexity,

$$\begin{aligned} \sum_{t=1}^{n} (\ell_t(a_t) - \ell_t(a)) &\leq \sum_{t=1}^{n} \nabla \ell_t(a_t) \cdot (a_t - a) \\ &\leq \frac{\|a_1 - a\|^2 - \|a_{n+1} - a\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{n} \|\nabla \ell_t(a_t)\|^2 \end{aligned}$$

Online Convex Optimization

1. Problem formulation
2. Empirical minimization fails.
3. Gradient algorithm.
4. Regularized minimization
   - Bregman divergence
   - Regularized minimization equivalent to minimizing latest loss and divergence from previous decision
   - Constrained minimization equivalent to unconstrained plus Bregman projection
   - Linearization
   - Mirror descent
5. Regret bounds
   - Unconstrained minimization
   - Seeing the future
   - Strong convexity
   - Examples (gradient, exponentiated gradient)
   - Extensions

Online Convex Optimization: A Regularization Viewpoint

- Suppose ℓ_t is linear: $\ell_t(a) = g_t \cdot a$.
- Suppose $\mathcal{A} = \mathbb{R}^d$.
- Then minimizing the regularized criterion

$$a_{t+1} = \arg\min_{a \in \mathcal{A}} \left(\eta \sum_{s=1}^{t} \ell_s(a) + \frac{1}{2}\|a\|^2 \right)$$

corresponds to the gradient step

$$a_{t+1} = a_t - \eta \nabla \ell_t(a_t).$$

Online Convex Optimization: Regularization

Regularized minimization

Consider the family of strategies of the form:

$$a_{t+1} = \arg\min_{a \in \mathcal{A}} \left(\eta \sum_{s=1}^{t} \ell_s(a) + R(a) \right).$$

The regularizer $R : \mathbb{R}^d \to \mathbb{R}$ is strictly convex and differentiable.

Online Convex Optimization: Regularization

Regularized minimization

$$a_{t+1} = \arg\min_{a \in \mathcal{A}} \left(\eta \sum_{s=1}^{t} \ell_s(a) + R(a) \right).$$

- R keeps the sequence of a_ts stable: it diminishes ℓ_t's influence.
- We can view the choice of a_{t+1} as trading off two competing forces: making $\ell_t(a_{t+1})$ small, and keeping a_{t+1} close to a_t.
- This is a perspective that motivated many algorithms in the literature. We'll investigate why regularized minimization can be viewed this way.

Properties of Regularization Methods

In the unconstrained case ($\mathcal{A} = \mathbb{R}^d$), regularized minimization is equivalent to minimizing the latest loss and the distance to the previous decision. The appropriate notion of distance is the Bregman divergence $D_{\Phi_{t-1}}$:
Define

$$\Phi_0 = R,$$
$$\Phi_t = \Phi_{t-1} + \eta \ell_t,$$

so that

$$a_{t+1} = \arg\min_{a \in \mathcal{A}} \left(\eta \sum_{s=1}^{t} \ell_s(a) + R(a) \right)$$
$$= \arg\min_{a \in \mathcal{A}} \Phi_t(a).$$

Bregman Divergence

Definition

For a strictly convex, differentiable $\Phi : \mathbb{R}^d \to \mathbb{R}$, the Bregman divergence wrt Φ is defined, for $a, b \in \mathbb{R}^d$, as

$$D_\Phi(a, b) = \Phi(a) - (\Phi(b) + \nabla\Phi(b) \cdot (a - b)) .$$

$D_\Phi(a, b)$ is the difference between $\Phi(a)$ and the value at a of the linear approximation of Φ about b.

Bregman Divergence

$$D_\Phi(a, b) = \Phi(a) - (\Phi(b) + \nabla\Phi(b) \cdot (a - b)) .$$

Example

For $a \in \mathbb{R}^d$, the squared euclidean norm, $\Phi(a) = \frac{1}{2}\|a\|^2$, has

$$\begin{aligned} D_\Phi(a, b) &= \frac{1}{2}\|a\|^2 - \left(\frac{1}{2}\|b\|^2 + b \cdot (a - b)\right) \\ &= \frac{1}{2}\|a - b\|^2, \end{aligned}$$

the squared euclidean norm.

Bregman Divergence

$$D_\Phi(a, b) = \Phi(a) - (\Phi(b) + \nabla\Phi(b) \cdot (a - b)) .$$

Example

For $a \in [0, \infty)^d$, the unnormalized negative entropy, $\Phi(a) = \sum_{i=1}^d a_i (\ln a_i - 1)$, has

$$\begin{aligned} D_\Phi(a, b) &= \sum_i (a_i(\ln a_i - 1) - b_i(\ln b_i - 1) - \ln b_i(a_i - b_i)) \\ &= \sum_i \left(a_i \ln \frac{a_i}{b_i} + b_i - a_i \right), \end{aligned}$$

the unnormalized KL divergence.
Thus, for $a \in \Delta^d$, $\Phi(a) = \sum_i a_i \ln a_i$ has

$$D_\phi(a, b) = \sum_i a_i \ln \frac{a_i}{b_i}.$$

Bregman Divergence

When the range of Φ is $\mathcal{A} \subset \mathbb{R}^d$, in addition to differentiability and strict convexity, we make two more assumptions:

- The interior of $\mathcal{A}$ is convex,
- For a sequence approaching the boundary of $\mathcal{A}$, $\|\nabla\Phi(a_n)\| \to \infty$.

We say that such a Φ is a *Legendre function*.

Bregman Divergence

Properties:

1. $D_\Phi \geq 0$, $D_\Phi(a, a) = 0$.
2. $D_{A+B} = D_A + D_B$.
3. *Bregman projection*, $\Pi_{\mathcal{A}}^{\Phi}(b) = \arg\min_{a \in \mathcal{A}} D_\Phi(a, b)$ is uniquely defined for closed, convex $\mathcal{A}$.
4. *Generalized Pythagorus*: for closed, convex $\mathcal{A}$, $b^* = \Pi_{\mathcal{A}}^{\Phi}(b)$, and $a \in \mathcal{A}$,

$$D_\Phi(a, b) \geq D_\Phi(a, a^*) + D_\Phi(a^*, b).$$

5. $\nabla_a D_\Phi(a, b) = \nabla\Phi(a) - \nabla\Phi(b)$.
6. For ℓ linear, $D_{\Phi+\ell} = D_\Phi$.
7. For Φ^* the Legendre dual of Φ,

$$\nabla\Phi^* = (\nabla\Phi)^{-1},$$
$$D_\Phi(a, b) = D_{\Phi^*}(\nabla\phi(b), \nabla\phi(a)).$$

Legendre Dual

For a Legendre function $\Phi : \mathcal{A} \to \mathbb{R}$, the Legendre dual is

$$\Phi^*(u) = \sup_{v \in \mathcal{A}} (u \cdot v - \Phi(v)) .$$

- Φ^* is Legendre.
- $\mathrm{dom}(\Phi^*) = \nabla\Phi(\mathrm{int\ dom}\ \Phi)$.
- $\nabla\Phi^* = (\nabla\Phi)^{-1}$.
- $D_\Phi(a, b) = D_{\Phi^*}(\nabla\phi(b), \nabla\phi(a))$.
- $\Phi^{**} = \Phi$.

Legendre Dual

Example

For $\Phi = \frac{1}{2}\|\cdot\|_p^2$, the Legendre dual is $\Phi^* = \frac{1}{2}\|\cdot\|_q^2$, where $1/p + 1/q = 1$.

Example

For $\Phi(a) = \sum_{i=1}^{d} e^{a_i}$,

$$\nabla\Phi(a) = (e^{a_1}, \ldots, e^{a_d})',$$

so

$$(\nabla\Phi)^{-1}(u) = \nabla\Phi^*(u) = (\ln u_1, \ldots, \ln u_d)',$$

and $\Phi^*(u) = \sum_i u_i(\ln u_i - 1)$.

Online Convex Optimization

1. Problem formulation
2. Empirical minimization fails.
3. Gradient algorithm.
4. Regularized minimization
   - Bregman divergence
   - Regularized minimization equivalent to minimizing latest loss and divergence from previous decision
   - Constrained minimization equivalent to unconstrained plus Bregman projection
   - Linearization
   - Mirror descent
5. Regret bounds
   - Unconstrained minimization
   - Seeing the future
   - Strong convexity
   - Examples (gradient, exponentiated gradient)
   - Extensions

Properties of Regularization Methods

In the unconstrained case ($\mathcal{A} = \mathbb{R}^d$), regularized minimization is equivalent to minimizing the latest loss and the distance (Bregman divergence) to the previous decision.

Theorem

Define $\tilde{a}_1$ *via* $\nabla R(\tilde{a}_1) = 0$, *and set*

$$\tilde{a}_{t+1} = \arg\min_{a \in \mathbb{R}^d} \left(\eta \ell_t(a) + D_{\Phi_{t-1}}(a, \tilde{a}_t) \right).$$

Then

$$\tilde{a}_{t+1} = \arg\min_{a \in \mathbb{R}^d} \left(\eta \sum_{s=1}^{t} \ell_s(a) + R(a) \right).$$

Properties of Regularization Methods

Proof.

By the definition of Φ_t,

$$\eta \ell_t(a) + D_{\Phi_{t-1}}(a, \tilde{a}_t) = \Phi_t(a) - \Phi_{t-1}(a) + D_{\Phi_{t-1}}(a, \tilde{a}_t).$$

The derivative wrt a is

$$\begin{aligned} &\nabla \Phi_t(a) - \nabla \Phi_{t-1}(a) + \nabla_a D_{\Phi_{t-1}}(a, \tilde{a}_t) \\ &= \nabla \Phi_t(a) - \nabla \Phi_{t-1}(a) + \nabla \Phi_{t-1}(a) - \nabla \Phi_{t-1}(\tilde{a}_t) \end{aligned}$$

Setting to zero shows that

$$\nabla \Phi_t(\tilde{a}_{t+1}) = \nabla \Phi_{t-1}(\tilde{a}_t) = \cdots = \nabla \Phi_0(\tilde{a}_1) = \nabla R(\tilde{a}_1) = 0,$$

So $\tilde{a}_{t+1}$ minimizes Φ_t. □

Properties of Regularization Methods

Constrained minimization is equivalent to unconstrained minimization, followed by Bregman projection:

Theorem

For

$$a_{t+1} = \arg\min_{a \in \mathcal{A}} \Phi_t(a),$$
$$\tilde{a}_{t+1} = \arg\min_{a \in \mathbb{R}^d} \Phi_t(a),$$

we have

$$a_{t+1} = \Pi_{\mathcal{A}}^{\Phi_t}(\tilde{a}_{t+1}).$$

Properties of Regularization Methods

Proof.

Let a'_{t+1} denote $\Pi_{\mathcal{A}}^{\Phi_t}(\tilde{a}_{t+1})$. First, by definition of a_{t+1},

$$\Phi_t(a_{t+1}) \leq \Phi_t(a'_{t+1}).$$

Conversely,

$$D_{\Phi_t}(a'_{t+1}, \tilde{a}_{t+1}) \leq D_{\Phi_t}(a_{t+1}, \tilde{a}_{t+1}).$$

But $\nabla\Phi_t(\tilde{a}_{t+1}) = 0$, so

$$D_{\Phi_t}(a, \tilde{a}_{t+1}) = \Phi_t(a) - \Phi_t(\tilde{a}_{t+1}).$$

Thus, $\Phi_t(a'_{t+1}) \leq \Phi_t(a_{t+1})$. □

Properties of Regularization Methods

Example

For linear ℓ_t, regularized minimization is equivalent to minimizing the last loss plus the Bregman divergence wrt R to the previous decision:

$$\begin{aligned}
&\arg\min_{a\in\mathcal{A}} \left(\eta \sum_{s=1}^{t} \ell_s(a) + R(a) \right) \\
&= \Pi_{\mathcal{A}}^{R} \left(\arg\min_{a\in\mathbb{R}^d} \left(\eta \ell_t(a) + D_R(a, \tilde{a}_t) \right) \right),
\end{aligned}$$

because adding a linear function to Φ does not change D_Φ.

Online Convex Optimization

1. Problem formulation
2. Empirical minimization fails.
3. Gradient algorithm.
4. Regularized minimization
   - Bregman divergence
   - Regularized minimization equivalent and Bregman divergence from previous
   - Constrained minimization equivalent to unconstrained plus Bregman projection
   - Linearization
   - Mirror descent
5. Regret bounds
   - Unconstrained minimization
   - Seeing the future
   - Strong convexity
   - Examples (gradient, exponentiated gradient)
   - Extensions

Properties of Regularization Methods: Linear Loss

We can replace ℓ_t by $\nabla \ell_t(a_t)$, and this leads to an upper bound on regret.

Theorem

Any strategy for online linear optimization, with regret satisfying

$$\sum_{t=1}^{n} g_t \cdot a_t - \min_{a \in \mathcal{A}} \sum_{t=1}^{n} g_t \cdot a \leq C_n(g_1, \ldots, g_n)$$

can be used to construct a strategy for online convex optimization, with regret

$$\sum_{t=1}^{n} \ell_t(a_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{n} \ell_t(a) \leq C_n(\nabla \ell_1(a_1), \ldots, \nabla \ell_n(a_n)).$$

Proof.

Convexity implies $\ell_t(a_t) - \ell_t(a) \leq \nabla \ell_t(a_t) \cdot (a_t - a)$. □

Properties of Regularization Methods: Linear Loss

Key Point:

We can replace ℓ_t by $\nabla \ell_t(a_t)$, and this leads to an upper bound on regret.

Thus, we can work with linear ℓ_t.

Regularization Methods: Mirror Descent

Regularized minimization for linear losses can be viewed as mirror descent—taking a gradient step in a dual space:

Theorem

The decisions

$$\tilde{a}_{t+1} = \arg\min_{a \in \mathbb{R}^d} \left(\eta \sum_{s=1}^{t} g_s \cdot a + R(a) \right)$$

can be written

$$\tilde{a}_{t+1} = (\nabla R)^{-1} \left(\nabla R(\tilde{a}_t) - \eta g_t \right).$$

This corresponds to first mapping from $\tilde{a}_t$ through ∇R, then taking a step in the direction $-g_t$, then mapping back through $(\nabla R)^{-1} = \nabla R^*$ to $\tilde{a}_{t+1}$.

Regularization Methods: Mirror Descent

Proof.

For the unconstrained minimization, we have

$$\nabla R(\tilde{a}_{t+1}) = -\eta \sum_{s=1}^{t} g_s,$$

$$\nabla R(\tilde{a}_t) = -\eta \sum_{s=1}^{t-1} g_s,$$

so $\nabla R(\tilde{a}_{t+1}) = \nabla R(\tilde{a}_t) - \eta g_t$, which can be written

$$\tilde{a}_{t+1} = \nabla R^{-1} \left(\nabla R(\tilde{a}_t) - \eta g_t \right).$$

Online Convex Optimization

1. Problem formulation
2. Empirical minimization fails.
3. Gradient algorithm.
4. Regularized minimization and Bregman divergences
5. Regret bounds
   - Unconstrained minimization
   - Seeing the future
   - Strong convexity
   - Examples (gradient, exponentiated gradient)
   - Extensions

Online Convex Optimization: Regularization

Regularized minimization

$$a_{t+1} = \arg\min_{a \in \mathcal{A}} \left(\eta \sum_{s=1}^{t} \ell_s(a) + R(a) \right).$$

The regularizer $R : \mathbb{R}^d \to \mathbb{R}$ is strictly convex and differentiable.

Regularization Methods: Regret

Theorem

For $\mathcal{A} = \mathbb{R}^d$, regularized minimization suffers regret against any $a \in \mathcal{A}$ of

$$\sum_{t=1}^{n} \ell_t(a_t) - \sum_{t=1}^{n} \ell_t(a) = \frac{D_R(a, a_1) - D_{\Phi_n}(a, a_{n+1})}{\eta} + \frac{1}{\eta} \sum_{t=1}^{n} D_{\Phi_t}(a_t, a_{t+1}),$$

and thus

$$\hat{L}_n \leq \inf_{a \in \mathbb{R}^d} \left(\sum_{t=1}^{n} \ell_t(a) + \frac{D_R(a, a_1)}{\eta} \right) + \frac{1}{\eta} \sum_{t=1}^{n} D_{\Phi_t}(a_t, a_{t+1}).$$

So the sizes of the steps $D_{\Phi_t}(a_t, a_{t+1})$ determine the regret bound.

Regularization Methods: Regret

Theorem

For $\mathcal{A} = \mathbb{R}^d$, regularized minimization suffers regret

$$\hat{L}_n \leq \inf_{a \in \mathbb{R}^d} \left(\sum_{t=1}^{n} \ell_t(a) + \frac{D_R(a, a_1)}{\eta} \right) + \frac{1}{\eta} \sum_{t=1}^{n} D_{\Phi_t}(a_t, a_{t+1}).$$

Notice that we can write

$$\begin{aligned} D_{\Phi_t}(a_t, a_{t+1}) &= D_{\Phi_t^*}(\nabla \Phi_t(a_{t+1}), \nabla \Phi_t(a_t)) \\ &= D_{\Phi_t^*}(0, \nabla \Phi_{t-1}(a_t) + \eta \nabla \ell_t(a_t)) \\ &= D_{\Phi_t^*}(0, \eta \nabla \ell_t(a_t)). \end{aligned}$$

So it is the size of the gradient steps, $D_{\Phi_t^*}(0, \eta \nabla \ell_t(a_t))$, that determines the regret.

Regularization Methods: Regret Bounds

Example

Suppose $R = \frac{1}{2}\|\cdot\|^2$. Then we have

$$\hat{L}_n \leq L_n^* + \frac{\|a^* - a_1\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{n}\|g_t\|^2.$$

And if $\|g_t\| \leq G$ and $\|a^* - a_1\| \leq D$, choosing η appropriately gives $\hat{L}_n \leq L_n^* \leq DG\sqrt{n}$.

Online Convex Optimization

1. Problem formulation
2. Empirical minimization fails.
3. Gradient algorithm.
4. Regularized minimization and Bregman divergences
5. Regret bounds
   - Unconstrained minimization
   - Seeing the future
   - Strong convexity
   - Examples (gradient, exponentiated gradient)
   - Extensions

Regularization Methods: Regret Bounds

Seeing the future gives small regret:

Theorem

For all $a \in \mathcal{A}$,

$$\sum_{t=1}^{n} \ell_t(a_{t+1}) - \sum_{t=1}^{n} \ell_t(a) \leq \frac{1}{\eta}(R(a) - R(a_1)).$$

Regularization Methods: Regret Bounds

Proof.

Since a_{t+1} minimizes Φ_t,

$$\eta \sum_{s=1}^{t} \ell_s(a) + R(a) \geq \eta \sum_{s=1}^{t} \ell_s(a_{t+1}) + R(a_{t+1})$$

$$= \eta \ell_t(a_{t+1}) + \eta \sum_{s=1}^{t-1} \ell_s(a_{t+1}) + R(a_{t+1})$$

$$\geq \eta \ell_t(a_{t+1}) + \eta \sum_{s=1}^{t-1} \ell_s(a_t) + R(a_t)$$

$$\vdots$$

$$\geq \eta \sum_{s=1}^{t} \ell_s(a_{s+1}) + R(a_1).$$

Regularization Methods: Regret Bounds

Theorem

For all $a \in \mathcal{A}$,

$$\sum_{t=1}^{n} \ell_t(a_{t+1}) - \sum_{t=1}^{n} \ell_t(a) \leq \frac{1}{\eta}(R(a) - R(a_1)).$$

Thus, if a_t and a_{t+1} are close, then regret is small:

Corollary

For all $a \in \mathcal{A}$,

$$\sum_{t=1}^{n} (\ell_t(a_t) - \ell_t(a)) \leq \sum_{t=1}^{n} (\ell_t(a_t) - \ell_t(a_{t+1})) + \frac{1}{\eta} (R(a) - R(a_1)) .$$

So how can we control the increments $\ell_t(a_t) - \ell_t(a_{t+1})$?

Online Convex Optimization

1. Problem formulation
2. Empirical minimization fails.
3. Gradient algorithm.
4. Regularized minimization
   - Bregman divergence
   - Regularized minimization equivalent and Bregman divergence from previous
   - Constrained minimization equivalent to unconstrained plus Bregman projection
   - Linearization
   - Mirror descent
5. Regret bounds
   - Unconstrained minimization
   - Seeing the future
   - Strong convexity
   - Examples (gradient, exponentiated gradient)
   - Extensions

Regularization Methods: Regret Bounds

Definition

We say R is strongly convex wrt a norm $\|\cdot\|$ if, for all a, b,

$$R(a) \geq R(b) + \nabla R(b) \cdot (a - b) + \frac{1}{2}\|a - b\|^2.$$

For linear losses and strongly convex regularizers, the dual norm of the gradient is small:

Theorem

If R is strongly convex wrt a norm $\|\cdot\|$, and $\ell_t(a) = g_t \cdot a$, then

$$\|a_t - a_{t+1}\| \leq \eta \|g_t\|_*,$$

where $\|\cdot\|_*$ is the dual norm to $\|\cdot\|$:

$$\|v\|_* = \sup\{|v \cdot a| : a \in \mathcal{A}, \|a\| \leq 1\}.$$

Regularization Methods: Regret Bounds

Proof.

$$R(a_t) \geq R(a_{t+1}) + \nabla R(a_{t+1}) \cdot (a_t - a_{t+1}) + \frac{1}{2}\|a_t - a_{t+1}\|^2,$$

$$R(a_{t+1}) \geq R(a_t) + \nabla R(a_t) \cdot (a_{t+1} - a_t) + \frac{1}{2}\|a_t - a_{t+1}\|^2.$$

Combining,

$$\|a_t - a_{t+1}\|^2 \leq (\nabla R(a_t) - \nabla R(a_{t+1})) \cdot (a_t - a_{t+1})$$

Hence,

$$\|a_t - a_{t+1}\| \leq \|\nabla R(a_t) - \nabla R(a_{t+1})\|_* = \|\eta g_t\|_*.$$

□

Regularization Methods: Regret Bounds

This leads to the regret bound:

Corollary

For linear losses, if R is strongly convex wrt $\|\cdot\|$*, then for all* $a \in \mathcal{A}$*,*

$$\sum_{t=1}^{n}(\ell_t(a_t) - \ell_t(a)) \leq \eta \sum_{t=1}^{n} \|g_t\|_*^2 + \frac{1}{\eta}\left(R(a) - R(a_1)\right).$$

Thus, for $\|g_t\|_* \leq G$ and $R(a) - R(a_1) \leq D^2$, choosing η appropriately gives regret no more than $2GD\sqrt{n}$.

Regularization Methods: Regret Bounds

Example

Consider $R(a) = \frac{1}{2}\|a\|^2$, $a_1 = 0$, and $\mathcal{A}$ contained in a Euclidean ball of diameter D.
Then R is strongly convex wrt $\|\cdot\|$ and $\|\cdot\|_* = \|\cdot\|$. And the mapping between primal and dual spaces is the identity.
So if $\sup_{a\in\mathcal{A}} \|\nabla \ell_t(a)\| \leq G$, then regret is no more than $2GD\sqrt{n}$.

Regularization Methods: Regret Bounds

Example

Consider $\mathcal{A} = \Delta^m$, $R(a) = \sum_i a_i \ln a_i$. Then the mapping between primal and dual spaces is $\nabla R(a) = \ln(a)$ (component-wise). And the divergence is the KL divergence,

$$D_R(a, b) = \sum_i a_i \ln(a_i / b_i).$$

And R is strongly convex wrt $\|\cdot\|_1$ (check!).
Suppose that $\|g_t\|_\infty \leq 1$. Also, $R(a) - R(a_1) \leq \ln m$, so the regret is no more than $2\sqrt{n \ln m}$.

Regularization Methods: Regret Bounds

Example

$\mathcal{A} = \Delta^m$, $R(a) = \sum_i a_i \ln a_i$.
What are the updates?

$$\begin{aligned} a_{t+1} &= \Pi^R_{\mathcal{A}}(\tilde{a}_{t+1}) \\ &= \Pi^R_{\mathcal{A}}(\nabla R^*(\nabla R(\tilde{a}_t) - \eta g_t)) \\ &= \Pi^R_{\mathcal{A}}(\nabla R^*(\ln(\tilde{a}_t \exp(-\eta g_t))) \\ &= \Pi^R_{\mathcal{A}}(\tilde{a}_t \exp(-\eta g_t)), \end{aligned}$$

where the ln and exp functions are applied component-wise.
This is exponentiated gradient: mirror descent with $\nabla R = \ln$.
It is easy to check that the projection corresponds to normalization, $\Pi^R_{\mathcal{A}}(\tilde{a}) = \tilde{a}/\|a\|_1$.

Regularization Methods: Regret Bounds

Notice that when the losses are linear, exponentiated gradient is exactly the exponential weights strategy we discussed for a finite comparison class.
Compare $R(a) = \sum_i a_i \ln a_i$ with $R(a) = \frac{1}{2}\|a\|^2$,
for $\|g_t\|_\infty \leq 1$, $\mathcal{A} = \Delta^m$:

$O(\sqrt{n \ln m})$ versus $O(\sqrt{mn})$.

Online Convex Optimization

1. Problem formulation
2. Empirical minimization fails.
3. Gradient algorithm.
4. Regularized minimization
   - Bregman divergence
   - Regularized minimization equivalent and Bregman divergence from previous
   - Constrained minimization equivalent to unconstrained plus Bregman projection
   - Linearization
   - Mirror descent
5. Regret bounds
   - Unconstrained minimization
   - Strong convexity
   - Examples (gradient, exponentiated gradient)
   - Extensions

Regularization Methods: Extensions

- Instead of

$$a_{t+1} = \arg\min_{a \in \mathcal{A}} \left(\eta \ell_t(a) + D_{\Phi_{t-1}}(a, \tilde{a}_t) \right),$$

  we can use

$$a_{t+1} = \arg\min_{a \in \mathcal{A}} \left(\eta \ell_t(a) + D_{\Phi_{t-1}}(a, a_t) \right).$$

  And analogous results apply. For instance, this is the approach used by the first gradient method we considered.
- We can get faster rates with stronger assumptions on the losses...

Regularization Methods: Varying η

Theorem

Define

$$a_{t+1} = \arg\min_{a \in \mathbb{R}^d} \left(\sum_{t=1}^{n} \eta_t \ell_t(a) + R(a) \right).$$

For any $a \in \mathbb{R}^d$,

$$\hat{L}_n - \sum_{t=1}^{n} \ell_t(a) \leq \sum_{t=1}^{n} \frac{1}{\eta_t} \left(D_{\Phi_t}(a_t, a_{t+1}) + D_{\Phi_{t-1}}(a, a_t) - D_{\Phi_t}(a, a_{t+1}) \right).$$

If we linearize the ℓ_t, we have

$$\hat{L}_n - \sum_{t=1}^{n} \ell_t(a) \leq \sum_{t=1}^{n} \frac{1}{\eta_t} \left(D_R(a_t, a_{t+1}) + D_R(a, a_t) - D_R(a, a_{t+1}) \right).$$

But what if ℓ_t are strongly convex?

Regularization Methods: Strongly Convex Losses

Theorem

If ℓ_t is σ-strongly convex wrt R, that is, for all $a, b \in \mathbb{R}^d$,

$$\ell_t(a) \geq \ell_t(b) + \nabla \ell_t(b) \cdot (a - b) + \frac{\sigma}{2} D_R(a, b),$$

then for any $a \in \mathbb{R}^d$, this strategy with $\eta_t = \frac{2}{t\sigma}$ has regret

$$\hat{L}_n - \sum_{t=1}^{n} \ell_t(a) \leq \sum_{t=1}^{n} \frac{1}{\eta_t} D_R(a_t, a_{t+1}).$$

Strongly Convex Losses: Proof idea

$$
\begin{aligned}
&\sum_{t=1}^{n} \left(\ell_t(a_t) - \ell_t(a)\right) \\
&\leq \sum_{t=1}^{n} \left(\nabla \ell_t(a_t) \cdot (a_t - a) - \frac{\sigma}{2} D_R(a, a_t)\right) \\
&\leq \sum_{t=1}^{n} \frac{1}{\eta_t} \left(D_R(a_t, a_{t+1}) + D_R(a, a_t) - D_R(a, a_{t+1}) - \frac{\eta_t \sigma}{2} D_R(a, a_t)\right) \\
&\leq \sum_{t=1}^{n} \frac{1}{\eta_t} D_R(a_t, a_{t+1}) + \sum_{t=2}^{n} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \frac{\sigma}{2}\right) D_R(a, a_t) \\
&\qquad + \left(\frac{1}{\eta_1} - \frac{\sigma}{2}\right) D_R(a, a_1).
\end{aligned}
$$

And choosing η_t appropriately eliminates the second and third terms.

Strongly Convex Losses

Example

For $R(a) = \frac{1}{2}\|a\|^2$, we have

$$\hat{L}_n - L_n^* \leq \frac{1}{2}\sum_{t=1}^{n} \frac{1}{\eta_t}\|\eta_t \nabla \ell_t\|^2 \leq \sum_{t=1}^{n} \frac{G^2}{t\sigma} = O\left(\frac{G^2}{\sigma}\log n\right).$$

Strongly Convex Losses

Key Point: When the loss is strongly convex wrt the regularizer, the regret rate can be faster; in the case of quadratic R (and ℓ_t), it is $O(\log n)$, versus $O(\sqrt{n})$.

Course Synopsis

- A finite comparison class: $\mathcal{A} = \{1, \ldots, m\}$.
- Converting online to batch.
- Online convex optimization.
- Log loss.
  - Three views of log loss.
  - Normalized maximum likelihood.
  - Sequential investment.
  - Constantly rebalanced portfolios.

Log Loss

A family of decision problems with several equivalent interpretations:

- Maximizing long term rate of growth in portfolio optimization.
- Minimizing redundancy in data compression.
- Minimizing likelihood ratio in sequential probability assignment.

See Nicolò Cesa-Bianchi and Gàbor Lugosi, *Prediction, Learning and Games*, Chapters 9, 10.

Log Loss

- Consider a finite outcome space $\mathcal{Y} = \{1, \ldots, m\}$.
- The comparison class $\mathcal{A}$ is a set of sequences $f_1, f_2, \ldots$ of maps $f_t : \mathcal{Y}^t \to \Delta^{\mathcal{Y}}$.
- We write $f_t(y_t | y_1, \ldots, y_{t-1})$, notation that is suggestive of a conditional probability distribution.
- The adversary chooses, at round t, a value $y_t \in \mathcal{Y}$, and the loss function for a particular sequence f is

$$\ell_t(f) = -\ln(f_t(y_t | y_1, \ldots, y_{t-1})).$$

Log Loss: Notation

$$y^n = y_1^n = (y_1, \ldots, y_n),$$
$$f_n(y^n) = \prod_{t=1}^{n} f_t(y_t|y^{t-1}),$$
$$a_n(y^n) = \prod_{t=1}^{n} a_t(y_t|y^{t-1}).$$

Again, this notation is suggestive of probability distributions. Check:

$$f_n(y^n) \geq 0 \qquad \sum_{y^n \in \mathcal{Y}^n} f_n(y^n) = 1.$$

Log Loss: Three applications

- Sequential probability assignment.
- Gambling/investment.
- Data compression.

Log Loss: Sequential Probability Assignment

Think of y_t as the indicator for the event that it rains on day t. Minimizing log loss is forecasting $\Pr(y_t|y^{t-1})$ sequentially:

$$L_n^* = \inf_{f \in F} \sum_{t=1}^{n} \ln \frac{1}{f_t(y_t|y^{t-1})}$$

$$\hat{L}_n = \sum_{t=1}^{n} \ln \frac{1}{a_t(y_t|y^{t-1})}$$

$$L_n^* - \hat{L}_n = \sup_{f \in F} \ln \frac{f_n(y^n)}{a_n(y^n)},$$

which is the worst ratio of log likelihoods.

Log Loss: Gambling

Suppose we are investing our initial capital C in proportions

$$a_t(1), \ldots, a_t(m)$$

across m horses. If horse i wins, it pays odds $o_t(i) \geq 0$. In that case, our capital becomes $Ca_t(i)o_t(i)$.
Let $y_t \in \{1, \ldots, m\}$ denote the winner of race t.
Suppose that $a_t(y|y_1, \ldots, y_{t-1})$ depends on the previous winners. Then our capital goes from C to

$$C \prod_{t=1}^{n} a_t(y_t|y^{t-1})o_t(y_t).$$

Log Loss: Gambling

Compared to a set F of experts (who also start with capital C), the ratio of the best expert's final capital to ours is

$$\begin{aligned}
&\sup_{f \in F} \frac{C \prod_{t=1}^{n} f_t(y_t | y^{t-1}) o_t(y_t)}{C \prod_{t=1}^{n} a_t(y_t | y^{t-1}) o_t(y_t)} \\
&= \sup_{f \in F} \frac{f_n(y^n)}{a_n(y^n)} \\
&= \exp\left(\sup_{f \in F} \ln \frac{f_n(y^n)}{a_n(y^n)} \right).
\end{aligned}$$

Log Loss: Data Compression

We can identify probability distributions with codes, and view $\ln p(y^n)$ as the length (in *nats*) of an optimal sequentially constructed codeword encoding the sequence y^n, under the assumption that y^n is generated by p.
Then

$$-\ln p_n(y^n) - \inf_{f \in F} \left(-\ln f_n(y^n)\right) = \hat{L} - L^*$$

is the *redundancy* (excess length) of the code with respect to a family F of codes.

Log Loss: Optimal Prediction

The minimax regret for a class F is

$$V_n(F) = \inf_a \sup_{y^n \in \mathcal{Y}^n} \ln \frac{\sup_{f \in F} f_n(y^n)}{a_n(y^n)}.$$

For a class F and $n > 0$, define the normalized maximum likelihood strategy a^* by

$$a_n^*(y^n) = \frac{\sup_{f \in F} f_n(y^n)}{\sum_{x^n \in \mathcal{Y}^n} \sup_{f \in F} f_n(x^n)}.$$

Log Loss: Optimal Prediction

Theorem

1. a^* is the unique strategy that satisfies

$$\sup_{y^n \in \mathcal{Y}^n} \ln \frac{\sup_{f \in F} f_n(y^n)}{a_n^*(y^n)} = V_n(F).$$

2. For all $y^n \in \mathcal{Y}^n$,

$$\ln \frac{\sup_{f \in F} f_n(y^n)}{a_n^*(y^n)} = \ln \sum_{x^n \in \mathcal{Y}^n} \sup_{f \in F} f_n(x^n).$$

Log Loss: Optimal Prediction

Proof.

2. By the definition of a_n^*,

$$\ln \frac{\sup_{f \in F} f_n(y^n)}{a_n^*(y^n)} = \ln \sum_{x^n \in \mathcal{Y}^n} \sup_{f \in F} f_n(x^n).$$

1. For any other a, there must be a $y^n \in \mathcal{Y}^n$ with $a_n(y^n) < a_n^*(y^n)$. Then

$$\ln \frac{\sup_{f \in F} f_n(y^n)}{a_n(y^n)} > \ln \frac{\sup_{f \in F} f_n(y^n)}{a_n^*(y^n)},$$

which implies the sup over y^n is bigger than its value for a^*. □

Log Loss: Optimal Prediction

How do we compute the normalized maximum likelihood strategy?

$$a_n^*(y^n) = \frac{\sup_{f \in F} f_n(y^n)}{\sum_{x^n \in \mathcal{Y}^n} \sup_{f \in F} f_n(x^n)}.$$

This a_n^* is a probability distribution on $\mathcal{Y}^n$. We can calculate it sequentially via

$$a_t^*(y_t | y^{t-1}) = \frac{a_t^*(y^t)}{a_{t-1}^*(y^{t-1})},$$

where

$$a_t^*(y^t) = \sum_{y_{t+1}^n \in \mathcal{Y}^{n-t}} a_n^*(y^n).$$

Log Loss: Optimal Prediction

- In general, these are big sums.
- The normalized maximum likelihood strategy does not exist if we cannot sum $\sup_{f \in F} f_n(x^n)$ over $x^n \in \mathcal{Y}^\backslash$.
- We need to know the horizon n: it is not possible to extend the strategy for $n-1$ to the strategy for n.
- In many cases, there are efficient strategies that approximate the performance of the optimal (normalized maximum likelihood) strategy.

Log Loss: Minimax Regret

Example

Suppose $|F| = m$. Then we have

$$\begin{aligned} V_n(F) &= \ln \sum_{y^n \in \mathcal{Y}^n} \sup_{f \in F} f_n(y^n) \\ &\leq \ln \sum_{y^n \in \mathcal{Y}^n} \sum_{f \in F} f_n(y^n) \\ &= \ln \sum_{f \in F} \sum_{y^n \in \mathcal{Y}^n} f_n(y^n) \\ &= \ln N. \end{aligned}$$

Log Loss: Minimax Regret

Example

Consider the class F of all constant experts:

$$f_t(y|y^{t-1}) = f_t(y).$$

For $|\mathcal{Y}| = 2$,

$$V_n(F) = \frac{1}{2}\ln n + \frac{1}{2}\ln\frac{\pi}{2} + o(1).$$

Minimax Regret: Proof Idea

$$V_n(F) = \ln \sum_{y^n \in \mathcal{Y}^n} \sup_{f \in F} f_n(y^n).$$

Suppose that $f(1) = q$, $f(0) = 1 - q$. Clearly, $f_n(y^n)$ depends only on the number n_1 of 1s in y^n, and it's easy to check that the maximizing value of q is n_1/n, so

$$\sup_{f \in F} f_n(y^n) = \max_q (1-q)^{n-n_1} q^{n_1} = \left(\frac{n-n_1}{n}\right)^{n-n_1} \left(\frac{n_1}{n}\right)^{n_1}.$$

Thus (using Stirling's approximation),

$$V_n(F) = \ln \sum_{n_1=1}^{n-1} \binom{n}{n_1} \left(\frac{n-n_1}{n}\right)^{n-n_1} \left(\frac{n_1}{n}\right)^{n_1}$$

$$\vdots$$

$$= \ln \left((1 + o(1)) \sqrt{\frac{n\pi}{2}} \right).$$

Log Loss

- Three views of log loss.
- Normalized maximum likelihood.
- Sequential investment.
- Constantly rebalanced portfolios.

Sequential Investment

Suppose that we have n financial instruments (let's call them $1, 2, \ldots, n$), and at each period we need to choose how to spread our capital. We invest a proportion p_i in instrument i (with $p_i \geq 0$ and $\sum_i p_i = 1$). During the period, the value of instrument i increases by a factor of $x_i \geq 0$ and so our wealth increases by a factor of

$$p'x = \sum_{i=1}^{n} p_i x_i.$$

For instance, $x_1 = 1$ and $x_2 \in \{0, 2\}$ corresponds to a choice between doing nothing and placing a fair bet at even odds.

Asymptotic growth rate optimality of logarithmic utility

Logarithmic utility has the attractive property that, if the vectors of market returns $X_1, X_2, \ldots X_t, \ldots$ are random, then maximizing expected log wealth leads to the optimal asymptotic growth rate.

We'll illustrate with a simple example, and then state a general result. Suppose that we are betting on two instruments many times. Their one-period returns (that is, the ratio of the instrument's value after period t to that before period t) satisfy

$$\begin{aligned} \Pr(X_{t,1} = 1) &= 1, \\ \Pr(X_{t,2} = 0) &= p, \\ Pr(X_{t,2} = 2) &= 1 - p. \end{aligned}$$

Clearly, one is risk free, and the other has two possible outcomes: complete loss of the investment, and doubling of the investment.

Asymptotic growth rate optimality of logarithmic utility

For instance, suppose that we start with wealth at $t = 0$ of $V_0 > 0$, and $0 < p < 1$. If we bet all of our money on instrument 2 at each step, then after T rounds we end up with expected wealth of

$$\mathbf{E}\, V_T = (2(1-p))^T V_0,$$

and this is the maximum value of expected wealth over all strategies. But with probability one, we will eventually have wealth zero if we follow this strategy. What should we do?

Asymptotic growth rate optimality of logarithmic utility

Suppose that, for period t, we bet a fraction b_t of our wealth on instrument 2. Then if we define

$$W_t = 1[X_{t,2} = 2] \qquad \text{(that is, we win the bet)},$$

then we have

$$V_{t+1} = (1 + b_t)^{W_t}(1 - b)^{1-W_t} V_t.$$

Consider the asymptotic growth rate of wealth,

$$G = \lim_{T\to\infty} \frac{1}{T} \log_2 \frac{V_T}{V_0}.$$

(This extracts the exponent.)

Asymptotic growth rate optimality of logarithmic utility

By the weak law of large numbers, we have

$$G = \lim_{T\to\infty} \left(\frac{1}{T} \sum_{t=1}^{T} \left(W_t \log_2(1 + b_t) + (1 - W_t) \log_2(1 - b_t) \right) \right)$$
$$= \lim_{T\to\infty} \left(\frac{1}{T} \sum_{t=1}^{T} \left((1 - p) \log_2(1 + b_t) + p \log_2(1 - b_t) \right) \right).$$

For what values of b_t is this maximized? Well, the concavity of $\log_2$, together with Jensen's inequality, implies that, for all $x_i \geq 0$ with $\sum_i x_i = x$,

$$\begin{aligned} \max \quad & \sum x_i \log y_i \\ \text{s.t.} \quad & \sum y_i = y \end{aligned}$$

has the solution $y_i = x_i y / x$. Thus, we should set $b_t = 1 - 2p$.

Asymptotic growth rate optimality of logarithmic utility

That is, if we choose the proportion b_t to allocate to each instrument so as to maximize the expected log return,

$$((1-p)\log_2(1+b_t) + p\log_2(1-b_t)),$$

then we obtain the optimal exponent in the asymptotic growth rate, which is

$$G = (1-p)\log_2(2(1-p)) + p\log_2(2p).$$

Notice that if p is strictly less than $1/2$, $G > 0$. That is, we have exponential growth. Compare this with the two individual alternatives: choosing instrument 1 gives no growth, whereas choosing instrument 2 gives expected wealth that grows exponentially, but it leads to ruin, almost surely.

Asymptotic growth rate optimality of logarithmic utility

This result was first pointed out by Kelly [5]. Kelly viewed p as the probability that a one-bit message containing the future outcome X_t was transmitted through a communication channel incorrectly, and then the optimal exponent G is equal to the channel capacity,

$$G = 1 - \left((1-p) \log_2 \frac{1}{1-p} + p \log_2 \frac{1}{p} \right).$$

Asymptotic growth rate optimality of logarithmic utility

Maximizing expected log return is asymptotically optimal much more generally. To define the general result, suppose that, in period t, we need to distribute our wealth over m instruments. We allocate proportion $b_{t,i}$ to the ith, and assume the $b_t \in \Delta_m$, the m-simplex. Then, if the period t returns are $X_{t,1}, \ldots, X_{t,m} \geq 0$, the yield per dollar invested is $b_t \cdot X_t$, so that our initial capital of V_t becomes

$$V_{t+1} = V_t b_t \cdot X_t.$$

By a strategy, we mean a sequence of functions $\{b_t\}$ which, at time t, uses the allocation $b_t(X_1, \ldots, X_{t-1}) \in \Delta_m$.

Asymptotic growth rate optimality of logarithmic utility

Definition

If $X_t \in \mathbb{R}_+^m$ denotes the random returns of m instruments during period t, we say that strategy b^* is log-optimal if

$$b_t^*(X_0, \ldots, X_{t-1}) = \arg\max_{b \in \Delta_m} \mathbf{E}\left[\log(b \cdot X_t) | X_0, \ldots, X_{t-1}\right].$$

Asymptotic growth rate optimality of logarithmic utility

Breiman [3] proved the following result for i.i.d. discrete-valued returns; Algoet and Cover [1] proved the general case.

Theorem

Suppose that the log-optimal strategy b^* has capital growth $V_0, V_1^*, \ldots, V_T^*$ over T periods and some strategy b has capital growth $V_0, V_1, \ldots, V_T$. Then almost surely

$$\limsup_{T\to\infty} \frac{1}{T} \log \frac{V_T}{V_T^*} \leq 0.$$

In particular, if the returns are i.i.d., then in each period the optimal strategy (at least, optimal to first order in the exponent) allocates its capital according to some fixed mixture $b^* \in \Delta_m$. This mixture is the one that maximizes the expected logarithm of the one-period yield.

Asymptotic growth rate optimality of logarithmic utility

This is an appealing property: if we are interested in what happens asymptotically, then we should use log as a utility function, and maximize the expected log return during each period.

Constantly rebalanced portfolios

A constantly rebalanced portfolio (CRP) is an investment strategy defined by a mixture vector $b \in \Delta_m$. At every time step, it allocates proportion b_j of the total capital to instrument j. We have seen that, for i.i.d. returns, the asymptotic growth rate is maximized by a particular CRP. The Dow Jones Industrial Average measures the performance of another CRP (the one that allocates one thirtieth of its capital to each of thirty stocks). Investing in a single stock is another special case of a CRP. (As an illustration of the benefits provided by rebalancing, consider an i.i.d. market with two instruments and return vectors chosen uniformly from $\{(2, 1/2), (1/2, 2)\}$. Investing in any single instrument leads to a growth rate of 0, whereas a $(1/2, 1/2)$ CRP will have wealth that increases by a factor of $5/4$ in each period.)

Constantly rebalanced portfolios

Now that we've motivated CRPs, we'll drop all probabilistic assumptions and move back to an online setting. Suppose that the market is adversarial (a reasonable assumption), and consider the problem of competing with the best CRP *in hindsight*. That is, at each step t we must choose an allocation of our capital b_t so that, after T rounds, the logarithm of our wealth is close to that of the best CRP.

Constantly rebalanced portfolios

The following theorem is due to Cover [4] (the proof we give is due to Blum and Kalai [2]). It shows that there is a *universal portfolio strategy*, that is, one that competes with the best CRP.

Theorem

There is a strategy (call it b_U) for which

$$\log(V_T) \geq \log(V_T(b^*)) - (m-1)\log(T+1) - 1,$$

where b^* is the best CRP.

The strategy is conceptually very simple. It involves distributing capital uniformly across all CRPs at each period.

Constantly rebalanced portfolios

Consider competing with the m single instrument portfolios. We could just place our money uniformly across the m instruments at the start, and leave it there. Then we have

$$\begin{aligned}\log(V_T) &= \log\left(\sum_{j=1}^{m}\prod_{t=1}^{T} X_{t,j}(V_0/m)\right)\\ &\geq \max_j \log\left(\prod_{t=1}^{T} X_{t,j}(V_0/m)\right)\\ &= \max_j \log\left(\prod_{t=1}^{T} X_{t,j} V_0\right) - \log(m),\end{aligned}$$

that is, our regret with respect to the best single instrument portfolio (in hindsight) is no more than $\log m$.

Constantly rebalanced portfolios

To compete with the set of CRPs, we adopt a similar strategy: we allocate our capital uniformly over Δ_m, and then calculate the mixture b_t that corresponds at time t to this initial distribution. Consider an infinitesimal region around a point $b \in \Delta_m$. If μ is the uniform measure on Δ_m, the initial investment in CRP b is $d\mu(b)V_0$. By time $t-1$, this has grown to $V_{t-1}(b)d\mu(b)V_0$, and so this is the contribution to the overall mixture b_t. And of course we need to appropriately normalize (by the total capital at time $t-1$):

$$b_t = \frac{\int_{\Delta_m} bV_{t-1}(b)d\mu(b)}{\int_{\Delta_m} V_{t-1}(b)d\mu(b)}.$$

Constantly rebalanced portfolios

How does this strategy perform? Suppose that b^* is the best CRP in hindsight. Then the region around b^* contains very similar mixtures, and provided that there is enough volume of sufficiently similar CRPs, our strategy should be able to compete with b^*. Indeed, consider the set of mixtures of b^* with some other vector $a \in \Delta_m$,

$$S = \{(1-\epsilon)b^* + \epsilon a : a \in \Delta_m\}.$$

For every $b \in S$, we have

$$\frac{V_1(b)}{V_0} = \frac{V_1((1-\epsilon)b^* + \epsilon a)}{V_0} \geq \frac{(1-\epsilon)V_1(b^*)}{V_0}.$$

Thus, after T steps,

$$\frac{V_T(b)}{V_T(b^*)} \geq (1-\epsilon)^T.$$

Constantly rebalanced portfolios

Also, the proportion of initial wealth allocated to CRPs in S is

$$\mu(S) = \mu(\{\epsilon a : a \in \Delta_m\}) = \epsilon^{m-1}.$$

Combining these two facts, we have that

$$\log\left(\frac{V_T(b_U)}{V_T(b^*)}\right) \geq \log\left((1-\epsilon)^T \epsilon^{m-1}\right).$$

Setting $\epsilon = 1/(T+1)$ gives a regret of

$$\log\left((1-1/(T+1))^T (T+1)^{-(m-1)}\right) > -1-(m-1)\log(T+1).$$

Constantly rebalanced portfolios

There are other approaches to portfolio optimization based on the online prediction strategies that we have seen earlier. For instance, the exponential weights algorithm can be used in this setting, although it leads to $\sqrt{T}$ regret, rather than $\log T$. Also, gradient descent approaches have also been investigated. For a Newton update method, logarithmic regret bounds have been proved.


Paul H. Algoet and Thomas M. Cover.
Asymptotic optimality and asymptotic equipartition properties of log-optimum investment.
The Annals of Probability, 16(2):876–898, 1988.


Avrim Blum and Adam Kalai.
Universal portfolios with and without transaction costs.
Machine Learning, 35:193–205, 1999.


Leo Breiman.
Optimal gambling systems for favorable games.
In *Proc. Fourth Berkeley Symp. Math. Statist. Probab.*, volume 1, pages 60–77. Univ. California Press, 1960.


Thomas M. Cover.
Universal portfolios.
Mathematical Finance, 1(1):1–29, 1991.


Jr. J. L. Kelly.
A new interpretation of information rate.
J. Oper. Res. Soc., 57:975–985, 1956.

Course Synopsis

- A finite comparison class: $\mathcal{A} = \{1, \ldots, m\}$.
- Converting online to batch.
- Online convex optimization.
- Log loss.