# The Online Approach to Machine Learning

Nicolò Cesa-Bianchi

Università degli Studi di Milano

# Summary

1. My beautiful regret
2. A supposedly fun game I'll play again
3. A graphic novel
4. The joy of convex
5. The joy of convex (without the gradient)

# Summary

1. My beautiful regret
2. A supposedly fun game I'll play again
3. A graphic novel
4. The joy of convex
5. The joy of convex (without the gradient)

# Machine learning

## Classification/regression tasks

- Predictive models $h$ mapping data instances $X$ to labels $Y$ (e.g., binary classifier)
- Training data $S_T = \big((X_1, Y_1), \dots, (X_T, Y_T)\big)$ (e.g., email messages with spam vs. nonspam annotations)
- Learning algorithm $A$ (e.g., Support Vector Machine) maps training data $S_T$ to model $h = A(S_T)$

Evaluate the risk of the trained model $h$ with respect to a given loss function

# Two notions of risk

## View data as a statistical sample: statistical risk

$$\mathbb{E}\Big[\mathrm{loss}\Big(\underbrace{A(S_T)}_{\substack{\text{trained}\\\text{model}}},\ \underbrace{(X,Y)}_{\substack{\text{test}\\\text{example}}}\Big)\Big]$$

Training set $S_T = \big((X_1,Y_1),\dots,(X_T,Y_T)\big)$ and test example $(X,Y)$ drawn i.i.d. from the same unknown and fixed distribution

## View data as an arbitrary sequence: sequential risk

$$\sum_{t=1}^{T} \mathrm{loss}\Big(\underbrace{A(S_{t-1})}_{\substack{\text{trained}\\\text{model}}},\ \underbrace{(X_t,Y_t)}_{\substack{\text{test}\\\text{example}}}\Big)$$

Sequence of models trained on growing prefixes
$S_t = \big((X_1,Y_1),\dots,(X_t,Y_t)\big)$ of the data sequence

# Regrets, I had a few

Learning algorithm $A$ maps datasets to models in a given class $\mathcal{H}$

## Variance error in statistical learning

$$\mathbb{E}\Big[\operatorname{loss}\big(A(S_T),(X,Y)\big)\Big] - \inf_{h\in\mathcal{H}} \mathbb{E}\Big[\operatorname{loss}\big(h,(X,Y)\big)\Big]$$

compare to expected loss of best model in the class

## Regret in online learning

$$\sum_{t=1}^{T} \operatorname{loss}\big(A(S_{t-1}),(X_t,Y_t)\big) - \inf_{h\in\mathcal{H}} \sum_{t=1}^{T} \operatorname{loss}\big(h,(X_t,Y_t)\big)$$

compare to cumulative loss of best model in the class

# Incremental model update

A natural blueprint for online learning algorithms

**For $t = 1, 2, \ldots$**

1. Apply current model $h_{t-1}$ to next data element $(X_t, Y_t)$
2. Update current model: $h_{t-1} \to h_t \in \mathcal{H}$

**Goal: control regret**

$$\sum_{t=1}^{T} \mathrm{loss}\big(h_{t-1}, (X_t, Y_t)\big) - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathrm{loss}\big(h, (X_t, Y_t)\big)$$

View this as a repeated game between a player generating predictors $h_t \in \mathcal{H}$ and an opponent generating data $(X_t, Y_t)$

# Summary

# Theory of repeated games

James Hannan
(1922–2010)

David Blackwell
(1919–2010)

**Learning to play a game (1956)**

Play a game repeatedly against a possibly suboptimal opponent

# Zero-sum 2-person games played more than once

| | 1 | 2 | $\dots$ | M |
|---|---|---|---|---|
| 1 | $\ell(1,1)$ | $\ell(1,2)$ | $\dots$ | |
| 2 | $\ell(2,1)$ | $\ell(2,2)$ | $\dots$ | |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | |
| N | | | | |

**$N \times M$ known loss matrix**

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

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

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

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

# Prediction with expert advice

Volodya Vovk

Manfred Warmuth

| | $t=1$ | $t=2$ | $\dots$ |
|---|---|---|---|
| 1 | $\ell_1(1)$ | $\ell_2(1)$ | $\dots$ |
| 2 | $\ell_1(2)$ | $\ell_2(2)$ | $\dots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ |
| N | $\ell_1(N)$ | $\ell_2(N)$ | |

Opponent's moves $y_1, y_2, \dots$ define a sequential prediction problem with a time-varying loss function $\boxed{\ell(i_t, y_t) = \ell_t(i_t)}$

# Playing the experts game

$N$ actions


**For $t = 1, 2, \ldots$**

1. Loss $\ell_t(i) \in [0,1]$ is assigned to every action $i = 1, \ldots, N$ (hidden from the player)

# Playing the experts game

$N$ actions

? ? ? ? ? ? ? ? ?

**For $t = 1, 2, \ldots$**

1. Loss $\ell_t(i) \in [0,1]$ is assigned to every action $i = 1, \ldots, N$ (hidden from the player)
2. Player picks an action $I_t$ (possibly using randomization) and incurs loss $\ell_t(I_t)$

# Playing the experts game

$N$ actions

7 3 2 4 1 6 7 4 9

**For $t = 1, 2, \ldots$**

1. Loss $\ell_t(i) \in [0,1]$ is assigned to every action $i = 1, \ldots, N$ (hidden from the player)
2. Player picks an action $I_t$ (possibly using randomization) and incurs loss $\ell_t(I_t)$
3. Player gets feedback information: $\ell_t(1), \ldots, \ell_t(N)$

# Oblivious opponents

Losses $\ell_t(1), \dots, \ell_t(N)$ for all $t = 1, 2, \dots$ are fixed beforehand, and unknown to the (randomized) player

## Oblivious regret minimization

$$R_T \stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(I_t)\right] - \min_{i=1,\dots,N} \sum_{t=1}^{T} \ell_t(i) \stackrel{\text{want}}{=} o(T)$$

## Lower bound using random losses

- $\ell_t(i) \to L_t(i) \in \{0,1\}$ independent random coin flip
- For any player strategy $\mathbb{E}\left[\sum_{t=1}^{T} L_t(I_t)\right] = \frac{T}{2}$
- Then the expected regret is

$$\mathbb{E}\left[\max_{i=1,\dots,N} \sum_{t=1}^{T} \left(\frac{1}{2} - L_t(i)\right)\right] = \big(1 - o(1)\big)\sqrt{\frac{T \ln N}{2}}$$

# Exponentially weighted forecaster

At time $t$ pick action $I_t = i$ with probability proportional to

$$\exp\left(-\eta \sum_{s=1}^{t-1} \ell_s(i)\right)$$

the sum at the exponent is the total loss of action $i$ up to now

## Regret bound

[Experts' paper, 1997]

- If $\eta = \sqrt{(\ln N)/(8T)}$ then $\boxed{R_T \leqslant \sqrt{\frac{T \ln N}{2}}}$
- Matching lower bound including constants
- Dynamic choice $\eta_t = \sqrt{(\ln N)/(8t)}$ only loses small constants

# The bandit problem: playing an unknown game

$N$ actions


**For $t = 1, 2, \ldots$**

1. Loss $\ell_t(i) \in [0,1]$ is assigned to every action $i = 1, \ldots, N$ (hidden from the player)

# The bandit problem: playing an unknown game

$N$ actions


**For $t = 1, 2, \dots$**

1. Loss $\ell_t(i) \in [0,1]$ is assigned to every action $i = 1, \dots, N$ (hidden from the player)
2. Player picks an action $I_t$ (possibly using randomization) and incurs loss $\ell_t(I_t)$

# The bandit problem: playing an unknown game

$N$ actions

? 3 ? ? ? ? ? ? ?

**For $t = 1, 2, \dots$**

1. Loss $\ell_t(i) \in [0,1]$ is assigned to every action $i = 1, \dots, N$ (hidden from the player)
2. Player picks an action $I_t$ (possibly using randomization) and incurs loss $\ell_t(I_t)$
3. Player gets feedback information: Only $\ell_t(I_t)$ is revealed

# The bandit problem: playing an unknown game

$N$ actions

? 3 ? ? ? ? ? ? ?

**For $t = 1, 2, \ldots$**

1. Loss $\ell_t(i) \in [0,1]$ is assigned to every action $i = 1, \ldots, N$ (hidden from the player)
2. Player picks an action $I_t$ (possibly using randomization) and incurs loss $\ell_t(I_t)$
3. Player gets feedback information: Only $\ell_t(I_t)$ is revealed

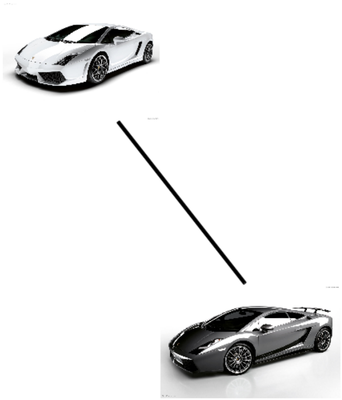
**Many applications**

Ad placement, dynamic content adaptation, routing, online auctions

# Summary

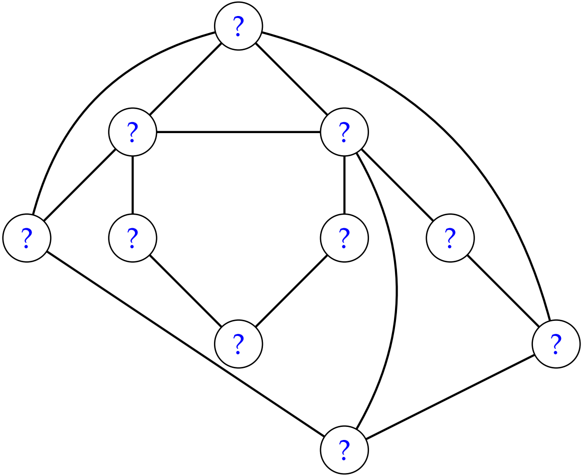
# Relationships between actions [Mannor and Shamir, 2011]

Undirected

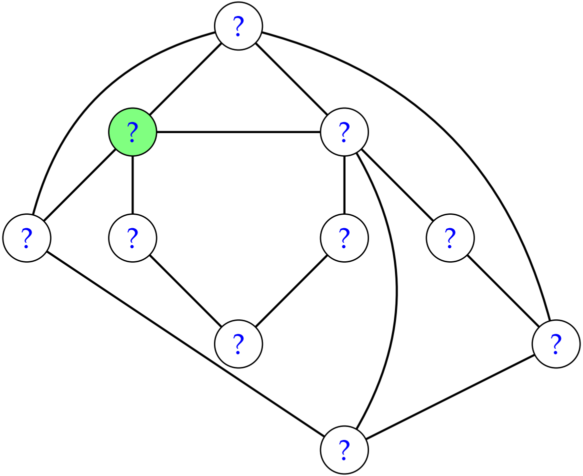


Directed

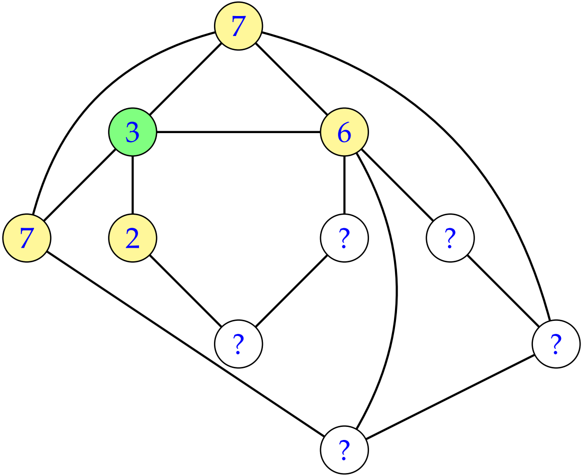
# A graph of relationships over actions



# A graph of relationships over actions

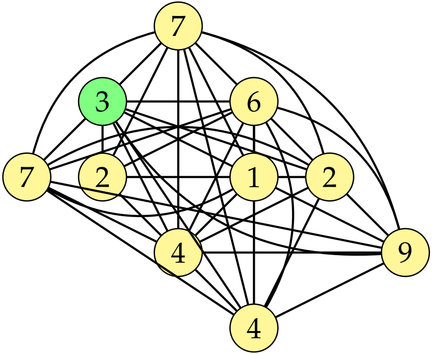


# A graph of relationships over actions

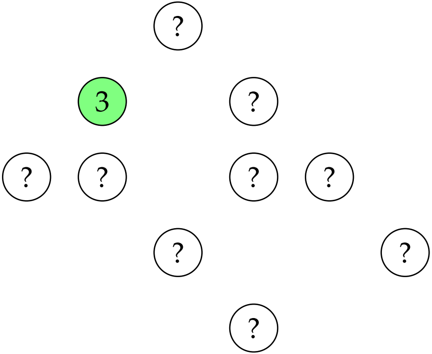


# Recovering expert and bandit settings

Experts: clique




Bandits: empty graph



# Exponentially weighted forecaster — Reprise

**Player's strategy** [Alon, C-B, Gentile, Mannor, Mansour and Shamir, 2013]

- $\mathbb{P}_t(I_t = i) \quad \propto \quad \exp\left(-\eta \sum_{s=1}^{t-1} \widehat{\ell}_s(i)\right) \qquad i = 1, \dots, N$
- $\widehat{\ell}_t(i) = \begin{cases} \dfrac{\ell_t(i)}{\mathbb{P}_t\big(\ell_t(i) \text{ observed}\big)} & \text{if } \ell_t(i) \text{ is observed} \\ 0 & \text{otherwise} \end{cases}$

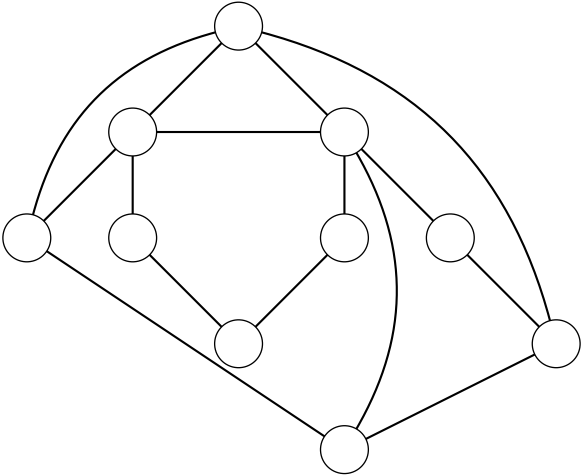
**Importance sampling estimator**

$$\mathbb{E}_t\left[\widehat{\ell}_t(i)\right] = \ell_t(i) \qquad \text{unbiased}$$

$$\mathbb{E}_t\left[\widehat{\ell}_t(i)^2\right] \leqslant \frac{1}{\mathbb{P}_t\big(\ell_t(i) \text{ observed}\big)} \qquad \text{variance control}$$

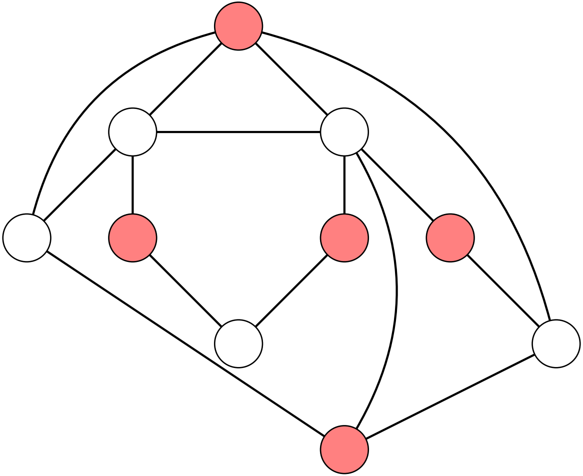
# Independence number $\alpha(G)$

The size of the largest independent set



# Independence number $\alpha(G)$

The size of the largest independent set



# Regret bounds

## Analysis (undirected graphs)

$$R_T \leqslant \frac{\ln N}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \overbrace{\sum_{i=1}^{N} \mathbb{P}\big(I_t = i \mid \ell_t(i) \text{ observed}\big)}^{\leqslant \alpha(G)}$$

$$= \sqrt{\alpha(G) T \ln N} \qquad \text{by tuning } \eta$$

If graph is directed, then bound worsens only by log factors

## Special cases

- Experts: $\alpha(G) = 1$ $\quad R_T \leqslant \sqrt{T \ln N}$
- Bandits: $\alpha(G) = N$ $\quad R_T \leqslant \sqrt{TN \ln N}$

# Reactive opponents [Dekel, Koren and Peres, 2014]

The loss of action $i$ at time $t$ depends on the player's past $m$ actions
$\ell_t(i) \to L_t(I_{t-m}, \ldots, I_{t-1}, i)$

## Adaptive regret

$$R_T^{\mathrm{ada}} = \mathbb{E}\left[\sum_{t=1}^{T} L_t(I_{t-m}, \ldots, I_{t-1}, I_t) - \min_{i=1,\ldots,K} \sum_{t=1}^{T} L_t(\underbrace{i, \ldots, i}_{m \text{ times}}, i)\right]$$

## Minimax adaptive regret (for any constant $m > 1$)

$$R_T^{\mathrm{ada}} = \Theta\left(T^{2/3}\right)$$

# Partial monitoring: not observing any loss

## Dynamic pricing: Perform as the best fixed price

1. Post a T-shirt price
2. Observe if next customer buys or not
3. Adjust price

Feedback does not reveal the player's loss

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 2 | 3 | 4 |
| 2 | c | 0 | 1 | 2 | 3 |
| 3 | c | c | 0 | 1 | 2 |
| 4 | c | c | c | 0 | 1 |
| 5 | c | c | c | c | 0 |

Loss matrix

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 1 | 1 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 0 | 0 | 1 | 1 |
| 5 | 0 | 0 | 0 | 0 | 1 |

Feedback matrix

# A characterization of minimax regret

**Special case**

Multiarmed bandits: loss and feedback matrix are the same

**A general gap theorem** [Bartok, Foster, Pál, Rakhlin and Szepesvári, 2013]

- A constructive characterization of the minimax regret for any pair of loss/feedback matrix
- Only three possible rates for nontrivial games:
  1. Easy games (e.g., bandits): $\Theta(\sqrt{T})$
  2. Hard games (e.g., revealing action): $\Theta(T^{2/3})$
  3. Impossible games: $\Theta(T)$

# Summary

# A game equivalent to prediction with expert advice

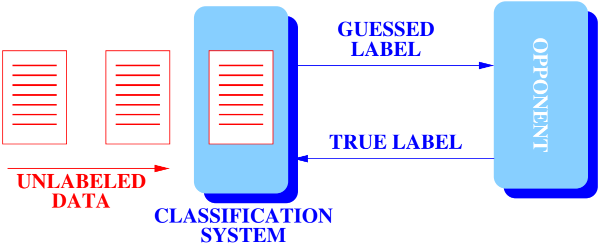
## Online linear optimization in the simplex

1. Play point $\mathbf{p}_t$ from the $N$-dimensional simplex $\Delta_N$
2. Incur linear loss $\mathbb{E}\big[\ell_t(I_t)\big] = \mathbf{p}_t^\top \widetilde{\boldsymbol{\ell}}_t$
3. Observe loss gradient $\widetilde{\boldsymbol{\ell}}_t$

## Regret: compete against the best point in the simplex

$$\sum_{t=1}^{T} \mathbf{p}_t^\top \widetilde{\boldsymbol{\ell}}_t - \underbrace{\min_{\mathbf{q}\in\Delta_N} \sum_{t=1}^{T} \mathbf{q}^\top \widetilde{\boldsymbol{\ell}}_t}_{= \min_{i=1,\dots,N} \frac{1}{T} \sum_{t=1}^{T} \widetilde{\ell}_t(i)}$$

# From game theory to machine learning




- Opponent's moves $y_t$ are viewed as values or labels assigned to observations $x_t \in \mathbb{R}^d$ (e.g., categories of documents)
- A repeated game between the player choosing an element $\boldsymbol{w}_t$ of a linear space and the opponent choosing a label $y_t$ for $\boldsymbol{x}_t$
- Regret with respect to best element in the linear space

# Online convex optimization [Zinkevich, 2003]

1. Play point $w_t$ from a convex linear space $S$
2. Incur convex loss $\ell_t(w_t)$
3. Observe loss gradient $\nabla \ell_t(w_t)$
4. Update point: $w_t \to w_{t+1} \in S$

## Example

- Regression with square loss: $\ell_t(w) = \left(w^\top x_t - y_t\right)^2 \quad y_t \in \mathbb{R}$
- Classification with hinge loss: $\ell_t(w) = \left[1 - y_t\, w^\top x_t\right]_+$
  $y_t \in \{-1, +1\}$

## Regret

$$\sum_{t=1}^{T} \ell_t(w_t) - \inf_{u \in S} \sum_{t=1}^{T} \ell_t(u)$$

# Finding a good online algorithm

## Follow the leader

$$\boldsymbol{w}_{t+1} = \operatorname*{arginf}_{\boldsymbol{w}\in S} \sum_{s=1}^{t} \ell_s(\boldsymbol{w})$$

Regret can be linear due to lack of stability

## Example

$$S = [-1,+1] \qquad \ell_1(w) = 1 + \frac{w}{2} \qquad \ell_t(w) = \begin{cases} -w & \text{if } t \text{ is even} \\ +w & \text{if } t \text{ is odd} \end{cases}$$

# Regularized online learning

## Strong convexity

$\Phi : S \to \mathbb{R}$ is $\beta$-strongly convex w.r.t. a norm $\|\cdot\|$ if for all $\boldsymbol{u}, \boldsymbol{v} \in S$

$$\Phi(\boldsymbol{v}) \geqslant \Phi(\boldsymbol{u}) + \nabla\Phi(\boldsymbol{u})^\top(\boldsymbol{v} - \boldsymbol{u}) + \frac{\beta}{2}\|\boldsymbol{u} - \boldsymbol{v}\|^2$$

Example: $\Phi(\boldsymbol{v}) = \frac{1}{2}\|\boldsymbol{v}\|^2$

## Follow the regularized leader

[Shalev-Shwartz, 2007; Abernethy, Hazan and Rakhlin, 2008]

$$\boldsymbol{w}_{t+1} = \operatorname*{argmin}_{\boldsymbol{w} \in S}\left[\eta \sum_{s=1}^{t} \ell_s(\boldsymbol{w}) + \Phi(\boldsymbol{w})\right]$$

$\Phi$ is a strongly convex regularizer defined on $S$

# Deriving an incremental update

## Linearization of convex losses

$$\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u}) \leqslant \underbrace{\nabla \ell_t(\boldsymbol{w}_t)}_{\widetilde{\boldsymbol{\ell}}_t}^\top \boldsymbol{w}_t - \underbrace{\nabla \ell_t(\boldsymbol{w}_t)}_{\widetilde{\boldsymbol{\ell}}_t}^\top \boldsymbol{u}$$

## Follow the regularized leader with linearized losses

$$\boldsymbol{w}_{t+1} = \operatorname*{argmin}_{\boldsymbol{w} \in S} \left( \eta \underbrace{\sum_{s=1}^{t} \widetilde{\boldsymbol{\ell}}_s}_{\theta_t}{}^\top \boldsymbol{w} + \Phi(\boldsymbol{w}) \right) = \operatorname*{argmax}_{\boldsymbol{w} \in S} \left( -\eta \theta_t^\top \boldsymbol{w} - \Phi(\boldsymbol{w}) \right)$$

$$= \nabla \Phi^*\left(-\eta\, \theta_t\right)$$

$\Phi^*$ is the convex dual of $\Phi$

# The Mirror Descent algorithm [Nemirovsky and Yudin, 1983]

**Recall:**

$$w_{t+1} = \nabla\Phi^*\big(-\eta\,\theta_t\big) = \nabla\Phi^*\left(-\eta\sum_{s=1}^{t}\nabla\ell_s(w_s)\right)$$

**Online Mirror Descent**

**Parameters:** Strongly convex regularizer $\Phi$ and learning rate $\eta > 0$

**Initialize:** $\theta_1 = \mathbf{0}$ `// primal parameter`

**For** $t = 1, 2, \ldots$

1. Use $\boxed{w_t = \nabla\Phi^*(\theta_t)}$ `// dual parameter (via mirror step)`
2. Suffer loss $\ell_t(w_t)$
3. Observe loss gradient $\nabla\ell_t(w_t)$
4. Update $\boxed{\theta_{t+1} = \theta_t - \eta\nabla\ell_t(w_t)}$ `// gradient step`

# Some examples

- Exponentiated gradient: $S = \Delta_N$ and $\Phi(\boldsymbol{w}) = \sum_{i=1}^{d} w_i \ln w_i$
  [Kivinen and Warmuth, 1997]
- Online Gradient Descent: $S = \mathbb{R}^d$ and $\Phi(\boldsymbol{w}) = \frac{1}{2} \|\boldsymbol{w}\|^2$
  [Zinkevich, 2003]
- p-norm Gradient Descent: $S = \mathbb{R}^d$ and $\Phi(\boldsymbol{w}) = \frac{1}{2(p-1)} \|\boldsymbol{w}\|_p^2$
  [Gentile, 2003]
- Matrix gradient descent
  [Cavallanti, C-B and Gentile, 2010; Kakade, Shalev-Shwartz and Tewari, 2012]

# General regret bound

Analysis relies on smoothness of $\Phi^*$ in order to bound increments $\Phi^*(\theta_{t+1}) - \Phi^*(\theta_t)$ via $\|\nabla\ell_t(\boldsymbol{w}_t)\|_*^2$

**Oracle bound** [Kakade, Shalev-Shwartz and Tewari, 2012]

$$\underbrace{\sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t)}_{\text{cumulative loss}} \leqslant \inf_{\boldsymbol{u}\in S}\left( \underbrace{\sum_{t=1}^{T} \ell_t(\boldsymbol{u})}_{\text{model fit}} + \underbrace{\frac{\Phi(\boldsymbol{u})}{\eta}}_{\text{model cost}} \right) + \frac{\eta}{2}\sum_{t=1}^{T} \frac{\|\nabla\ell_t(\boldsymbol{w}_t)\|_*^2}{\beta}$$

$\ell_1, \ell_2, \dots$ are arbitrary convex losses

- If gradients are bounded, then $R_T = \mathcal{O}\big(\sqrt{T}\big)$
- This is optimal for general convex losses $\ell_t$
- If all $\ell_t$ are strongly convex, then $R_T = \mathcal{O}\big(\ln T\big)$

# Regularization via stochastic smoothing

**Follow the perturbed leader** [Kalai and Vempala, 2005]

$$w_{t+1} = \mathbb{E}\left[\operatorname*{argmin}_{w \in S} \left(\eta \theta_t^\top w + Z^\top w\right)\right]$$

- The distribution of $Z$ must be "stable" (small variance and small average sensitivity)
- For some choices of $Z$, FPL becomes equivalent to OMD [Abernethy, Lee, Sinha and Tewari, 2014]

# Adaptive regularization

## Online Ridge Regression [Vovk, 2001; Azoury and Warmuth, 2001]

$$\sum_{t=1}^{T} \left( w_t^\top x_t - y_t \right)^2 \leqslant \inf_{u \in \mathbb{R}^d} \left( \sum_{t=1}^{T} \left( u^\top x_t - y_t \right)^2 + \|u\|^2 \right) + d \ln \left( 1 + \frac{T}{d} \right)$$

$$\Phi_t(w) = \frac{1}{2} \|w\|_{A_t}^2 \qquad A_t = I + \sum_{s=1}^{t} x_s x_s^\top$$

## More examples

- Online Newton Step [Hazan, Agarwal and Kale, 2007]
  Logarithmic regret for exp-concave loss functions
- AdaGrad [Duchi, Hazan and Singer, 2010]
  Competitive with "optimal" fixed regularizer
- Scale-invariant algorithms [Ross, Mineiro and Langford, 2013]
  Regret invariant w.r.t. rescaling of single features

# Shifting regret [Herbster and Warmuth, 2001]

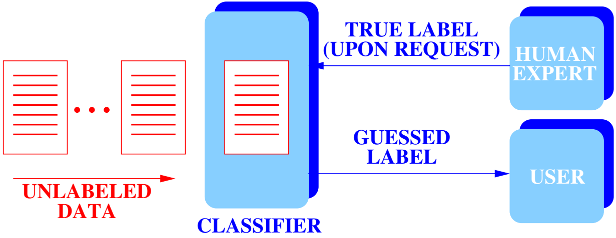
## Nonstationarity

- If data source is not fitted well by any model in the class, then comparing to the best model $\mathbf{u} \in S$ is trivial
- Compare instead to the best sequence $\mathbf{u}_1, \mathbf{u}_2, \dots \in S$ of models

## Shifting Regret for Online Mirror Descent [Zinkevich, 2003]

$$\underbrace{\sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t)}_{\text{cumulative loss}} \leqslant \inf_{\mathbf{u}_1,\dots,\mathbf{u}_T \in S} \underbrace{\sum_{t=1}^{T} \ell_t(\mathbf{u}_t)}_{\text{model fit}} + \underbrace{\sum_{t=1}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|}_{\text{shifting model cost}} + \operatorname{diam}(S) + \Box$$

# Online active learning

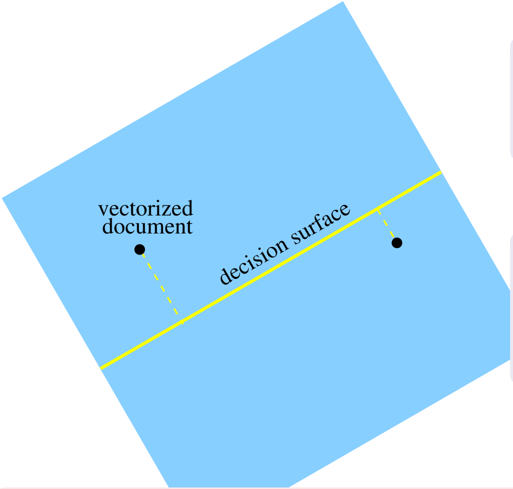



- Observing the data process is cheap
- Observing the label process is expensive
  $\rightarrow$ need to query the human expert

## Question

How much better can we do by subsampling adaptively the label process?

# A game with the opponent [C-B, Gentile, Zaniboni, 2006]

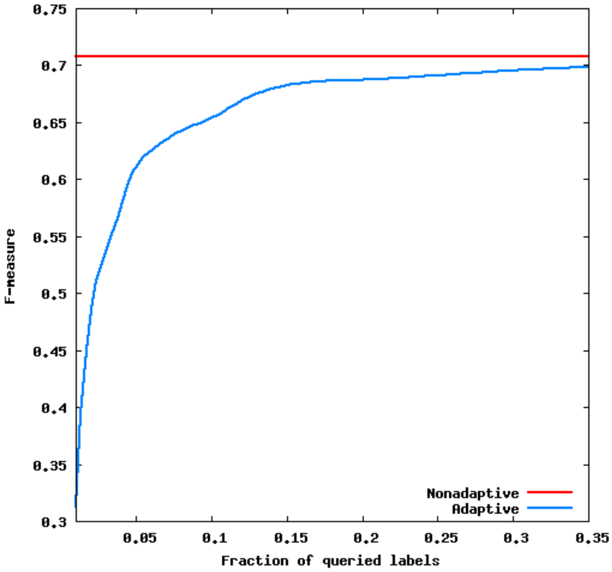



Opponent avoids causing mistakes on documents far away from decision surface

Probability of querying a document proportional to inverse distance to decision surface

Binary classification performance guarantee remains identical (in expectation) to the full sampling case

# Experiments on document categorization



# Stochastic Online Mirror Descent

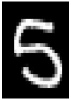
**Parameters:** Strongly convex regularizer $\Phi$ and learning rate $\eta > 0$
**Initialize:** $\theta_1 = \mathbf{0}$ `// primal parameter`

**For** $t = 1, 2, \dots$

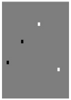
1. Use $\boldsymbol{w}_t = \nabla\Phi^*(\theta_t)$ `// mirror step with projection on S`
2. Suffer loss $\ell_t(\boldsymbol{w}_t)$
3. Compute estimate $\widehat{\boldsymbol{g}}_t$ of loss gradient $\nabla\ell_t(\boldsymbol{w}_t)$
4. Update $\theta_{t+1} = \theta_t - \eta\widehat{\boldsymbol{g}}_t$ `// gradient step`

Typically, $\Phi(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$ (stochastic OGD)

# Attribute efficient learning [C-B, Shamir, Shalev-Shwartz, 2011]



original



sampled

Obtain a few attributes
from each training example

- Use Stochastic OGD with square loss $\ell_t(\boldsymbol{w}) = \frac{1}{2}\left(\boldsymbol{w}^\top \boldsymbol{x}_t - y_t\right)^2$
- $\nabla \ell_t(\boldsymbol{w}) = \left(\boldsymbol{w}^\top \boldsymbol{x}_t - y_t\right)\boldsymbol{x}_t$

## Unbiased estimate of square loss gradient using two attributes

1. Estimate of $\boldsymbol{w}^\top \boldsymbol{x}$: query $x_i$ according to $p(i) = \dfrac{|w_i|}{\|\boldsymbol{w}\|_1}$
2. Estimate of $\boldsymbol{x}$: query $x_j$ uniformly at random
3. Gradient estimate: $\widehat{g} = \Big(\|\boldsymbol{w}\|_1 \operatorname{sgn}(w_i)\, x_i - y\Big) d\, x_j \boldsymbol{e}_j$

# Summary

1. My beautiful regret
2. A supposedly fun game I'll play again
3. A graphic novel
4. The joy of convex
5. The joy of convex (without the gradient)

# Online convex optimization with bandit feedback

**For $T = 1, 2, \dots$**

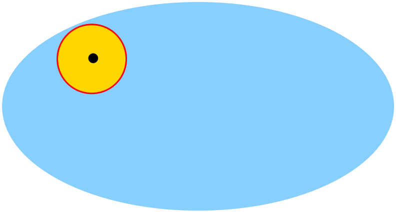
1. Play point $\boldsymbol{w}_t$ from a convex linear space $S$
2. Incur and observe convex loss $\ell_t(\boldsymbol{w}_t)$
3. Update point: $\boldsymbol{w}_t \to \boldsymbol{w}_{t+1} \in S$

**Regret**

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t)\right] - \inf_{\boldsymbol{u} \in S} \sum_{t=1}^{T} \ell_t(\boldsymbol{u})$$

# Gradient descent without a gradient
[Flaxman, Kalai and McMahan, 2004]

- Run stochastic OGD using a perturbed version of $\boldsymbol{w}_t$: $\boldsymbol{w}_t + \delta\,\mathbf{U}$ ($\mathbf{U}$ is a random unit vector and $\delta > 0$)
- Gradient estimate $\widehat{g}_t = \dfrac{d}{\delta}\,\ell_t(\boldsymbol{w}_t + \delta\,\mathbf{U})\,\mathbf{U}$
- Fact (Stokes' Theorem): If $\ell_t$ were differentiable, then
$$\mathbb{E}\big[\widehat{g}_t\big] = \nabla\mathbb{E}\big[\ell_t(\boldsymbol{w}_t + \delta\,\mathbf{B})\big]$$
where $\mathbf{B}$ is a random vector in the unit ball



$\widehat{g}_t$ estimates the gradient of a locally smoothed version of $\ell_t$

# Guarantees

- If $\ell_t$ is Lipschitz, then the smoothed version is a good approximation of $\ell_t$
- Radius $\delta$ of perturbation controls bias/variance trade-off

**Regret of stochastic OGD for convex and Lipschitz loss sequences**

$$R_T = \mathcal{O}\left(T^{3/4}\right)$$

# Guarantees

- If $\ell_t$ is Lipschitz, then the smoothed version is a good approximation of $\ell_t$
- Radius $\delta$ of perturbation controls bias/variance trade-off

**Regret of stochastic OGD for convex and Lipschitz loss sequences**

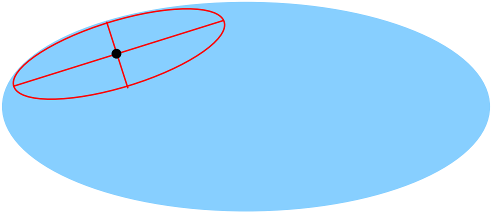
$$R_T = \mathcal{O}\left(T^{3/4}\right)$$

**The linear case**

- Assume losses are linear functions on $S$, $\ell_t(\boldsymbol{w}) = \boldsymbol{\ell}_t^\top \boldsymbol{w}$
- Can we achieve a better rate?

# Self-concordant functions [Abernethy, Hazan and Rakhlin, 2008]

- Run stochastic OGD regularized with a self-concordant function for $S$
- Variance control through the associated Dikin ellipsoid
- Loss estimate $\widehat{\ell}_t$ obtained via perturbed point $W_t \pm \boldsymbol{e}_i \sqrt{\lambda_i}$
  $\{\boldsymbol{e}_i, \lambda_i\}$ is a randomly drawn eigenvector-eigenvalue pair of Dikin ellipsoid



**Regret for linear functions**

$$R_T = \mathcal{O}\left(d^{3/2}\sqrt{T \ln T}\right)$$

- Build an $\varepsilon$-cover $S_0 \subseteq S$ of size $\varepsilon^{-d}$
- Use experts algorithm (e.g., exponential weights) to draw actions $W_t \in S_0$ and use unbiased linear estimator for the loss

$$\widehat{\ell}_t = P_t^{-1} W_t \boxed{W_t^\top \ell_t} \qquad \text{where} \qquad P_t = \mathbb{E}\big[W_t W_t^\top\big]$$

- Mix exponential weights with exploration distribution $\mu$ over the actions in the cover:

$$p_t(\boldsymbol{w}) = (1-\gamma)\underbrace{q_t(\boldsymbol{w})}_{\text{exp. distrib.}} + \boxed{\gamma\,\mu(\boldsymbol{w})} \qquad (0 \leqslant \gamma \leqslant 1)$$

- $\mu$ controls the variance of the loss estimates by ensuring all directions are sampled often enough

## Regret bound

$$R_T = \mathcal{O}\left(d\sqrt{\left(\frac{1}{d\,\lambda_{\min}}+1\right)T\ln T}\right)$$

$\lambda_{\min}$ = smallest eigenvalue of $\mathbb{E}_{\mu}\left[W W^{\top}\right]$

- $\lambda_{\min}^{-1}$ is proportional to the variance of loss estimates
- When $\lambda_{\min} \approx \frac{1}{d}$ we get the optimal bound $\Theta\left(d\sqrt{T\ln T}\right)$
- If $\mu$ is uniform over all actions, the above happens when action space is approximately isotropic

# Good exploration bases [Bubeck, C-B and Kakade, 2012]

## Choose a basis under which the action set looks isotropic

- There are at most $\mathcal{O}(d^2)$ contact points between $S_0$ and Löwner ellipsoid (the min volume ellipsoid enclosing $S_0$)
- Put exploration distribution $\mu$ on these contact points
- This ensures that $\mathbb{E}_\mu\left[W W^\top\right]$ is isotropic: $\lambda_{\min} = \frac{1}{d}$

- Exploration on contact points of Löwner ellipsoid achieves optimal regret

$$R_T = \mathcal{O}\left(d\sqrt{T \ln T}\right)$$

- However, this construction is not efficient in general
- An efficient construction uses volumetric ellipsoids [Hazan, Gerber and Meka, 2014]

# Conclusions

## More applications

- Portfolio management
- Matrix completion
- Competitive analysis of algorithms
- Recommendation systems

## Some open problems

- Exact rates for bandit convex optimization
- Trade-offs between regret bounds and running times
- Online tensor and spectral learning
- Problems with states