Exponential weights

Expert prediction

Suppose $b_t \in [0,1]^d$ is a vector of **d** experts predictions of tomorrow's temperature.

	t=1	*t*=2	*t*=3	*t*=4	*t*=5	…
Expert 1						
Expert 2						
Expert 3						

Expert prediction

Suppose $b_t \in [0,1]^d$ is a vector of **d** experts predictions of tomorrow's temperature.

	t=1	*t=2*	*t=3*	*t=4*	*t=5*	…
Expert 1						
Expert 2						
Expert 3						

$$z_t(i) = |b_t(i) - y_t|$$

*i*th expert's prediction

True temperature

Input: d experts
for $t = 1, 2, \ldots$
    Player picks $p_t \in \Delta_d$ and plays $I_t \sim p_t$
    Adversary simultaneously reveals expert losses $z_t \in [0,1]^d$
    Player pays loss $\langle p_t, z_t \rangle = \mathbb{E}[z_t(I_t)]$

Expert prediction

Suppose $b_t \in [0,1]^d$ is a vector of **d** experts predictions of tomorrow's temperature.

	t=1	*t*=2	*t*=3	*t*=4	*t*=5	...
Expert 1						
Expert 2						
Expert 3						

$$z_t(i) = |b_t(i) - y_t|$$

*i*th expert's prediction

True temperature

$\texttt{Input:}$ d experts
$\texttt{for}$ $t = 1, 2, \ldots$
Player picks $p_t \in \triangle_d$ and plays $I_t \sim p_t$
Adversary simultaneously reveals expert losses $z_t \in [0,1]^d$
Player pays loss $\langle p_t, z_t \rangle = \mathbb{E}[z_t(I_t)]$

Goal: Minimize regret wrt best $\max_{i \in [d]} \sum_{t=1}^{T} \langle p_t, z_t \rangle - \langle \mathbf{e}_i, z_t \rangle$

Expert prediction

Goal: Minimize regret wrt best $\max_{i \in [d]} \sum_{t=1}^{T} \langle p_t, z_t \rangle - \langle \mathbf{e}_i, z_t \rangle$

`Input:` d experts
`for` $t = 1, 2, \ldots$
  Player picks $p_t \in \triangle_d$ and plays $I_t \sim p_t$
  Adversary simultaneously reveals expert losses $z_t \in [0,1]^d$
  Player pays loss $\langle p_t, z_t \rangle = \mathbb{E}[z_t(I_t)]$

Exponential weights algorithm

`Input:` d experts, $\eta > 0$
`Initialize:` $w_1 \in [1, \ldots, 1]^\top \in \mathbb{R}^d$
`for` $t = 1, 2, \ldots$
  Player plays $I_t \sim p_t$ where $p_t(i) = w_t(i) / \sum_{j=1}^{d} w_t(j)$
  Adversary simultaneously reveals expert losses $z_t \in [0,1]^d$
  Player pays loss $\langle p_t, z_t \rangle = \mathbb{E}[z_t(I_t)]$
  Player updates weights $w_{t+1}(i) = w_t(i) \exp(-\eta z_t(i))$

Expert prediction

Goal: Minimize regret wrt best $\max_{i\in[d]} \sum_{t=1}^{T} \langle p_t, z_t \rangle - \langle \mathbf{e}_i, z_t \rangle$

Exponential weights algorithm

Input: d experts, $\eta > 0$

Initialize: $w_1 \in [1, \ldots, 1]^\top \in \mathbb{R}^d$

for $t = 1, 2, \ldots$

Player plays $I_t \sim p_t$ where $p_t(i) = w_t(i) / \sum_{j=1}^{d} w_t(j)$

Adversary simultaneously reveals expert losses $z_t \in [0,1]^d$

Player pays loss $\langle p_t, z_t \rangle = \mathbb{E}[z_t(I_t)]$

Player updates weights $w_{t+1}(i) = w_t(i) \exp(-\eta z_t(i))$

$\exp(\beta) \overset{?}{\geq} 1 + \beta$

$(1 \pm \beta)$

Theorem: If $z_t \in [0,1]^d \; \forall t$, and I_t, p_t are chosen by exponential weights then
$\max_{i\in[d]} \mathbb{E}\left[\sum_{t=1}^{T} \langle I_t, z_t \rangle - \langle \mathbf{e}_i, z_t \rangle\right] = \max_{i\in[d]} \sum_{t=1}^{T} \langle p_t, z_t \rangle - \langle \mathbf{e}_i, z_t \rangle \leq \frac{\log(d)}{\eta} + \frac{T\eta}{8}$

Choosing $\eta = \sqrt{\frac{8\log(d)}{T}}$ gives regret bound of $\sqrt{T\log(d)/2}$

Expert prediction

Goal: Minimize regret wrt best

$$\max_{i \in [d]} \sum_{t=1}^{T} \langle p_t, z_t \rangle - \langle \mathbf{e}_i, z_t \rangle$$

Exponential weights algorithm, proof: Let $W_t = \sum_{i=1}^{d} w_t(i)$ so that

Expert prediction

Goal: Minimize regret wrt best

$$\max_{i\in[d]} \sum_{t=1}^{T} \langle p_t, z_t\rangle - \langle \mathbf{e}_i, z_t\rangle$$

Exponential weights algorithm, proof: Let $W_t = \sum_{i=1}^d w_t(i)$ so that

$$\log \frac{W_{T+1}}{W_1} = \sum_{t=1}^T \log \frac{W_{t+1}}{W_t}$$
$$= \sum_{t=1}^T \log\left(\sum_{i=1}^d \frac{w_{t+1}(i)}{W_t}\right)$$
$$= \sum_{t=1}^T \log\left(\sum_{i=1}^d \frac{w_t(i)\exp(-\eta z_t(i))}{W_t}\right)$$
$$= \sum_{t=1}^T \log\left(\sum_{i=1}^d p_t(i)\exp(-\eta z_t(i))\right)$$
$$= \sum_{t=1}^T \log\left(\exp(-\eta \mathbb{E}[z_t(I_t)]) \sum_{i=1}^d p_t(i)\exp(-\eta(z_t(i) - \mathbb{E}[z_t(I_t)]))\right)$$
$$= \sum_{t=1}^T -\eta \mathbb{E}[z_t(I_t)] + \log\left(\mathbb{E}[\exp(-\eta(z_t(I_t) - \mathbb{E}[z_t(I_t)]))]\right)$$
$$\leq \sum_{t=1}^T -\eta \mathbb{E}[z_t(I_t)] + \eta^2/8$$

$$\log \frac{W_{T+1}}{W_1} \geq \log \frac{w_{T+1}(i)}{W_1}$$
$$= -\log(d) + \log\left(\prod_{t=1}^T \exp(-\eta z_t(i))\right)$$
$$= -\log(d) - \sum_{t=1}^T \eta z_t(i)$$

Lemma (Hoeffding's Lemma). Let X be a real-valued random variable such that $X \in [a, b]$ almost surely, and let $\mathbb{E}[X] = \mu$. Then, for any $t \in \mathbb{R}$,

$$\mathbb{E}\left[e^{t(X-\mu)}\right] \leq \exp\left(\frac{t^2(b-a)^2}{8}\right)$$

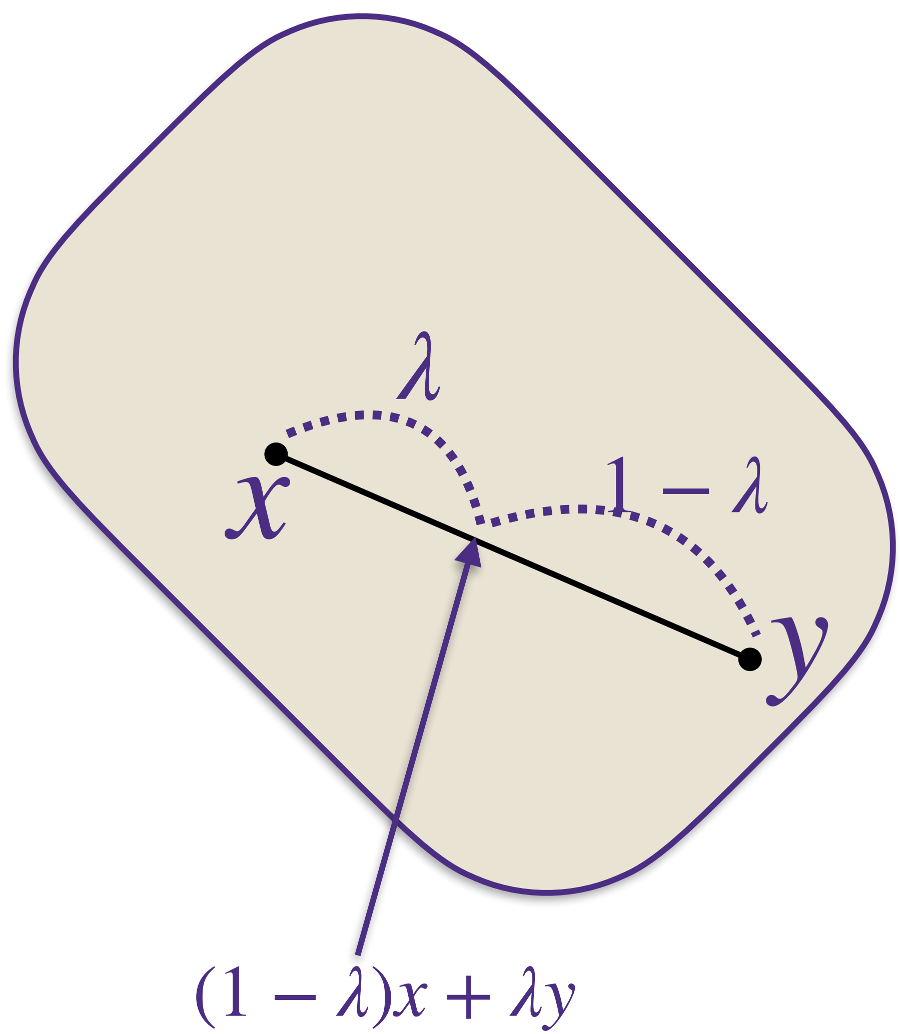
$$\implies \sum_{t=1}^T \eta \mathbb{E}[z_t(I_t)] - \sum_{t=1}^T \eta z_t(i) \leq \log(d) + \eta^2 T/8$$

Convexity

- When is an optimization (or learning) easy/fast to solve?

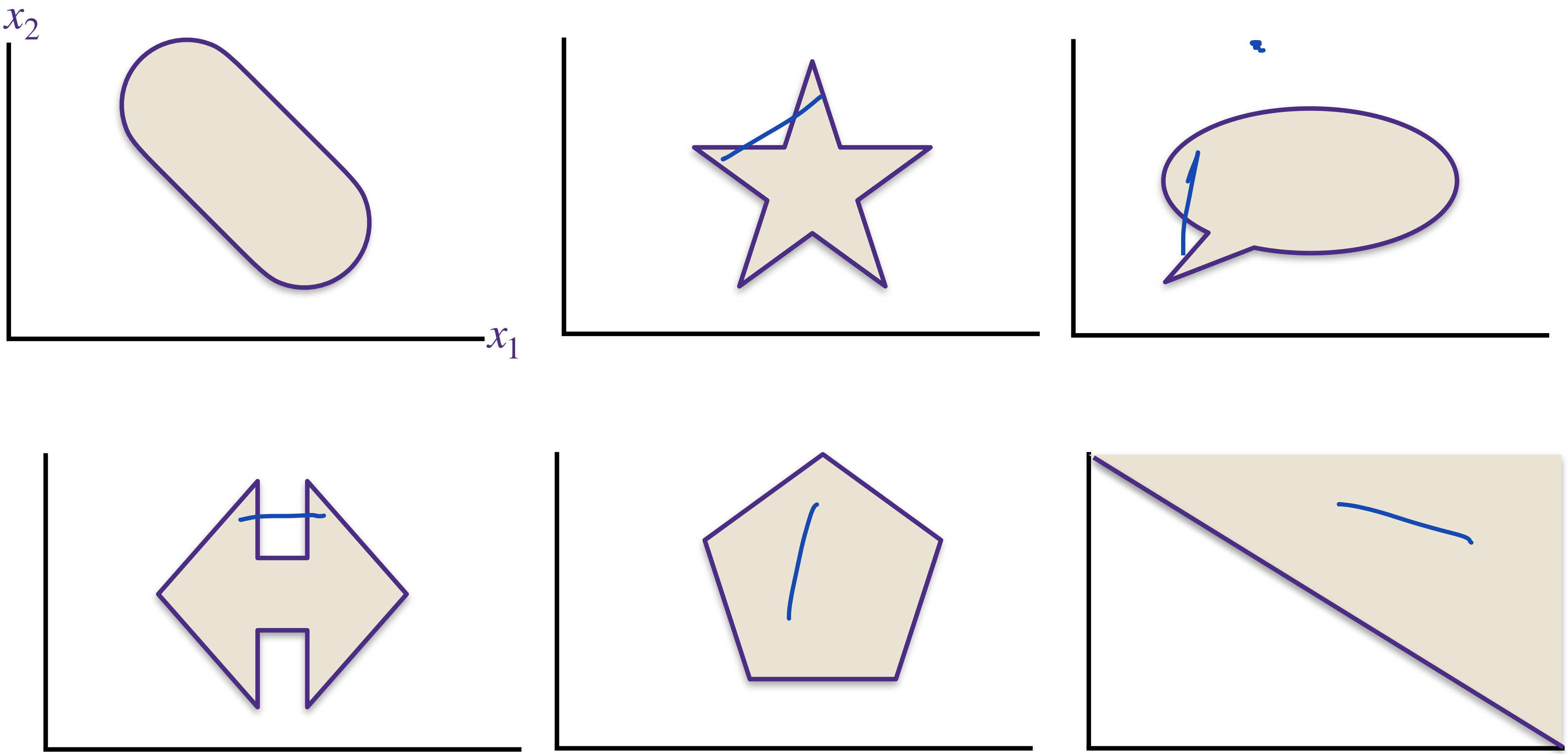
What is a convex set?

A set $K \subset \mathbb{R}^d$ is convex if $(1-\lambda)x + \lambda y \in K$ for all $x, y \in K$ and $\lambda \in [0,1]$



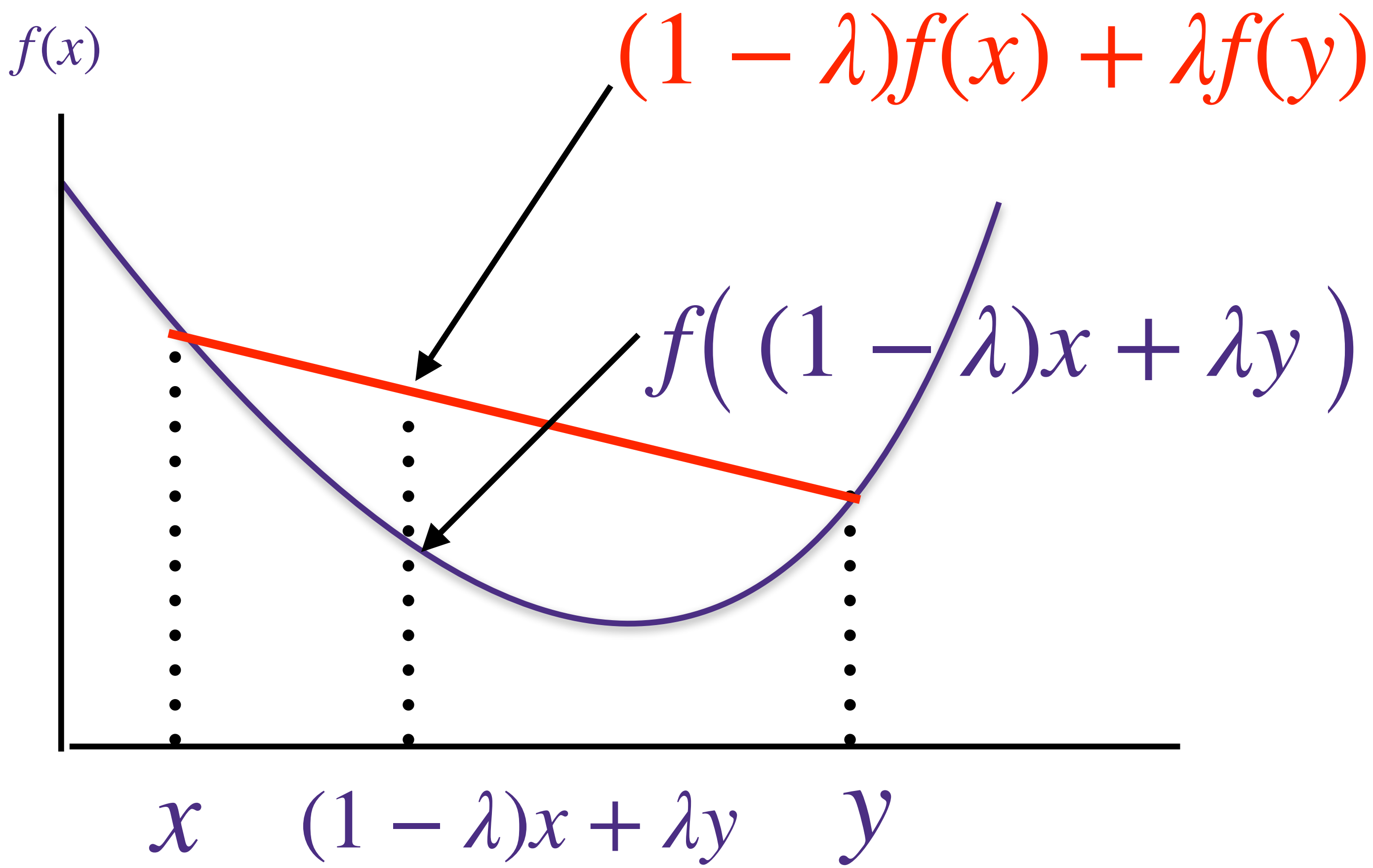
What is a convex set?

A set $K \subset \mathbb{R}^d$ is convex if $(1-\lambda)x + \lambda y \in K$ for all $x, y \in K$ and $\lambda \in [0,1]$



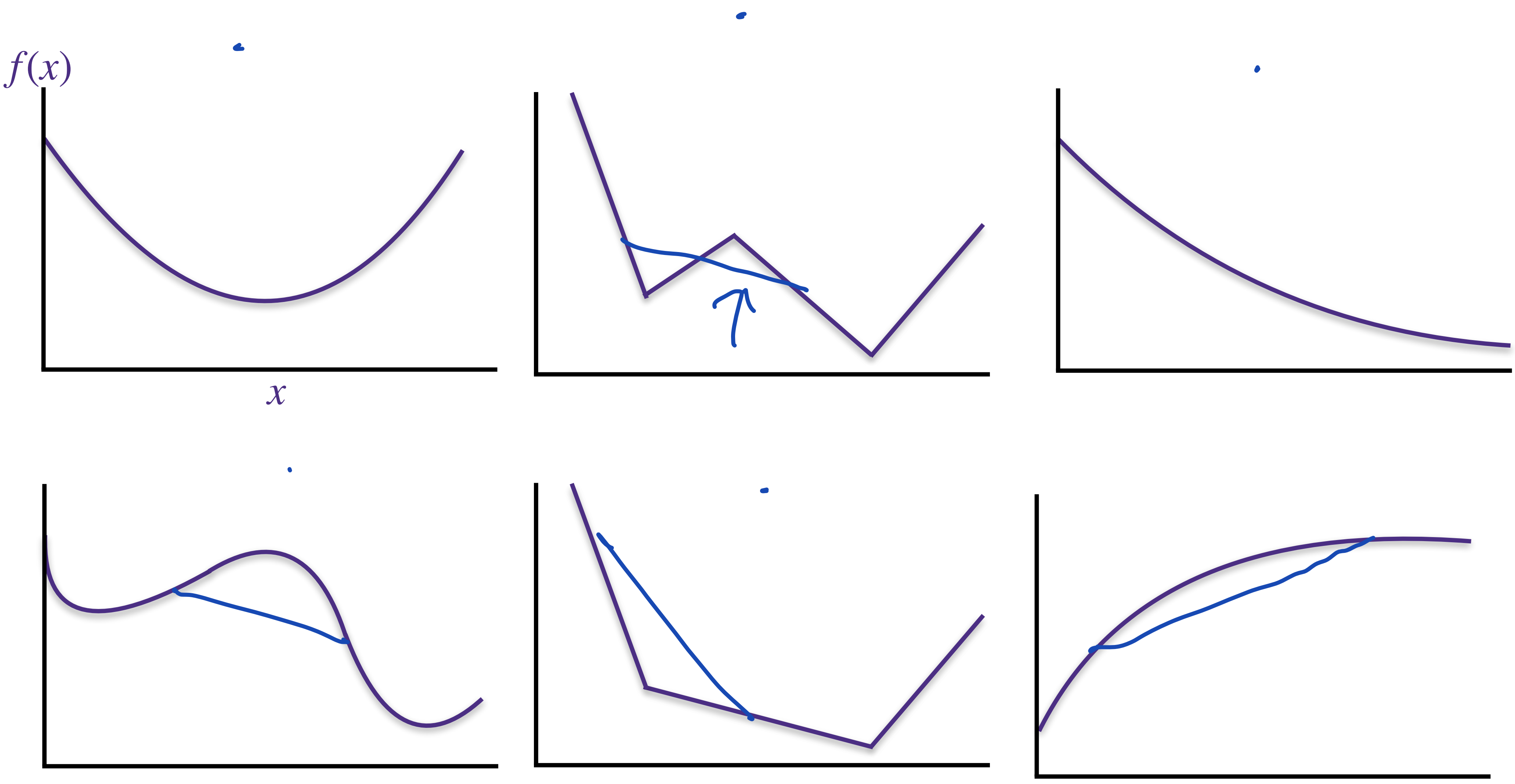
What is a convex function?

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if $f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y)$ for all $x, y \in \mathbb{R}^d$ and $\lambda \in [0,1]$



What is a convex function?

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if $f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y)$ for all $x, y \in \mathbb{R}^d$ and $\lambda \in [0,1]$



Convex functions and convex sets?

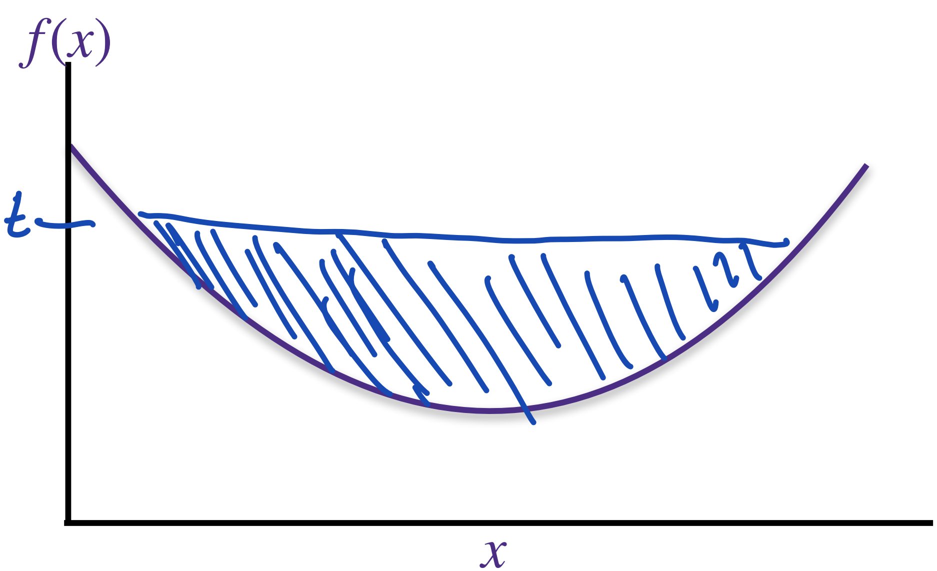
A set $K \subset \mathbb{R}^d$ is convex if $(1-\lambda)x + \lambda y \in K$ for all $x, y \in K$ and $\lambda \in [0,1]$

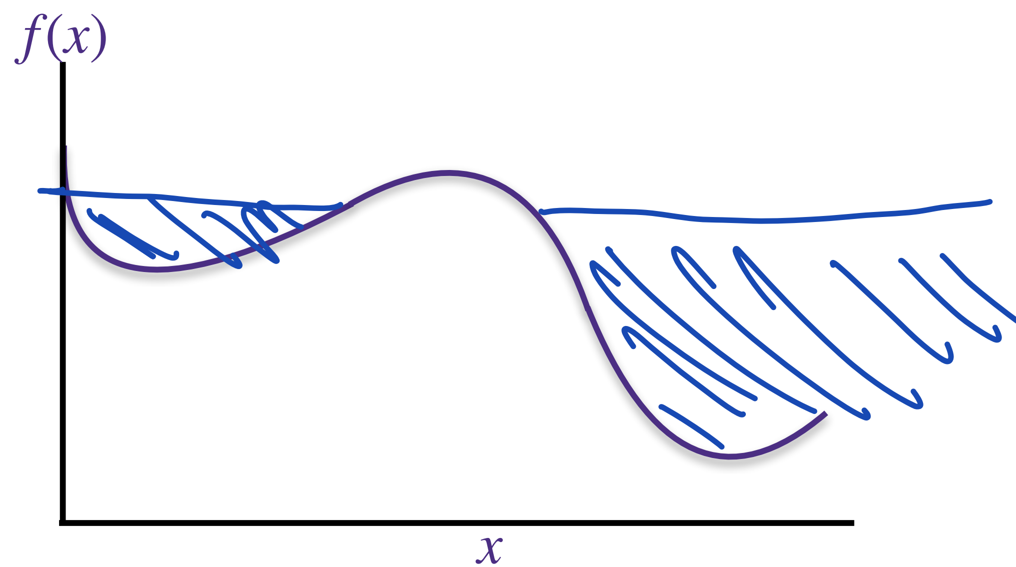
A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if $f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y)$ for all $x, y \in \mathbb{R}^d$ and $\lambda \in [0,1]$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if the set $\{(x,t) \in \mathbb{R}^{d+1} : f(x) \leq t\}$ is convex

Graph of f id defined as $\{(x,t) \; : \; f(x) = t\}$

Epigraph of f is defined as $\{(x,t) \; : \; f(x) \leq t\}$



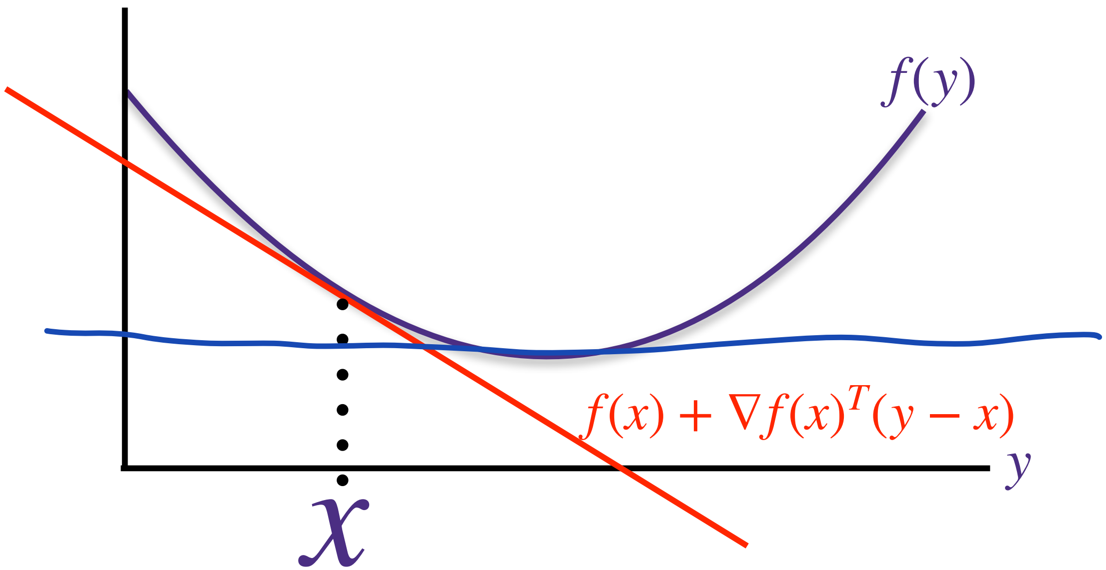


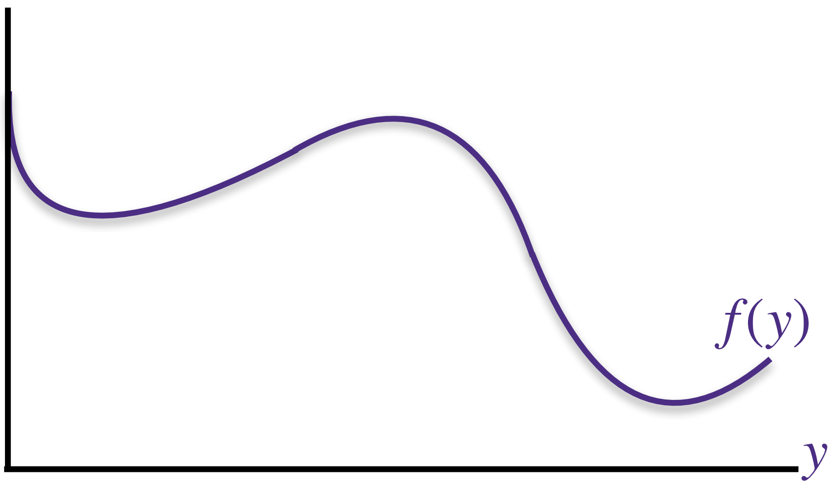
More definitions of convexity

A set $K \subset \mathbb{R}^d$ is convex if $(1-\lambda)x + \lambda y \in K$ for all $x, y \in K$ and $\lambda \in [0,1]$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if the set $\{(x,t) \in \mathbb{R}^{d+1} : f(x) \leq t\}$ is convex

A function $f : \mathbb{R}^d \to \mathbb{R}$ that is differentiable everywhere is convex if $f(y) \geq f(x) + \nabla f(x)^\top (y-x)$ for all $x, y \in dom(f)$



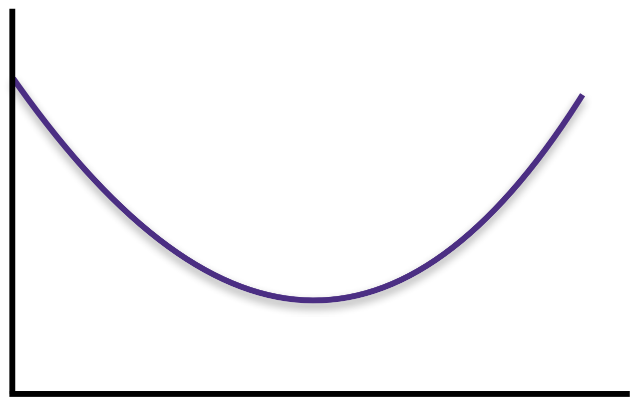


Why do we care about convexity?

Convex functions

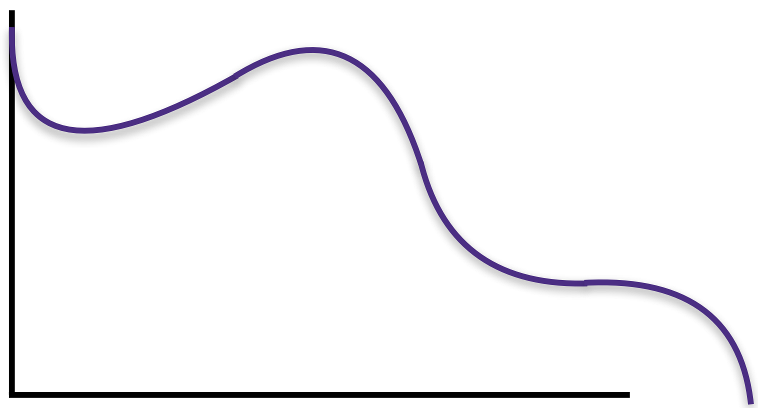
- All local minima are global minima
- Efficient to optimize (e.g., gradient descent)

Convex Function



We only need to find a point with $\nabla f(x) = 0$, which for convex functions implies that it is a local minima and a global minima

Non-convex Function



For non-convex functions, a stationary point with $\nabla f(x) = 0$ could be a local minima, a local maxima, or a saddle point

Online Convex Optimization

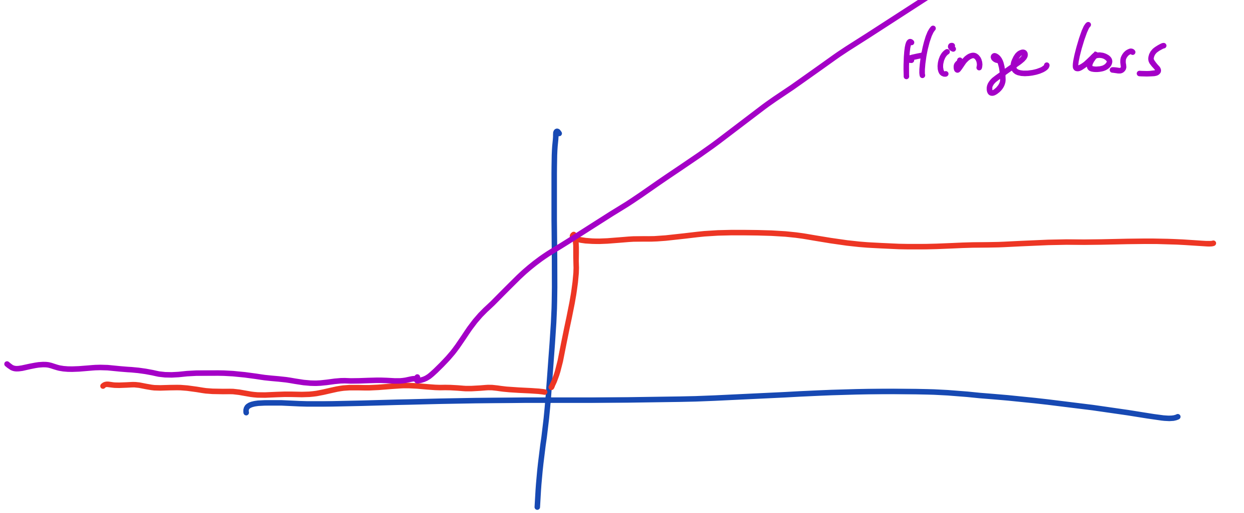
UNIVERSITY of WASHINGTON

Convex surrogate loss functions

Previous section for the adversarial case suggested using multiplicative weights over the |H| hypotheses, which is completely intractable in practice.

And in the stochastic case we used $h_t \in \arg\min_{h \in \mathcal{H}} \sum_{s=1}^{t-1} \mathbf{1}\{h(x_s) \neq y_s\}$ which is also intractable to compute!

So it seems we have no practical algorithm! Solution: relax the objective.



Convex surrogate loss functions

Previous section for the adversarial case suggested using multiplicative weights over the |H| hypotheses, which is completely intractable in practice.

And in the stochastic case we used $h_t \in \arg\min_{h \in \mathcal{H}} \sum_{s=1}^{t-1} \mathbf{1}\{h(x_s) \neq y_s\}$ which is also intractable to compute!

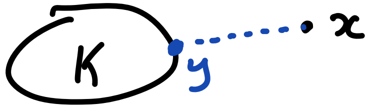
$\ell(h(x_s), y_s)$

So it seems we have no practical algorithm! Solution: relax the objective.

Instead of $$\max_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbf{1}\{h_t(x_t) \neq y_t\} - \mathbf{1}\{h(x_t) \neq y_t\}$$

We use $$\max_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell(h_t, (x_t, y_t)) - \ell(h, (x_t, y_t))$$ with $\mathcal{H}$ convex

Example: Linear classification takes $\mathcal{H} \subset \mathbb{R}^d$ and $\ell(h, (x_t, y_t)) = \log(1 + \exp(-y_t h^\top x_t))$



$\Pi_K(x) = \arg\min_{y \in K} \|y - x\|_2$

Convex surrogate loss functions

Goal: $\max_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell(h_t, (x_t, y_t)) - \ell(h, (x_t, y_t))$ with $\mathcal{H}$ convex

Online gradient descent

`Input:` $\mathcal{H} \subset \mathbb{R}^d$, convex loss function ℓ, step size $\eta > 0$

`Initialize:` Choose any $h_1 \in \mathcal{H}$

`for` $t = 1, 2, \ldots$

  Player plays $h_t \in \mathcal{H}$

  Adversary simultaneously reveals (x_t, y_t)

  Player pays loss $\ell_t(h_t) := \ell(h_t, (x_t, y_t))$

  Player updates $h_{t+1} = \Pi_{\mathcal{H}}(h_t - \eta \nabla_h \ell_t(h_t))$

Theorem Online gradient descent satisfies for any $h_* \in \mathcal{H}$

$$\sum_{t=1}^{T} \ell(h_t, (x_t, y_t)) - \ell(h_*, (x_t, y_t)) \leq \frac{\|h_*\|_2^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla_h \ell_t(h_t)\|_2^2$$

if $\max_{h \in \mathcal{H}} \|h_*\|_2 \leq R$ and $\ell(\cdot)$ is G-Lipschitz then regret $\leq RG\sqrt{T}$

Proof

$\leq \frac{R^2}{2\eta} + \frac{\eta G^2 T}{2}$

Theorem Online gradient descent satisfies for any $h_* \in \mathcal{H}$

$$\sum_{t=1}^T \ell(h_t, (x_t, y_t)) - \ell(h_*, (x_t, y_t)) \leq \frac{\|h_*\|_2^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^T \|\nabla_h \ell_t(h_t)\|_2^2$$

Proof

Theorem Online gradient descent satisfies for any $h_* \in \mathcal{H}$
$\sum_{t=1}^T \ell(h_t, (x_t, y_t)) - \ell(h_*, (x_t, y_t)) \le \frac{\|h_*\|_2^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^T \|\nabla_h \ell_t(h_t)\|_2^2$

$$
\begin{aligned}
\|h_{t+1} - h_*\|_2^2 &= \|\Pi_{\mathcal{H}}(h_{t+1}) - \Pi_{\mathcal{H}}(h_*)\|_2^2 \\
&= \|\Pi_{\mathcal{H}}(h_t - \eta \nabla \ell_t(h_t)) - \Pi_{\mathcal{H}}(h_*)\|_2^2 \\
&\le \|h_t - \eta \nabla \ell_t(h_t) - h_*\|_2^2 \\
&= \|h_t - h_*\|_2^2 - 2\eta \nabla \ell_t(h_t)^\top (h_t - h_*) + \eta^2 \|\nabla \ell_t(h_t)\|_2^2 \\
&\le \|h_t - h_*\|_2^2 - 2\eta \big(\ell_t(h_t) - \ell_t(h_*)\big) + \eta^2 \|\nabla \ell_t(h_t)\|_2^2
\end{aligned}
$$

$$
\begin{aligned}
\sum_{t=1}^T (\ell_t(h_t) - \ell_t(h_*)) &\le \sum_{t=1}^T \frac{\|h_t - h_*\|_2^2 - \|h_{t+1} - h_*\|_2^2}{2\eta} + \sum_{t=1}^T \frac{\eta}{2} \|\nabla \ell_t(h_t)\|_2^2 \\
&\le \frac{\|h_1 - h_*\|_2^2}{2\eta} + \sum_{t=1}^T \frac{\eta}{2} \|\nabla \ell_t(h_t)\|_2^2
\end{aligned}
$$

Universal Portfolio Optimization

UNIVERSITY *of* WASHINGTON

Universal Portfolio Optimization

Given a collection of stocks, let the ith stock have price $S_t(i)$ over time t.
You start with v_1 dollars and fractionally invest it into d stocks according to $p_1 \in \triangle_d$.

Your portfolio at time 2 is worth $v_2 := \sum_{i=1}^{d} v_1 p_1(i) r_1(i) = v_1 \langle p_1, r_1 \rangle$ dollars

where $r_t(i) = \dfrac{S_{t+1}(i)}{S_t(i)} = \dfrac{\text{price of GOOG at time t+1}}{\text{price of GOOG at time t}}$.

Universal Portfolio Optimization

Given a collection of stocks, let the ith stock have price $S_t(i)$ over time t.
You start with v_1 dollars and fractionally invest it into d stocks according to $p_1 \in \triangle_d$.

Your portfolio at time 2 is worth $v_2 := \sum_{i=1}^{d} v_1 p_1(i) r_1(i) = v_1 \langle p_1, r_1 \rangle$ dollars

where $r_t(i) = \frac{S_{t+1}(i)}{S_t(i)} = \frac{\text{price of GOOG at time t+1}}{\text{price of GOOG at time t}}$.

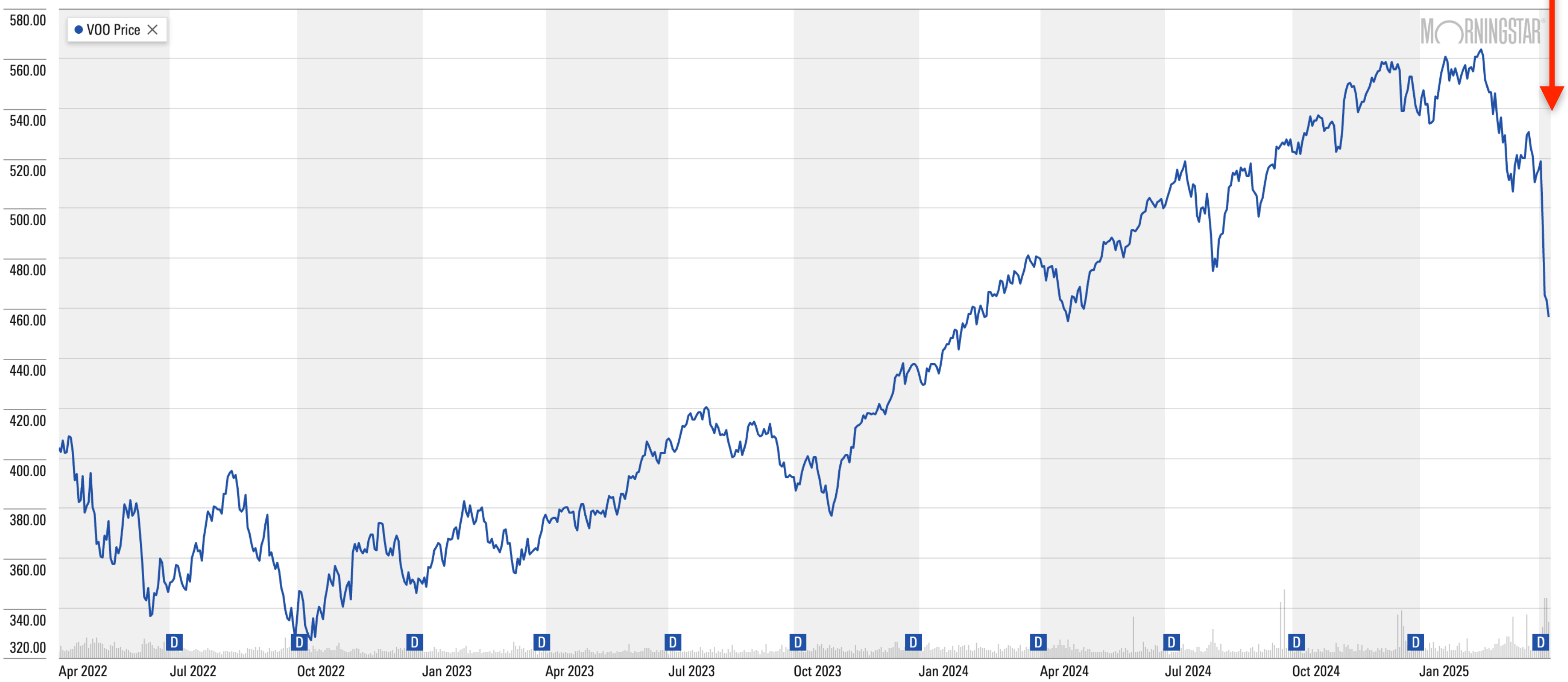
Classical Portfolio Theory (Markowitz 1952): Assume returns $r_t \in \mathbb{R}^n_+$ are IID with mean $\mu = \mathbb{E}[r_t]$ and covariance $\Sigma = \mathbb{E}[(r_t - \mu)(r_t - \mu)^\top]$. The for a return target $\bar{r} \geq 0$ solve

$$\min_{p \in \triangle_d} p^\top \Sigma p \quad \text{subject to} \quad p^\top \mu \geq \bar{r}$$

In practice, estimate μ, Σ from data. What could possibly go wrong?

Universal Portfolio Optimization

Trump administration announces Tariffs



Returns are not an IID stochastic random walk!

Can we model the stock market as an online learning problem and develop an algorithm that is robust to even adversarial returns?

Universal Portfolio Optimization

You start with v_1 dollars and fractionally invest it into d stocks according to $p_1 \in \triangle_d$

Your portfolio at time 2 is worth $v_2 := \sum_{i=1}^{d} v_1 p_1(i) r_1(i) = v_1 \langle p_1, r_1 \rangle$ dollars

where $r_t(i) = \dfrac{\text{price of GOOG at time t+1}}{\text{price of GOOG at time t}}$.

After T times your portfolio is worth $v_T = v_1 \prod_{t=1}^{T-1} \langle p_t, r_t \rangle$.

Universal Portfolio Optimization

You start with v_1 dollars and fractionally invest it into d stocks according to $p_1 \in \triangle_d$

Your portfolio at time 2 is worth $v_2 := \sum_{i=1}^{d} v_1 p_1(i) r_1(i) = v_1 \langle p_1, r_1 \rangle$ dollars

where $r_t(i) = \dfrac{\text{price of GOOG at time t+1}}{\text{price of GOOG at time t}}$.

After T times your portfolio is worth $v_T = v_1 \prod_{t=1}^{T-1} \langle p_t, r_t \rangle$.

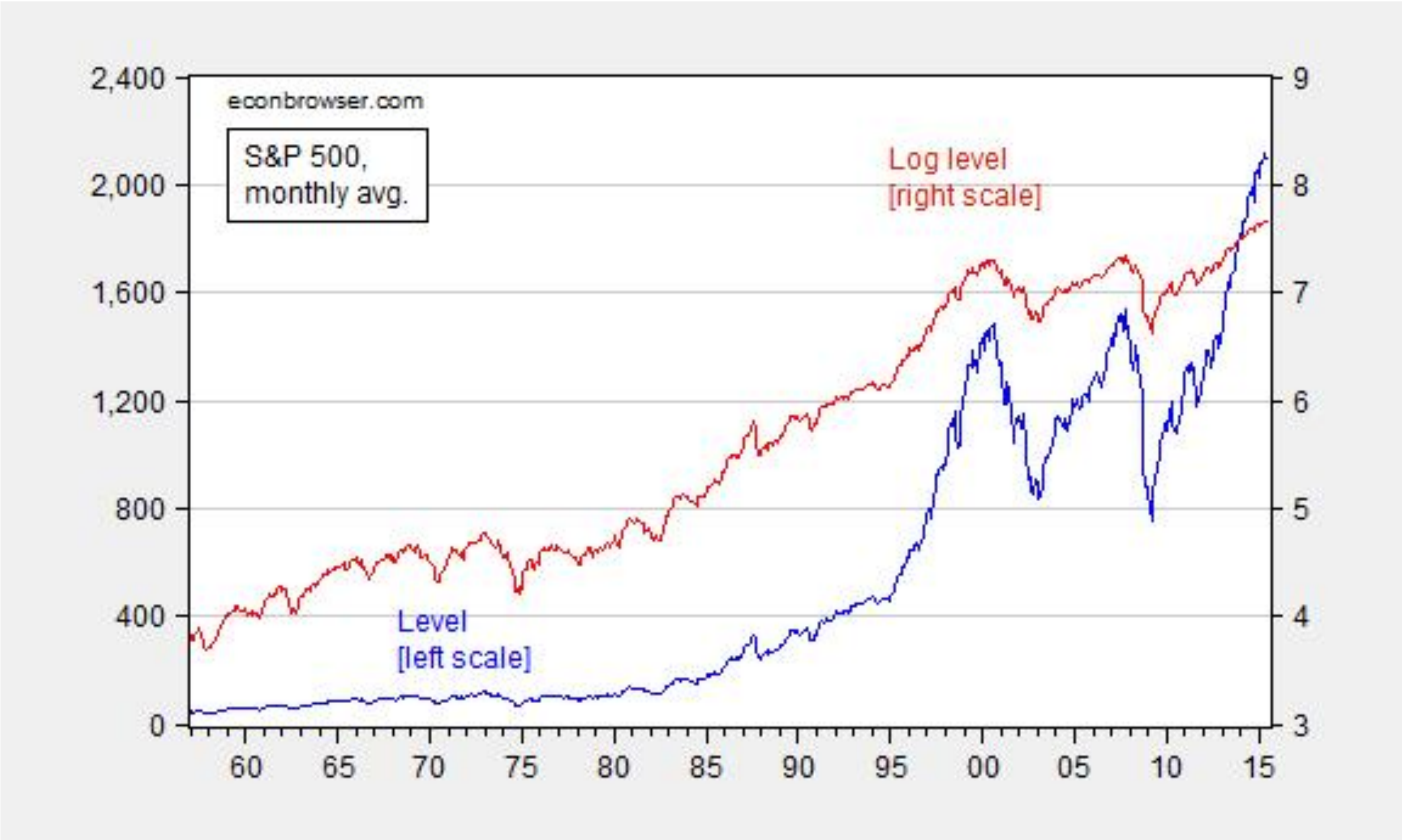
Goal: Maximize your return $\dfrac{v_T}{v_1}$, equivalent to $\log\left(\dfrac{v_T}{v_1}\right) = \sum_{t=1}^{T-1} \log \langle p_t, r_t \rangle$

$$\text{Regret} = \max_{p \in \triangle_d} \sum_{t=1}^{T-1} \log \langle p, r_t \rangle - \sum_{t=1}^{T-1} \log \langle p_t, r_t \rangle$$

Universal Portfolio Optimization

$$\text{Regret} = \max_{p \in \triangle_d} \sum_{t=1}^{T-1} \log\langle p, r_t \rangle - \sum_{t=1}^{T-1} \log\langle p_t, r_t \rangle$$

The SP500 (VOO) is an index that weights 500 stocks by their market capitalization. An alternative index (RSP) weights these 500 stocks uniformly $p = (\frac{1}{500}, \ldots, \frac{1}{500})$.



Universal Portfolio Optimization

$$\text{Regret} = \max_{p \in \triangle_d} \sum_{t=1}^{T-1} \log\langle p, r_t \rangle - \sum_{t=1}^{T-1} \log\langle p_t, r_t \rangle$$

for $t = 1, 2, \ldots$

Player picks $p_t \in \triangle_d$

Adversary simultaneously reveals $r_t \in \mathbb{R}_+^d$

Player pays loss $\ell_t(p_t) = -\log\langle p_t, r_t \rangle$

Exponential weights algorithm

Initialize: $w_1 = (1, \ldots, 1) \in \mathbb{R}^d$

for $t = 1, 2, \ldots$

Player plays $p_t(i) = w_t(i) / \sum_{j=1}^d w_t(j)$

Adversary simultaneously reveals convex loss $\ell_t(\cdot)$

Player pays loss $\ell_t(p_t)$

Player updates weights $w_{t+1}(i) = w_t(i) \exp(-\eta \ell_t(\mathbf{e}_i))$

Universal Portfolio Optimization

$$\text{Regret} = \max_{p \in \triangle_d} \sum_{t=1}^{T-1} \log\langle p, r_t \rangle - \sum_{t=1}^{T-1} \log\langle p_t, r_t \rangle$$

Competes with the single best stock in hindsight!

Theorem: With $\eta = 1$ and $l_t(p) = -\log\langle p, r_t \rangle$, $\max_{i \in [d]} \sum_{t=1}^{T-1} \log\langle \mathbf{e}_i, r_t \rangle - \log\langle p_t, r_t \rangle \leq \log(d)$

Exponential weights algorithm

Initialize: $w_1 = (1, \ldots, 1) \in \mathbb{R}^d$

for $t = 1, 2, \ldots$

Player plays $p_t(i) = w_t(i) / \sum_{j=1}^{d} w_t(j)$

Adversary simultaneously reveals convex loss $\ell_t(\cdot)$

Player pays loss $\ell_t(p_t)$

Player updates weights $w_{t+1}(i) = w_t(i) \exp(-\eta \ell_t(\mathbf{e}_i))$

Proof

Theorem: With $\eta = 1$ and $l_t(p) = -\log\langle p, r_t\rangle$, $\max_{i\in[d]} \sum_{t=1}^{T-1} \log\langle \mathbf{e}_i, r_t\rangle - \log\langle p_t, r_t\rangle \leq \log(d)$

$$\begin{aligned}
\log \frac{W_{T+1}}{W_1} &= \sum_{t=1}^{T} \log \frac{W_{t+1}}{W_t} \\
&= \sum_{t=1}^{T} \log\left(\sum_{i=1}^{d} \frac{w_{t+1}(i)}{W_t}\right) \\
&= \sum_{t=1}^{T} \log\left(\sum_{i=1}^{d} \frac{w_t(i)\exp(-\eta \ell_t(\mathbf{e}_i))}{W_t}\right) \\
&= \sum_{t=1}^{T} \log\left(\sum_{i=1}^{d} p_t(i)\exp(-\eta \ell_t(\mathbf{e}_i))\right) \\
&= \sum_{t=1}^{T} \log\left(\sum_{i=1}^{d} p_t(i)\exp(\log\langle \mathbf{e}_i, r_t\rangle)\right) \\
&= \sum_{t=1}^{T} \log\langle p_t, r_t\rangle
\end{aligned}$$

$$\begin{aligned}
\log \frac{W_{T+1}}{W_1} &\geq \log \frac{w_{T+1}(i)}{W_1} \\
&= -\log(d) + \log\left(\prod_{t=1}^{T} \exp(-\eta \ell_t(\mathbf{e}_i))\right) \\
&= -\log(d) - \sum_{t=1}^{T} \eta \ell_t(\mathbf{e}_i) \\
&= -\log(d) + \sum_{t=1}^{T} \log\langle \mathbf{e}_i, r_t\rangle
\end{aligned}$$

$$\Longrightarrow \max_{i\in[d]} \sum_{t=1}^{T} \log\langle \mathbf{e}_i, r_t\rangle - \log\langle p_t, r_t\rangle \leq \log(d)$$

Universal Portfolio Optimization

$$\text{Regret} = \max_{p \in \triangle_d} \sum_{t=1}^{T-1} \log\langle p, r_t \rangle - \sum_{t=1}^{T-1} \log\langle p_t, r_t \rangle$$

Competes with the single best stock in hindsight!

Theorem: With $\eta = 1$ and $l_t(p) = -\log\langle p, r_t \rangle$, $\max_{i \in [d]} \sum_{t=1}^{T-1} \log\langle \mathbf{e}_i, r_t \rangle - \log\langle p_t, r_t \rangle \leq \log(d)$

Is competing against single best stock a good benchmark? Consider just 2 stocks:

$$r_t(1) = (2, \tfrac{1}{2}, 2, \tfrac{1}{2}, 2, \tfrac{1}{2}, \ldots)$$

$$r_t(2) = (\tfrac{1}{2}, 2, \tfrac{1}{2}, 2, \tfrac{1}{2}, 2 \ldots)$$

$$\prod_{t=1}^{T} \langle \mathbf{e}_i, r_t \rangle = 1 \qquad \prod_{t=1}^{T} \langle \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}, r_t \rangle = \left((\tfrac{1}{2})^2 + 1 \right)^{T/2}$$

How do we compete with any $p \in \triangle_d$?

Example due to Elad Hazan's *Introduction to Online Convex Optimization*