15-888 Computational Game Solving (Fall 2021) Tue, September 21th 2021

Lecture 5

Regret circuits and the Counterfactual Regret Minimization (CFR) paradigm

Instructor: Gabriele Farina*

In Lecture 4 we studied online optimization for *simplex* domains. In this section we show that it is possible to construct an efficient regret minimizer that outputs sequence-form strategies, by combining *any* local regret minimizers at each decision point of the tree-form sequential decision making, including RM and RM$^+$ discussed before. The resulting algorithm has been known since the seminal work by Zinkevich et al. [2007] under the name *counterfactual regret minimization (CFR)*.

The analysis we present is based on the formalism of *regret circuits* [Farina et al., 2019].

1 Regret circuits

Regret circuits are a constructive answer to the question: *"given regret minimizers for two sets $\mathcal{X}, \mathcal{Y}$, can we compose these regret minimizers for various operations performed on the sets (e.g., intersection, convex hull, Cartesian product), in order to arrive at a regret minimizer for the resulting composite set?"*. In particular, in the following subsections, we will show that once regret minimizers for two sets $\mathcal{X}$ and $\mathcal{Y}$ are known, they can always be efficiently combined to come up with composite regret minimizers for the Cartesian product $\mathcal{X} \times \mathcal{Y}$ and the convex hull $\text{co}\{\mathcal{X}, \mathcal{Y}\}$.

Since, as we say in Lecture 2, any sequence-form strategy space Q can be defined inductively starting from convex hulls and Cartesian products, we can keep combining regret circuits until we construct an algorithm for Q. The resulting family of algorithms is known as counterfactual regret minimization (CFR).

1.1 Regret circuit for Cartesian product

Let $\mathcal{X} \subseteq \mathbb{R}^m$ and $\mathcal{Y} \subseteq \mathbb{R}^n$ be two sets, and let $\mathcal{R}_\mathcal{X}$ and $\mathcal{R}_\mathcal{Y}$ be regret minimizers for $\mathcal{X}$ and $\mathcal{Y}$ respectively. We can combine $\mathcal{R}_\mathcal{X}$ and $\mathcal{R}_\mathcal{Y}$ to form a regret minimizer for the Cartesian product $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^{m+n}$ by doing the following:

- To output the next strategy, we ask $\mathcal{R}_\mathcal{X}$ and $\mathcal{R}_\mathcal{Y}$ for their next strategies—call them $\boldsymbol{x}^t$ and $\boldsymbol{y}^t$—and return the pair $(\boldsymbol{x}^t, \boldsymbol{y}^t) \in \mathcal{X} \times \mathcal{Y}$.
- Any utility vector $\boldsymbol{\ell} \in \mathbb{R}^{m+n}$ can be written as $\boldsymbol{\ell} = (\boldsymbol{\ell}_\mathcal{X}, \boldsymbol{\ell}_\mathcal{Y}) \in \mathbb{R}^m \times \mathbb{R}^n$. When at time t we receive the utility vector $\boldsymbol{\ell}^t = (\boldsymbol{\ell}^t_\mathcal{X}, \boldsymbol{\ell}^t_\mathcal{Y})$ we forward $\boldsymbol{\ell}^t_\mathcal{X}$ to $\mathcal{R}_\mathcal{X}$ and $\boldsymbol{\ell}^t_\mathcal{Y}$ to $\mathcal{Y}$.

The algorithm we just described is summarized in Algorithm 1 and given pictorially in Figure 1.

*Computer Science Department, Carnegie Mellon University. ✉ gfarina@cs.cmu.edu.

Algorithm 1: Regret minimizer for $\mathcal{X} \times \mathcal{Y}$

function NEXTSTRATEGY()
  $\boldsymbol{x}^t \leftarrow \mathcal{R}_{\mathcal{X}}$.NEXTSTRATEGY()
  $\boldsymbol{y}^t \leftarrow \mathcal{R}_{\mathcal{Y}}$.NEXTSTRATEGY()
  return $(\boldsymbol{x}^t, \boldsymbol{y}^t) \in \mathcal{X} \times \mathcal{Y}$

function OBSERVEUTILITY($\boldsymbol{\ell}^t = (\boldsymbol{\ell}^t_{\mathcal{X}}, \boldsymbol{\ell}^t_{\mathcal{Y}}) \in \mathbb{R}^m \times \mathbb{R}^n$)
  $\mathcal{R}_{\mathcal{X}}$.OBSERVEUTILITY($\boldsymbol{\ell}^t_{\mathcal{X}}$)
  $\mathcal{R}_{\mathcal{Y}}$.OBSERVEUTILITY($\boldsymbol{\ell}^t_{\mathcal{Y}}$)

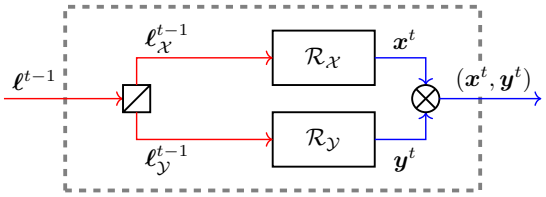

Figure 1: Regret circuit for the Cartesian product $\mathcal{X} \times \mathcal{Y}$.

Analysis The regret minimizers $\mathcal{R}_{\mathcal{X}}$ and $\mathcal{R}_{\mathcal{Y}}$ output strategies $\boldsymbol{x}^t, \boldsymbol{y}^t$ and observe utilities $\boldsymbol{\ell}^t_{\mathcal{X}}, \boldsymbol{\ell}^t_{\mathcal{Y}}$ at all time t. Hence, the regret cumulated by $\mathcal{R}_{\mathcal{X}}$ and $\mathcal{R}_{\mathcal{Y}}$ up to time T is given by

$$R^T_{\mathcal{X}} = \max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_{t=1}^{T} (\boldsymbol{\ell}^t_{\mathcal{X}})^\top \hat{\boldsymbol{x}} - (\boldsymbol{\ell}^t_{\mathcal{X}})^\top \boldsymbol{x}^t \right\}$$
$$R^T_{\mathcal{Y}} = \max_{\hat{\boldsymbol{y}} \in \mathcal{Y}} \left\{ \sum_{t=1}^{T} (\boldsymbol{\ell}^t_{\mathcal{Y}})^\top \hat{\boldsymbol{y}} - (\boldsymbol{\ell}^t_{\mathcal{Y}})^\top \boldsymbol{y}^t \right\}.$$

In order to verify that Algorithm 1 indeed yields a regret minimizer we study the regret cumulated by Algorithm 1 up to any time T. Starting from the definition of regret, we have

$$\begin{aligned} R^T_{\mathcal{X}\times\mathcal{Y}} &= \max_{(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \in \mathcal{X}\times\mathcal{Y}} \left\{ \sum_{t=1}^{T} (\boldsymbol{\ell}^t_{\mathcal{X}}, \boldsymbol{\ell}^t_{\mathcal{Y}})^\top (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) - (\boldsymbol{\ell}^t_{\mathcal{X}}, \boldsymbol{\ell}^t_{\mathcal{Y}})^\top (\boldsymbol{x}^t, \boldsymbol{y}^t) \right\} \\ &= \max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_{t=1}^{T} (\boldsymbol{\ell}^t_{\mathcal{X}})^\top \hat{\boldsymbol{x}} - \sum_{t=1}^{T} (\boldsymbol{\ell}^t_{\mathcal{X}})^\top \boldsymbol{x}^t \right\} + \min_{\hat{\boldsymbol{y}} \in \mathcal{Y}} \left\{ \sum_{t=1}^{T} (\boldsymbol{\ell}^t_{\mathcal{Y}})^\top \hat{\boldsymbol{y}} - \sum_{t=1}^{T} (\boldsymbol{\ell}^t_{\mathcal{Y}})^\top \boldsymbol{y}^t \right\} \\ &= R^T_{\mathcal{X}} + R^T_{\mathcal{Y}}. \end{aligned}$$

So, if $\mathcal{R}_{\mathcal{X}}$ and $\mathcal{R}_{\mathcal{Y}}$ guarantee sublinear regret, then so does Algorithm 1. In other words, it is possible to minimize regret on $\mathcal{X} \times \mathcal{Y}$ by simply minimizing it on $\mathcal{X}$ and $\mathcal{Y}$ independently and then combining the decisions.

In summary, we have proven the following result.

Theorem 1.1. The regret circuit of Figure 1 provides a regret minimizer for the Cartesian product $\mathcal{X} \times \mathcal{Y}$ that satisfies the regret bound

$$R^T_{\mathcal{X}\times\mathcal{Y}} = R^T_{\mathcal{X}} + R^T_{\mathcal{Y}}.$$

The extension to the Cartesian product of more than two sets is direct.

1.2 Regret circuit for convex hull

We now show how to combine $\mathcal{R}_{\mathcal{X}}$ and $\mathcal{R}_{\mathcal{Y}}$ for two sets $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{Y} \subseteq \mathbb{R}^n$ to construct a regret minimizer for the convex hull $\text{co}\{\mathcal{X}, \mathcal{Y}\}$. The construction is harder than in the case of Cartesian products, but some of the ideas carry over. Then, we will use a third regret minimizer $\mathcal{R}_\Delta$ for the 2-simplex Δ^2 decide how to "mix" $\boldsymbol{x}^t$ and $\boldsymbol{y}^t$. More precisely, we will do the following.

- To output the next strategy, we ask $\mathcal{R}_{\mathcal{X}}$ and $\mathcal{R}_{\mathcal{Y}}$ for their next strategies—call them $\boldsymbol{x}^t$ and $\boldsymbol{y}^t$. Then, we will ask $\mathcal{R}_\Delta$ for its next strategy $\boldsymbol{\lambda}^t = (\lambda^t_1, \lambda^t_2) \in \Delta^2$, and return the convex combination $\lambda^t_1 \boldsymbol{x}^t + \lambda^t_2 \boldsymbol{y}^t \in \text{co}\{\mathcal{X}, \mathcal{Y}\}$.

- When at time t we receive the utility vector $\boldsymbol{\ell}^t \in \mathbb{R}^n$, we forward $\boldsymbol{\ell}^t$ to $\mathcal{R}_\mathcal{X}$ and $\mathcal{R}_\mathcal{Y}$. Then, we forward the utility vector

$$\boldsymbol{\ell}_\Delta^t \coloneqq \begin{pmatrix} (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t \\ (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \end{pmatrix}. \tag{1}$$

  to $\mathcal{R}_\Delta$.

The algorithm we just described is summarized in Algorithm 2 and given pictorially in Figure 2.

Algorithm 2: Regret minimzer for $\mathrm{co}\{\mathcal{X}, \mathcal{Y}\}$

function NEXTSTRATEGY()
  $\boldsymbol{x}^t \leftarrow \mathcal{R}_\mathcal{X}$.NEXTSTRATEGY()
  $\boldsymbol{y}^t \leftarrow \mathcal{R}_\mathcal{Y}$.NEXTSTRATEGY()
  $\boldsymbol{\lambda}^t = (\lambda_1^t, \lambda_2^t) \in \Delta^2 \leftarrow \mathcal{R}_\Delta$.NEXTSTRATEGY()
  return $\lambda_1^t \boldsymbol{x}^t + \lambda_2^t \boldsymbol{y}^t \in \mathrm{co}\{\mathcal{X}, \mathcal{Y}\}$

function OBSERVEUTILITY($\boldsymbol{\ell}^t$)
  $\mathcal{R}_\mathcal{X}$.OBSERVEUTILITY($\boldsymbol{\ell}^t$)
  $\mathcal{R}_\mathcal{Y}$.OBSERVEUTILITY($\boldsymbol{\ell}^t$)
  $\mathcal{R}_\Delta$.OBSERVEUTILITY$\left(\boldsymbol{\ell}_\Delta^t \coloneqq \begin{pmatrix} (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t \\ (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \end{pmatrix}\right)$

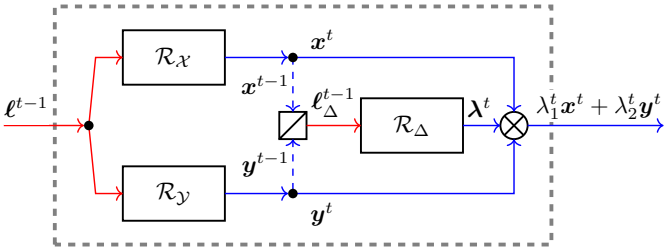

Figure 2: Regret circuit for the convex hull $\mathrm{co}\{\mathcal{X}, \mathcal{Y}\}$. The utility vector ℓ_λ^t is defined in Equation (1).

Analysis The regret minimizers $\mathcal{R}_\mathcal{X}$ and $\mathcal{R}_\mathcal{Y}$ output strategies $\boldsymbol{x}^t, \boldsymbol{y}^t$ and observe utilities $\boldsymbol{\ell}^t$ at all time t. Hence, the regret cumulated by $\mathcal{R}_\mathcal{X}$ and $\mathcal{R}_\mathcal{Y}$ up to time T is given by

$$R_\mathcal{X}^T = \max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{x}} - (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t \right\} \tag{2}$$

$$R_\mathcal{Y}^T = \max_{\hat{\boldsymbol{y}} \in \mathcal{Y}} \left\{ \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{y}} - (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right\}. \tag{3}$$

The regret minimizer $\mathcal{R}_\Delta$ outputs strategies $\boldsymbol{\lambda}^t = (\lambda_1^t, \lambda_2^t)$ at all times t and observes utility vectors $\boldsymbol{\ell}_\Delta^t$ at all times t. So, the regret cumulated by $\mathcal{R}_\Delta$ is given by

$$R_\Delta^T \coloneqq \max_{\hat{\boldsymbol{\lambda}} \in \Delta^2} \left\{ \left(\sum_{t=1}^T \hat{\lambda}_1 (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t + \hat{\lambda}_2 (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right) \right\} - \left(\sum_{t=1}^T \lambda_1^t (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t + \lambda_2^t (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right).$$

In order to verify that Algorithm 2 indeed yields a regret minimizer we study the regret it cumulates up to any time T. Starting from the definition of regret, we have

$$\begin{aligned} R^T &= \max_{\substack{\hat{\boldsymbol{\lambda}} \in \Delta^2 \\ \hat{\boldsymbol{x}} \in \mathcal{X}, \hat{\boldsymbol{y}} \in \mathcal{Y}}} \left\{ \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top (\hat{\lambda}_1 \hat{\boldsymbol{x}} + \hat{\lambda}_2 \hat{\boldsymbol{y}}) - (\boldsymbol{\ell}^t)^\top (\lambda_1^t \boldsymbol{x}^t + \lambda_2^t \boldsymbol{y}^t) \right\} \\ &= \max_{\substack{\hat{\boldsymbol{\lambda}} \in \Delta^2 \\ \hat{\boldsymbol{x}} \in \mathcal{X}, \hat{\boldsymbol{y}} \in \mathcal{Y}}} \left\{ \hat{\lambda}_1 \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{x}} + \hat{\lambda}_2 \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{y}} \right\} - \left(\sum_{t=1}^T \lambda_1^t \, (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t + \lambda_2^t \, (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right). \end{aligned} \tag{4}$$

Now, we make two important observations. First,

$$\max_{\substack{\hat{\boldsymbol{\lambda}} \in \Delta^2 \\ \hat{\boldsymbol{x}} \in \mathcal{X}, \hat{\boldsymbol{y}} \in \mathcal{Y}}} \left\{ \hat{\lambda}_1 \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{x}} + \hat{\lambda}_2 \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{y}} \right\} = \max_{\hat{\boldsymbol{\lambda}} \in \Delta^2} \left\{ \hat{\lambda}_1 \max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{x}} \right\} + \hat{\lambda}_2 \max_{y \in \mathcal{Y}} \left\{ \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{y}} \right\} \right\}, \tag{5}$$

since all components of $\hat{\boldsymbol{\lambda}} = (\hat{\lambda}_1, \hat{\lambda}_2) \in \Delta^2$ are non-negative. Second, the inner minimization problem over $\mathcal{X}$ is related to the cumulative regret $R_{\mathcal{X}}^T$ (2) of the regret minimizer $\mathcal{R}_{\mathcal{X}}$ that observes the utility functions ℓ^t as follows:

$$\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_{t=1}^{T} (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{x}} \right\} = R_{\mathcal{X}}^T + \sum_{t=1}^{T} (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t. \tag{6}$$

Similarly, for $\mathcal{Y}$ we have from (3) that

$$\max_{\hat{\boldsymbol{y}} \in \mathcal{Y}} \left\{ \sum_{t=1}^{T} (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{y}} \right\} = R_{\mathcal{Y}}^T + \sum_{t=1}^{T} (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t. \tag{7}$$

Plugging Equations (5), (6), and (7) into (4), we obtain

$$\begin{aligned}
R^T &= \max_{\hat{\boldsymbol{\lambda}} \in \Delta^2} \left\{ \hat{\lambda}_1 \max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_{t=1}^{T} (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{x}} \right\} + \hat{\lambda}_2 \max_{y \in \mathcal{Y}} \left\{ \sum_{t=1}^{T} (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{y}} \right\} \right\} - \left(\sum_{t=1}^{T} \lambda_1^t (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t + \lambda_2^t (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right) \\
&= \max_{\hat{\boldsymbol{\lambda}} \in \Delta^2} \left\{ \hat{\lambda}_1 \left(R_{\mathcal{X}}^T + \sum_{t=1}^{T} (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t \right) + \hat{\lambda}_2 \left(R_{\mathcal{Y}}^T + \sum_{t=1}^{T} (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right) \right\} - \left(\sum_{t=1}^{T} \lambda_1^t (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t + \lambda_2^t (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right) \\
&= \max_{\hat{\boldsymbol{\lambda}} \in \Delta^2} \left\{ \left(\sum_{t=1}^{T} \hat{\lambda}_1 (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t + \hat{\lambda}_2 (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right) + \left(\hat{\lambda}_1 R_{\mathcal{X}}^T + \hat{\lambda}_2 R_{\mathcal{Y}}^T \right) \right\} - \left(\sum_{t=1}^{T} \lambda_1^t (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t + \lambda_2^t (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right).
\end{aligned}$$

No matter the choice of $(\hat{\lambda}_1, \hat{\lambda}_2) \in \Delta^2$, the convex combination $\hat{\lambda}_1 R_{\mathcal{X}}^T + \hat{\lambda}_2 R_{\mathcal{Y}}^T$ satisfies

$$\hat{\lambda}_1 R_{\mathcal{X}}^T + \hat{\lambda}_2 R_{\mathcal{Y}}^T \leq \max\{R_{\mathcal{X}}^T, R_{\mathcal{Y}}^T\}.$$

Hence, we can write

$$\begin{aligned}
R^T &\leq \max_{\hat{\boldsymbol{\lambda}} \in \Delta^2} \left\{ \left(\sum_{t=1}^{T} \hat{\lambda}_1 (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t + \hat{\lambda}_2 (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right) + \max\{R_{\mathcal{X}}^T, R_{\mathcal{Y}}^T\} \right\} - \left(\sum_{t=1}^{T} \lambda_1^t (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t + \lambda_2^t (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right) \\
&= \max_{\hat{\boldsymbol{\lambda}} \in \Delta^2} \left\{ \left(\sum_{t=1}^{T} \hat{\lambda}_1 (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t + \hat{\lambda}_2 (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right) \right\} + \max\{R_{\mathcal{X}}^T, R_{\mathcal{Y}}^T\} - \left(\sum_{t=1}^{T} \lambda_1^t (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t + \lambda_2^t (\boldsymbol{\ell}^t)^\top \boldsymbol{y}^t \right) \\
&= R_{\Delta}^T + \max\{R_{\mathcal{X}}^T, R_{\mathcal{Y}}^T\}.
\end{aligned} \tag{8}$$

Equation (8) guarantees that if all three regrets $\{R_{\Delta}^T, R_{\mathcal{X}}^T, R_{\mathcal{Y}}^T\}$ grow sublinearly, then so does R^T. Figure 2 shows the regret circuit that corresponds to our construction above.

In summary, we have proven the following result.

Theorem 1.2. The regret circuit of Figure 2 provides a regret minimizer for the convex hull $\mathrm{co}\{\mathcal{X}, \mathcal{Y}\}$ that satisfies the regret bound

$$R^T \leq R_{\Delta^2}^T + \max\{R_{\mathcal{X}}^T, R_{\mathcal{Y}}^T\}.$$

Extension to multiple sets The construction shown in Algorithm 2 and Figure 2 can be readily extended to handle the convex hull $\mathrm{co}\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ of n sets as follows.

- At each time instant t, we let n regret minimizers $\mathcal{R}_{\mathcal{X}_1}, \ldots, \mathcal{R}_{\mathcal{X}_n}$ (for $\mathcal{X}_1, \ldots, \mathcal{X}_n$, respectively) output their next strategies $\boldsymbol{x}_1^t, \ldots, \boldsymbol{x}_n^t$. Those strategies are combined according to the convex combination coefficients $\boldsymbol{\lambda}^t$ output by the additional regret minimizer $\mathcal{R}_{\Delta}$ for the n-simplex Δ^n to form the convex combination strategy $\lambda_1^t \boldsymbol{x}_1^t + \cdots + \lambda_n^t \boldsymbol{x}_n^t \in \mathrm{co}\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$.

- The utility vector $\boldsymbol{\ell}^t$ is fed into all the regret minimizers $\mathcal{R}_{\mathcal{X}_i}$ $(i = 1, \dots, n)$. Then, the utility vector $\boldsymbol{\ell}^t_\Delta$, defined as

$$\boldsymbol{\ell}^t_\Delta = \begin{pmatrix} (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t_1 \\ \vdots \\ (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t_n \end{pmatrix},$$

  is input into a regret minimizer for the n-dimensional simplex Δ^n.

With the same argument used earlier, one can easily show that the regret bound in that case is

$$R^T \leq R^T_{\Delta^n} + \max\{R^T_{\mathcal{X}_1}, \dots, R^T_{\mathcal{X}_n}\}.$$

2 Counterfactual Regret Minimization

We have seen in Lecture 2 that a sequence-form strategy space can be characterized recursively by composing convex hulls and Cartesian products operations. By applying the regret circuits described above, we can then construct a regret minimizers for any sequence-form strategy space. The resulting regret minimizer is called CFR. In a nutshell, CFR decomposes the problem of minimizing regret on the whole tree-form decision problem into local regret minimization problems at each of the individual decision points $j \in \mathcal{J}$. Any regret minimizer $\mathcal{R}_j$ for simplex domains can be used to solve the local regret minimization problems. Popular options are the regret matching algorithm, and the regret matching plus algorithm (Lecture 4).

Before giving pseudocode, we recall and introduce a bit of notation to deal with tree-form sequential decision processes.

Notation for tree-form sequential decision processes We use the following notation for dealing with tree-form sequential decision processes (TFSDPs), most of which was already introduced in Lecture 2. The notation is also summarized in Table 1.

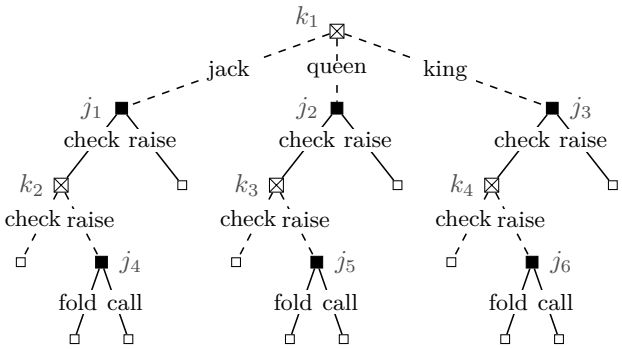

Figure 3: Tree-form sequential decision making process of the first acting player in the game of Kuhn poker.

- We denote the set of decision points in the TFSDP as $\mathcal{J}$, and the set of observation points as $\mathcal{K}$. At each decision point $j \in \mathcal{J}$, the agent selects an action from the set A_j of available actions. At each observation point $k \in \mathcal{K}$, the agent observes a signal s_k from the environment out of a set of possible signals S_k.
- (**new!**) We denote by ρ the transition function of the process. Picking action $a \in A_j$ at decision point $j \in \mathcal{J}$ results in the process transitioning to $\rho(j, a) \in \mathcal{J} \cup \mathcal{K} \cup \{\perp\}$, where $\perp$ denotes the end of the decision process. Similarly, the process transitions to $\rho(k, s) \in \mathcal{J} \cup \mathcal{K} \cup \{\perp\}$ after the agent observes signal $s \in S_k$ at observation point $k \in \mathcal{K}$.
- A pair (j, a) where $j \in \mathcal{J}$ and $a \in A_j$ is called a *sequence*. The set of all sequences is denoted as $\Sigma \coloneqq \{(j, a) : j \in \mathcal{J}, a \in A_j\}$. For notational convenience, we will often denote an element (j, a) in Σ as ja without using parentheses.

- Given a decision point $j \in \mathcal{J}$, we denote by p_j its *parent sequence*, defined as the last sequence (that is, decision point-action pair) encountered on the path from the root of the decision process to j. If the agent does not act before j (that is, j is the root of the process or only observation points are encountered on the path from the root to j), we let $p_j = \varnothing$.

Symbol	Description
$\mathcal{J}$	Set of decision points
A_j	Set of legal actions at decision point $j \in \mathcal{J}$
$\mathcal{K}$	Set of observation points
S_k	Set of possible signals at observation point $k \in \mathcal{K}$
ρ	Transition function: • given $j \in \mathcal{J}$ and $a \in A_j$, $\rho(j, a)$ returns the next point $v \in \mathcal{J} \cup \mathcal{K}$ in the decision tree that is reached after selecting legal action a in j, or $\perp$ if the decision process ends; • given $k \in \mathcal{K}$ and $s \in S_k$, $\rho(k, s)$ returns the next point $v \in \mathcal{J} \cup \mathcal{K}$ in the decision tree that is reached after observing signal s in k, or $\perp$ if the decision process ends
Σ	Set of sequences, defined as $\Sigma \coloneqq \{(j, a) : j \in \mathcal{J}, a \in A_j\}$
p_j	Parent sequence of decision point $j \in \mathcal{J}$, defined as the last sequence (decision point-action pair) on the path from the root of the TFSDP to decision point j; if the agent does not act before j, $p_j = \varnothing$

Table 1: Summary of notation in TFSDPs.

As an example, consider the TFSDP faced by Player 1 in the game of Kuhn poker [Kuhn, 1950], depicted in Figure 3, which we already introduced in Lecture 2. There, we have that $\mathcal{J} = \{j_1, \dots, j_6\}$ and $\mathcal{K} = \{k_1, \dots, k_4\}$. $A_{j_1} = S_{k_4} = \{\text{check}, \text{raise}\}$. $A_{j_5} = \{\text{fold}, \text{call}\}$. $S_{k_1} = \{\text{jack}, \text{queen}, \text{king}\}$. $\rho(k_3, \text{check}) = \rho(j_2, \text{raise}) = \perp$. $\rho(k_1, \text{king}) = j_3$. $\rho(j_2, \text{check}) = k_3$. $p_{j_4} = (j_1, \text{check})$. $p_{j_6} = (j_3, \text{check})$. $p_{j_1} = p_{j_2} = p_{j_3} = \varnothing$.

Notation for the components of vectors Any vector $\boldsymbol{x} \in \mathbb{R}^{|\Sigma|}$ has, by definition, as many components as sequences Σ. The component corresponding to a specific sequence $ja \in \Sigma$ is denoted as $\boldsymbol{x}[ja]$. Similarly, given any decision point $j \in \mathcal{J}$, any vector $\boldsymbol{x} \in \mathbb{R}^{|A_j|}$ has as many components as the number of actions at j. The component corresponding to a specific action $a \in A_j$ is denoted $\boldsymbol{x}[a]$.

CFR algorithm Pseudocode for CFR is given in Algorithm 3. Note that the implementation is parametric on the regret minimization algorithms $\mathcal{R}_j$ run locally at each decision point. Any regret minimizer $\mathcal{R}_j$ for simplex domains can be used to solve the local regret minimization problems. Popular options are the regret matching algorithm, and the regret matching plus algorithm (Lecture 4).

It can be shown that the regret cumulated by the CFR algorithm satisfies the following bound.

Proposition 2.1. Let R_j^T $(j \in \mathcal{J})$ denote the regret cumulated up to time T by each of the regret minimizers $\mathcal{R}_j$. Then, the regret R^T cumulated by Algorithm 3 up to time T satisfies

$$R^T \leq \sum_{j \in \mathcal{J}} \max\{0, R_j^T\}.$$

It is then immediate to see that if each R_j^T grows sublinearly in T, then so does R^T.

Algorithm 3: CFR regret minimizer

Data: $\mathcal{R}_j$, one regret minimizer for $\Delta^{|A_j|}$; one for each decision point $j \in \mathcal{J}$ of the TFSDP.

function NEXTSTRATEGY()
  [▷ Step 1: we ask each of the $\mathcal{R}_j$ for their next strategy local at each decision point]
  for each decision point $j \in \mathcal{J}$ **do**
    $\boldsymbol{b}_j^t \in \Delta^{|A_j|} \leftarrow \mathcal{R}_j$.NEXTSTRATEGY()
  [▷ Step 2: we construct the sequence-form representation of the strategy that plays according to the distribution $\boldsymbol{b}_j^t$ at each decision point $j \in \mathcal{J}$]
  $\boldsymbol{x}^t = \mathbf{0} \in \mathbb{R}^{|\Sigma|}$
  for each decision point $j \in \mathcal{J}$ in *top-down traversal* order in the TFSDP **do**
    for each action $a \in A_j$ **do**
      if $p_j = \varnothing$ **then**
        $\boldsymbol{x}^t[ja] \leftarrow \boldsymbol{b}_j^t[a]$
      else
        $\boldsymbol{x}^t[ja] \leftarrow \boldsymbol{x}^t[p_j] \cdot \boldsymbol{b}_j^t[a]$
  [▷ You should convince yourself that the vector $\boldsymbol{x}^t$ we just filled in above is a valid sequence-form strategy, that is, it satisfies the required consistency constraints we saw in Lecture 2. In symbols, $\boldsymbol{x}^t \in Q$]
  return $\boldsymbol{x}^t$

function OBSERVEUTILITY($\boldsymbol{\ell}^t \in \mathbb{R}^{|\Sigma|}$)
  [▷ Step 1: we compute the expected utility for each subtree rooted at each node $v \in \mathcal{J} \cup \mathcal{K}$]
  $V^t \leftarrow$ empty dictionary [▷ eventually, it will map keys $\mathcal{J} \cup \mathcal{K} \cup \{\perp\}$ to real numbers]
  $V^t[\perp] \leftarrow 0$
  for each node in the tree $v \in \mathcal{J} \cup \mathcal{K}$ in *bottom-up traversal* order in the TFSDP **do**
    if $v \in \mathcal{J}$ **then**
      Let $j = v$
      $V^t[j] \leftarrow \sum_{a \in A_j} \boldsymbol{b}_j^t[a] \cdot \left(\boldsymbol{\ell}^t[ja] + V^t[\rho(j,a)]\right)$
    else
      Let $k = v$
      $V^t[k] \leftarrow \sum_{s \in S_k} V^t[\rho(k,s)]$
  [▷ Step 2: at each decision point $j \in \mathcal{J}$, we now construct a local utility vector $\boldsymbol{\ell}_j^t$ called *counterfactual utility*]
  for each decision point $j \in \mathcal{J}$ **do**
    $\boldsymbol{\ell}_j^t \leftarrow \mathbf{0} \in \mathbb{R}^{|A_j|}$
    for each action $a \in A_j$ **do**
      $\boldsymbol{\ell}_j^t[a] \leftarrow \boldsymbol{\ell}^t[ja] + V^t[\rho(j,a)]$
    $\mathcal{R}_j$.OBSERVEUTILITY($\boldsymbol{\ell}_j^t$)

References

Martin Zinkevich, Michael Bowling, Michael Johanson, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Regret circuits: Composability of regret minimizers. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

H. W. Kuhn. A simplified two-person poker. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, volume 1 of *Annals of Mathematics Studies, 24*, pages 97–103. Princeton University Press, Princeton, New Jersey, 1950.