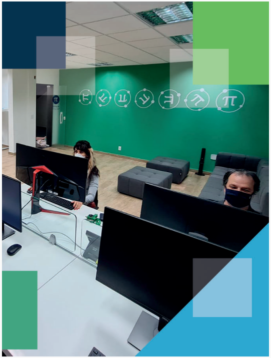
Similarity Learning via Boosting

Thiago Rodrigo Ramos

05 de Julho de 2024

My PhD

I completed my PhD at IMPA, at the Centro PI (Center for Projects and Innovation)



Projects at Centro Pi

My thesis is result of three applications of Machine Learning to real industrial problems

- Stone Pagamentos (2020): Credit Scoring
    - ExactBoost: directly boosting the margin in combinatorial and non-decomposable metrics
- Dasa (2021): Uncertainty Quantification
    - Split conformal prediction for dependent data
- Rede Globo (2022): Record Linkage
    - Similarity Learning via Boosting

Projects at Centro Pi

My thesis is result of three applications of Machine Learning to real industrial problems

- Stone Pagamentos (2020): Credit Scoring
  - ExactBoost: directly boosting the margin in combinatorial and non-decomposable metrics
- Dasa (2021): Uncertainty Quantification
  - Split conformal prediction for dependent data
- Rede Globo (2022): Record Linkage
  - Similarity Learning via Boosting

Projects at Centro Pi

My thesis is result of three applications of Machine Learning to real industrial problems

- Stone Pagamentos (2020): Credit Scoring
  - ExactBoost: directly boosting the margin in combinatorial and non-decomposable metrics
- Dasa (2021): Uncertainty Quantification
  - Split conformal prediction for dependent data
- Rede Globo (2022): Record Linkage
  - Similarity Learning via Boosting

Projects at Centro Pi

My thesis is result of three applications of Machine Learning to real industrial problems

- Stone Pagamentos (2020): Credit Scoring
  - ExactBoost: directly boosting the margin in combinatorial and non-decomposable metrics
- Dasa (2021): Uncertainty Quantification
  - Split conformal prediction for dependent data
- Rede Globo (2022): Record Linkage
  - Similarity Learning via Boosting

IMPA and TV

Figure: This scene from “Tropa de Elite” was filmed on IMPA

IMPA and TV

Figure: Filme “Ricos de amor” da Netflix

The problem

Multiple very large datasets with movies information (e.g. IMDB, TMDB, Rotten Tomatoes)

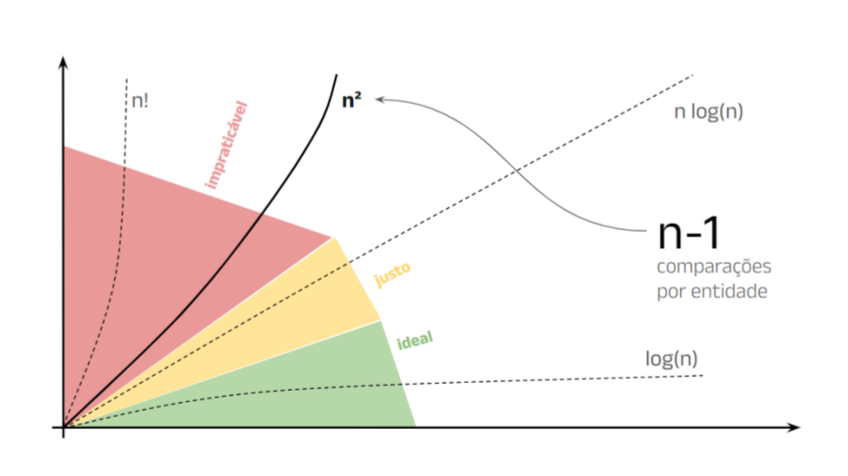
The problem

How can we match similar entries from potentially large datasets to create a richer dataset?

The problem

Naive solution: Check all pairs

Naive solution: Check all pairs

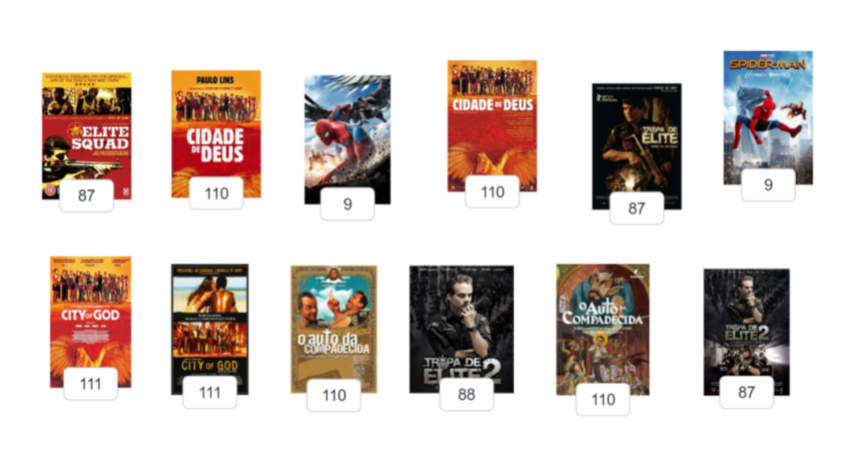


The problem

How to filter out entries that are highly dissimilar?

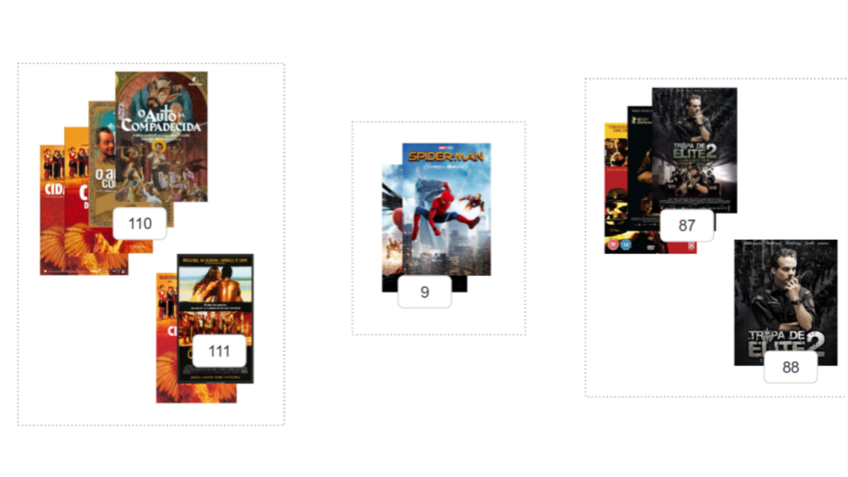
Similarity hashing

One possible solution is to define a hash code for each entry



Similarity hashing

One possible solution is to define a hash code for each entry and then block similar movies together



Evaluation metrics

Given datasets $\mathcal{A}, \mathcal{B}$ for $A \in \mathcal{A}$ we want to find similar items $B \in \mathcal{B}$ while doing as few pairwise comparisons as possible.

$$\begin{aligned}
\text{Recall} &:= \frac{1}{|\mathcal{M}|} \sum_{(\ell,r)\in\mathcal{M}} \mathbf{1}_{[A_\ell \text{ and } B_r \text{ share a block}]};\\
\text{RR} &:= 1 - \frac{1}{|\mathcal{N}|} \sum_{(\ell,r)\in\mathcal{N}} \mathbf{1}_{[A_\ell \text{ and } B_r \text{ share a block}]}\\
\text{H} &:= 2\frac{\text{Recall} \cdot \text{RR}}{\text{Recall} + \text{RR}}
\end{aligned}$$

where $\mathcal{N} := [N_\mathcal{A}] \times [N_\mathcal{B}]$ denotes all possible pairs and $\mathcal{M}$ denotes the set of matching pairs:

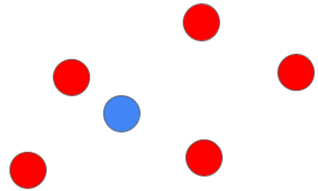
$$\mathcal{M} := \{(\ell, r) \in \mathcal{N}, A_\ell \sim_R B_r, (A_\ell, B_r) \in \mathcal{A} \times \mathcal{B}\}.$$

Similarity hashing: LSH

A possible solution is to use Locality Sensitive Hashing (LSH)

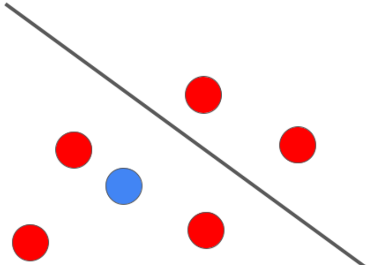
Similarity hashing: LSH

How to find out which red point is closest to the blue point?



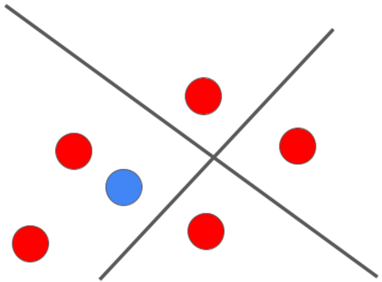
Similarity hashing: LSH

Select a random hyperplane...



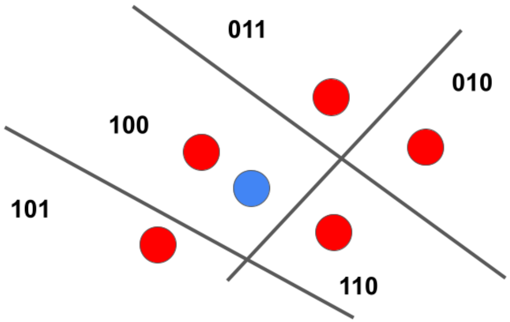
Similarity hashing: LSH

and another...



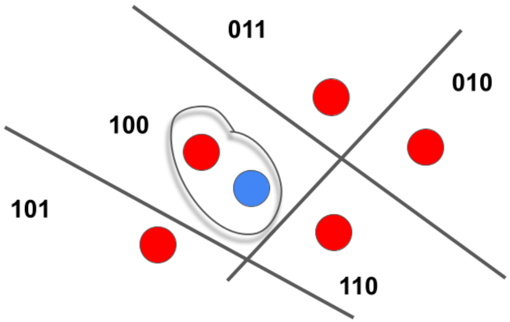
Similarity hashing: LSH

This creates a partition



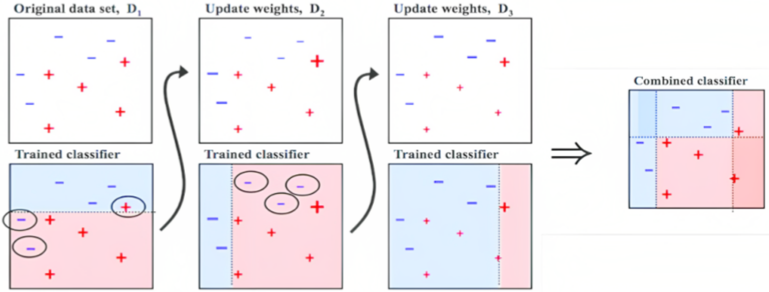
Similarity hashing: LSH

Compare only points in the same partition!

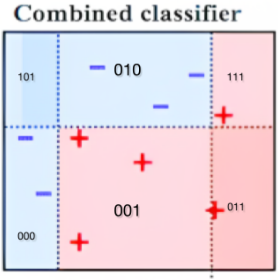


Boost can be used to learn these hyperplanes effectively using a data-driven approach!

Similarity hashing: Boosting



Similarity hashing: Boosting



Similarity hashing: Boosting

Step 1: The model learns simple similarity rules from the data

Movies			
name	year	genre	country
cidade de deus	2002	action	brazil
spider-man	2021	adventure	usa
robocop	2014	action	usa
cidade de deus	2002	drama	brazil

- Rule : Is it from 2002?
- Rule : Is it from Brazil?
- Rule : Name starts with "c"?

  ⋮

- Rule : The second letter in its name is an "i"?

Similarity hashing: Boosting

Step 1: The model learns simple similarity rules from the data

Movies			
name	year	genre	country
cidade de deus	2002	action	brazil
spider-man	2021	adventure	usa
robocop	2014	action	usa
cidade de deus	2002	drama	brazil

- **Rule$_1$:** Is it from 2002?
- **Rule$_2$:** Is it from Brazil?
- **Rule$_3$:** Name starts with "c"?

  ⋮

- **Rule$_T$:** The second letter in its name is an "i"?

Similarity hashing: Boosting

Step 1: The model learns simple similarity rules from the data

Movies			
name	year	genre	country
cidade de deus	2002	action	brazil
spider-man	2021	adventure	usa
robocop	2014	action	usa
cidade de deus	2002	drama	brazil

- $\textbf{Rule}_1$: Is it from 2002?
- $\textbf{Rule}_2$: Is it from Brazil?
- $\textbf{Rule}_3$: Name starts with "c"?

  ⋮

- $\textbf{Rule}_T$: The second letter in its name is an "i"?

Similarity hashing: Boosting

Step 1: The model learns simple similarity rules from the data

Movies			
name	year	genre	country
cidade de deus	2002	action	brazil
spider-man	2021	adventure	usa
robocop	2014	action	usa
cidade de deus	2002	drama	brazil

- $\mathbf{Rule}_1$: Is it from 2002?
- $\mathbf{Rule}_2$: Is it from Brazil?
- $\mathbf{Rule}_3$: Name starts with "c"?

  ⋮
- $\mathbf{Rule}_T$: The second letter in its name is an "i"?

Similarity hashing: Boosting

Step 1: The model learns simple similarity rules from the data

Movies			
name	year	genre	country
cidade de deus	2002	action	brazil
spider-man	2021	adventure	usa
robocop	2014	action	usa
cidade de deus	2002	drama	brazil

- $\mathbf{Rule}_1$: Is it from 2002?
- $\mathbf{Rule}_2$: Is it from Brazil?
- $\mathbf{Rule}_3$: Name starts with "c"?

  ⋮

- $\mathbf{Rule}_T$: The second letter in its name is an "i"?

Similarity hashing: Boosting

Step 1: The model learns simple similarity rules from the data

Movies			
name	year	genre	country
cidade de deus	2002	action	brazil
spider-man	2021	adventure	usa
robocop	2014	action	usa
cidade de deus	2002	drama	brazil

- $\mathbf{Rule}_1$: Is it from 2002?
- $\mathbf{Rule}_2$: Is it from Brazil?
- $\mathbf{Rule}_3$: Name starts with "c"?

  ⋮
- $\mathbf{Rule}_T$: The second letter in its name is an "i"?

Similarity hashing: Boosting

Step 1: For each Rule, the model learns positive weights associated to its relevance and an error value

Movies			
name	year	genre	country
cidade de deus	2002	action	brazil
spider-man	2021	adventure	usa
robocop	2014	action	usa
cidade de deus	2002	drama	brazil

- $\mathbf{Rule}_1$ has relevance $\alpha_1 = 0.31$
- $\mathbf{Rule}_2$ has relevance $\alpha_2 = 0.29$
- $\mathbf{Rule}_3$ has relevance $\alpha_3 = 0.25$

  $\vdots$
- $\mathbf{Rule}_T$ has relevance $\alpha_T = 0.015$

Similarity hashing: Boosting

Step 1: For each Rule, the model learns positive weights associated to its relevance and an error value

Movies			
name	year	genre	country
cidade de deus	2002	action	brazil
spider-man	2021	adventure	usa
robocop	2014	action	usa
cidade de deus	2002	drama	brazil

- $\mathbf{Rule}_1$ has relevance $\alpha_1 = 0.31$
- $\mathbf{Rule}_2$ has relevance $\alpha_2 = 0.29$
- $\mathbf{Rule}_3$ has relevance $\alpha_3 = 0.25$

  $\vdots$
- $\mathbf{Rule}_T$ has relevance $\alpha_T = 0.015$

Similarity hashing: Boosting

- Using such rules and its weights, we can construct a similarity function between items A and B given by:

$$f^*(A, B) = \sum_{i=1}^{T} \alpha_i \mathrm{Rule}_i(A)\mathrm{Rule}_i(B)$$

  where $\mathrm{Rule}_i(x) = 1$ if x satisfies Rule_i and -1 otherwise
- Intuitively, for we want for some $\theta \in (0, 1)$:
  - if A and B are match then $f^*(A, B) \geq \theta \approx 1$
  - if A and B are not match then $f^*(A, B) \leq -\theta \approx -1$,

Similarity hashing: Boosting

- Using such rules and its weights, we can construct a similarity function between items A and B given by:

$$f^*(A,B) = \sum_{i=1}^{T} \alpha_i \mathrm{Rule}_i(A)\mathrm{Rule}_i(B)$$

  where $\mathrm{Rule}_i(x) = 1$ if x satisfies Rule_i and -1 otherwise
- Intuitively, for we want for some $\theta \in (0,1)$:
  - if A and B are match then $f^*(A,B) \geq \theta \approx 1$
  - if A and B are not match then $f^*(A,B) \leq -\theta \approx -1$,

Similarity hashing: Boosting

Theorem

If $\theta > 0$, then f^* satisfies the previous condition with probability at least $1 - \varepsilon$, where:

$$\varepsilon := 2^T \prod_{t=1}^{T} \mathrm{error}_t^{1/2-\theta} (1 - \mathrm{error}_t)^{\theta - 1/2} \tag{1}$$

$$+ \frac{8}{\theta} \left(\mathcal{R}_{\mathcal{S}_{\mathcal{A},n}}(\mathcal{K}) + \mathcal{R}_{\mathcal{S}_{\mathcal{B},n}}(\mathcal{K}) \right). \tag{2}$$

Furthermore, if there exists $\gamma > 0$ such that for all $t \in [T]$, $\gamma \leq (1/2 - \mathrm{error}_t)$ and $\theta \leq 2\gamma$, then the term in (1) decreases exponentially with T.

The proof relies on concentration of measure, margin theory and Rademacher complexity properties

Similarity hashing: Boosting

Step 2: Given rules Rule_t, weights $\alpha_t > 0$ and the similarity function f^* from the previous step

- To construct a single-bit hash we draw a random rule R following the distribution $\mathbb{P}\left(R = \mathrm{Rule}_i\right) = \alpha_i$
- We say that two items A and B have the same single-bit hash if $R(A) = R(B)$
- It is easy to show that

$$\mathbb{P}\left[R(A) = R(B)\right] = \frac{1 + f^*(A, B)}{2}.$$

- We concatenate several single-bit hashes to construct a final hash function

Similarity hashing: Boosting

Step 2: Given rules Rule_t, weights $\alpha_t > 0$ and the similarity function f^* from the previous step

- To construct a single-bit hash we draw a random rule R following the distribution $\mathbb{P}\left(R = \mathrm{Rule}_i\right) = \alpha_i$
- We say that two items A and B have the same single-bit hash if $R(A) = R(B)$
- It is easy to show that

$$\mathbb{P}\left[R(A) = R(B)\right] = \frac{1 + f^*(A, B)}{2}.$$

- We concatenate several single-bit hashes to construct a final hash function

Similarity hashing: Boosting

Step 2: Given rules Rule_t, weights $\alpha_t > 0$ and the similarity function f^* from the previous step

- To construct a single-bit hash we draw a random rule R following the distribution $\mathbb{P}\left(R = \mathrm{Rule}_i\right) = \alpha_i$
- We say that two items A and B have the same single-bit hash if $R(A) = R(B)$
- It is easy to show that

$$\mathbb{P}\left[R(A) = R(B)\right] = \frac{1 + f^*(A, B)}{2}.$$

- We concatenate several single-bit hashes to construct a final hash function

Similarity hashing: Boosting

Step 2: Given rules Rule_t, weights $\alpha_t > 0$ and the similarity function f^* from the previous step

- To construct a single-bit hash we draw a random rule R following the distribution $\mathbb{P}\left(R = \mathrm{Rule}_i\right) = \alpha_i$
- We say that two items A and B have the same single-bit hash if $R(A) = R(B)$
- It is easy to show that

$$\mathbb{P}\left[R(A) = R(B)\right] = \frac{1 + f^*(A, B)}{2}.$$

- We concatenate several single-bit hashes to construct a final hash function

Step 2: Given rules Rule_t, weights $\alpha_t > 0$ and the similarity function f^* from the previous step

- To construct a single-bit hash we draw a random rule R following the distribution $\mathbb{P}\left(R = \mathrm{Rule}_i\right) = \alpha_i$
- We say that two items A and B have the same single-bit hash if $R(A) = R(B)$
- It is easy to show that

$$\mathbb{P}\left[R(A) = R(B)\right] = \frac{1 + f^*(A, B)}{2}.$$

- We concatenate several single-bit hashes to construct a final hash function

Similarity hashing: Boosting

We combine several single-bit hashes via the following algorithm:

Algorithm Algorithm to construct the hash codes

Require: $k, L \in \mathbb{N}$, convex weights$(\alpha_t)_{t=1}^T$, Rules $(\mathrm{Rule}_t)_{t=1}^T$

for $i \leftarrow 1$ to L **do**
    for $j \leftarrow 1$ to k **do**
        $g_{i,j} \leftarrow \mathrm{Rule}_t$ with probability α_t
    end for
    $g_i \leftarrow (g_{i,1}, \ldots, g_{i,k})$
end for
$g \leftarrow (g_1, \ldots, g_L)$
return g

Two elements A and B are tested for similarity if $g_i(A) = g_i(B)$ for some $i = 1, \ldots, L$

Similarity hashing: Boosting

Theorem

Consider datasets $\mathcal{A}$ and $\mathcal{B}$ such that $|\mathcal{A}| = N_{\mathcal{A}}$ and $|\mathcal{B}| = N_{\mathcal{B}}$. Suppose our condition holds for $\theta > 0$. Then, given $\gamma \in (0,1)$, if we set:

$$\rho := \frac{\log\left(\frac{2}{1+\theta}\right)}{\log\left(\frac{2}{1-\theta}\right)},\ k := \lceil \log_{\frac{2}{1+\theta}} N_{\mathcal{A}} \cdot N_{\mathcal{B}} \rceil \text{ and } L := \left\lceil \frac{2(N_{\mathcal{A}} \cdot N_{\mathcal{B}})^{\rho} \log(1/\gamma)}{1+\theta} \right\rceil,$$

then

$$\mathbb{E}\left[\mathrm{Recall}\right] \geq (1-\gamma)(1-\varepsilon), \quad \mathbb{E}\left[\mathrm{RR}\right] \geq \left(1 - \frac{|\mathcal{M}| + L}{N_{\mathcal{A}} \cdot N_{\mathcal{B}}}\right)(1-\varepsilon)\,.$$

Both expectations are with respect to the randomness in the hash code.

Similarity hashing: Boosting

Dataset	BB	Canopy	KLSH	TLSH	Spect	AG	CTT	Hybrid
ABT_BUY	**0.911**	0.761	0.365	0.625	0.263	0.503	0.907	0.822
AMZ_GG	**0.877**	0.605	0.515	0.281	0.518	0.539	0.810	0.849
DBLP_ACM	0.993	0.850	0.895	0.861	0.662	0.696	0.993	**0.998**
DBLP_SCH	0.989	0.891	0.691	0.543	0.602	0.670	**0.991**	0.983
RESTAURANT	0.988	0.785	0.937	0.838	0.519	0.728	**0.997**	0.997
RLDATA500	**0.992**	0.829	0.969	0.982	0.691	0.717	0.966	0.966
RLDATA10K	**0.999**	0.929	0.926	0.987	0.755	0.800	0.957	0.926
MUSICBRAINZ	0.991	0.101	0.944	0.950	0.662	0.737	**0.994**	0.992
WM_AMZ	**0.943**	0.017	0.495	0.005	0.577	0.558	**0.943**	0.942
AVERAGE	**0.965**	0.641	0.749	0.675	0.583	0.660	0.951	0.942

Table: Harmonic mean for each model and dataset.

Similarity hashing: Boosting

At each iteration $t = 1, \ldots, T$ of the boosting process, our model assigns a weight α_t^* to a specific feature of the dataset to block. This weight serves as an indicator of the significance of this feature in matching entities.

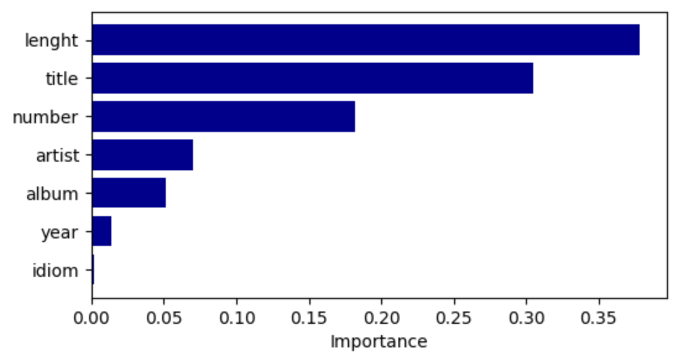


Figure: Feature relevance identified by the model during the boosting step for the `musicbrainz` dataset.

Thank you!