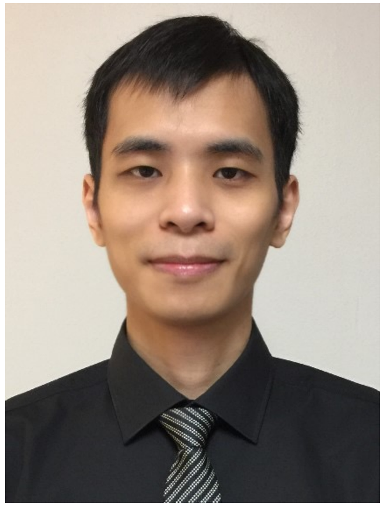
Efficient PAC Learning from the Crowd with Pairwise Comparisons



Shiwei Zeng
Stevens Institute of Technology

Jie Shen
Stevens Institute of Technology

Crowdsourced PAC Learning

Crowdsourced PAC Learning

Standard PAC learning [Valiant 84]

Crowdsourced PAC Learning

Standard PAC learning [Valiant 84]

- Given samples (x, y)

Crowdsourced PAC Learning

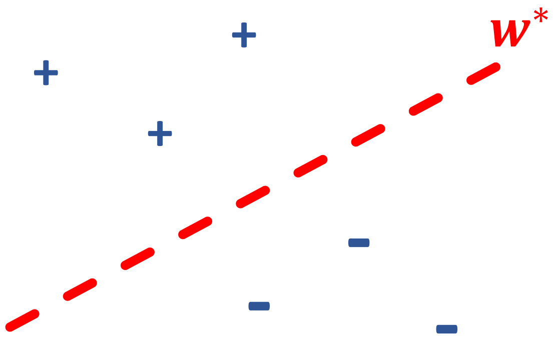
Standard PAC learning [Valiant 84]

- Given samples (x, y)
- Assume $y = h(x)$ for some unknown hypothesis $h \in H$

Crowdsourced PAC Learning

Standard PAC learning [Valiant 84]

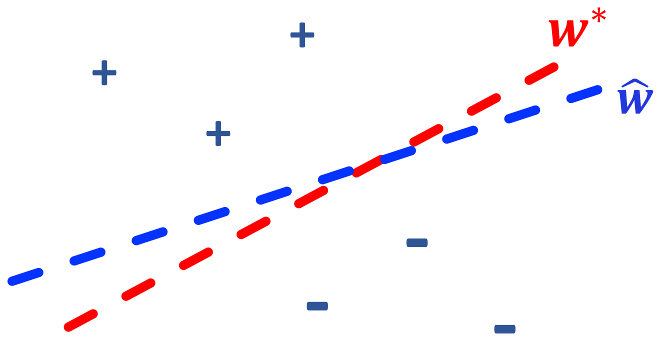
- Given samples (x, y)
- Assume $y = h(x)$ for some unknown hypothesis $h \in H$



Crowdsourced PAC Learning

Standard PAC learning [Valiant 84]

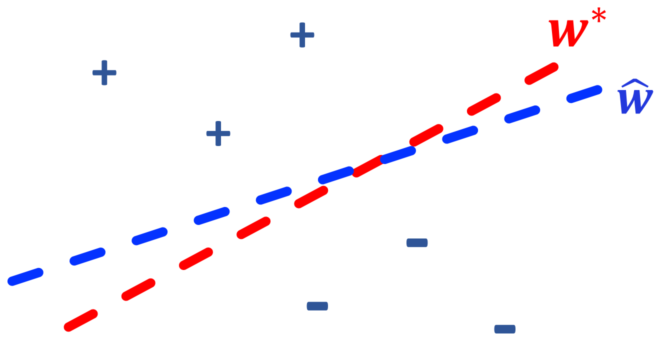
- Given samples (x, y)
- Assume $y = h(x)$ for some unknown hypothesis $h \in H$



Crowdsourced PAC Learning

Standard PAC learning [Valiant 84]

- Given samples (x, y)
- Assume $y = h(x)$ for some unknown hypothesis $h \in H$

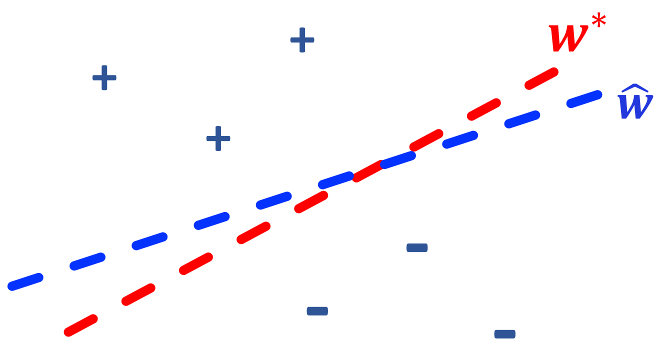


- Goal: $\widehat{\boldsymbol{w}}$ is **P**robably **A**pproximately **C**orrect.

Crowdsourced PAC Learning

Standard PAC learning [Valiant 84]

- Given samples (x, y)
- Assume $y = h(x)$ for some unknown hypothesis $h \in H$



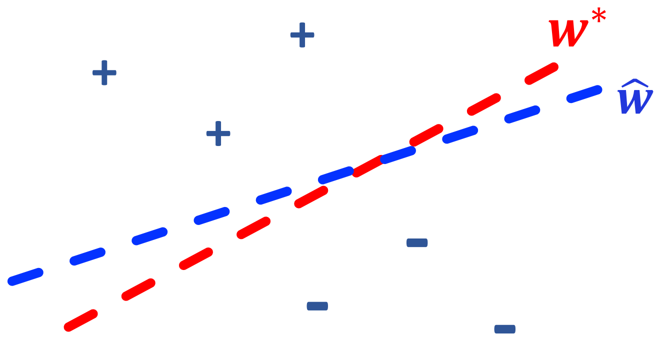
- Goal: $\widehat{\boldsymbol{w}}$ is **P**robably **A**pproximately **C**orrect.

Crowdsourced PAC learning [ABHM17]

Crowdsourced PAC Learning

Standard PAC learning [Valiant 84]

- Given samples (x, y)
- Assume $y = h(x)$ for some unknown hypothesis $h \in H$



- Goal: $\hat{\boldsymbol{w}}$ is **P**robably **A**pproximately **C**orrect.

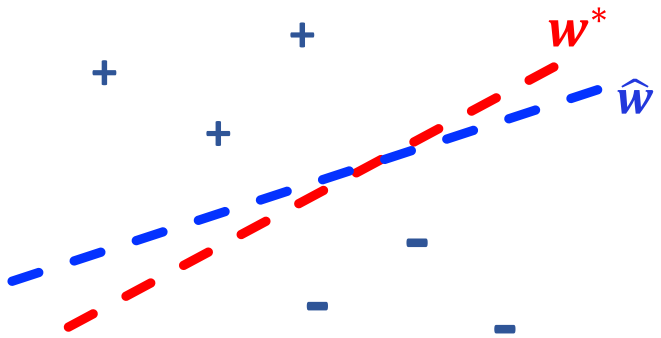
Crowdsourced PAC learning [ABHM17]

- y is not given

Crowdsourced PAC Learning

Standard PAC learning [Valiant 84]

- Given samples (x, y)
- Assume $y = h(x)$ for some unknown hypothesis $h \in H$



- Goal: $\widehat{\boldsymbol{w}}$ is **P**robably **A**pproximately **C**orrect.

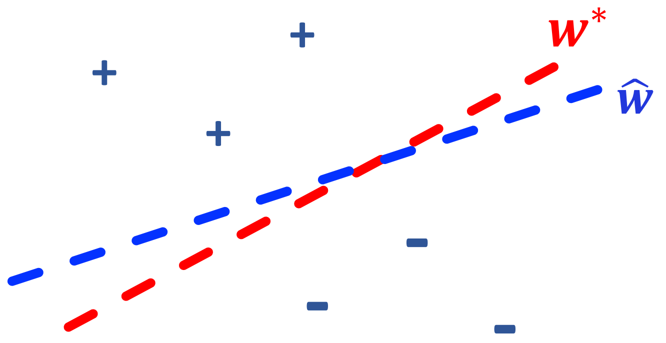
Crowdsourced PAC learning [ABHM17]

- y is not given
- But can collect $\{y_1, \dots, y_k\}$ from crowd

Crowdsourced PAC Learning

Standard PAC learning [Valiant 84]

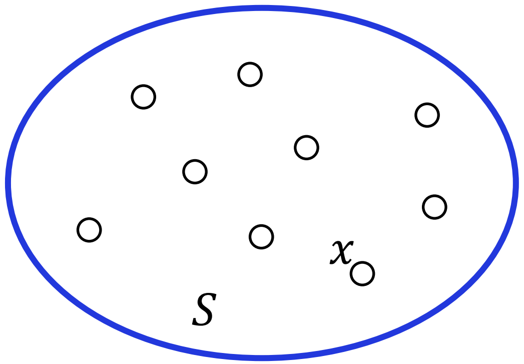
- Given samples (x, y)
- Assume $y = h(x)$ for some unknown hypothesis $h \in H$



- Goal: $\hat{w}$ is **P**robably **A**pproximately **C**orrect.

Crowdsourced PAC learning [ABHM17]

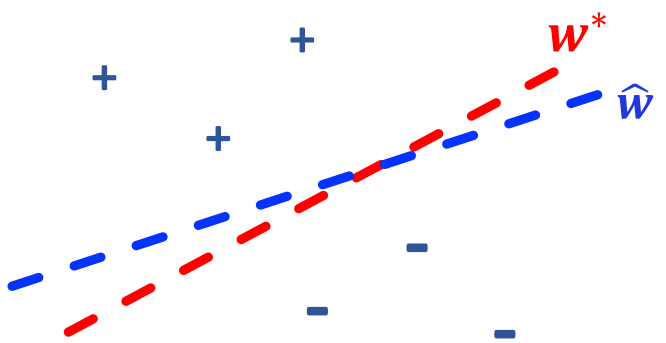
- y is not given
- But can collect $\{y_1, \ldots, y_k\}$ from crowd



Crowdsourced PAC Learning

Standard PAC learning [Valiant 84]

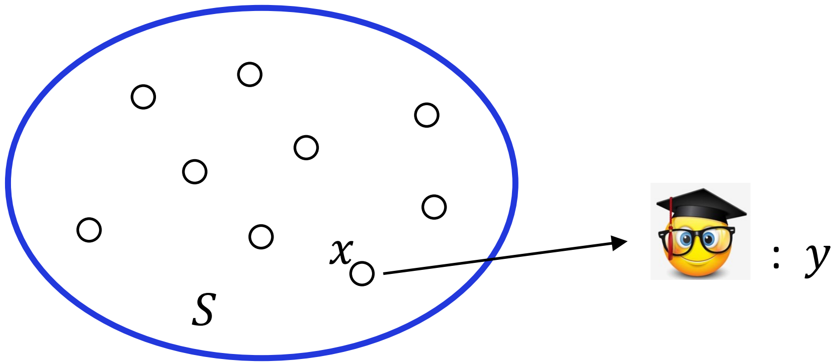
- Given samples (x, y)
- Assume $y = h(x)$ for some unknown hypothesis $h \in H$



- Goal: $\hat{w}$ is **P**robably **A**pproximately **C**orrect.

Crowdsourced PAC learning [ABHM17]

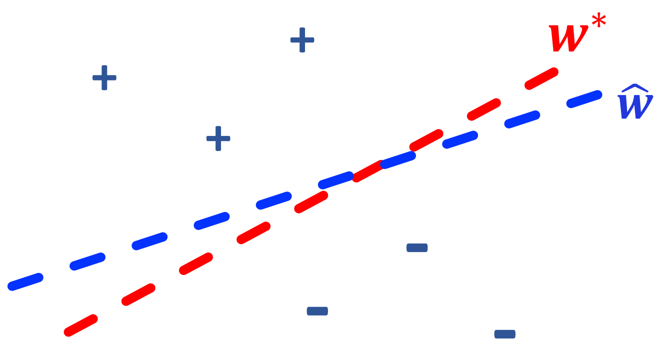
- y is not given
- But can collect $\{y_1, \dots, y_k\}$ from crowd



Crowdsourced PAC Learning

Standard PAC learning [Valiant 84]

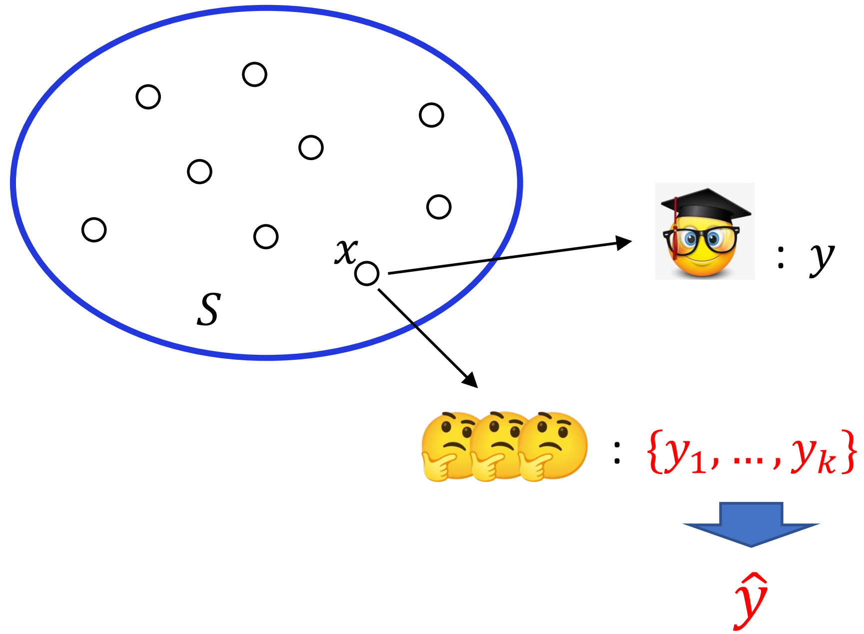
- Given samples (x, y)
- Assume $y = h(x)$ for some unknown hypothesis $h \in H$



- Goal: $\hat{w}$ is **P**robably **A**pproximately **C**orrect.

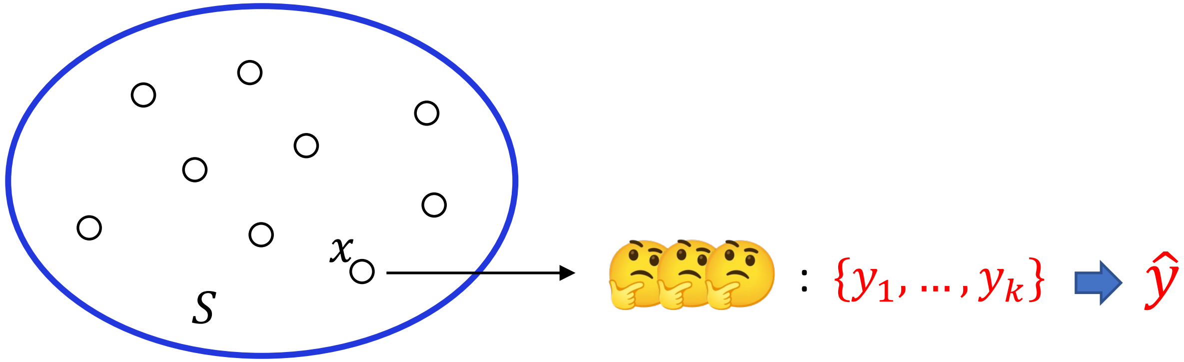
Crowdsourced PAC learning [ABHM17]

- y is not given
- But can collect $\{y_1, \dots, y_k\}$ from crowd

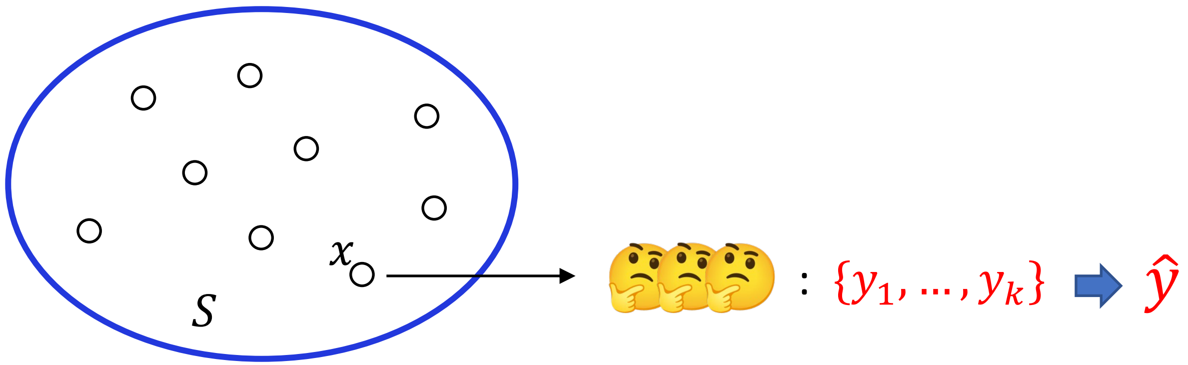


Motivation

Motivation

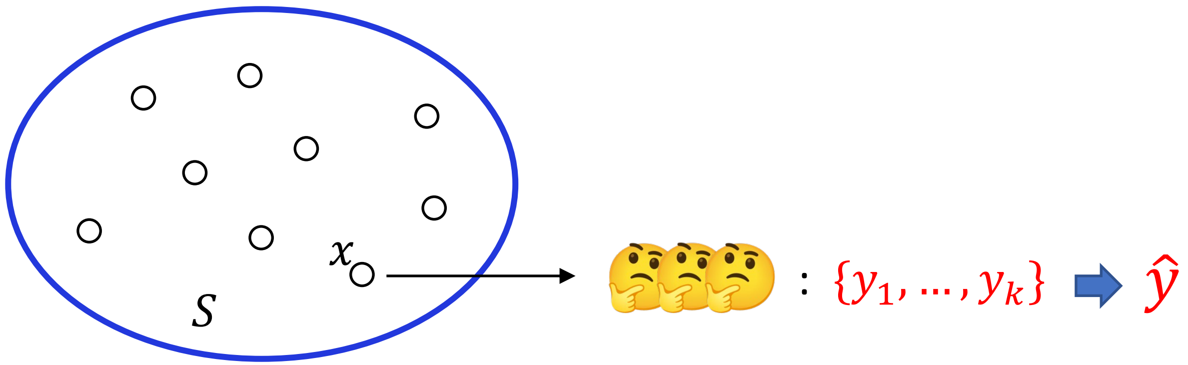


Motivation



- Label-and-train: $k = \log m$ for each x.

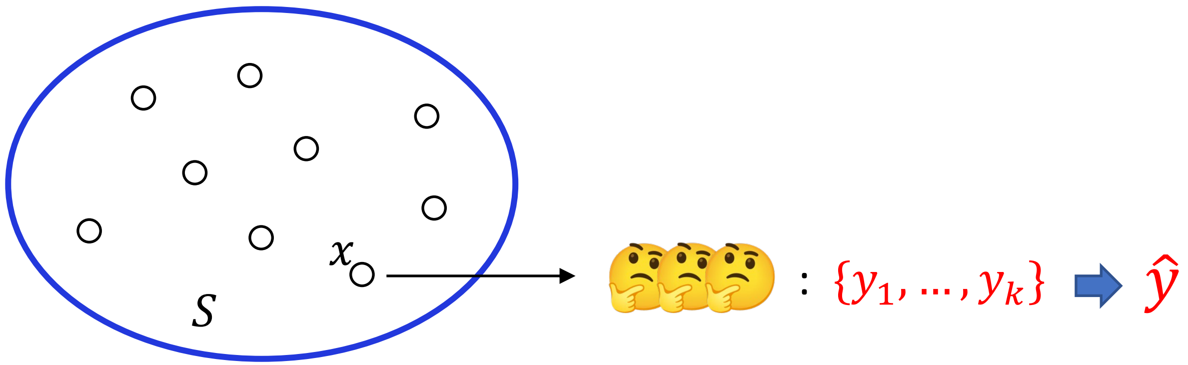
Motivation



- Label-and-train: $k = \log m$ for each x.

- Can we achieve $\boldsymbol{k} = \boldsymbol{O(1)}$?

Motivation



- Label-and-train: $k = \log m$ for each x.

- Can we achieve $\boldsymbol{k} = \boldsymbol{O}(\mathbf{1})$? $\rightarrow$ **Query-efficient.**

Motivation

Motivation

- *Recommendation system*:

Motivation

- *Recommendation system*:

Motivation

- *Recommendation system*:

Prefer which one?

Motivation

- *Recommendation system*:

Prefer which one?

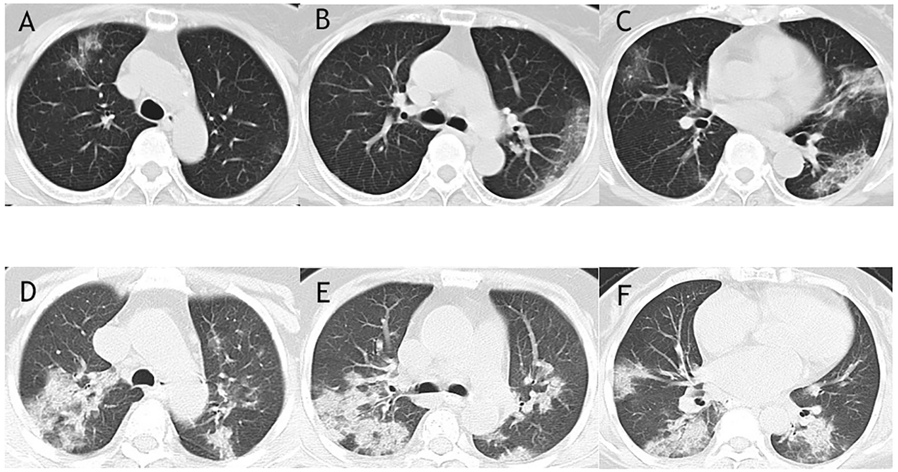
- *Covid-19 diagnosis*:

Motivation

- *Recommendation system*:

Prefer which one?

- *Covid-19 diagnosis*:

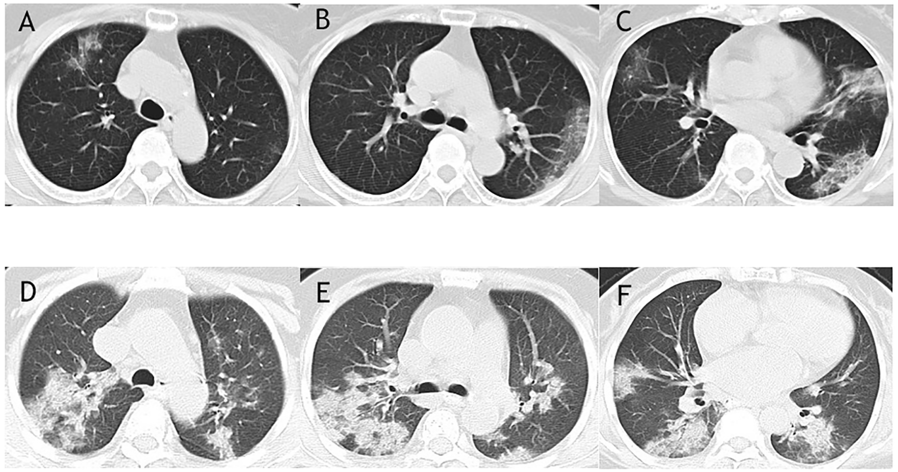


Motivation

- *Recommendation system*:

Prefer which one?

- *Covid-19 diagnosis*:



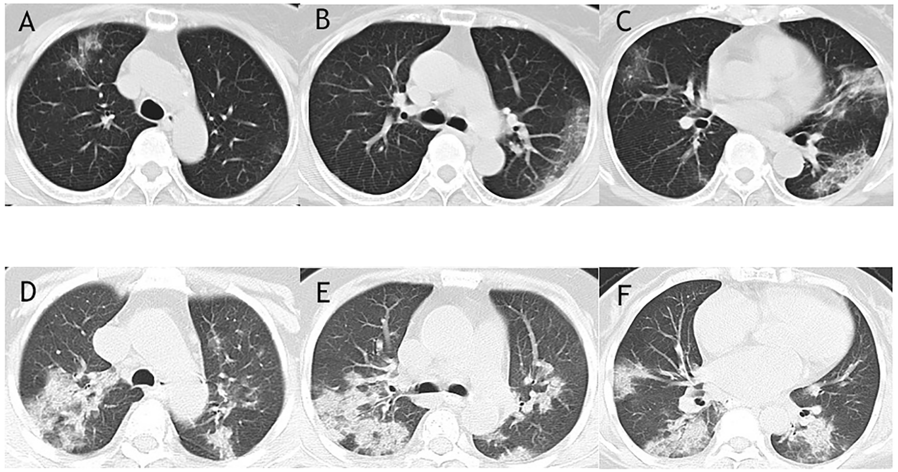
Which one is healthier?

Motivation

- *Recommendation system*:

Prefer which one?

- *Covid-19 diagnosis*:



Which one is healthier?

- **Pairwise comparisons**: $f(x) >^{?} f(x')$

Our results

Our results

- PAC guarantees: w.p. **1-$\boldsymbol{\delta}$**, error $\leq \boldsymbol{\epsilon}$.

Our results

- PAC guarantees: w.p. **1-$\boldsymbol{\delta}$**, error $\leq \boldsymbol{\epsilon}$.
- No distributional assumptions.

Our results

- PAC guarantees: w.p. **1-$\boldsymbol{\delta}$**, error $\leq \boldsymbol{\epsilon}$.
- No distributional assumptions.
- **Query** and **label-efficient**.

Our results

- PAC guarantees: w.p. **1-$\boldsymbol{\delta}$**, error $\leq \boldsymbol{\epsilon}$.
- No distributional assumptions.
- **Query** and **label-efficient.**
- #Labels = $\tilde{O}(\log \frac{d+1/\delta}{\epsilon})$, #pairwise comparisons = $\tilde{O}(\frac{d+(1/\delta)^{1/1000}}{\epsilon})$.

Our results

- PAC guarantees: w.p. **1-$\boldsymbol{\delta}$**, error $\leq \boldsymbol{\epsilon}$.
- No distributional assumptions.
- **Query** and **label-efficient.**
- #Labels = $\tilde{O}(\log \frac{d+1/\delta}{\epsilon})$, #pairwise comparisons = $\tilde{O}(\frac{d+(1/\delta)^{1/1000}}{\epsilon})$.

#Label per instance:

$$\boldsymbol{\Lambda_L = \frac{\epsilon}{d}\tilde{O}(\log\frac{d}{\epsilon})}$$

Our results

- PAC guarantees: w.p. **1-δ**, error $\leq \epsilon$.
- No distributional assumptions.
- **Query** and **label-efficient**.
- #Labels = $\tilde{O}(\log \frac{d+1/\delta}{\epsilon})$, #pairwise comparisons = $\tilde{O}(\frac{d+(1/\delta)^{1/1000}}{\epsilon})$.

#Label per instance:

$$\boldsymbol{\Lambda_L = \frac{\epsilon}{d}\tilde{O}(\log \frac{d}{\epsilon})}$$

#Comparison per instance:

$$\boldsymbol{\Lambda_C = \tilde{O}(\sqrt{\epsilon}\log^2 \frac{d}{\epsilon} + 1)}$$

Our results

- PAC guarantees: w.p. **1-$\boldsymbol{\delta}$**, error $\leq \boldsymbol{\epsilon}$.
- No distributional assumptions.
- **Query** and **label-efficient**.
- #Labels = $\tilde{O}(\log \frac{d+1/\delta}{\epsilon})$, #pairwise comparisons = $\tilde{O}(\frac{d+(1/\delta)^{1/1000}}{\epsilon})$.

#Label per instance:

$$\boldsymbol{\Lambda_L = \frac{\epsilon}{d}\tilde{O}(\log\frac{d}{\epsilon})}$$

#Comparison per instance:

$$\boldsymbol{\Lambda_C = \tilde{O}(\sqrt{\epsilon}\log^2\frac{d}{\epsilon} + 1)}$$

- When $\epsilon \to 0$, $\Lambda_L = o(1)$, $\Lambda_C = O(1)$.

Main techniques

Main techniques

- Comparison-based ***sorting*** algorithm.

Main techniques

- Comparison-based ***sorting*** algorithm.
- Query and label-efficient ***filtering*** algorithm.

Main techniques

- Comparison-based ***sorting*** algorithm.
- Query and label-efficient ***filtering*** algorithm.

See you at poster session!