Discovering Voting Power for Ensemble Methods

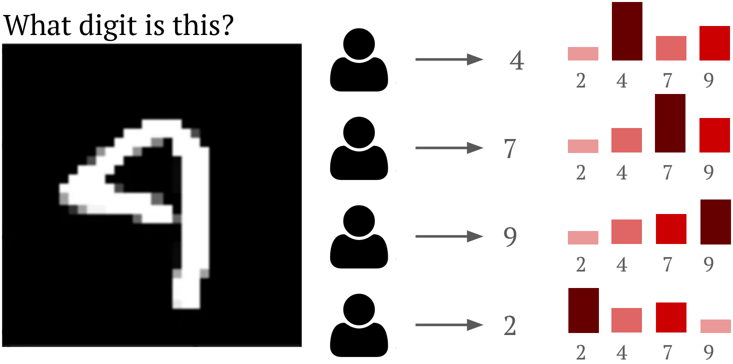
Pratik Karmakar[1,3] Angelo Saadeh[1]
Pierre Senellart[1,2,4,5] Stéphane Bressan[3,6*]

1 cnrs @CREATE Singapore
2 ENS ÉCOLE NORMALE SUPÉRIEURE | PSL
3 NUS National University of Singapore
4 cnrs
5 IUF institut universitaire de France
6 ipal

DEXA 2025

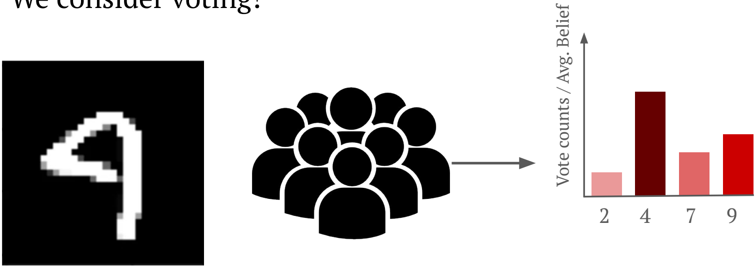
[*] In memory of Stéphane Bressan

What digit is this?



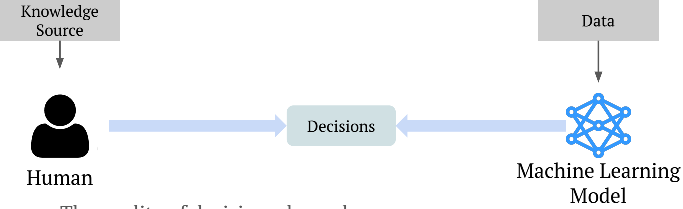
What do we consider?

We consider voting!



We resort to ‘4’ in a democratic way!

Humans and Machine Learning Models



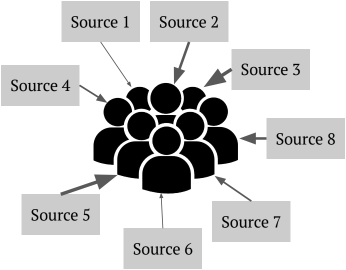
The quality of decisions depends on:

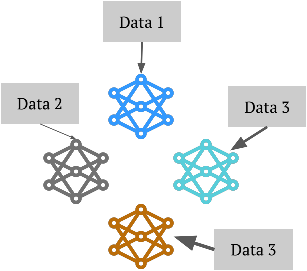
1. Quality of knowledge (data)
2. Capability of learning (model capacity and complexity)
3. Ways (algorithms) of learning

etc.

And also on the **problem** to solve.

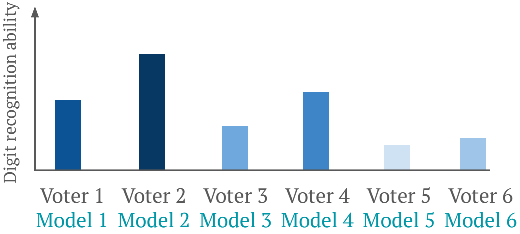
ML Models can vote too (like us)!





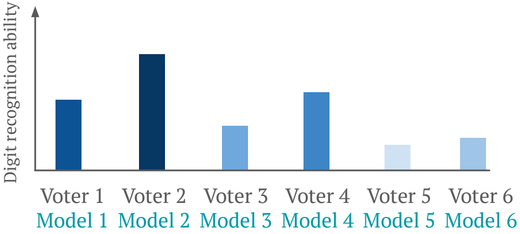
Now, like people, these models give us bunch of decisions for every question asked!

But can all of them recognize digits equally well?



Should we consider simple vote counts in this case?

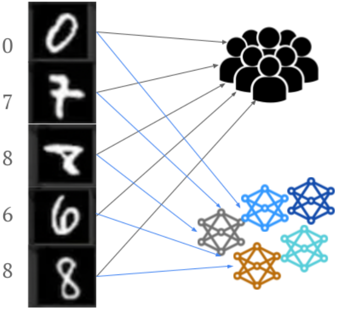
Should we consider simple vote counts?



More importantly, how do we quantify ‘digit recognition ability’?

How do we quantify 'digit recognition ability'?

One way is to look at track records: Supervised Methods



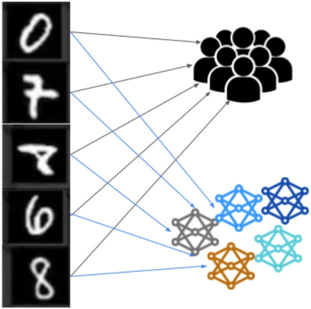
We use the performance metrics of the human/model to quantify its 'ability'.

Possible metrics:

- Accuracy
- Shapley value
- Leave One Out score
- Regression weights

How do we quantify ‘digit recognition ability’?

The other way is to rely on confidence and consensus: Unsupervised Methods



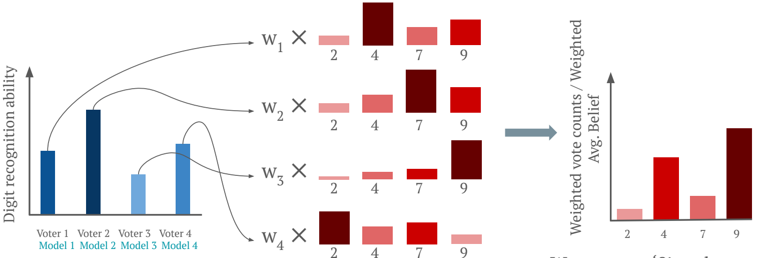
We use the performance metrics of the human/model to quantify its ‘ability’.

Possible metrics:

- Inverse Entropy (Confidence based)
- CRH (Consensus based)

Weighted voting is the way out!

These digit recognition abilities are now called weights (w)



We resort to '9' as the answer.

Why voting power for ensembles?

- Ensemble methods: commonly used way to aggregate the predictions of different machine learning models (e.g., classifiers)
- Ensembles often use equal weights – may treat weak and strong models equally.
- Assigning different voting power can improve aggregate accuracy.
- We explore supervised and unsupervised methods to discover voting power.

Contributions

- Propose and evaluate multiple voting-power discovery methods (incl. two original proposals: Shapley for voting power and inverse-entropy).
- Compare supervised (Accuracy, Regression, Shapley, LOO) and unsupervised (CRH truth discovery, Inverse Entropy) approaches.
- Empirical evaluation across MNIST, CINIC-10, URL classification (DMOZ), Phishing datasets; study accuracy vs. running time.

Voting mechanisms

Plurality (weighted): winner $= \arg\max_k \sum_i w_i \mathbb{1}_{M_i(q)=k}$

Borda (weighted): winner $= \arg\max_k \sum_i w_i \left(\vec{M}_i(q)\right)_k$

- Plurality uses hard labels; Borda uses soft scores (vectors).
- Choice of mechanism affects which prediction format you need.

Supervised methods (require labeled validation set $\mathcal{F}$)

Accuracy $w_i = \frac{1}{|\mathcal{F}|} \sum_{q \in \mathcal{F}} 1_{M_i(q)=\gamma(q)}$.

Regression Solve $\min_w \mathbb{E}_{q \in \mathcal{F}} \|w \times \mathrm{M}(q) - \gamma(q)\|_2^2$.

Shapley values Attribution of contribution to ensemble accuracy (exponential cost).

Leave-One-Out (LOO) $\tilde{w}_i = v(\{1..n\}) - v(\{1..n\} \setminus \{i\})$ (practical proxy).

Unsupervised methods (no ground-truth labels)

Inverse Entropy $w_i = \frac{|\mathcal{F}|}{\sum_{q \in \mathcal{F}} H(\vec{M}_i(q))}$ – models with lower mean entropy get higher weight.

CRH Truth Discovery Iteratively estimate truths y_q and weights w_i by minimizing $\sum_i w_i \sum_q 1_{y_q \neq M_i(q)}$ with a normalization constraint.

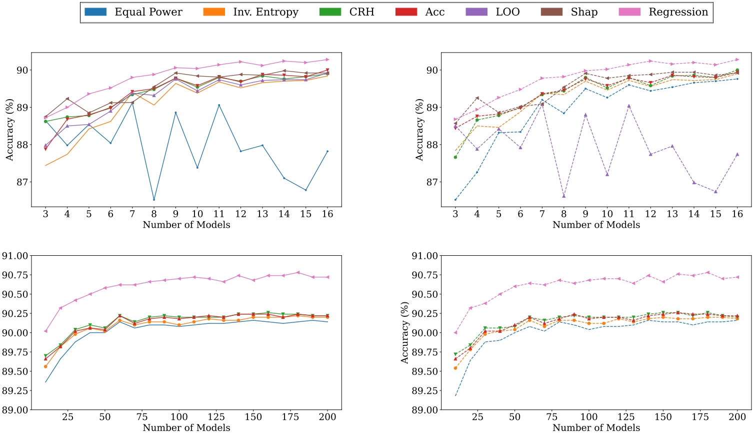
Datasets and model variations

- Datasets: MNIST, CINIC-10, DMOZ (URL classification), Webpage Phishing.
- Create ensembles of models with different quality by: **Label flipping** and **Class imbalance**.
- Validation set $= \mathcal{F}$ (used to compute weights). Test set for final evaluation.
- Repeated runs for statistical significance (MNIST: 75 runs; others: 20 runs).

Implementation details

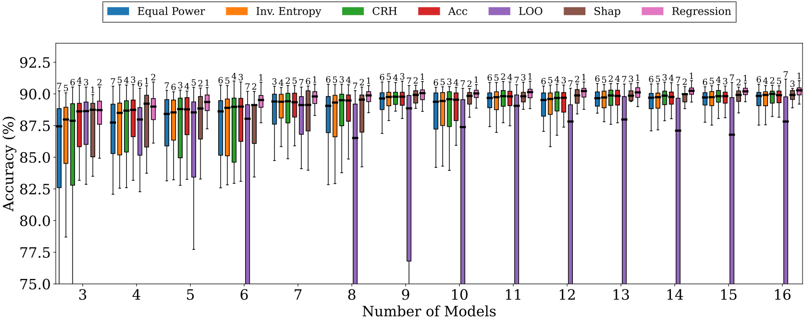
- Classifiers: logistic regression (MNIST), CNN/VGG16 (CINIC-10), MNB (DMOZ), 3-layer NN (Phishing).
- Regression solved with gradient descent; Shapley computed where feasible (small n); LOO used as approximation.
- Measure: test accuracy and runtime for weight computation (log-scale).

MNIST – up to 16 / 200 models (Borda and Plurality)



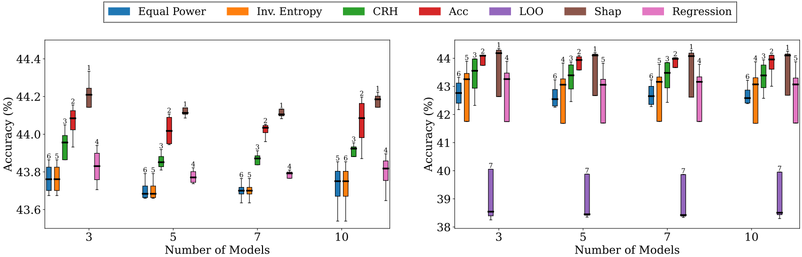
Performance of the power-assigning methods in Borda (left) and Plurality (right) settings for up to 16 models (top) and up to 200 models (bottom).

MNIST – boxplot (method comparison)



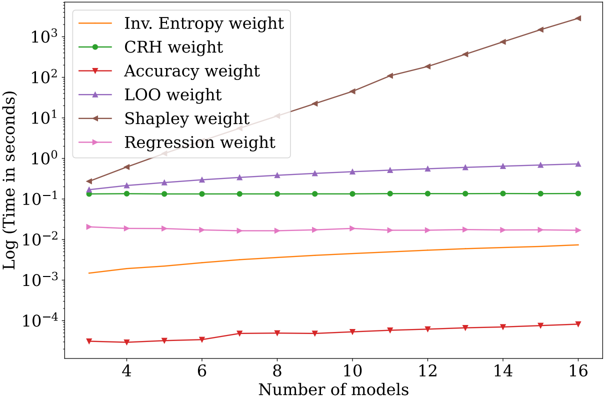
Borda voting with different voting power assigning methods.

URL Classification (DMOZ) results



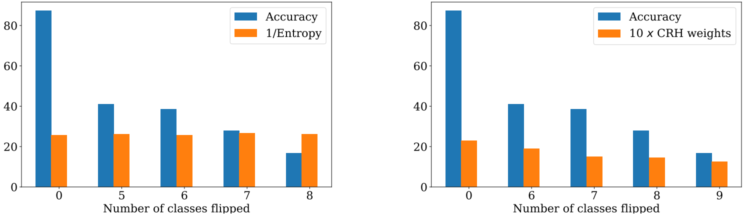
Performance of the power-assigning methods in Borda (left) and Plurality (right) settings for label-flipping noise (DMOZ data)

Time comparison for weight computation (MNIST)



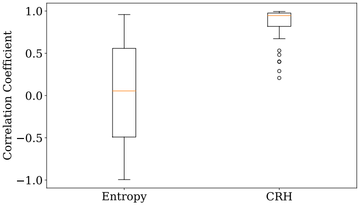
Time comparison of the power-assigning methods. The y-axis uses a logarithmic time scale.

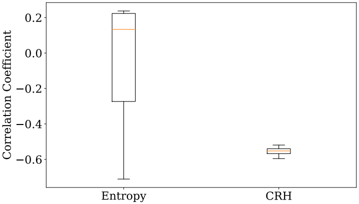
Failure modes: inverse entropy vs CRH



Comparison of the effect of label-interchange in training data on inverse entropy weights (left) and CRH weights (right) (MNIST data)

Failure modes: inverse entropy vs CRH





Correlation between model accuracy and model voting powers computed using inverse entropy and CRH when the complete label interchange pattern is random (left) or deterministic (right) across different training datasets (MNIST data)

Key takeaways

- Learning weights (regression, Shapley) generally outperforms equal voting.
- Shapley is strong but costly; regression is a practical top performer.
- Unsupervised CRH is competitive and more robust than inverse entropy in some failure modes.
- Choice of voting mechanism (Borda vs Plurality) has limited effect compared to weight discovery choice.

Conclusion & Future Work

- Power-assigned voting improves ensembles; regression and Shapley are top supervised methods.
- CRH offers a good unsupervised alternative; inverse entropy is cheap but brittle.
- Future: apply in privacy-preserving knowledge transfer (PATE-like setups), analyze sensitivity and noise addition.

Questions?

Thank you!

Questions welcome.

Contact: `pratik.karmakar@u.nus.edu`