SUPPLEMENTAL MATERIAL
Filtering participants improves generalization in competitions and benchmarks

1 Estimation of k*

Our analysis makes a common assumption for real and synthetic data: rankings are drawn in all phases from the same distribution of rankings.

A first result applicable to real and synthetic data is that **the first and the last points of the Meta-generalization curves have identical score, which corresponds to the performance of the vanilla method**. Trivially, the result for $k = n$ is that of the vanilla method (select winner in Final phase among all participants). For $k = 1$, we select the participant winning the Development phase. But, since the rankings in the Final phase and the Development phase are drawn from the same distribution, in expectation, the performance is going to be the same as that of the vanilla method.

We propose and simple empirical formula for the optimum of k: $k^* \simeq 1+d/n$, with d the Kendall τ distance between D and F. In Figure 1, we validate the formula with simulations using synthetic data and position the optimal k observed in real data, for comparison. We observe that, on synthetic data, the formula fares well, but there is quite a bit of variance. On real data, for all meta-datasets, except OpenML, the true optimum is between $1 + d/n$ and $1 + 2d/n$. For OpenML, the optimum is obtained for a smaller value of k. These results suggest that choosing $k^* \simeq 1 + d/n$ should provide best meta-generalization, on average, but with a risk of biasing results, due to the large variance. Hence a more conservative choice of k, such as $1 + 2d/n$ would be safer.

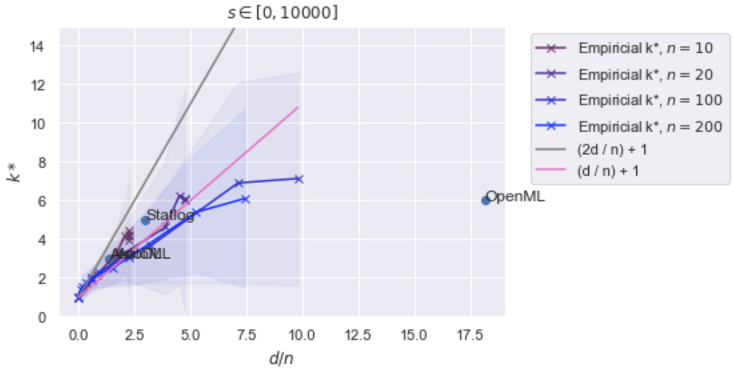

Fig. 1: **Experimental evaluation of** k^*. k^* is both estimated empirically (mean and std shown) and predicted by the empirical formula $k^* \simeq 1 + d/n$.

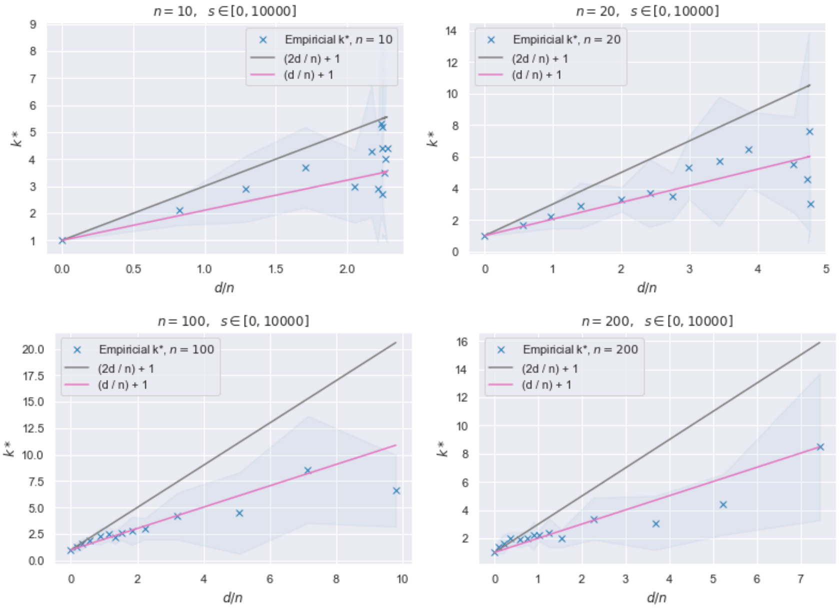

Fig. 2: Experimental evaluation of k^*, separated in for plots for $n = 10$, $n = 20$, $n = 100$ and $n = 200$.

2 Formal framework

To formally compare the top-k method with the vanilla method, we use our synthetic example obtained by swapping neighbors, starting from an ideal true ranking. We first formalize the problem as that of maximizing the probability of finding the winner with the top-k method (Section 2.1). We then decompose the probability of finding the winner with the top-k method (Section 2.2), which we call method “accuracy” $acc(k)$ in two factors playing the role of training accuracy and generalization gap. We compute the accuracy of the vanilla method and show that it is equal to $acc(1) = acc(n)$, for n participants. We (try to) prove that, under some conditions, $acc(k)$ goes through an optimum as a function of k and we propose and simple empirical formula for the optimum of k: $k^* \simeq 1 + d/n$.

2.1 Problem definition

We consider a competition with $n \in \mathbb{N}$ participants. We name the n participants as $\{1, 2, ..., n\}$ where their name corresponds to their true but unknown ranking:

$$g = [1, 2, ..., n]$$

An empirical ranking obtained in a competition phase is assumed to be obtained from g by repeated permutations of pairs of neighbors. A position i is drawn at random from $\{1, \cdots, n-1\}$ and the participants i and $i+1$ are inverted. We repeat this operation s times. The smaller s, the more the empirical rankings will be correlated to the true ranking.

We call D the random variable (RV) corresponding to a ranking drawn from the previously described process, for the Development phase and F the RV corresponding the that of the Final phase, drawn similarly.

The goal is now to maximize the probability of picking the winner with the top-k method.

The degree of correlation between D and F is governed by $\phi = \frac{s}{n}$. In practice, we estimate the correlation by computing the Kendall τ distance d between D and F, which is interesting because we can compute it in real case scenarios.

The participant i^* selected by the top-k method has rank j^* :

$$j^* = \arg\min_{j \leq k} F^{-1}(D(j)).$$

Using:

$$D(j) = i$$
$$D^{-1}(i) = j$$

we get:

$$i^* = \arg\min_{D^{-1}(i) \leq k} F^{-1}(i).$$

The choice of the top-k method is the REAL winner *iff* $i^* = 1$, that is the identity if the winner selected with the top-k method is the true winner.

The problem is to maximize the probability $acc(k)$ that the declared winner in the Final phase (using the top-k method) is the true winner (*acc* stands for accuracy). The problem is formalized as follows:

$$k^* = \arg\max_{k} acc(k)$$

with:

$$\boxed{acc(k) = \texttt{Proba}[\arg\min_{D^{-1}(i) \leq k} F^{-1}(i) = 1]} \tag{1}$$

2.2 Calculation of acc(k)

2.2.1 Vanilla method

We first evaluate the value of the first and the last point of the curve $acc(k=1)$ and $acc(k=n)$, corresponding to the *vanilla* method. If $k=1$, we choose the winner in the *development phase* as our winning candidate. If $k=n$, we choose the winner in the *final phase.*

$$acc(k=1) = \texttt{Proba}[D^{-1}(1) = 1]$$

$$acc(k=n) = \texttt{Proba}[F^{-1}(1) = 1]$$

The probability that this is the true winner is identical in both cases since the processes to generate D and F are identical:

$$acc(k=1) = acc(k=n) = \texttt{Proba}[D^{-1}(1) = 1] = \texttt{Proba}[F^{-1}(1) = 1]$$

We can notice that $D^{-1}(1) = 1$ occurs if the winner (that is the participant with first position in g) does not move in the s swapping trials, or if it moves forward then backward to return to its original position.

We can therefore model the movements of the candidate by a Markov chain with each state representing a possible position, and a transition matrix T as represented by Figure 3.

$$T = \begin{pmatrix} 1-p & p & 0 & 0 & ... & 0 \\ p & 1-2p & p & 0 & ... & 0 \\ 0 & p & 1-2p & p & ... & 0 \\ ... & ... & ... & ... & ... & p \\ 0 & 0 & 0 & 0 & p & 1-p \end{pmatrix}$$

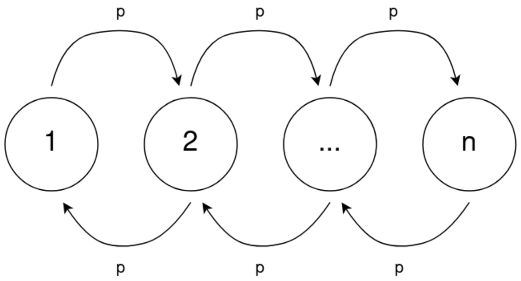

Fig. 3: Definition of the transition matrix (left) and scheme of the Markov chain (right). $p = \frac{1}{(n-1)}$. The probabilities of staying in the same state are not shown for simplicity.

The probability $P_{ij}(s)$, the probability to reach the position j from the position i after s steps on the Markov chain can be computed using a matrix power, according to the Chapman-Kolmogorov equation:

$$P_{ij}(s) = (T^s)_{ij}$$

Considering that the probability of being involved in a swap at each time step is $\frac{1}{n-1}$ at the bounds and $\frac{2}{n-1}$ for any other nodes, the probability of transition to a different state is given by $p = \frac{1}{n-1}$. The probability that the winner stays the winner (going from 1 to 1) after s swaps is given by:

$$\boxed{acc(k=1) = acc(k=n) = P_{11}(s)}$$

More generally we can compute the probability of ending in a position j for any candidate i, after s swaps, using:

$$\texttt{Proba}[D^{-1}(i) = j] = P_{ij}(s) \tag{2}$$

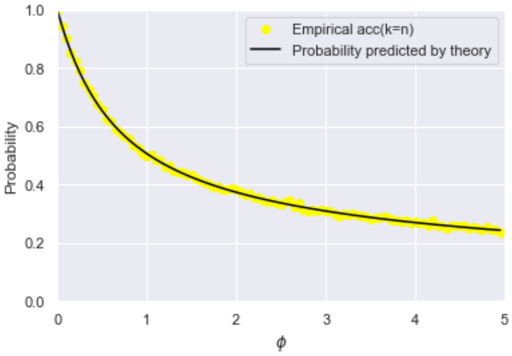

Fig. 4: Simulated results. The yellow dots represent the empirical probability acc(1) or acc(n) and the black line represents the value predicted by the formula. Performed with $n = 20$. $\phi = \frac{s}{n}$ and represents the degree of perturbation of rankings.

2.3 Decomposition of acc(k)

Need to invoke Bayes theorem – Isabelle

We decompose Equation 1 into 2 factors: the probability that the candidate selected is the true winner selected, when we know that the true winner is in the top-k of the development phase, and the probability that the the true winner is in the top-k. Finding the winner in the top-k of the development phase can be written as $D^{-1}(1) \leq k$, hence:

$$\boxed{\mathbf{acc(k)} = \texttt{Proba}[\arg \min_{\mathbf{D^{-1}(i) \leq k}} \mathbf{F^{-1}(i) = 1 \mid D^{-1}(1) \leq k}] \ \texttt{Proba}[\mathbf{D^{-1}(1) \leq k}]} \tag{3}$$

Prouver que le deuxieme facteur est une sorte de trainign error. – Isabelle

Justifier le fait d'avoir enlevÃ© de la formule le cas ou le gagnant n'est pas dans le top-k (proba = 0)
Ajouter accolades pour dÃ©finir les termes
Noter cas phi=0 et phi=inf : courbe plate – Adrien

The probabilities given by Equation 2 can be added to obtain the probability of ending in a set of position (e.g. between 0 and k). Thus, the probability $\texttt{Proba}[D^{-1}(1) \leq k]$ that the true winner is in the development top-k is given by:

$$
\begin{aligned}
P_{topk} &= \texttt{Proba}[D^{-1}(1) \leq k] \\
P_{topk} &= \texttt{Proba}[D^{-1}(1) = 1] + \texttt{Proba}[D^{-1}(1) = 2] + ... + \texttt{Proba}[D^{-1}(1) = k] \\
P_{topk} &= P_{11}(s) + P_{12}(s) + ... + P_{1k}(s) \\
P_{topk} &= \sum_{j=1}^{k} P_{1j}(s)
\end{aligned}
$$

Ca ne nous sert pas vraiment. Ce qu'il faut c'est une formule simple en fonction de ϕ. – Isabelle

To estimate the probability that the true winner is selected when we know it is in the dev top-k, we compute and add together:

- First, the probability that the true winner is ends up first (in F)

- Then, for each other candidates c, their probability of NOT being in development top-k ($1 - \texttt{Proba}[D^{-1}(c) \leq k]$) multiplied by their probability of ending up first ($\texttt{Proba}[D^{-1}(c) = 1]$)

As an approximation and for simplicity, the probabilities of these events are computed independently.

$$
\begin{aligned}
P_k &= \texttt{Proba}[\arg \min_{D^{-1}(i) \leq k} F^{-1}(i) = 1 \mid D^{-1}(1) \leq k] \\
P_k &\approx P[D^{-1}(1) = 1] + \sum_{i=1}^{n} \Big(1 - P[D^{-1}(i) \leq k]\Big) \times P[D^{-1}(i) = 1] \\
P_k &\approx P_{11}(s) + \sum_{i=1}^{n} \sum_{j=1}^{k} \big(1 - P_{ij}(s)\big) \times P_{i1}(s)
\end{aligned}
$$

A better approximation may be to compute$\texttt{Proba}[D^{-1}(1) = 1]$ and $\texttt{Proba}[D^{-1}(i) = 1]$ on a k size version of D, simulating F. Doing that should correct a bit the overestimation of p(k) – Adrien

All together, factorizing more:

$$
acc(k) \approx \sum_{j=1}^{k} P_{1j}(s) \Big(P_{11}(s) + \sum_{i=1}^{n} \sum_{j=1}^{k} \big(1 - P_{ij}(s)\big) \times P_{i1}(s) \Big)
$$

Or, using matrix power:

$$\boxed{P(k) \approx \sum_{j=1}^{k} T^s_{1j} \left(T^s_{11} + \sum_{i=1}^{n} \sum_{j=1}^{k} (1 - T^s_{ij}) \times T^s_{i1} \right)} \tag{4}$$

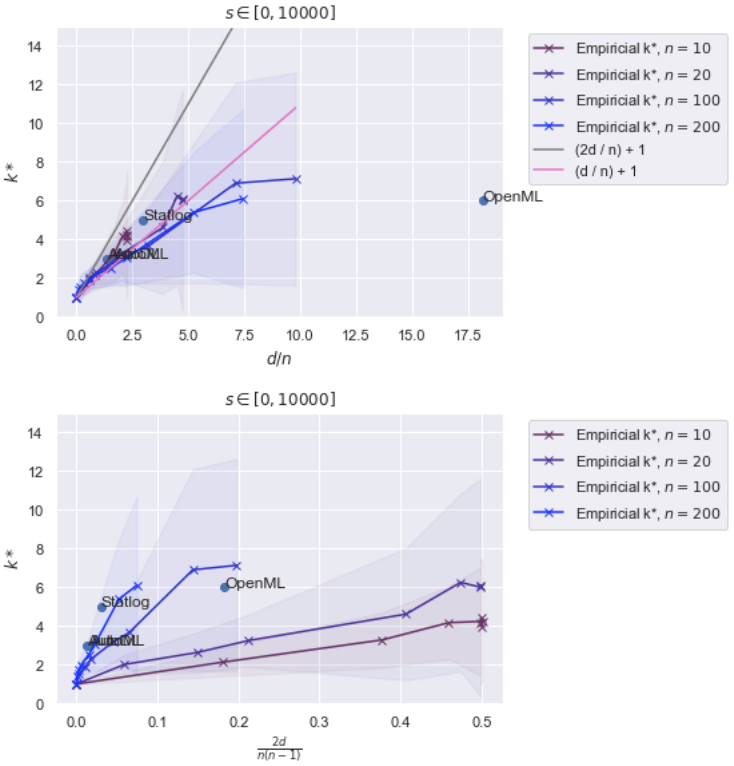

Fig. 5: $k*$ estimated empirically (mean and std showed) and predicted by the approximations.

Faire k* en fonction de d/n pour rank error et pour NLA. Faire varier n. Rajouter les points correpondant aux donnÃ©e rÃ©elles. – Isabelle

2.4 Analysis of $acc(k)$

In this section, we prove that, under certain conditions, $acc(k)$ goes through an optimum. We derive the optimal value k^* of k.

Justify, interpret, comment this approximation
On doit pouvoir trouver une justification “avec les mains”, notamment d’une borne supÃ©rieure, en reliant d/n au nombre de swaps subits par chaque candidat en moyenne. Avec k = 2d/n + 1 on devrait avoir une grande probabilitÃ© d’avoir le gagnant dans le top-k. Si on prend un k ¡ n et qu’on a une bonne garantie d’avoir le gagnant dans le top-k, alors on ne devrait pas pouvoir dÃ©grader nos rÃ©sultats.
Il faudrait aussi gÃ©nÃ©raliser au top 3 comme c’est le cas dans la plupart des compÃ©titions, mais nous n’avons pas assez de temps pour nous lancer lÃ -dedans. – Adrien

2.4.1 $acc(1) < acc(2)$

Let’s now compare

$$acc(2) = \mathbb{P}\left(\arg \min_{D^{-1}(i)\leq 2} F^{-1}(i) = 1\right) = \mathbb{P}\left(\min F^{-1}D(\{1,2\}) = F^{-1}(1)\right)$$

and

$$acc(1) = \left(\arg \min_{D^{-1}(i)\leq 1} F^{-1}(i) = 1\right) = \mathbb{P}(D(1) = 1).$$

Let’s first consider the index set

$$I := \{(i_1, i_2) | i_1 \neq i_2, 1 \leq i_1, i_2 \leq n\} \tag{5}$$

with $|I| = n(n-1)$ and define the transition matrix (note here T^s is a notation, not a power; we could also use the notation $T^{(s)}$ instead of T^s for clarity)

$$T^s_{(i_1,i_2),(j_1,j_2)} := \mathbb{P}\left(D(j_1) = i_1, D(j_2) = i_2\right) \tag{6}$$

or more conveniently

$$T^s_{ij} = \mathbb{P}(D(j) = i)$$

where $i = (i_1, i_2), j = (j_1, j_2) \in I$ and

$$D(j) = D((j_1, j_2)) := (D(j_1), D(j_2)).$$

By definition we immediately have

$$T^s_{(i_1,i_2),(j_1,j_2)} = T^s_{(i_2,i_1),(j_2,j_1)}.$$

We also note that $T^s \in \mathbb{R}^{n(n-1)\times n(n-1)}$ is a Markov matrix

$$\sum_{i\in I} T^s_{ij} = 1, \forall j \in I.$$

By the definition of the random variable D, one can easily use the property of transition matrices to prove following proposition.

Proposition 1. *For $D = \Sigma_1 \circ ... \circ \Sigma_s$ with $\Sigma_i \overset{iid}{\sim} \mathcal{U}(\{(1,2),(2,3),...,(n-1,n)\})$ and $T^s_{ij} = \mathbb{P}(D(j) = i)$ with $i, j \in I$ as defined in (5), then we have*

$$T^s = (T^1)^s$$

and

$$T^1_{ij} = \frac{1}{n-1} \sum_{k=1}^{n-1} \mathbb{1}\left(\sigma_k(j) = i\right)$$

with σ_k being the swap $(k, k+1)$. Furthermore, T^s is symmetric, i.e. $T^s_{ij} = T^s_{ji}, \forall i, j \in I$.

Proof. We have

$$T^s_{ij} = \sum_{k \in I} \mathbb{P}(\Sigma_s(j) = k, \Sigma_1 \circ ... \circ \Sigma_{s-1}(k) = i) = \sum_{k \in I} \mathbb{P}(\Sigma_s(j) = k) \cdot \mathbb{P}(\Sigma_1 \circ ... \circ \Sigma_{s-1}(k) = i) = \sum_{k \in I} T^{s-1}_{ik} T^1_{kj}$$

and the proposition follows from induction. The fact that T^s is symmetric easily follows from the fact $\sigma_k^{-1} = \sigma_k$. □

With the help of T^s_{ij} defined above, we can now write

$$\begin{aligned} acc(2) &= \mathbb{P}\left(\min F^{-1} D(\{1,2\}) = F^{-1}(1)\right) \\ &= \sum_{i_1 \neq i_2} \mathbb{P}(D(1) = i_1, D(2) = i_2, \min F^{-1} D(\{1,2\}) = F^{-1}(1)) \\ &= \sum_{i_1 \neq i_2} \mathbb{P}(D(1) = i_1, D(2) = i_2, \min F^{-1}(\{i_1, i_2\}) = F^{-1}(1)) \\ &= \sum_{i_1 \neq i_2} \mathbb{P}(D(1) = i_1, D(2) = i_2) \cdot \mathbb{P}(\min F^{-1}(\{i_1, i_2\}) = F^{-1}(1)) \\ &= \sum_{i_1 \neq i_2} T^s_{(i_1,i_2),(1,2)} \cdot \mathbb{P}(\min F^{-1}(\{i_1, i_2\}) = F^{-1}(1)) \end{aligned} \tag{7}$$

For the latter term, we have

$$\begin{aligned} &\mathbb{P}(\min F^{-1}(\{i_1, i_2\}) = F^{-1}(1)) \\ =&\mathbb{P}(F^{-1}(i_1) = F^{-1}(1), F^{-1}(i_1) < F^{-1}(i_2)) + \mathbb{P}(F^{-1}(i_2) = F^{-1}(1), F^{-1}(i_2) < F^{-1}(i_1)) \\ =&\delta_{i_1,1} \mathbb{P}(F^{-1}(1) < F^{-1}(i_2)) + \delta_{i_2,1} \mathbb{P}(F^{-1}(1) < F^{-1}(i_1)) \\ =&\delta_{i_1,1} \sum_{j_1 < j_2} T^s_{(1,i_2),(j_1,j_2)} + \delta_{i_2,1} \sum_{j_1 < j_2} T^s_{(1,i_1),(j_1,j_2)} \end{aligned} \tag{8}$$

So we have

$$\begin{aligned}
acc(2) &= \sum_{i_1\neq i_2} T^s_{(i_1,i_2),(1,2)} \cdot \left(\delta_{i_1,1} \sum_{j_1<j_2} T^s_{(1,i_2),(j_1,j_2)} + \delta_{i_2,1} \sum_{j_1<j_2} T^s_{(1,i_1),(j_1,j_2)}\right) \\
&= \sum_{1\neq i_2} T^s_{(1,i_2),(1,2)} \sum_{j_1<j_2} T^s_{(1,i_2),(j_1,j_2)} + \sum_{i_1\neq 1} T^s_{(i_1,1),(1,2)} \sum_{j_1<j_2} T^s_{(1,i_1),(j_1,j_2)} \\
&= \sum_{i\neq 1} \left(T^s_{(1,i),(1,2)} + T^s_{(i,1),(1,2)}\right) \cdot \left(\sum_{j_1<j_2} T^s_{(1,i),(j_1,j_2)}\right) \\
&= \sum_{i\neq 1} \left(T^s_{(1,i),(1,2)} + T^s_{(1,i),(2,1)}\right) \cdot \left(\sum_{j_1<j_2} T^s_{(1,i),(j_1,j_2)}\right)
\end{aligned} \tag{9}$$

For $acc(1)$, we have

$$\begin{aligned}
acc(1) &= \mathbb{P}(D(1)=1) \\
&= \sum_{i\neq 1} \mathbb{P}(D(1)=1, D(2)=i) \\
&= \sum_{i\neq 1} T^s_{(1,i),(1,2)}.
\end{aligned} \tag{10}$$

By (9) and (10), we can prove the following theorem.

Theorem 2. *For $s=1$, we have*

$$acc(2) > acc(1). \tag{11}$$

Proof. First we have

$$\begin{aligned}
T^1_{(1,i),(1,2)} &= \frac{1}{n-1} \sum_{k=1}^{n-1} \mathbb{1}\left(\sigma_k(1)=1, \sigma_k(2)=i\right) \\
&= \begin{cases} \frac{n-3}{n-1}, \text{ if } i=2, \\ \frac{1}{n-1}, \text{ if } i=3, \\ 0, \text{ otherwise.} \end{cases}
\end{aligned} \tag{12}$$

Thus

$$acc(1) = \frac{n-2}{n-1}.$$

For $acc(2)$, we have

$$\begin{aligned}
acc(2) &= \sum_{i\neq 1}\left(T^s_{(1,i),(1,2)} + T^s_{(1,i),(2,1)}\right)\cdot\left(\sum_{j_1<j_2} T^s_{(1,i),(j_1,j_2)}\right)\\
&= \sum_{i=2}^{3}\left(T^s_{(1,i),(1,2)} + T^s_{(1,i),(2,1)}\right)\cdot\left(\sum_{j_1<j_2} T^s_{(1,i),(j_1,j_2)}\right)\\
&= \left(T^s_{(1,2),(1,2)} + T^s_{(1,2),(2,1)}\right)\cdot\left(\sum_{j_1<j_2} T^s_{(1,2),(j_1,j_2)}\right) + \left(T^s_{(1,3),(1,2)} + T^s_{(1,3),(2,1)}\right)\cdot\left(\sum_{j_1<j_2} T^s_{(1,3),(j_1,j_2)}\right)\\
&= \left(\frac{n-3}{n-1} + \frac{1}{n-1}\right)\cdot\left(1 - \frac{1}{n-1}\right) + \left(\frac{1}{n-1} + 0\right)\cdot 1\\
&= \frac{n-2}{n-1} + \left(\frac{1}{n-1}\right)^2
\end{aligned} \tag{13}$$

Thus

$$acc(2) > \frac{n-2}{n-1} = acc(1).$$

□

Here is how to prove this in a simpler way: (1) acc(1) = the proba that the true winner does not move with 1 swap, ie 1-proba that it moves = $1 - \frac{1}{n-1} = \frac{n-2}{n-1}$; (2) According to formula 6, acc(2) = proba(winner is best in F in top 2 of D) proba(winner in top 2 of D). But proba(winner in top 2 of D) = 1, if only 1 swap. Hence acc(2) = proba(winner is best in F in top 2 of D). Two cases arise: (a) The top ranked in F is the winner. This arises with proba $1 - \frac{1}{n-1} = acc(1)$. (b) the winner moved to another place in F, but it is still the one with the smallest rank in top k of D. There is only 1 way this can happen: D = [1 3 2 ...] and F = [2 1 3 ...]. Then the top 2 are [1 3] and the winner is ranked second in F, but is selected because the top ranked in F is not in the top 2 of D. This happens with proba $\left(\frac{1}{n-1}\right)^2$. QED – Isabelle

Finally, we manage to prove the following theorem.

Theorem 3. *For $s = 1$, the curve $acc(k)$ achieves maximum for some k^* such that $1 < k^* < n$.*

Proof. This follows immediately from

$$acc(2) > acc(1)$$

by Theorem 2 and the fact

$$acc(1) = \mathbb{P}(D(1) = 1) = \mathbb{P}(F(1) = 1) = acc(n).$$

□