#### Vladimir Podolskii

Computer Science Department, Higher School of Economics

### Outline

**Unordered Pairs** 

#### **Ranking Problem**

Suppose we have n texts and we need to rank them by relevance. One of the standard approaches requires to compare each text with each other. How many comparisons we need to make?

#### Solution attempt:

• There are n documents

- There are n documents
- We need to compare each document to each of n-1 other documents

- There are n documents
- We need to compare each document to each of n-1 other documents
- By the rule of product there will be  $n \times (n-1)$  comparisons

- There are n documents
- We need to compare each document to each of n-1 other documents
- By the rule of product there will be  $n \times (n-1)$  comparisons
- Is everything right?

Consider n = 3 and denote texts by a, b and c

• Our solution gives  $3 \times 2 = 6$  comparisons

- Our solution gives  $3 \times 2 = 6$  comparisons
- But actually, there are only 3: a vs. b, a vs. c
  and b vs. c

- Our solution gives  $3 \times 2 = 6$  comparisons
- But actually, there are only 3: a vs. b, a vs. c and b vs. c
- Let's review our solution

- Our solution gives  $3 \times 2 = 6$  comparisons
- But actually, there are only 3: a vs. b, a vs. c
  and b vs. c
- Let's review our solution
- For each text we compare it with 2 others

- Our solution gives  $3 \times 2 = 6$  comparisons
- But actually, there are only 3: a vs. b, a vs. c
  and b vs. c
- Let's review our solution
- For each text we compare it with 2 others
- That is, a is compared to b and c; b is compared to a and c; c is compared to a and b

- Our solution gives  $3 \times 2 = 6$  comparisons
- But actually, there are only 3: a vs. b, a vs. c and b vs. c
- Let's review our solution
- For each text we compare it with 2 others
- That is, a is compared to b and c; b is compared to a and c; c is compared to a and b
- We counted comparisons ab, ac, ba, bc, ca, cb

- Our solution gives  $3 \times 2 = 6$  comparisons
- But actually, there are only 3: a vs. b, a vs. c and b vs. c
- Let's review our solution
- For each text we compare it with 2 others
- That is, a is compared to b and c; b is compared to a and c; c is compared to a and b
- We counted comparisons ab, ac, ba, bc, ca, cb
- We counted each comparison twice!

Now we can fix our solution:

• There are n documents

- There are *n* documents
- We need to compare to each of n-1 other documents

- There are *n* documents
- We need to compare to each of n-1 other documents
- By the rule of product there will be  $n \times (n-1)$  comparisons

- There are *n* documents
- We need to compare to each of n-1 other documents
- By the rule of product there will be  $n \times (n-1)$  comparisons
- We have counted every a vs. b comparison twice: as a vs. b and as b vs. a

- There are *n* documents
- We need to compare to each of n-1 other documents
- By the rule of product there will be  $n \times (n-1)$  comparisons
- We have counted every a vs. b comparison twice: as a vs. b and as b vs. a
- Let's just divide the result by 2

- There are n documents
- We need to compare to each of n-1 other documents
- By the rule of product there will be  $n \times (n-1)$  comparisons
- We have counted every a vs. b comparison twice: as a vs. b and as b vs. a
- Let's just divide the result by 2
- The answer is n(n-1)/2

 We have counted the number of unordered pairs of n objects

- We have counted the number of unordered pairs of n objects
- The answer is n(n-1)/2

- We have counted the number of unordered pairs of n objects
- The answer is n(n-1)/2
- When counting make sure that you count each object once

- We have counted the number of unordered pairs of n objects
- The answer is n(n-1)/2
- When counting make sure that you count each object once
- If you counted each object k times, just divide the result by k

### Outline

**Unordered Pairs** 

#### **Road Trip**

You are planning a road trip. You have 5 friends, but you can fit only 3 of them in your car. How many ways do you have to do it?

- Basically, we want to pick a subset of size 3 in the set of your 5 friends
- We are interested in how many subsets of size
  3 are there
- Recall that elements of subset are unordered

- We can pick the first friend in 5 ways
- We can pick the second friend in 4 ways
- We can pick the third friend in 3 ways
- By the rule of product there are  $5 \times 4 \times 3 = 60$  possibilities
- Is there a problem?
- · Again we counted each option several times

#### Fixing solution:

Note, that we actually counted ordered sequences of friends

#### Fixing solution:

- Note, that we actually counted ordered sequences of friends
- Each (unordered) group  $\{a,b,c\}$  is counted  $3 \times 2 = 6$  times: abc, acb, bac, bca, cab and cba

#### Fixing solution:

- Note, that we actually counted ordered sequences of friends
- Each (unordered) group  $\{a,b,c\}$  is counted  $3\times 2=6$  times: abc, acb, bac, bca, cab and cba
- We can just divide the previous result by 6

#### Fixing solution:

- Note, that we actually counted ordered sequences of friends
- Each (unordered) group  $\{a,b,c\}$  is counted  $3\times 2=6$  times: abc, acb, bac, bca, cab and cba
- ullet We can just divide the previous result by 6
- The actual result is  $5 \times 4 \times 3/(3 \times 2) = 10$

• For a set S its k-combination is a subset of S of size k

- For a set S its k-combination is a subset of S of size k
- The number of k-combinations of n element set is denoted  $\binom{n}{k}$

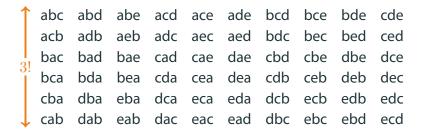
- For a set S its k-combination is a subset of S of size k
- The number of k-combinations of n element set is denoted  $\binom{n}{k}$
- Pronounced 'n choose k'

- For a set S its k-combination is a subset of S of size k
- The number of k-combinations of n element set is denoted  $\binom{n}{k}$
- Pronounced 'n choose k'
- We have shown  $\binom{5}{3} = 10$

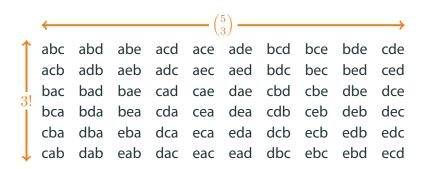
#### **3-Combinations and 3-Permutations**

abd abe acd ace ade bcd bce bde acb adb aeb adc aec aed bdc bec bed bad bae cad dae cbd cbe dbe dce bac cae dea cdb ceb bca bda bea cda cea deb dec dcb ecb edb cba dba eba dca eca eda edc dab eab dac eac ead dbc ebc ebd cab ecd

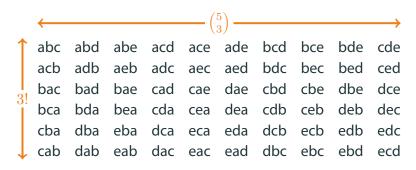
# 3-Combinations and 3-Permutations



## **3-Combinations and 3-Permutations**



## 3-Combinations and 3-Permutations



$$3! \binom{5}{3} = \frac{5!}{(5-3)!}$$

 Combinations is an important setting that is needed a lot

- Combinations is an important setting that is needed a lot
- How can it arise in Data Science? Here is an example

- Combinations is an important setting that is needed a lot
- How can it arise in Data Science? Here is an example
- Sometimes you need to use weak ML model that cannot in principal give a good prediction on your data

- Combinations is an important setting that is needed a lot
- How can it arise in Data Science? Here is an example
- Sometimes you need to use weak ML model that cannot in principal give a good prediction on your data
- One possible solution: enrichment by feature products

Suppose you have 5 numerical features a, b, c,
 d and e in your setting

- Suppose you have 5 numerical features a, b, c,
  d and e in your setting
- To enrich the set of features we can add new features consisting of products of any 3 distinct features

- Suppose you have 5 numerical features a, b, c, d and e in your setting
- To enrich the set of features we can add new features consisting of products of any 3 distinct features
- For example, we add  $a\cdot b\cdot c$ ,  $b\cdot c\cdot e$ ,  $d\cdot a\cdot c$  and all other triple products

- Suppose you have 5 numerical features a, b, c,
  d and e in your setting
- To enrich the set of features we can add new features consisting of products of any 3 distinct features
- For example, we add  $a\cdot b\cdot c$ ,  $b\cdot c\cdot e$ ,  $d\cdot a\cdot c$  and all other triple products
- After that we can run our weak model on the new set of features

- Suppose you have 5 numerical features a, b, c, d and e in your setting
- To enrich the set of features we can add new features consisting of products of any 3 distinct features
- For example, we add  $a\cdot b\cdot c$ ,  $b\cdot c\cdot e$ ,  $d\cdot a\cdot c$  and all other triple products
- After that we can run our weak model on the new set of features
- In many cases it gives much better results

• But how many features we have added?

- But how many features we have added?
- Basically, for each new feature we pick
  k-combination of initial features

## **Number of Combinations**

There is the following short formula for the number of combinations

#### **Theorem**

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

• Need to count the number of subsets of size  $\boldsymbol{k}$  of a set of size  $\boldsymbol{n}$ 

- Need to count the number of subsets of size  $\boldsymbol{k}$  of a set of size  $\boldsymbol{n}$
- There are n choices of the first element, (n-1) choices of the second element, ..., (n-k+1) choices of the k-th element

- Need to count the number of subsets of size k of a set of size n
- There are n choices of the first element, (n-1) choices of the second element, ..., (n-k+1) choices of the k-th element
- This gives  $n(n-1)\cdots(n-k+1)=\frac{n!}{(n-k)!}$

- Need to count the number of subsets of size k of a set of size n
- There are n choices of the first element, (n-1) choices of the second element, ..., (n-k+1) choices of the k-th element
- This gives  $n(n-1)\cdots(n-k+1)=\frac{n!}{(n-k)!}$
- But this is the number of k-permutations rather than the number of k-combinations

In other words , we counted ordered subset instead of unordered

- In other words , we counted ordered subset instead of unordered
- Each subset is counted as many times, as there are ways to order it

- In other words , we counted ordered subset instead of unordered
- Each subset is counted as many times, as there are ways to order it
- There are k! ways to order a subset of size k

- In other words , we counted ordered subset instead of unordered
- Each subset is counted as many times, as there are ways to order it
- There are k! ways to order a subset of size k
- Thus, each subset is counted k! times

- In other words , we counted ordered subset instead of unordered
- Each subset is counted as many times, as there are ways to order it
- There are k! ways to order a subset of size k
- Thus, each subset is counted k! times
- This finally gives  $\frac{n!}{k!(n-k)!}$

• In this week we started to study combinatorics

- In this week we started to study combinatorics
- We started with simple observations

- In this week we started to study combinatorics
- We started with simple observations
- We built on them more complicated constructions

- In this week we started to study combinatorics
- We started with simple observations
- We built on them more complicated constructions
- We reached a rather non-trivial question of counting combinations

- In this week we started to study combinatorics
- We started with simple observations
- We built on them more complicated constructions
- We reached a rather non-trivial question of counting combinations
- Next week we will proceed with combinations and will see that they have nice and interesting properties