TELECOM ParisTech

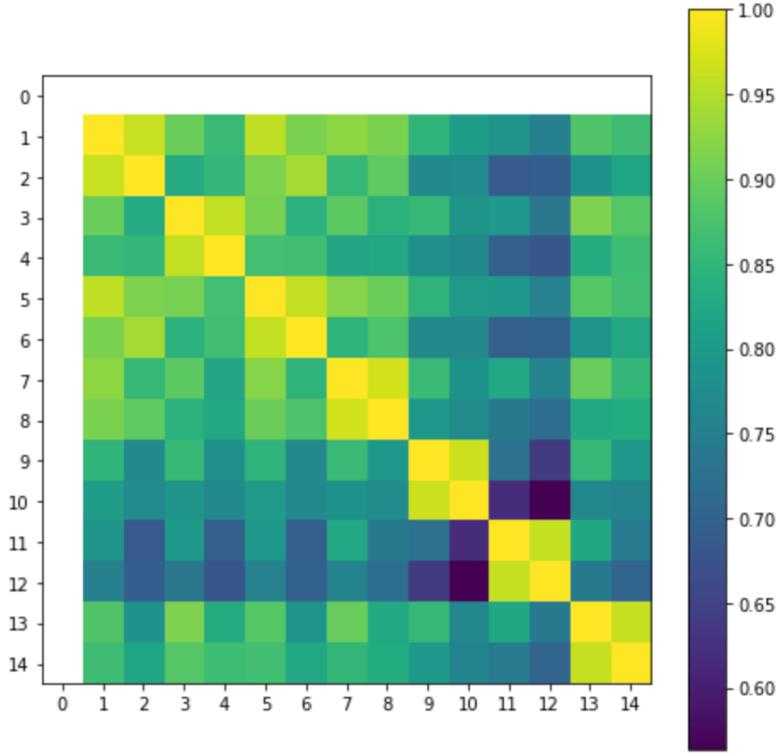
Une école de l'IMT

Challenge MDI343 2017-2018

Sparse Score Fusion for Classifying Mate Pairs of Images

Kaichen MA

Feature analysis and comparisons



correlation among algorithms

performance of each algorithm

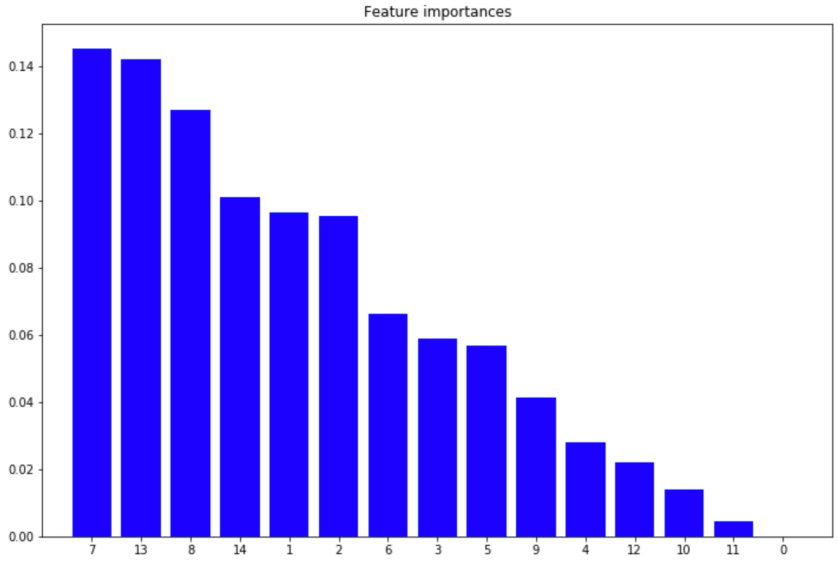
```
algorithm 1     frr : 0.253079046817
algorithm 2     frr : 0.253079046817
algorithm 3     frr : 0.255898650898
algorithm 4     frr : 0.255898650898
algorithm 5     frr : 0.231109631689
algorithm 6     frr : 0.231109631689
algorithm 7     frr : 0.165397192144
algorithm 8     frr : 0.165397192144
algorithm 9     frr : 0.174090971393
algorithm 10    frr : 0.174090971393
algorithm 11    frr : 0.313034794698
algorithm 12    frr : 0.313034794698
algorithm 13    frr : 0.163027941493
algorithm 14    frr : 0.163027941493
```

These pair algorithms are relatively correlated: Algo1 and 2, Algo3 and 4, algo5 and 6, Algo7 and 8, Algo9 and 10, Algo11 and 12, Algo13 and 14.

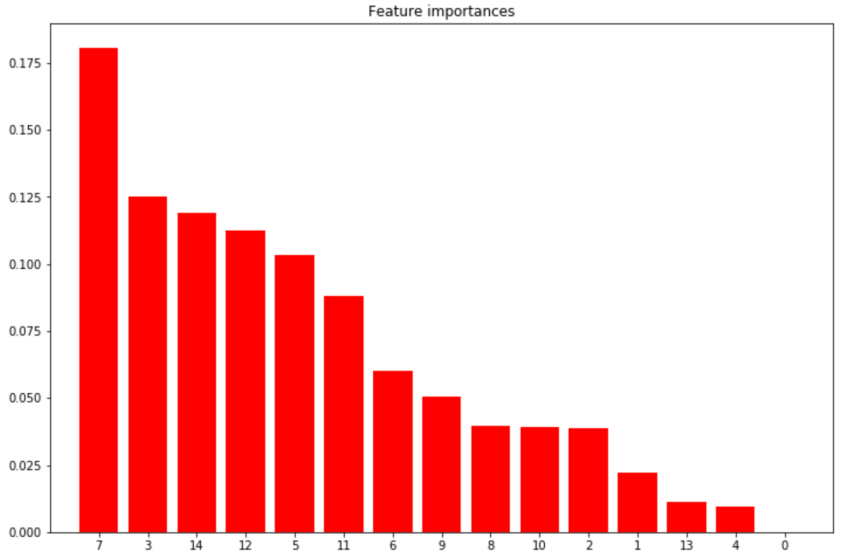
Algo 7, 8, 9, 10, 13, 14 display better performance than the others.

Feature analysis and comparisons

Feature importance analysis by logistic regression



Feature importance analysis by extra tree



In global, the algo 7, 8, 13, 14 are relatively having more weigh than other algorithms.

This is consistent with the algorithm's performance analysis.

Variable construction and processing

- **Enumerate all possible combinations among 14 algorithms**
  - 16384 combinations
- **Remove the combinations that exceed budgeted time**
  - 1024 combinations
- **Remove subsets in those combinations**
  - 577 combinations
- **Calculate false recognition rate (FRR) for each combinations**
  - coefficents for these combinations are simply identificly given 1
- **Algorithm combinations are filtered according to their FRR, those with 10 least errors are selected.**
  - The 10 top combinations are all among Algo 7 to 14.

Optimization of combination coefficients

- **Define a function which calculates FRR with fusion matrix M**

```
def combcoef(coef):

    M = np.zeros((15, 15))
    .....
    return frr
```

- **Use function *minimize* to optimize relative coefficients for selected combinations**

```
res = minimize(combcoef, coef0, args=(), method="COBYLA",
               jac=None, hess=None, hessp=None, tol=None, callback=None)
```

Validation and conclusion

- **Calculate FRR with optimized coefficients for training data**

	alg7	alg8	alg9	alg10	alg11	alg12	alg13	alg14	time	error
0	0	0.97863	0.92145	1.01491	0.992821	0	0	0.906982	539	0.082473
1	0.633101	0	0.641464	0.742667	0.814128	0	1	0	539	0.083198
4	0.562156	0	0.599002	1.03809	0	0.905884	0	0.984055	539	0.083629
7	0	1.48445	1.02796	2.02084	0	1.99793	0	0.996331	539	0.083648
3	0.851942	0	1.02801	1.19234	0.926618	0	0	1	539	0.083766
9	0.79514	2.28029	2.08815	0	1.30657	0	1.02888	0	576	0.084921
5	1.00705	0	0.993355	1.85703	0	1.17096	1	0	539	0.086527
8	0.750372	2.21014	1.55685	0	1.40503	0	0	0.99457	576	0.087388
6	0	1.43542	1.23062	1.95381	0	0.957369	1	0	539	0.089640
2	0	0.533291	0.429814	0.52981	0.355249	0	0.956879	0	539	0.090932

They are generally validated for training data, except the combination 3, i.e. among Algo 8, 9, 10, 11, 13.

Validation and conclusion

- **Validate optimized coefficients by test data**

```
M_pred_1.txt => 0.0710172744722
M_pred_2.txt => 0.0753358925144
M_pred_3.txt => 0.0830134357006
M_pred_4.txt => 0.0681381957774
M_pred_5.txt => 0.0714971209213
M_pred_6.txt => 0.0690978886756
M_pred_7.txt => 0.0700575815739
M_pred_8.txt => 0.0724568138196
M_pred_9.txt => 0.0748560460653
M_pred_10.txt => 0.0734165067179
```

Consistent with training data, algorithm combination 3 displays the worst performance of classification.

Combination 4, composed of Algo 7, 9, 10, 11, 14 has the best classification performance.

Thanks for your attention !