An adaptive probabilistic classification method for dynamic class hierarchies

C. Ferri, M.J. Ramírez-Quintana, M.Kull, A. Martínez-Usó and P. Flach
DSIC, Universidad Politécnica de Valencia, Spain
University of Bristol, U.K
{cferri,mramirez,admaru}@dsic.upv.es
{Mellis.Kull,Peter.Flach}@bristol.ac.uk

LMCE 2015 - 11th September 2015

Overview

- Introduction
- Reframing hierarchical classifiers
- Experimental Evaluation
- Conclusions and Future work

REFRAME

Introduction

- Many classification tasks involve a large number of categories
  - Usually these categories are ordered in a hierarchy:
    - Bioinformatic, text categorization, film or music genre classification

- In hierarchical classification a model labels an instance by using the structure of the hierarchy

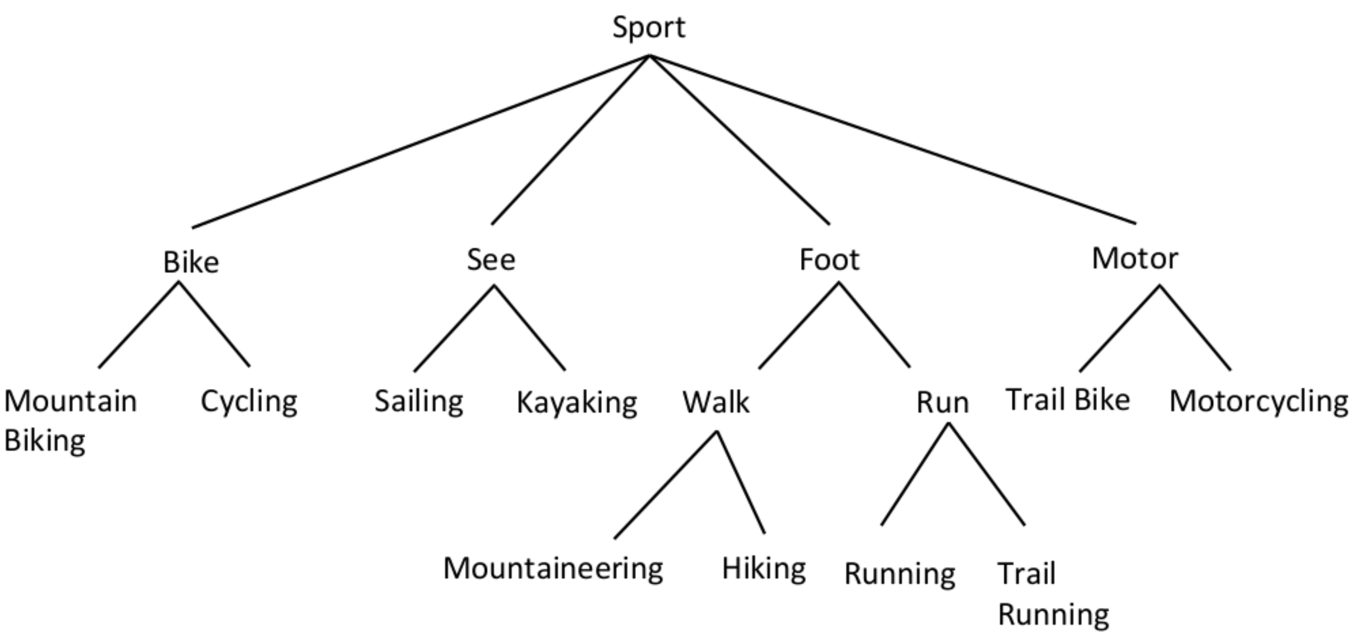
Introduction

- In many contexts the hierarchy of classes is dynamic, i.e., the structure of classes can change
  - Reordering of categories: Biotechnology can be under computer science or medicine
- We investigate a scenario (“mandatory leaf-node”) where class hierarchies can change from learning to deployment time
  - We do not contemplate deletion or creation of categories

REFRAME

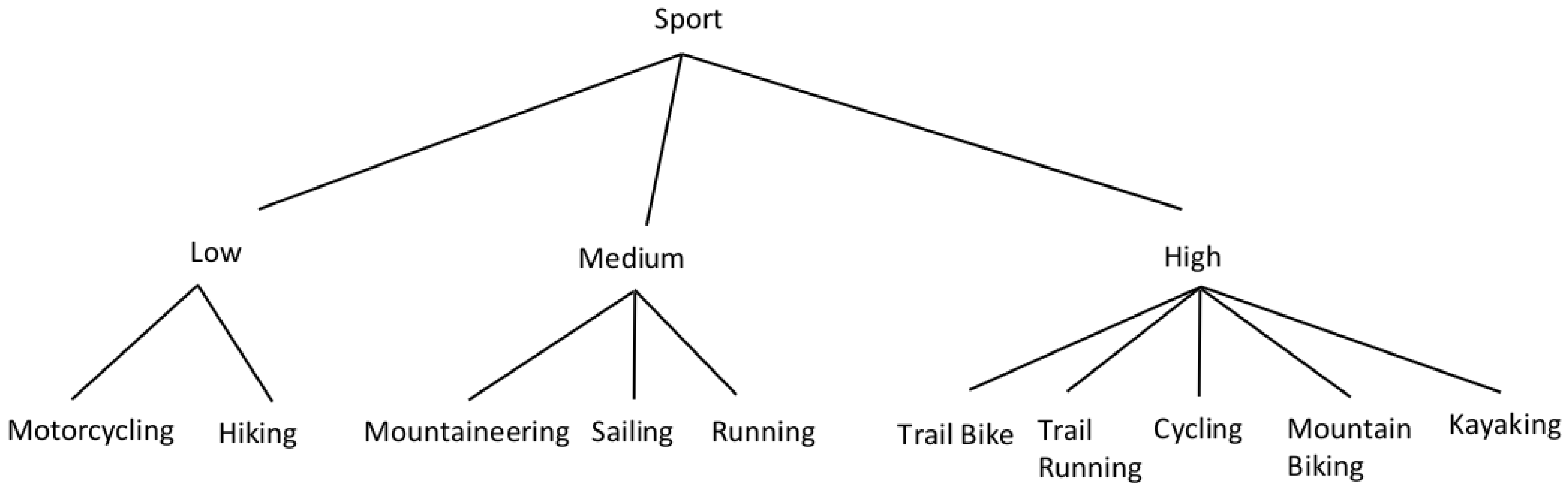
Introduction

- We analyse GPS tracks for medical reasons
- We learn a hierarchical classifier using the following model that considers: locomotion form, surface type and speed.



Introduction

- We now consider a new hierarchy based on the intensity of the sport:



Introduction

- In this situation, we can:
  - **Retrain**: Ignore previous models and learn a new classifier with the new context
  - **Reframe**: Try to adapt existing models to the new context

Reframing hierarchical classifiers

- Cost Sensitive-Classification:
  - A new context means a different skew
  - We adapt probabilistic models by assigning classes based on the minimisation of the expected cost
- Based on this approach we propose a a new hierarchical classification technique:
  - *Hprb*: hierarchical probabilistic method

Reframing hierarchical classifiers

- Hierarchical classification loss:
  - We use a distance-based metric that considers the distance between the predicted and the actual class in the hierarchy
  - Classes that are close to each other in the hierarchy tend to be similar

Reframing hierarchical classifiers

- Hierarchical classification loss:
  - Given a predicted class *p*, and an actual class *r* in a hierarcy *T*
  - $hlossT(p,r) = d(p,r) / dmax(T)$
  - *d(p,r)* is the number of edges of the shortest path between *p* and *r* in *T*
  - *dmax(T)* is the size in edges of the longest path between two classes in *T*

REFRAME

Reframing hierarchical classifiers

- Hierarchical classification loss:
  - *hlossT(Mountain Biking,Trail Bike)=4/5*
  - *hlossT(Trail Bike,Trail Bike)=0*
  - *hlossT(Motorcycling,Trail Bike)=5/5*

REFRAME

Reframing hierarchical classifiers

- *Hpbr*:
  - Given a probabilistic model *M* with *C* classes, a hierarchy *T*, and an instance *e* to be classified.
  - *p(c/e)* is the estimated probability by *M* that example *e* belongs to a class *c*

$$hprb(e) = argmin_{c \in C}\left(\sum_{\forall k \in C} p(k|e) * hloss_T(k, c)\right)$$

REFRAME

Experiments

- Contexts where the hierarchical structure is variable

- Two class hierarchies.
  - *T*: Old context is used in the training phase
  - *NT*: New context is employed in the test phase

- Hierachy is automatically induced by building a dendrogram from a confussion matrix
  - **ZeroR** for *T*
  - **J48** for *NT*

REFRAME

Experiments

- 12 Learning Methods
  - **J48**: decision tree
  - **J48Unp**: unpruned decision tree
  - **Jrip**: propositional rule learner
  - **NB**: Naive Bayes
  - **Logist**: logistic regression
  - **IBK**: K-nearest neighbours with ten neighbours
  - **RF**: random rorest
  - **Bagging**: ten J48 models combined by Bagging
  - **PART**: decision list
  - **Boosting**: boosting of J48 models
  - **Stump**: decision stump
  - **LB**: Boosted Logistic Regression

REFRAME

Experiments

- Datasets: 50% train, 50% test

	dataset	NumInst	NumAtt	Numclass
1	anneal	898	39	5
2	glass	214	10	6
3	zoo	101	18	7
4	autos	205	26	6
5	grub-damage	155	9	4
6	soybean	683	36	19
7	eucalyptus	736	20	5
8	vowel	990	14	11
9	pendigits	10992	17	10
10	segment	2310	20	7

REFRAME

Experiments

- **NB** 300 iterations.
- **PAVCal** Calibration

	dataset	flat_error	flat	hpnt	hp	hpntcal	hpcal	MSE	MSEcal
1	anneal	0.1216	0.0929	0.0929	0.0929	0.0748	0.0748	0.0449	0.0321
2	glass	0.5159	0.2917	0.2899	0.2902	0.3206	0.3215	0.1342	0.1228
3	zoo	0.0461	0.0261	0.0262	0.0265	0.0284	0.0300	0.0110	0.0145
4	autos	0.4613	0.3128	0.3123	0.3130	0.3203	0.3224	0.1347	0.1161
5	grub-damage	0.5189	0.3776	0.3757	0.3760	0.3742	0.3748	0.1806	0.1768
6	soybean	0.1627	0.0720	0.0673	0.0718	0.0593	0.0652	0.0129	0.0131
7	eucalyptus	0.5254	0.3709	0.3708	0.3707	0.3686	0.3695	0.1672	0.1380
8	vowel	0.4006	0.2184	0.2153	0.2170	0.2164	0.2194	0.0493	0.0498
9	pendigits	0.1426	0.0956	0.0957	0.0956	0.0937	0.0929	0.0257	0.0213
10	segment	0.1974	0.1048	0.1048	0.1048	0.0873	0.0879	0.0518	0.0320
	Average	0.3092	0.1963	0.1951	0.1959	0.1944	0.1958	0.0812	0.0716

Experiments

Average results of the classification methods:

	Method	flat_error	flat	hpnt	hpntcal	hpntcal vs flat	MSE	MSEcal	MSEcal vs MSE
1	J48	0.2558	0.1655	0.1655	0.1618	905-1284-811	0.0679	0.0609	1745-406-849
2	JRip	0.2844	0.1831	0.1830	0.1846	96-2607-297	0.0689	0.0664	1785-520-695
3	logist	0.2738	0.1760	0.1759	0.1798	876-750-1374	0.0861	0.0771	1884-161-955
4	NB	0.3092	0.1963	0.1951	0.1944	1993-139-868	0.0812	0.0716	2245-48-707
5	IBK	0.3862	0.2484	0.2383	0.2333	2102-52-846	0.0728	0.0731	1811-40-1149
6	RF	0.2098	0.1379	0.1393	0.1352	1362-176-1462	0.0542	0.0523	1999-38-963
7	bagging	0.2301	0.1497	0.1497	0.1475	1199-309-1492	0.0546	0.0555	1310-99-1591
8	PART	0.2609	0.1679	0.1684	0.1641	770-1689-541	0.0693	0.0621	1963-376-661
9	boosting	0.2198	0.1427	0.1426	0.1450	1006-799-1195	0.0638	0.0571	2340-326-334
10	lb	0.2446	0.1578	0.1566	0.1564	1269-294-1437	0.0568	0.0573	1298-88-1614
11	J48Unp	0.2542	0.1642	0.1692	0.1672	801-624-1575	0.0643	0.0619	2227-44-729
12	stump	0.6145	0.3953	0.3752	0.3761	1567-1259-174	0.0987	0.1001	613-1145-1242
Total						13946-9982-12072			21220-3291-11489

REFRAME

Conclusions

- We have addressed the problem of hierarchical classification in dynamic contexts
- New method based on predicting the label that minimises the expected loss with respect to the deployment context (class hierarchy)
  - It is able to adapt the predictions to a novel context.

Conclusions

- We have analysed the performance of the proposal over 10 datasets and 12 learning methods.
- Hierarchies are artificially induced by computing similarities between classes
- We have studied the effect of applying multiclass calibration over probabilities (PAV Calibration)

REFRAME

Future work

- Consider other scenarios:
  - “non-mandatory leaf-node” problem
  - Drastic changes in the hierarchy of classes

- We also want to explore the adaption of hierarchical methods such as the Top Down approach to dynamic contexts .

Thank you

REFRAME