A Scalable Noise Reduction Technique for Large Case-Based Systems

Nicola Segata
segata@disi.unitn.it

Enrico Blanzieri
blanzier@disi.unitn.it

Pádraig Cunningham
Padraig.Cunningham@ucd.ie

Dept. of Information Engineering
and Computer Science,
University of Trento, Italy.

Computer Science,
University College Dublin,
Dublin, Ireland.

8th International Conference on Case-Based Reasoning

Seattle, University of Washington, USA

July 22, 2009

Outline of the talk

Noise Reduction for Case-Based Reasoning (CBR)

- All (supervised) learning systems must deal with noise
- CBR is instance-based and it is thus particularly affected by noise
- Considering more instance for local learning (k-NN with $k > 1$) can alleviate noise problems but with large k the locality assumption is violated and the accuracy decreases
- Noisy examples can prevent case-based explanation

Noise Reduction (NR) for Instance-Based Learning (IBL)

Noise reduction is an important pre-processing step that allows higher accuracy performances for IBL and CBR systems.

N.B. noise reduction can be important in other machine learning fields:

- improve the quality of data in medical domains
- data cleansing in bioinformatics and high-throughput techniques
- simplify machine learning models and training phases

State-of-the-art noise reduction techniques for IBL

Following [Wilson & Martinez, 2000] the most effective NR techniques are:

Edited NN (ENN) Rule
: removes from the training set examples that do not agree with the majority of their k-NN [Wilson,1972]

Repeated Edited NN (RENN) Rule
: repeats the ENN algorithm until no further eliminations can be made [Tomek,1976]

All-kNN (AkNN) rule
: repeat the ENN rule for each example using all neighbourhood size between 1 and k [Tomek,1976]

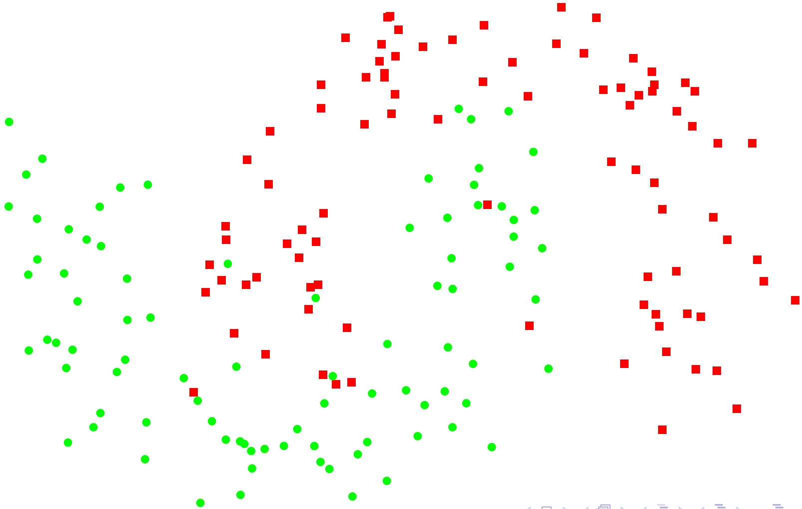
State-of-the-art for Noise Reduction

Despite their simplicity, ENN, RENN and AkNN are not overcome by more recent approaches for general CBR and IBL problems

N.B.: For specific field some NR approaches can perform particularly well (e.g. BBNR [Delany & Cunningham,2007] for spam classification and [Malossini, Blanzieri, Ng,2004] for microarray mislabelling detection)

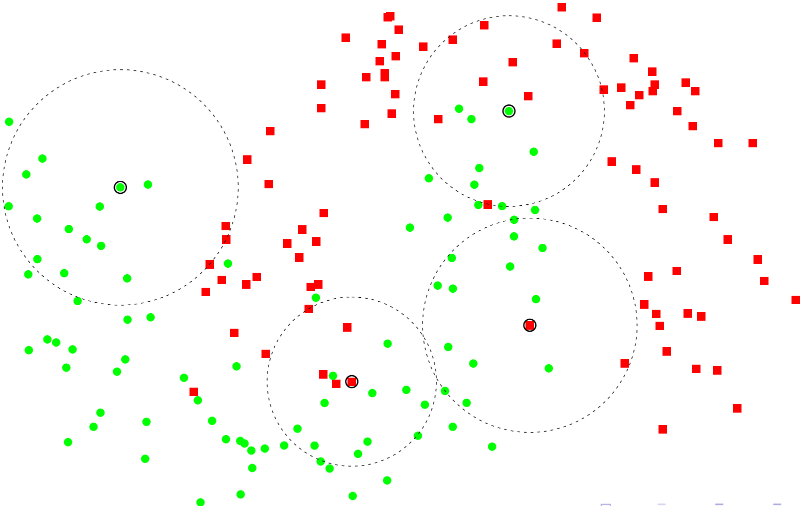
Noise Reduction with Local Support Vector Machines

1. for each training example. . .



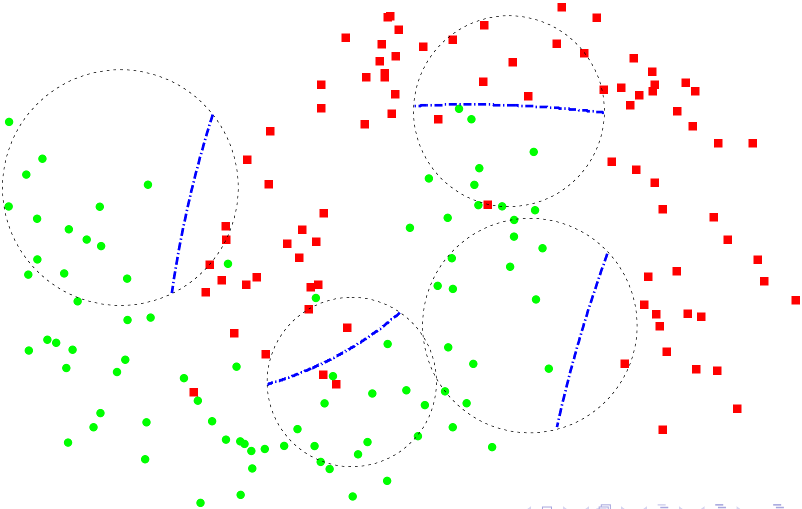
Noise Reduction with Local Support Vector Machines

2 . . . retrieve its neighbourhood ($k = 15$)



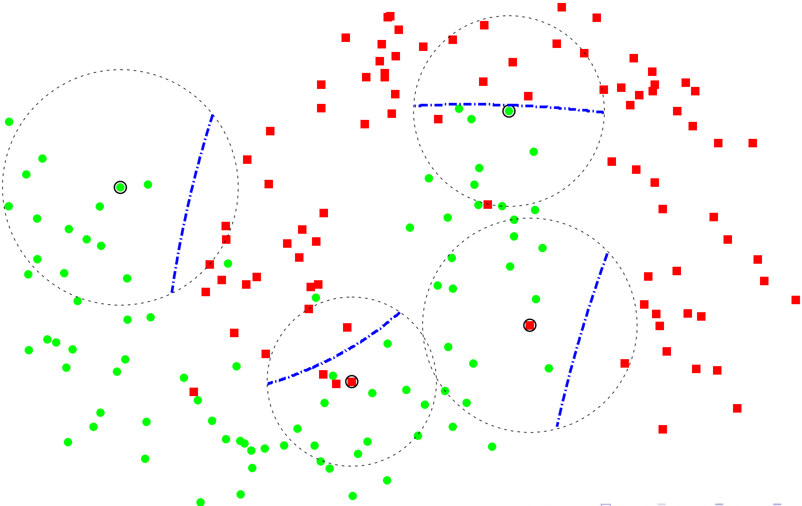
Noise Reduction with Local Support Vector Machines

3. train a Local SVM model for each neighbourhood ($C = 10$, RBF kernel with $\sigma = 1/10$)



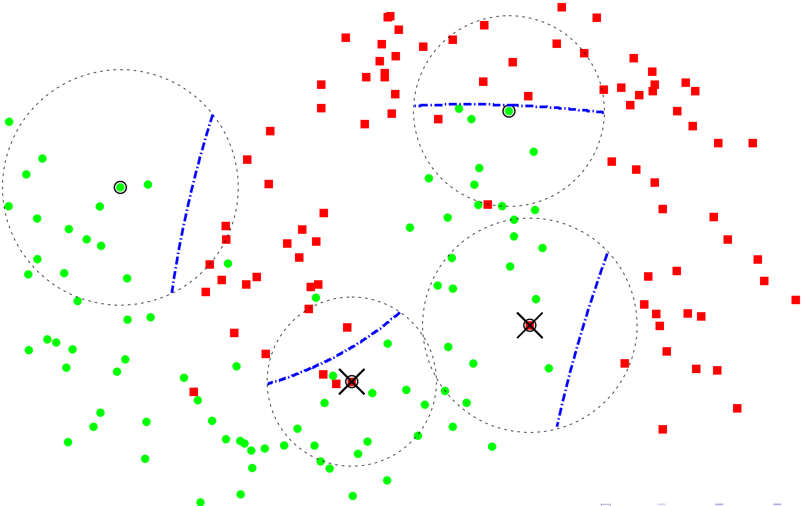
Noise Reduction with Local Support Vector Machines

4. predict the labels of the central examples of the neighbourhoods



Noise Reduction with Local Support Vector Machines

5. remove the examples that are misclassified by the Local SVM model



Noise Reduction with Local Support Vector Machines

The LSVM-nr algorithm [Segata, Blanzieri, Delany, Cunnigham, 2008] for noise reduction with Local SVM uses the probabilistic estimate of Local SVM to remove noisy training examples (in accordance to a threshold)

LSVM-nr is more accurate than ENN, RENN and AkNN

LSVM-nr showed statistically better results on the induced k-NN accuracies on 15 real datasets.

LSVM-nr is particularly effective also for:

- spam filtering (like BBNR)
- Gaussian noise in feature values
- uneven class densities

LSVM-nr is computationally less efficient than ENN and it is not suitable for large and very large CBR systems. . .

The Issue of Scalability for Noise Reduction

Practical motivations:

- Modern CBR systems can be very large
- Datasets in medical and bioinformatics can be huge
- Accuracies of IBL are higher when the training set density is rather high (and thus the data is abundant)

Theoretical motivations:

method	learning bound	condition
NN	$2\times$ Bayes Error	$n \mapsto \infty$
k-NN	Bayes Error	$n \mapsto \infty,\ k \mapsto \infty,\ k/n \mapsto 0$
edited NN	Bayes Error	$n \mapsto \infty$

How to improving classification accuracy of IBL and CBR

1. carefully remove noisy examples from the Case-Base: LSVM-nr
2. scale the noise reduction system in order to use as many examples as possible for NN classification: the topic of the present work

Noise reduction for large datasets: FaLKNR

Fast **L**ocal **K**ernel **N**oise **R**eduction (FaLKNR) is a noise reduction technique based on Local SVM noise reduction scalable for large datasets.
The main ideas contained in FaLKNR are:

- Adopting the Cover Tree data-structure
- Developing a set of strategies to lower the number of neighbourhoods that need to be retrieved and the number of local SVM that need to be trained
- Developing a local model selection approach to efficiently select the hyper-parameters of the technique

The Cover Tree Data-structure [Beygelzimer, Kakade, Langford, 2006]

Cover Trees (CT) are indexed trees satisfying the following invariants:

- **Nesting** $C_i \subset C_{i-1}$
- **Covering** $\forall p \in C_{i-1}$ there exists a $q \in C_i$ such that $dist(p,q) \leq 2^i$ and $\exists! q$ that is a parent of p
- **Separation** $\forall p, q \in C_i,\ dist(p,q) > 2^i$

they permits excellent performances:

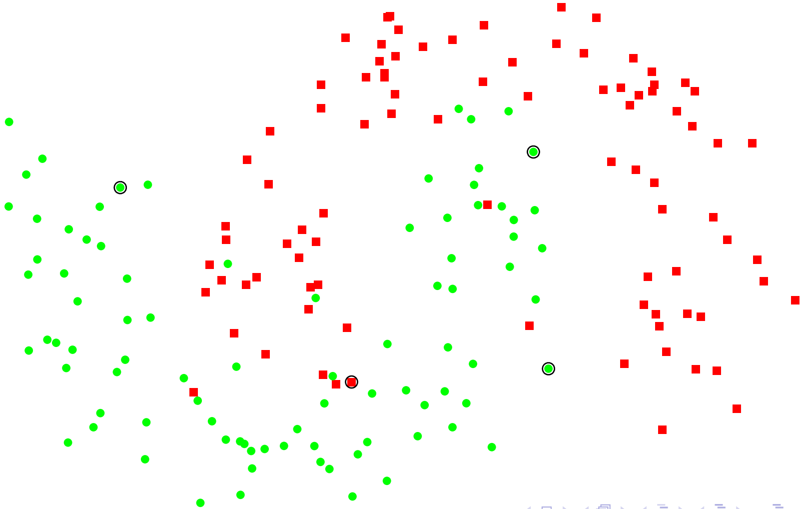
space requirements: $\mathcal{O}(n)$
construction time: $\mathcal{O}(n \log n)$
single insertion/removal/query: $\mathcal{O}(\log n)$

CT are applicable in metric spaces, thus in Hilbert spaces

$$||\Phi(x) - \Phi(x')||^2 = K(x,x) + K(x',x') - 2K(x,x')$$

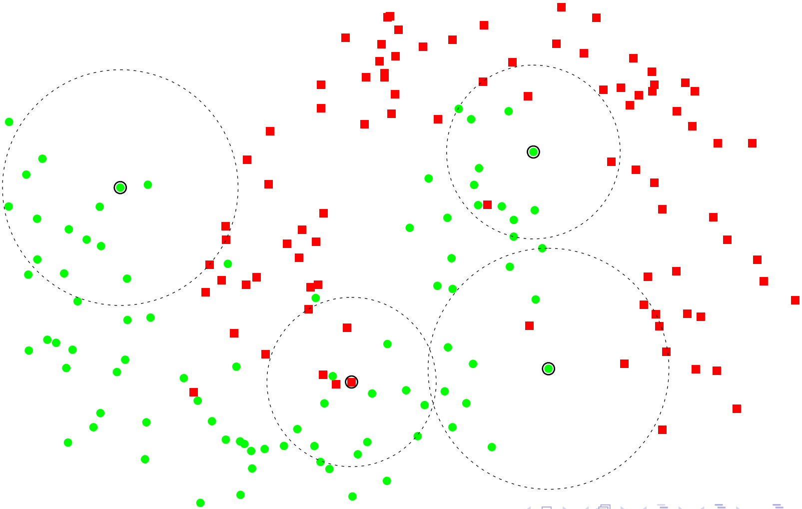
Lowering the Number of Local SVM Trained

1. for *some* training examples. . .



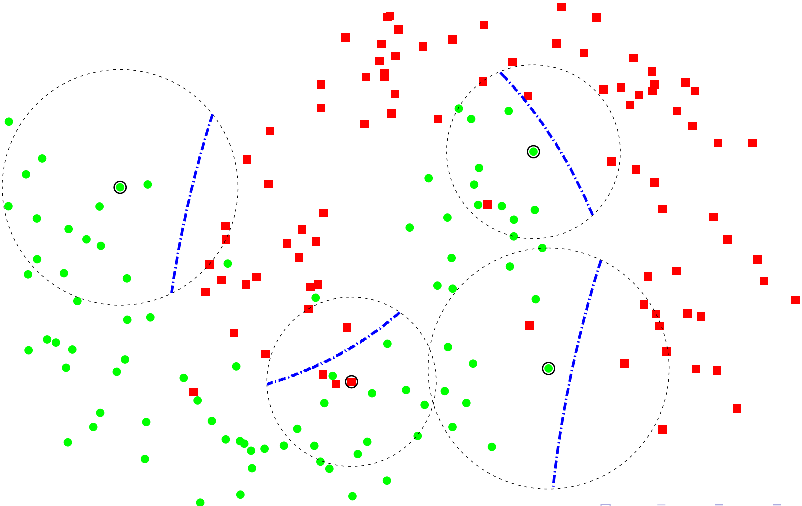
Lowering the Number of Local SVM Trained

2 . . . retrieve their neighbourhood ($k = 15$)



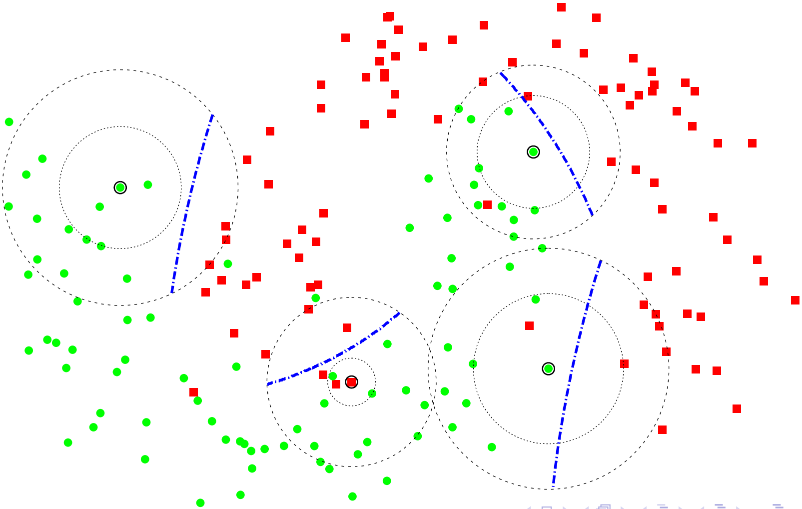
Lowering the Number of Local SVM Trained

3. train a Local SVM on each neighbourhood ($C = 10$, RBF kernel with $\sigma = 1/10$)



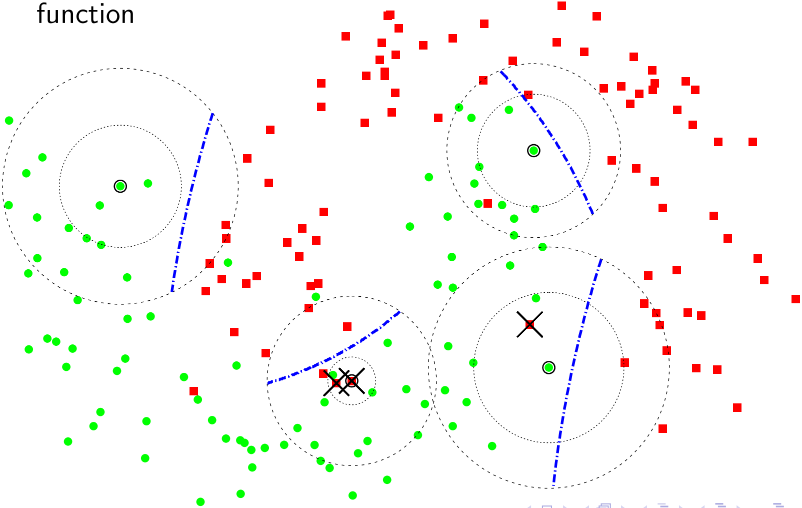
Lowering the Number of Local SVM Trained

4. consider the *assignment k'-neighbourhood* of each neighbourhood ($k' = 4$)



Lowering the Number of Local SVM Trained

5. remove the examples in the assignment neighbourhoods that are misclassified by the corresponding Local SVM decision function



FaLKNR: the choice of the assignment k'-neighbourhoods

Definition (k'-neighbourhood covering set)

Given $k' \in \mathbb{N}$, a k'-neighbourhood covering set of centers $\mathcal{C}_{k'} \subseteq X$ is a subset of the training set such that the following holds:

$$\bigcup_{c \in \mathcal{C}_{k'}} \{x_{r_c(i)} \mid i = 1, \ldots, k'\} = X.$$

- finding the *minimal* k'-neighbourhood covering set is NP-HARD!
- for FaLKNR is not so crucial to find the very minimal $\mathcal{C}$ (minimality vs redundancy)
- greedy approximated approaches for the related problems of *Set Cover Problem* and *Minimum Sphere Set Covering Problem* have been proposed
- a point can be in the k'neighbourhood of multiple centers

A greedy approach to approximate the minimal $\mathcal{C}$

The idea of the greedy approach for approximated minimal $\mathcal{C}$

Recursively take as centers those points which are not k'-neighbours of any point that has already been taken as center

The set of $c_i \in \mathcal{C}$ with $i = 1, \ldots, |\mathcal{C}|$ can be detected as:

$$c_i = x_j \in X \quad \text{with } j = \min\left(l \in \{1, \ldots, n\} \middle| x_l \in X \setminus X_{c_i}\right)$$
$$\text{where } X_{c_i} = \bigcup_{l<i} \left\{ x_{r_{c_l}(h)} \mid h = 1, \ldots, k' \right\}.$$

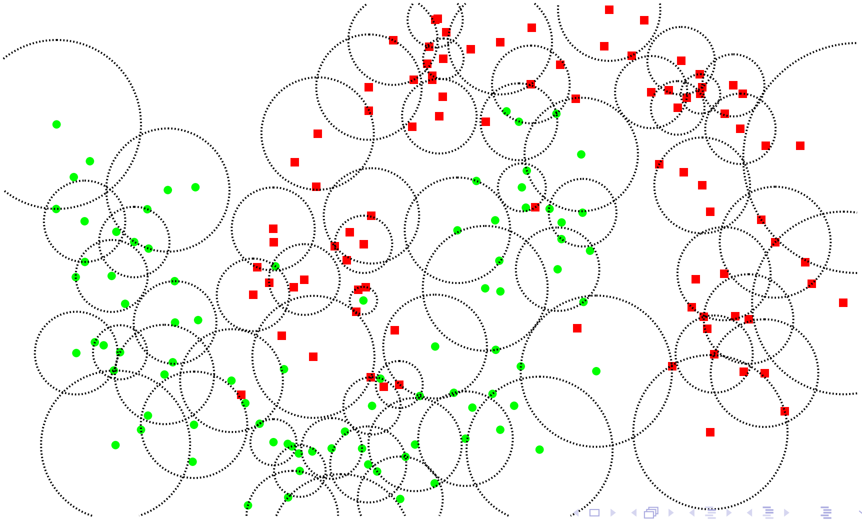
The function $cnt(t) : X \rightarrow \mathcal{C}$ assigns each training example to a center:

$$cnt(x_i) = x_j \in \mathcal{C} \quad \text{with } j = \min\left(l \in \{1, \ldots, n\} \middle| x_l \in \mathcal{C} \text{ and } x_i \in X_{x_l}\right)$$
$$\text{where } X_{x_l} = \left\{ x_{r_{x_l}(h)} \mid h = 1, \ldots, k' \right\}.$$

The separation invariant of Cover Trees can help in selecting the $(i+1) - th$ center...

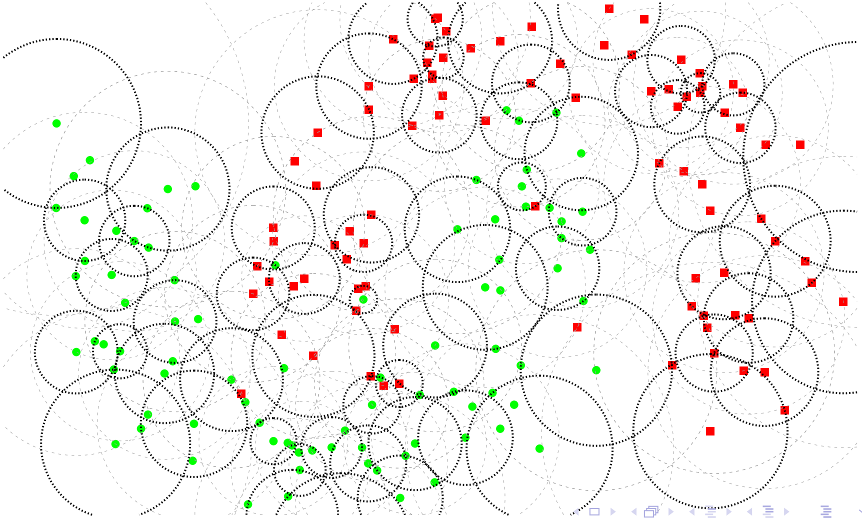
FaLKNR: the overall picture

- the training set covered by the $\mathcal{C}$ assignment k'-neighbourhoods ($k' = 4$)



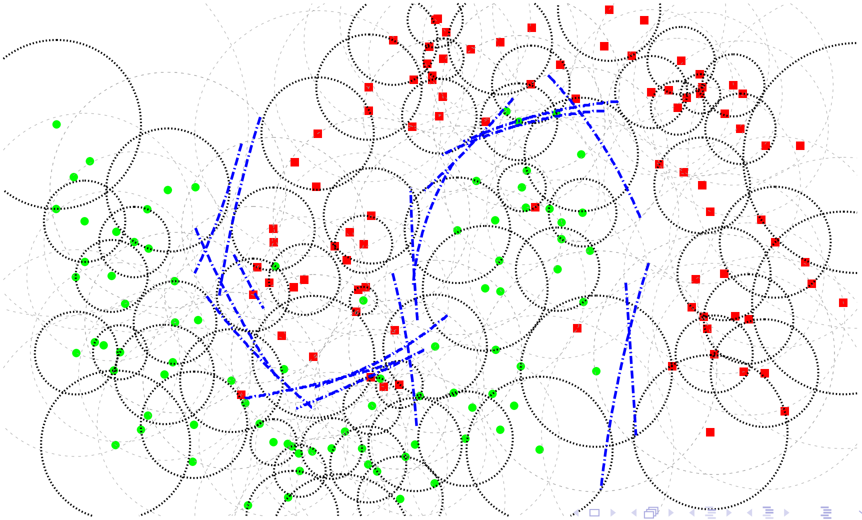
FaLKNR: the overall picture

- the training set covered by the $\mathcal{C}$ neighbourhoods ($k = 15$) and the corresponding $\mathcal{C}$ assignment neighbourhoods ($k' = 4$)



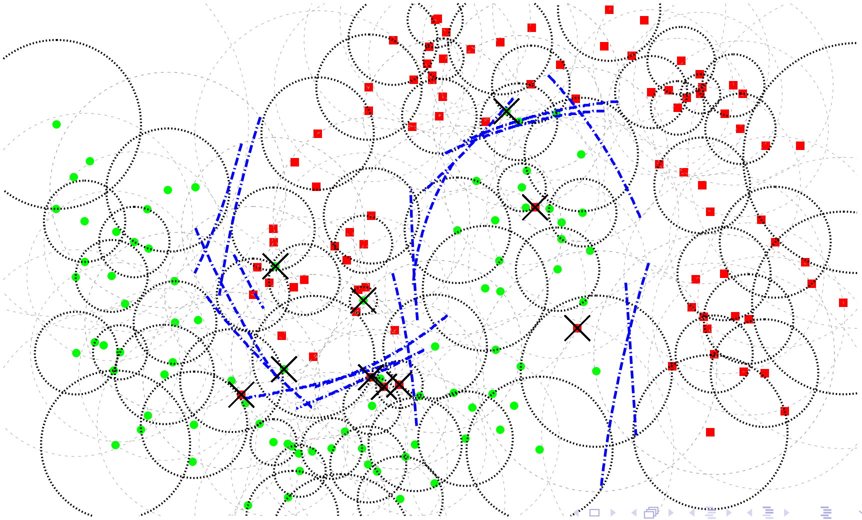
FaLKNR: the overall picture

- the Local SVM decision functions of FaLKNR (only 17 Local SVMs for 185 examples)



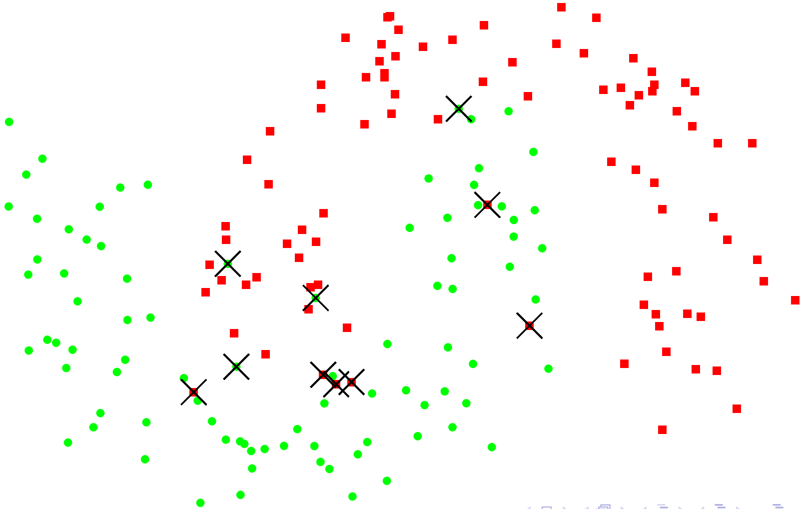
FaLKNR: the overall picture

- the examples removed by FaLKNR



FaLKNR: the overall picture

- the dataset edited with FaLKNR



Local Model Selection for FaLKNR Parameters

The idea of local model selection

Use a subset of the local neighbourhoods to select k and the parameters for all the local models that need to be trained

The Local Model Selection Procedure:

- for a random (small) subset of training examples and each parameter choice P
  1. separate the $k' < k$ nearest neighbours of x, called S, from the k nearest neighbours of x, called S^E;
  2. randomly split S in κ disjoint internal validation sets S_i with $0 < i < \kappa$;
  3. for each fold i train a model with the $S^E \cup (S \setminus S_i)$ using the P parameters set evaluating it on the correspondent S_i set, taking the mean of the accuracies on the κ folds.
- select the P parameter set giving the best average κ fold accuracy

Computational Complexity Analysis of FaLKNR

1. construct the Cover Tree[1]
2. retrieve the $|\mathcal{C}|$ neighbourhoods
3. train the $|\mathcal{C}|$ local SVMs (and perform local model selection)[1]
4. univocally assign each example to a k'-neighbourhood
5. predict the each training label with the corresponding model[1]
6. remove the misclassified examples[1]

FaLKNR computational complexity

Time Complexity: $\mathcal{O}(n \log n + |\mathcal{C}| \cdot k \log n + |\mathcal{C}| \cdot k^3 + n + n \cdot k + n) =$

$= \mathcal{O}(n \log n + |\mathcal{C}| \cdot k \log n + |\mathcal{C}| \cdot k^3) \stackrel{a}{=} \mathcal{O}(n \log n)$

Space Complexity: $\mathcal{O}(n + |\mathcal{C}| \cdot k^2) \stackrel{a}{=} \mathcal{O}(n)$

[a] Assuming k relatively small and fixed, and the **worst case** in which $k' = 1$ and thus $|\mathcal{C}| = n$

Note that the training of the $|\mathcal{C}|$ models can very easily be parallelised...

[1] The mathematical formulation of these steps is not discussed here.

The Experimental Setting

- Comparison between FaLKNR ENN RENN AkNN and AkNNc (a conservative variant of AkNN) on the basis of the induced NN generalization accuracies
- Neighbourhood sizes: $k = 3$ for ENN RENN AkNN and AkNNc, $k = 1000, k' = 250$ for FaLKNR
- C of FaLKNR estimated with local model selection, σ of RBF kernel estimated locally as the median distance in the neighbourhood
- Comparison performed on 9 datasets with large training sets (from 50k to more than 500k training examples)[1]

[1] Details of the used datasets can be found in the paper.

Accuracy and Computational Results of FaLKNR

	NN accuracies (in %)						comput. performances (s)		
dataset	uned.	FaLKNR	ENN	RENN	AkNN	AkNNc	FaLKNR[2]	ENN[3]	speed-up
ijcnn1	96.6	**96.7**	96.3	96.0	96.0	96.2	39	61	1.6
connect-4	66.2	**69.8**	69.3	68.3	69.3	69.4	455	1244	2.7
seismic	65.3	**73.3**	71.9	72.6	72.2	71.8	950	3025	3.2
acoustic	67.4	**75.3**	73.7	74.2	74.0	73.8	331	2641	8.0
2-spirals	83.2	**88.6**	87.6	88.1	87.9	87.7	97	44	0.5
census-inc	92.6	**94.5**	94.2	94.3	94.4	94.3	771	6965	9.0
poker-hand	56.6	**60.7**	57.8	58.3	58.3	58.0	2230	16905	7.6
rna	**96.3**	95.8	94.0	94.0	94.3	94.3	550	3340	6.1
cover-type	**95.8**	95.4	95.2	95.0	95.1	95.2	993	1538	1.5

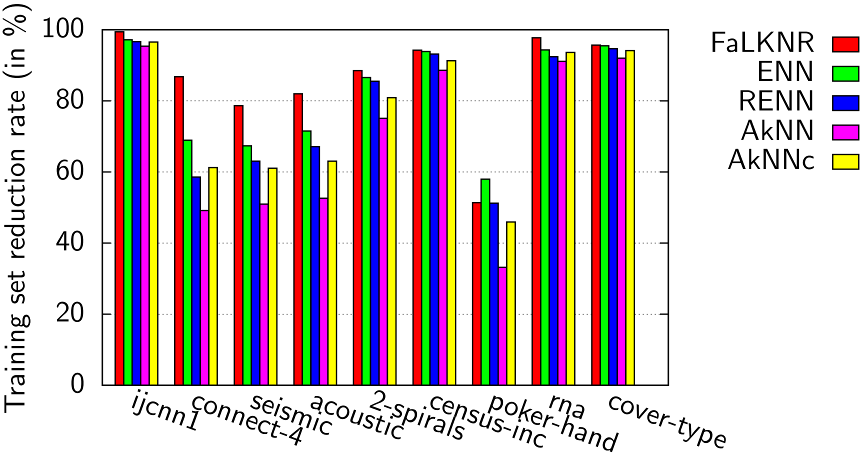
Wilcoxon Signed Rank Test ($\alpha = 0.05$)

NN accuracies using FaLKNR are better than unedited NN accuracies and accuracies of NN edited with ENN RENN AkNN AkNNc.

[2] The computational time of FaLKNR includes the local model selection.

[3] ENN is implemented using Cover Trees and it is faster than RENN, AkNN and AkNNc.

Reduction Rates of the Edited Training Sets



Final Remarks and Future Works

- The maximal margin principle is effective for noise reduction both for small [Segata, Blanzieri, Delany, Cunnigham, 2008] and large CBR systems
- FaLKNR overcome state-of-the-art noise reduction techniques with statistical significance for large CBR systems
- FaLKNR is generally faster than traditional approaches based on locality and CBR rules

Possible further developments

- Other principles different from the maximal margin principle can be exploited locally with the same framework
- FaLKNR is promising also for cleansing bioinformatics datasets and as preprocessing step for other machine learning approaches
- FaLKNR can integrate a competence preserving step to decrease the size of the case base without decreasing NN accuracies

A SW Library for Fast Local Kernel Machines: FaLKM-lib

FaLKM-lib v1.0 [Segata, 2009] is a software library for fast local kernel machine implemented in C++. It contains the following modules:

- **FkNN** a (kernel-space) kNN implementation using Cover Trees
- **FkNNSVM** the kNNSVM algorithm for Local SVM
- **FkNNSVM-nr** a noise reduction algorithm based on kNNSVM
- **FaLK-SVM** very fast and scalable learning with local kernel machines
- **FaLKNR** the fast and scalable noise reduction algorithm

The modules share also tools for model selection, efficient local model selection, performance assessment...

FaLKM-lib is freely available for research purposes

You can download the code, datasets, benchmark, additional infos and papers, and examples at `http://disi.unitn.it/~segata/FaLKM-lib`
Any comments/suggestions are welcome!

A Scalable Noise Reduction Technique for Large Case-Based Systems

Nicola Segata
segata@disi.unitn.it

Enrico Blanzieri
blanzier@disi.unitn.it

Pádraig Cunningham
Padraig.Cunningham@ucd.ie

Dept. of Information Engineering
and Computer Science,
University of Trento, Italy.

Computer Science,
University College Dublin,
Dublin, Ireland.

8th International Conference on Case-Based Reasoning

Seattle, University of Washington, USA

July 22, 2009