Using Neural Networks for Dynamical Reduction of the Features Space Dimensionality

Yury Tsoy

Tomsk Polytechnic University
Tomsk, Russia

IFOST 2012, Tomsk (Sept. 18, 2012)

Table of contents

Introduction

Features space dimensionality reduction problem arises in many practical applications.
Principal Components Analysis (PCA) is one of the most popular methods.
Concerns computing of **eigenvectors** for the data covariance matrix.
Geometrically plausible, fast and efficient ($O(n^{2.36})$ with all the numerical tricks).

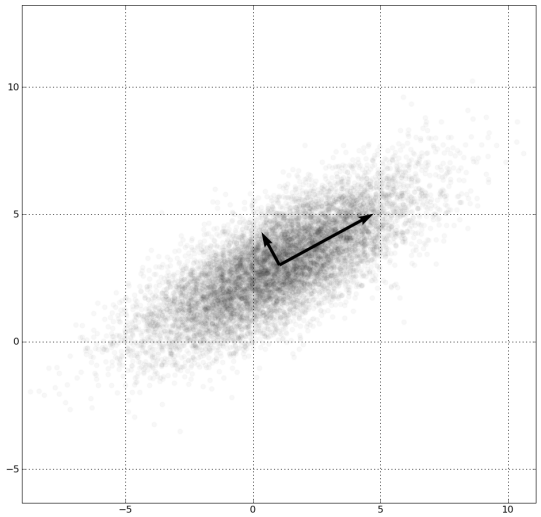


Figure: Illustrative example for eigenvectors.

Generalized Hebbian Algorithm

Algorithm

1. Initialization of the linear ANN without hidden nodes. The number of outputs = required dimensionality.
2. Update ANN weights. For each training sample:

$$
\begin{aligned}
y_j(t) &= \sum_{i=1}^{m} w_{ji}(t) x_i(t), \\
\Delta w_{ji}(t) &= \eta \left[y_j(t) x_i(t) - y_j(t) \sum_{k=1}^{j} w_{ki}(t) y_k(t) \right],
\end{aligned}
$$

3. If stopping criterion is failed go to **Step 2**.

Generalized Hebbian Algorithm

Algorithm

1. Initialization of the linear ANN without hidden nodes. The number of outputs = required dimensionality.
2. Update ANN weights. For each training sample:

$$y_j(t) = \sum_{i=1}^{m} w_{ji}(t)x_i(t),$$

$$\Delta w_{ji}(t) = \eta\left[y_j(t)x_i(t) - y_j(t)\sum_{k=1}^{j} w_{ki}(t)y_k(t)\right],$$

3. If stopping criterion is failed go to **Step 2**.

Problems with GHA

- Tricky for dimensionality reduction (explained below).
- Slow convergence (may take thousands of iterations).
- Takes much time.

Generalized Hebbian Algorithm

GHA for dimensionality reduction (two options)

1. Compute all eigenvectors and eigenvalues and apply selection mechanism to reduce dimensionality. **Higher computational complexity**.
2. Set the required dimensionality beforehand. **Requires guessing of "true" data set dimensionality.**

Generalized Hebbian Algorithm

GHA for dimensionality reduction (two options)

1. Compute all eigenvectors and eigenvalues and apply selection mechanism to reduce dimensionality. **Higher computational complexity**.
2. Set the required dimensionality beforehand. **Requires guessing of "true" data set dimensionality.**

GHA is relatively slow

Problem name	GHA (50 iterations), ms	MATLAB `cov+eig` time, ms
cancer1	218.74	0.07
card1	13113.56	4.4
horse1	8463.24	6.6
thyroid1	12206.75	1.8

Generalized Hebbian Algorithm

GHA for dimensionality reduction (two options)

1. Compute all eigenvectors and eigenvalues and apply selection mechanism to reduce dimensionality. **Higher computational complexity**.
2. Set the required dimensionality beforehand. **Requires guessing of "true" data set dimensionality.**

GHA is relatively slow

Problem name	GHA (50 iterations), ms	MATLAB `cov+eig` time, ms
cancer1	218.74	0.07
card1	13113.56	4.4
horse1	8463.24	6.6
thyroid1	12206.75	1.8

Sweet dreams

It would be good if we could remove output nodes dynamically.

1. Reduces computational complexity.
2. Does not require guessing the data dimensionality.

A bit of theory

If we want to remove outputs of ANN dynamically we've got to do it using approximate eigenvectors' coordinates.

Can we remove inexact non-informative eigenvectors?

A bit of theory

If we want to remove outputs of ANN dynamically we've got to do it using approximate eigenvectors' coordinates.

Can we remove inexact non-informative eigenvectors?

Proposition

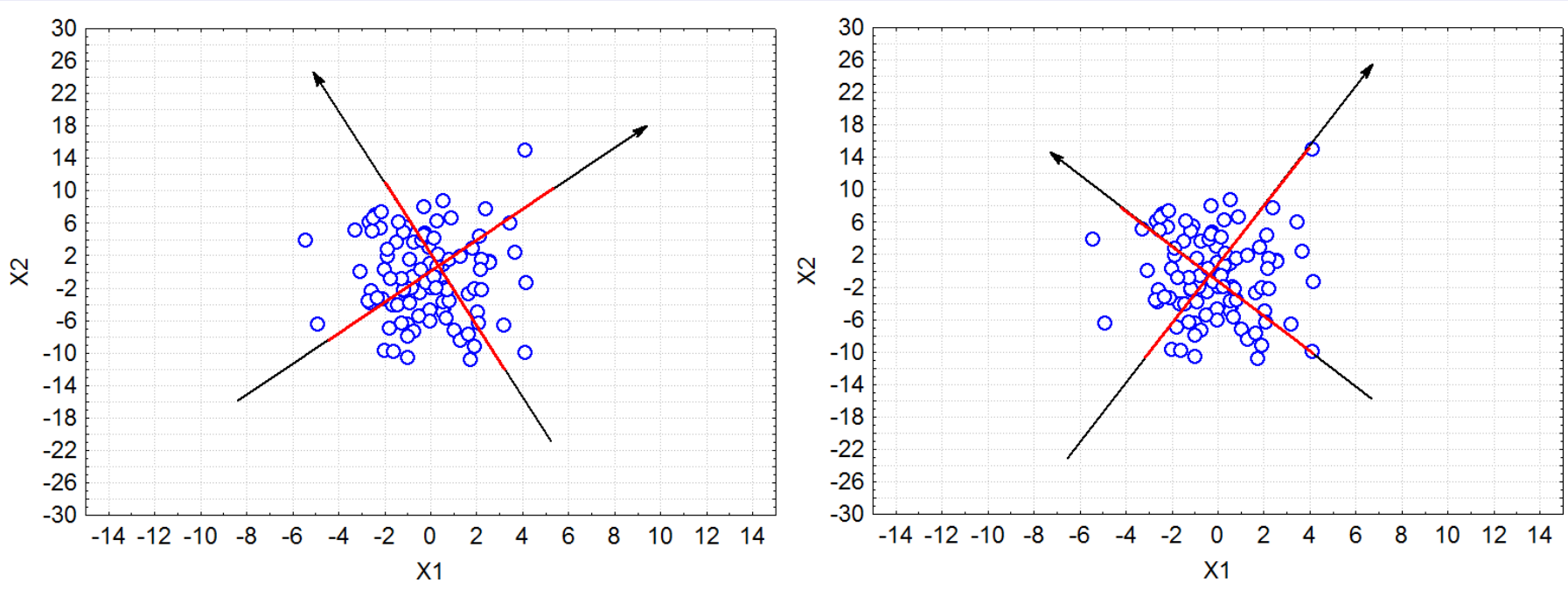
Let $\mathbf{X} = \{\mathbf{X}_i, i = 1, \ldots, N\}, \mathbf{X}_i \in \mathcal{R}^n$ be a set of data points and $\mathbf{Q} = \{\mathbf{q}_i, i = 1, \ldots, n\}$ is an orthogonal basis in $\mathcal{R}^n$. Denote $proj_{\mathbf{q}_i}(\mathbf{X})$ as projection of data points from $\mathbf{X}$ onto coordinate vector $\mathbf{q}_i$, and $Var(proj_{\mathbf{q}_i}(\mathbf{X}))$ as a variance of correspondent projections. Then summation over all dimensions

$$\sum_{i=1,\ldots,n} Var(proj_{\mathbf{q}_i}(\mathbf{X}))$$

is constant and doesn't depend on $\mathbf{Q}$.

In other words . . .

Illustrative example



Sum variance of projections can be treated as a finite resource, which is 'distributed' over coordinate vectors (*eigenvectors estimates*).

How to decide?

We suppose that all eigenvectors estimates are sorted by projection variances (e.g. significance).

Criterion for throwing away "bad" eigenvectors estimates:

$$\frac{Var(proj_{\hat{\mathbf{q}}_0}(\mathbf{X}))}{Var(proj_{\hat{\mathbf{q}}_i}(\mathbf{X}))} > \tau \tag{1}$$

where $\hat{\mathbf{q}}_i$ – estimate of the i-th eigenvector, τ is a threshold. Typical values for τ are 5, 10, 15, 20, ...

It is possible to truncate low-informative subspaces without knowing exact coordinates of principal eigenvectors $\Rightarrow$ **pseudo-PCA (pPCA)**.

How to decide?

We suppose that all eigenvectors estimates are sorted by projection variances (e.g. significance).

Criterion for throwing away "bad" eigenvectors estimates:

$$\frac{Var(proj_{\hat{\mathbf{q}}_0}(\mathbf{X}))}{Var(proj_{\hat{\mathbf{q}}_i}(\mathbf{X}))} > \tau \tag{1}$$

where $\hat{\mathbf{q}}_i$ – estimate of the i-th eigenvector, τ is a threshold. Typical values for τ are 5, 10, 15, 20, ...

It is possible to truncate low-informative subspaces without knowing exact coordinates of principal eigenvectors $\Rightarrow$ **pseudo-PCA (pPCA)**.

- Smaller values of $\tau \rightarrow$ smaller dimensionality (more features removed).
- Larger values of $\tau \rightarrow$ larger dimensionality (less features removed).

Dynamical GHA

1. Initialization of the linear ANN without hidden nodes. The number of outputs = required dimensionality.
2. Compute projections variances and remove output nodes, which satisfy to the criterion (1).
3. Update ANN weights. For each training sample:

$$
\begin{aligned}
y_j(t) &= \sum_{i=1}^{m} w_{ji}(t) x_i(t), \\
\Delta w_{ji}(t) &= \eta \left[y_j(t) x_i(t) - y_j(t) \sum_{k=1}^{j} w_{ki}(t) y_k(t) \right],
\end{aligned}
$$

4. If stopping criterion is failed go to **Step 2**.

The Neuroevolutionary Algorithm

1. **Initialize** random population, each individual is a candidate solution for pPCA (linear ANN without hidden nodes).
2. **Evaluate** each individual using the following fitness function:

$$\begin{aligned} f &= \alpha * \sum_{i=1,\ldots,n} Var(proj_{\hat{\mathbf{q}}_i}(\mathbf{X})) \rightarrow max, \\ \alpha &= (\hat{\mathbf{q}}_0^T \mathbf{r})^2, \mathbf{r} = \mathbf{C}\hat{\mathbf{q}}_0 / \|\mathbf{C}\hat{\mathbf{q}}_0\|. \end{aligned}$$

   and remove nodes, for which criterion (1) is satisfied.
3. **Selection**
4. **Crossing** and **Mutation**.
5. If algorithm's run is completed then proceed to **Step 6**, otherwise proceed to **Step 2**.
6. **Return** the best found individual.

Goals & Test Problems

Goals

1. It is important to find out whether efficient dimensionality reduction is possible.
2. Since pPCA doesn't yield linear subspaces associated with the principal components it's also important to know how this affects classification accuracy.

Proben1 data set

Proben1 problem name	# of features	# of classes	Training / Validation / Test sets sizes
cancer1	9	2	350 / 175 / 174
card1	51	2	345 / 173 / 172
diabetes1	8	2	384 / 192 / 192
glass1	9	6	107 / 54 / 53
heart1	35	2	460 / 230 / 230
horse1	58	3	182 / 91 / 91
thyroid1	21	3	3600 / 1800 / 1800

Comparison

Classification errors (%) for different values of τ

Problem	$\tau = 5$		$\tau = 10$	
	NE pPCA	DGHA	NE pPCA	DGHA
cancer1 (9)	2.3 (1)	2.30 (1)	1.7 (2.5)	2.82 (1.2)
card1 (51)	14.48 (8.3)	16.28 (28.5)	13.66 (11.7)	15.41 (50.7)
diabetes1 (8)	24.74 (7.6)	24.95 (7.6)	24.38 (8)	25.00 (8)
glass1 (9)	71.7 (1)	36.23 (5.5)	40.38 (4.3)	33.02 (6.7)
heart1 (35)	21.3 (9.9)	21.13 (22.3)	21.74 (15.7)	19.91 (31.5)
horse1 (58)	34.07 (1)	28.79 (35.3)	32.86 (5.3)	29.23 (57.7)
thyroid1 (21)	7.24 (7)	7.19 (8.9)	7.21 (8)	6.03 (16.3)

Problem	$\tau = 15$		$\tau = 20$	
	NE pPCA	DGHA	NE pPCA	DGHA
cancer1 (9)	1.44 (4)	1.78 (4.6)	1.67 (5.6)	1.84 (6.3)
card1 (51)	16.8 (16.4)	15.64 (51)	16.4 (19.8)	15.76 (51)
diabetes1 (8)	24.43 (8)	25.00 (8)	24.64 (8)	25.00 (8)
glass1 (9)	37.55 (6.8)	32.07 (7.9)	34.91 (7.4)	32.26 (8.4)
heart1 (35)	22.52 (17.7)	20.00 (34.2)	21.13 (19.1)	20.04 (35)
horse1 (58)	29.89 (27.3)	30.66 (58)	26.81 (32.4)	29.56 (58)
thyroid1 (21)	6.78 (14)	5.87 (18)	6.73 (15)	5.92 (19.8)

Change of averaged mean dimensionality

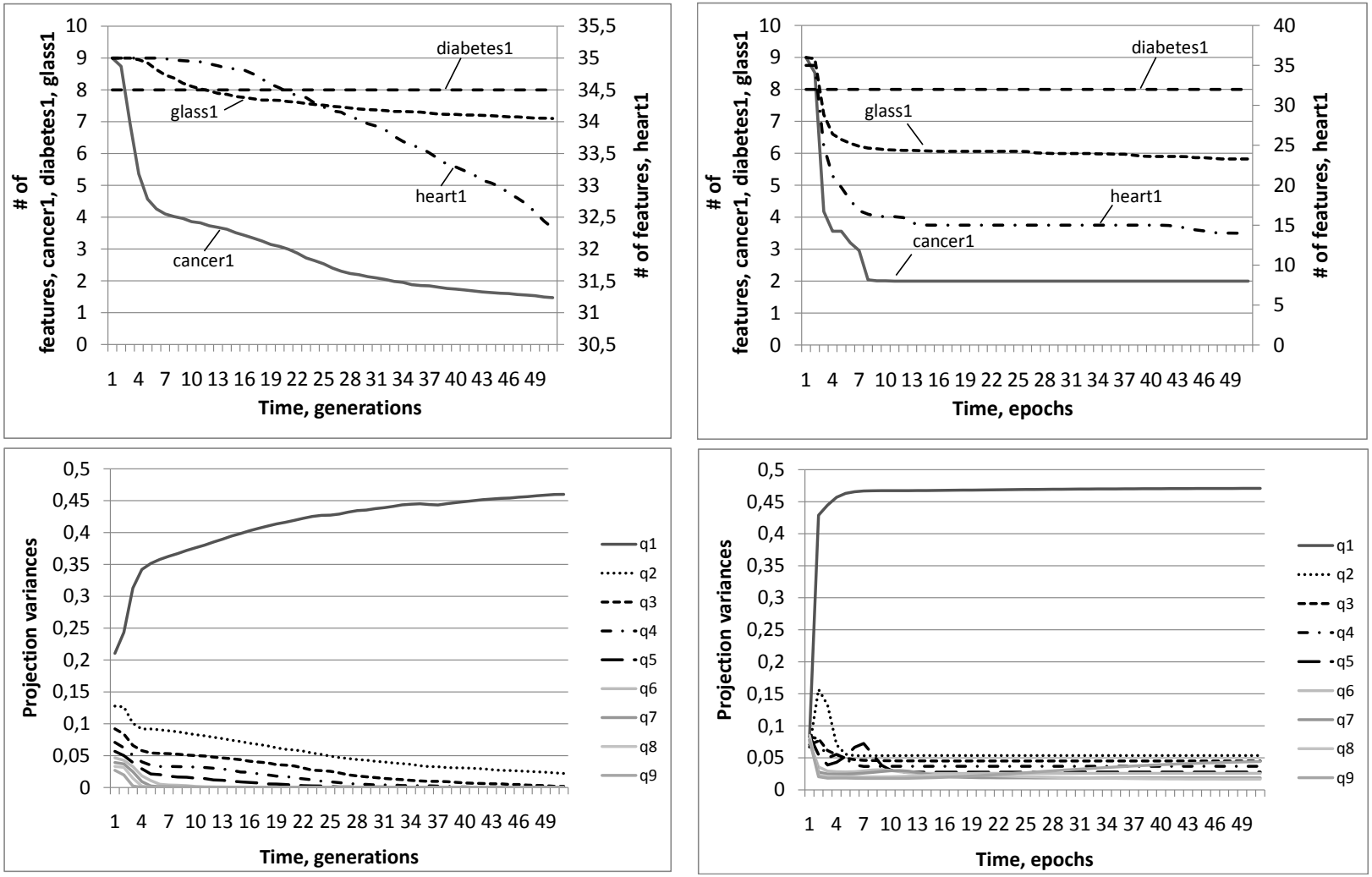


Figure: Change of averaged dimensionality and projection variances for NE pPCA (left) and DGHA (right). $\tau = 10$.

Comparison

- Layered Genetic Programming (FLGP) (Lin etal., 2008);
- Recursive Feature Elimination combined with multi-layered neural network (RFENN)
- . . . and support vector machines (RFESVM) (Windeatt, 2011).

Method	cancer1	diabetes1	heart1
FLGP	2.24 (5.2)	27.24 (6.1)	22.40 (11.0)
RFENN	4.00 (7)	24.90 (2)	21.00 (27)
RFESVM	3.70 (7)	24.50 (3)	20.00 (18)
NE pPCA, $\tau = 15$	1.78 (4.6)	25.00 (8)	20.00 (34.2)
DGHA, $\tau = 15$	1.84 (4)	24.32 (8)	21.78 (18)

Table: Comparison of the test set classification errors (%) obtained using different features selection methods for *cancer1*, *diabetes1* and *heart1* problems. Average dimensionality of the resulting features space is given in brackets.

Approximate eigenvectors

Ok, we can work with approximate covariance matrix eigenvectors. Sources of inexactness:

- Approximate methods to compute eigenvectors.
- Inexact covariance matrix.

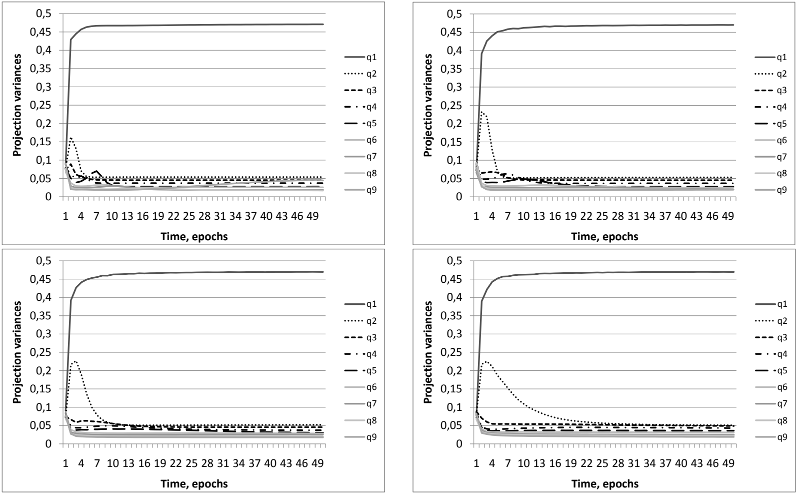
Dynamical GHA with reduced data set

1. Initialization of the linear ANN without hidden nodes. The number of outputs = required dimensionality.
2. Compute projections variances and remove output nodes, which satisfy to the criterion (1).
3. Sample $r\%$ of the data from the training set to update ANN weights.
4. Update ANN weights. For each training sample:

$$
\begin{aligned}
y_j(t) &= \sum_{i=1}^{m} w_{ji}(t) x_i(t), \\
\Delta w_{ji}(t) &= \eta \left[y_j(t) x_i(t) - y_j(t) \sum_{k=1}^{j} w_{ki}(t) y_k(t) \right],
\end{aligned}
$$

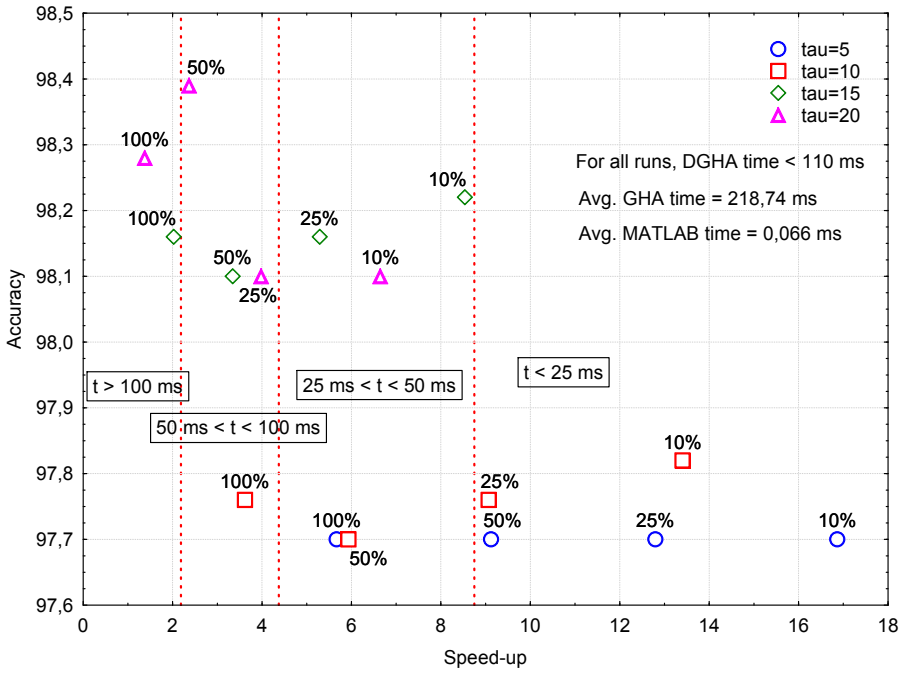
5. If stopping criterion is failed go to **Step 2**.

Change of projection variances (cancer1)

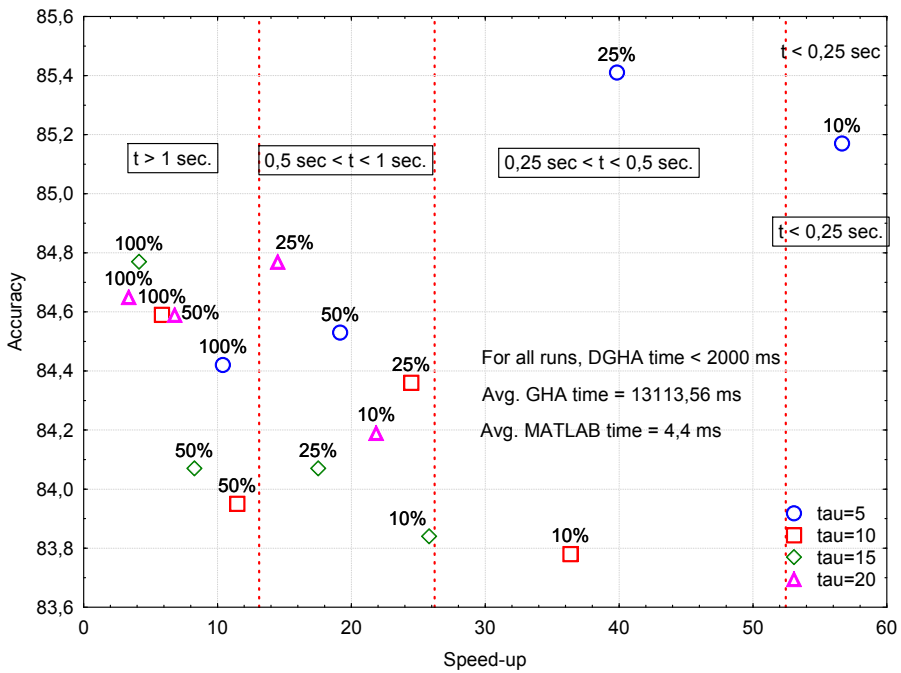


a) 100% of data; b) 50% of data;
c) 25% of data; d) 10% of data.

Speed-up VS Accuracy

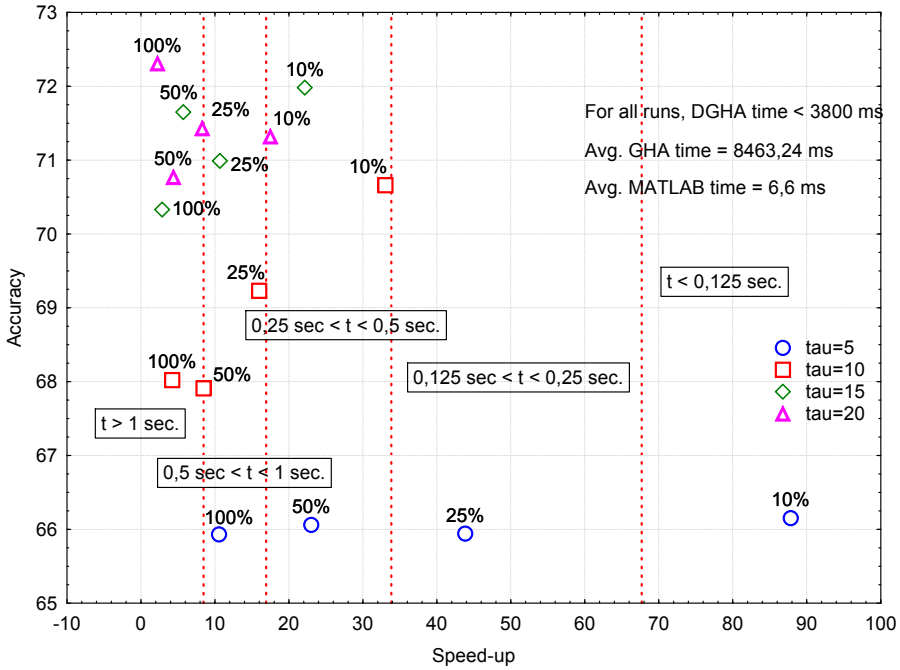


cancer1

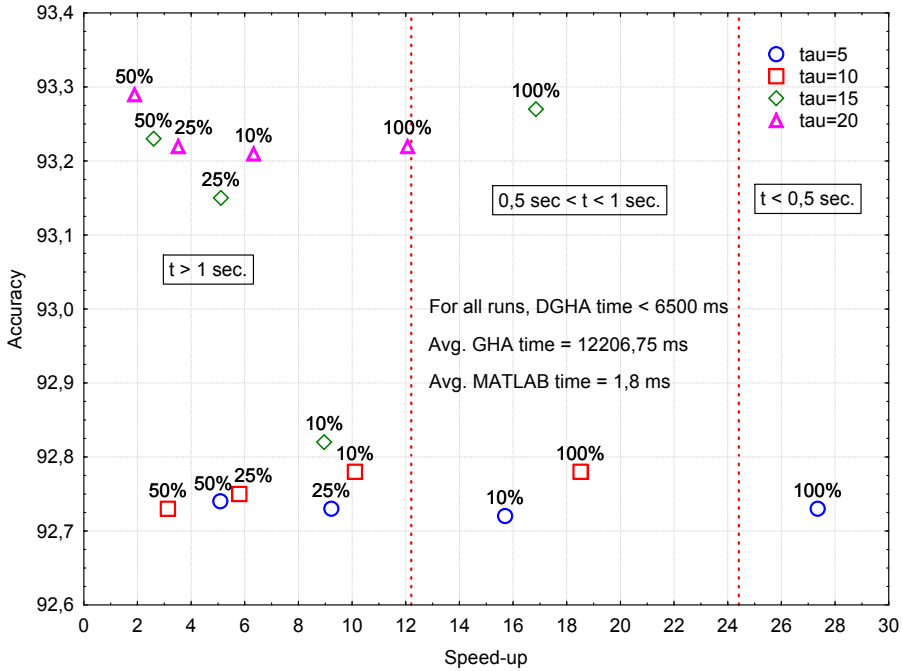


card1

Speed-up VS Accuracy



horse1



thyroid1

Change of averaged mean dimensionality

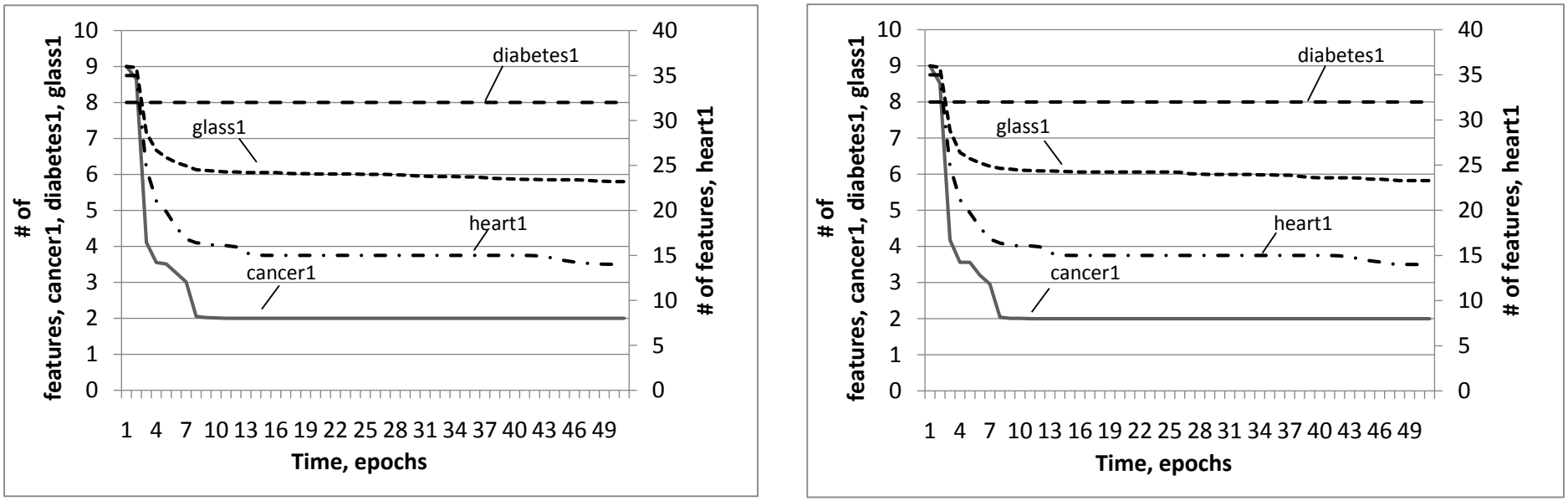


Figure: Comparison of change of averaged features space dimensionality for *cancer1*, *diabetes1*, *glass1* and *heart1* problems for DGHA with reduced data set (10%, left) and with full data set (right). $\tau = 10$.

Conclusion

Quite a simple proposition lead to:

- Novel way for dynamical dimensionality reduction using inexact coordinates of eigenvectors (pseudo-PCA).
- NE pPCA – way of evolutionary training of ANN with tractable and understandable results.
- Dynamical modification of the GHA algorithm (DGHA).
- Use of part of data to speed-up the GHA, DGHA and NE pPCA.
- DGHA is much more practically useful than GHA and NE pPCA due to its speed.

Future Research:

1. Parallelization of the NE pPCA. The most time consuming part is computation of fitness ($\tilde{7}$5-80% of time). Each individual can be evaluated in parallel.
2. Constraints for pPCA: use criteria from PCA and/or try to keep certain amount of information when performing nodes removal.

Acknowledgements

Grants

The research is supported by the Russian Foundation for Basic Researches (projects no. 11-07-00027-a, 12-08-00296-a).

Colleagues

Author thanks Dr. Yu. Burkatovskaya for her notes on the paper contents.

Source Code

Mental Alchemy (`http://code.google.com/p/mentalalchemy`) and Encog (`http://www.heatonresearch.com/encog`) open-source projects were used to implement all the algorithms and experiments.

Thank you for attention!

Yury Tsoy
yurytsoy@gmail.com