Model interpretability and deep learning

Stockholms universitet

Panagiotis Papapetrou, PhD

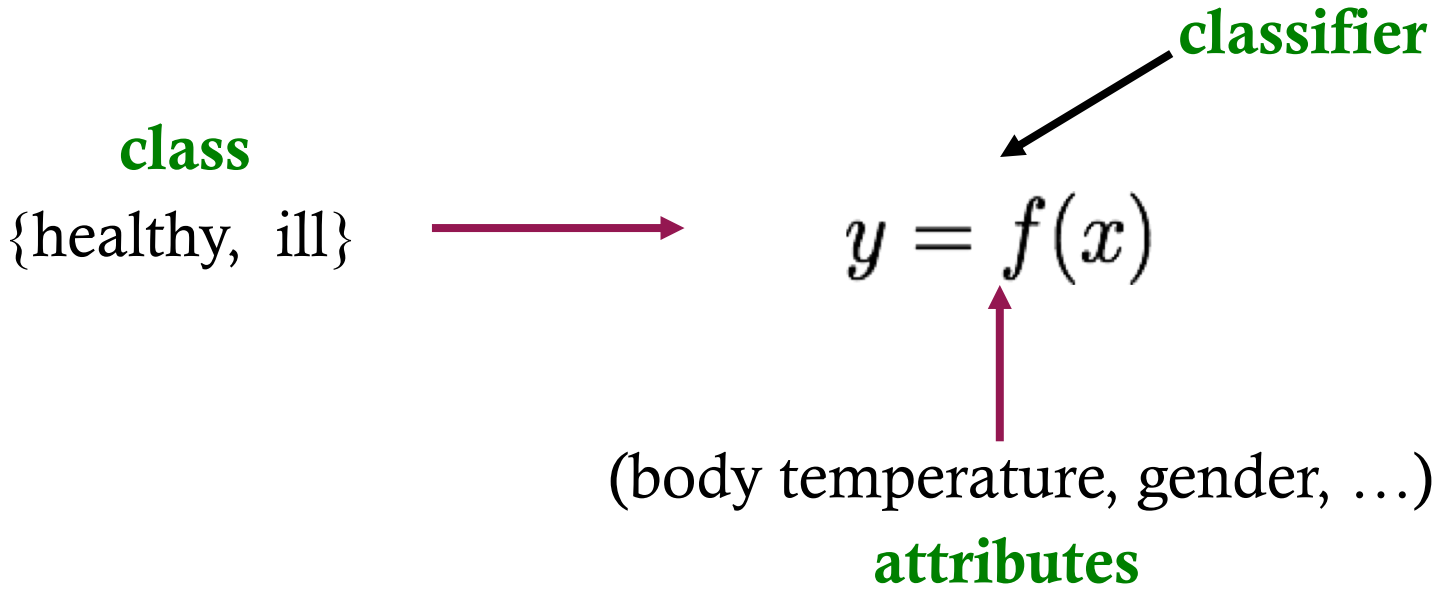
Professor, Stockholm University

Adjunct Professor, Aalto University

Part IV - Outline

- **Interpretable models** for EHRs
- Time series **feature tweaking**
- **Feature grouping** and explainability
- **Recurrent Neural Networks** for diagnosis prediction
- Deep learning with **attention mechanisms**

Classifiers



Properties of a good classifier include:

- high accuracy
- interpretability (*e.g., why is a person healthy?*)

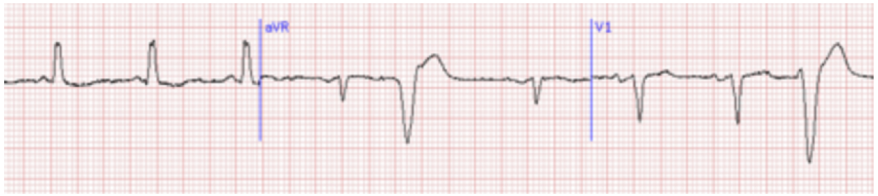
Interpretable and actionable models

- Trade-offs between interpretability + accura
- Ability to understand the irable outcomes without compromising p 2016]

I can tell you **what changes you need to make** to the patient record, so that I can **change my prediction** ☺

black box classifier

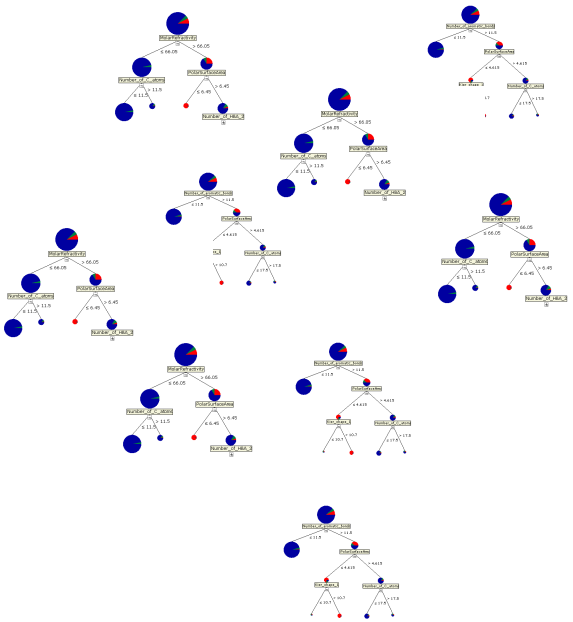
The patient will die from HF in **2 days**!



Now what? Please tell me **why**?

Actionable feature tweaking for random forests

Random Forest (opaque model)



example x

- Let x be a true-negative instance
- **Goal:** minimum number of feature tweaks (changes) so that x becomes true-positive, x'

prediction: -1

$$\hat{f}(\mathbf{x}) = -1 \iff \left(\sum_{k=1}^{K} \hat{h}_k(\mathbf{x})\right) \leq 0$$

$$\mathbf{x}' = \arg\min_{\mathbf{x}^*} \left\{ \delta(\mathbf{x}, \mathbf{x}^*) \mid \hat{f}(\mathbf{x}) = -1 \wedge \hat{f}(\mathbf{x}^*) = +1 \right\}$$

Tolomei et al. **Interpretable Predictions of Tree-based Ensembles via Actionable Feature Tweaking**, KDD (and ARXIV), 2017

Solution

- Let x be a true-negative instance
- **Goal:** minimum number of feature tweaks (changes) so that x becomes true-positive, x'

Observe:

- If the prediction of the RF is -1, then at least half of its trees predict -1
- If the prediction of a tree is -1, then the example is passed through a negative path, i.e., a path that predicts the class to be -1
- **Solution:** revert these paths and consequently the trees!

Note: if a single transformation results in changing another tree's decision, then ignore it!

- Focus on the trees that predict -1
- For each tree: explore the positive paths, i.e., those that predict +1
- **Apply the transformations imposed by the positive path**

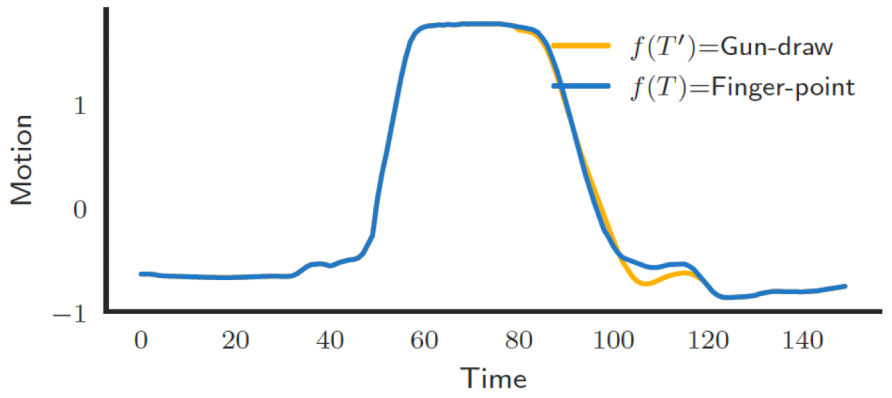
$$\mathbf{x}^{+}_{j(\epsilon)}[i] = \begin{cases} \theta_i - \epsilon & \text{if the } i\text{-th condition is } (x_i \leq \theta_i) \\ \theta_i + \epsilon & \text{if the } i\text{-th condition is } (x_i > \theta_i) \end{cases}$$

- Choose the transformation with the **minimum cost**

$$\mathbf{x}' = \underset{\mathbf{x}^{+}_{j(\epsilon)} \in \Gamma \mid \hat{f}(\mathbf{x}^{+}_{j(\epsilon)})=+1}{\arg\min} \left\{ \delta(\mathbf{x}, \mathbf{x}^{+}_{j(\epsilon)}) \right\}$$

Time series tweaking

What is the minimum number of changes to apply to a time series T so that a given opaque classifier changes its prediction?

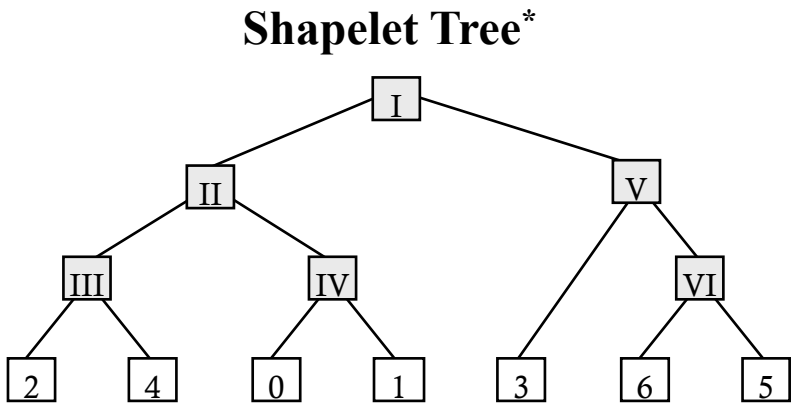


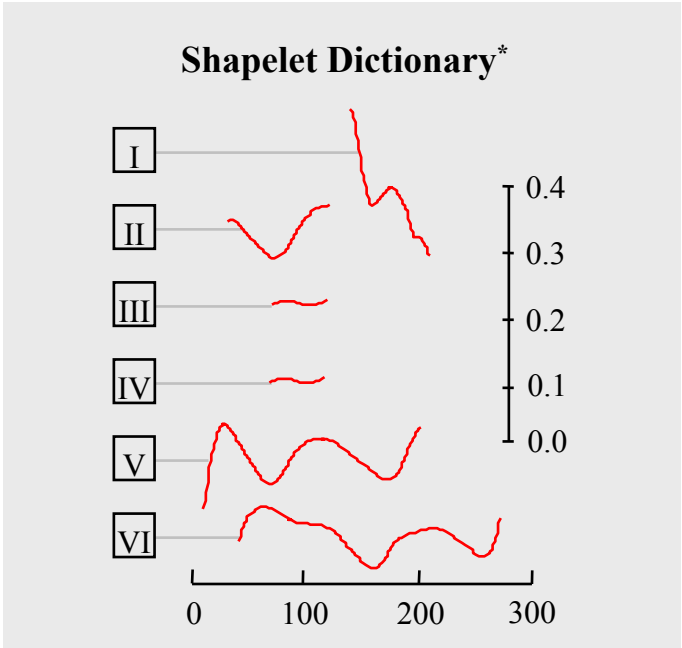
$$T \rightarrow T^1 \rightarrow T^2 \rightarrow \ldots \rightarrow T'$$

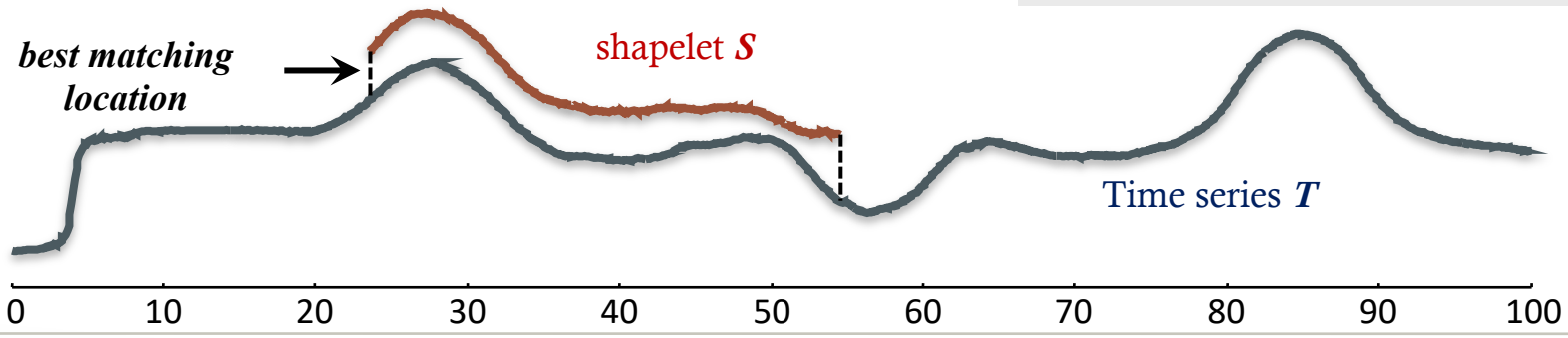
- **Reversible tweaking:** each subsequent transformation can override a previous one
- **Irreversible tweaking:** each subsequent transformation cannot override a previous one

Random Shapelet Forests

Shapelets: class-distinctive time series subsequences capturing local trends in time series





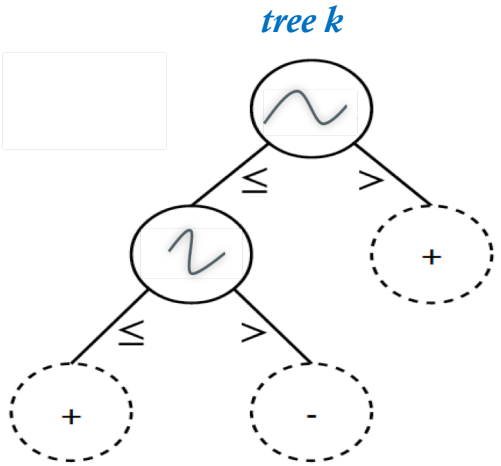


* *Figures taken from Eamonn Keogh*

Root-leaf path

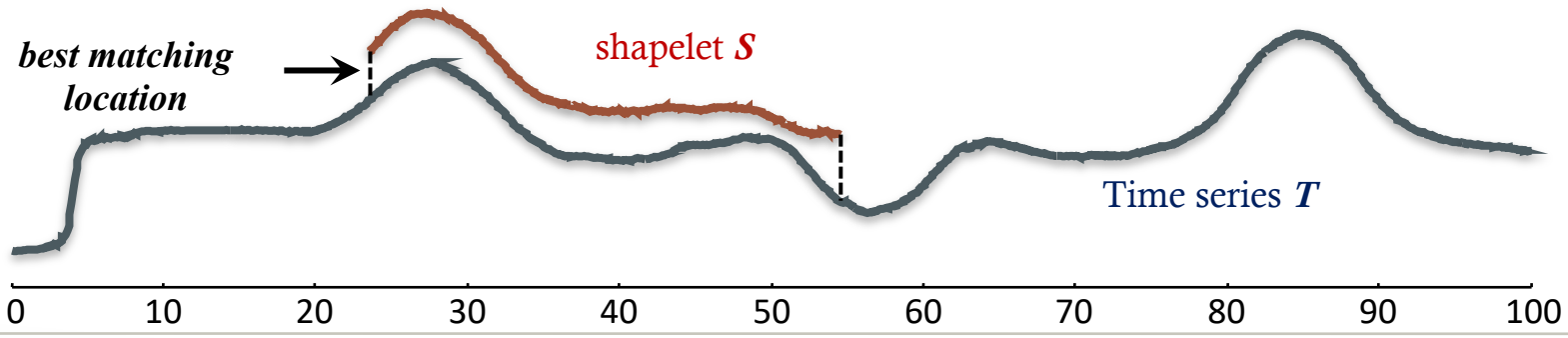
- Each tree defines several root-leaf paths
- The *j*th path of *the k*th *tree* is defined as follows:

$$p_{k,j} = \{(x_1 \lesseqgtr \theta_1), (x_2 \lesseqgtr \theta_2), \ldots, (x_n \lesseqgtr \theta_n)\}$$



Non-leaf node condition:
Euclidean distance, lowest scoring subsequence match of S in T

$$d_s(\mathcal{S}_k^j, \mathcal{T}) \leq \theta_k^j \qquad d_s(\mathcal{S}_k^j, \mathcal{T}) > \theta_k^j$$



Time series tweaking: solution

- Focus on the trees that predict -1
- For each tree, explore the positive paths, i.e., those that predict +1
- Try to force those trees to predict +1 by "tweaking" shapelet features of T

Given a non-leaf node (S^j_k, θ^j_k)

- **Increase distance:**
  - If S^j_k exists in T, that is $d_s(S^j_k, T) \leq \theta^j_k$
  - and the current node condition demands otherwise
  - ✓ Increase the distance of all matching instances of S^j_k, so that they all fall above the distance threshold θ^j_k

Time series tweaking: solution

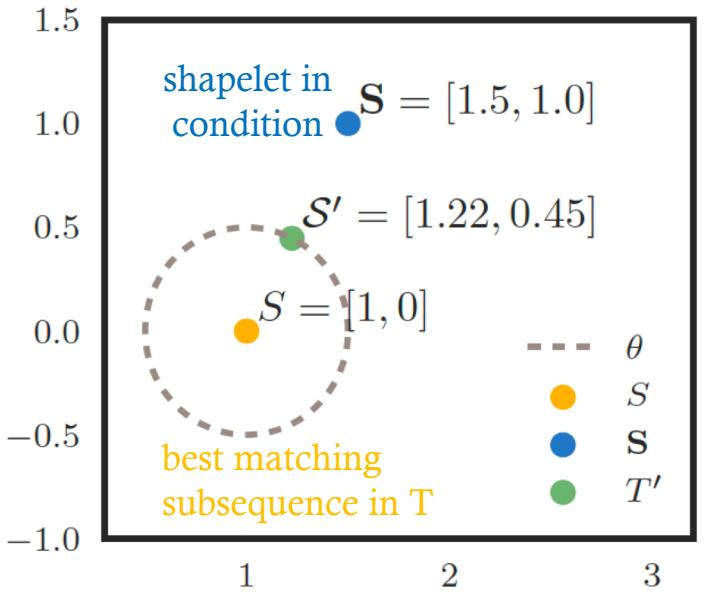
- Focus on the trees that predict -1
- For each tree, explore the positive paths, i.e., those that predict +1
- Try to force those trees to predict +1 by "tweaking" features of T

Given a non-leaf node (S^j_k, θ^j_k)

- **Decrease distance:**
  - If S^j_k does not exist in T, that is $d_s(\mathcal{S}^j_k, \mathcal{T}) > \theta^j_k$
  - and the current node condition demands otherwise
  - ✓ Decrease the distance of the best matching instance of S^j_k, so that it falls below the distance threshold θ^j_k

How to transform the time series?

- Consider S as an m-dimensional point
- Define an m-sphere with S as its center and radius θ



- The transformed time series counterpart of **S** is given by the following equation:

$$\tau_{\mathcal{S}}(\mathbf{S}, p_{ik}^{j}, \epsilon) = \mathcal{S}_k^j + \frac{\mathcal{S}_k^j - \mathbf{S}}{\|\mathcal{S}_k^j - \mathbf{S}\|_2}(\theta_k^j + (\epsilon\delta_{ik}^j))$$

Irreversible time series tweaking

- Changes cannot override earlier transformations
- Locking data structure: keeps track of the *changed* regions
- A transformation is allowed if and only if the region affected is **not** locked (has not been changed)
- *Early abandoning of transformations*: if the cumulative cost is above the current best successful transformation

Reversible time series tweaking

- Changes can override earlier transformations
- *Prediction ordering*: pre-order all transformations based on their cost (min to max)
- The first transformation that flips the class label is the solution

Experimental setup

- UCR time series repository:
  - all binary classification datasets (26 datasets)
- Competitor:
  - 1-NN under the Euclidean distance

$$\tau_{NN}(\mathcal{T}, y') = \underset{\{\mathcal{T}' | (\hat{y}, \mathcal{T}') \in \mathcal{D}, \hat{y} = y'\}}{\arg\min} d_E(\mathcal{T}, \mathcal{T}')$$

Evaluation – metrics

Average cost of successful transformation, i.e.,

how costly is the transformation?

$$c_\mu(\tau, y') = \frac{1}{n} \sum_{i=1}^{n} c(\mathcal{T}_i, \tau(\mathcal{T}_i, y'))$$

Compactness of transformation, i.e.,

how much of the time series is changed?

$$compact(\mathcal{T}, \mathcal{T}') = \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} diff\,(T_i, T_i') \ ,$$

where

$$diff\,(T_i, T_i') = \begin{cases} 1, \text{ if } |T_i - T_i'| \leq e \\ 0, \text{ otherwise.} \end{cases}$$

Evaluation – result

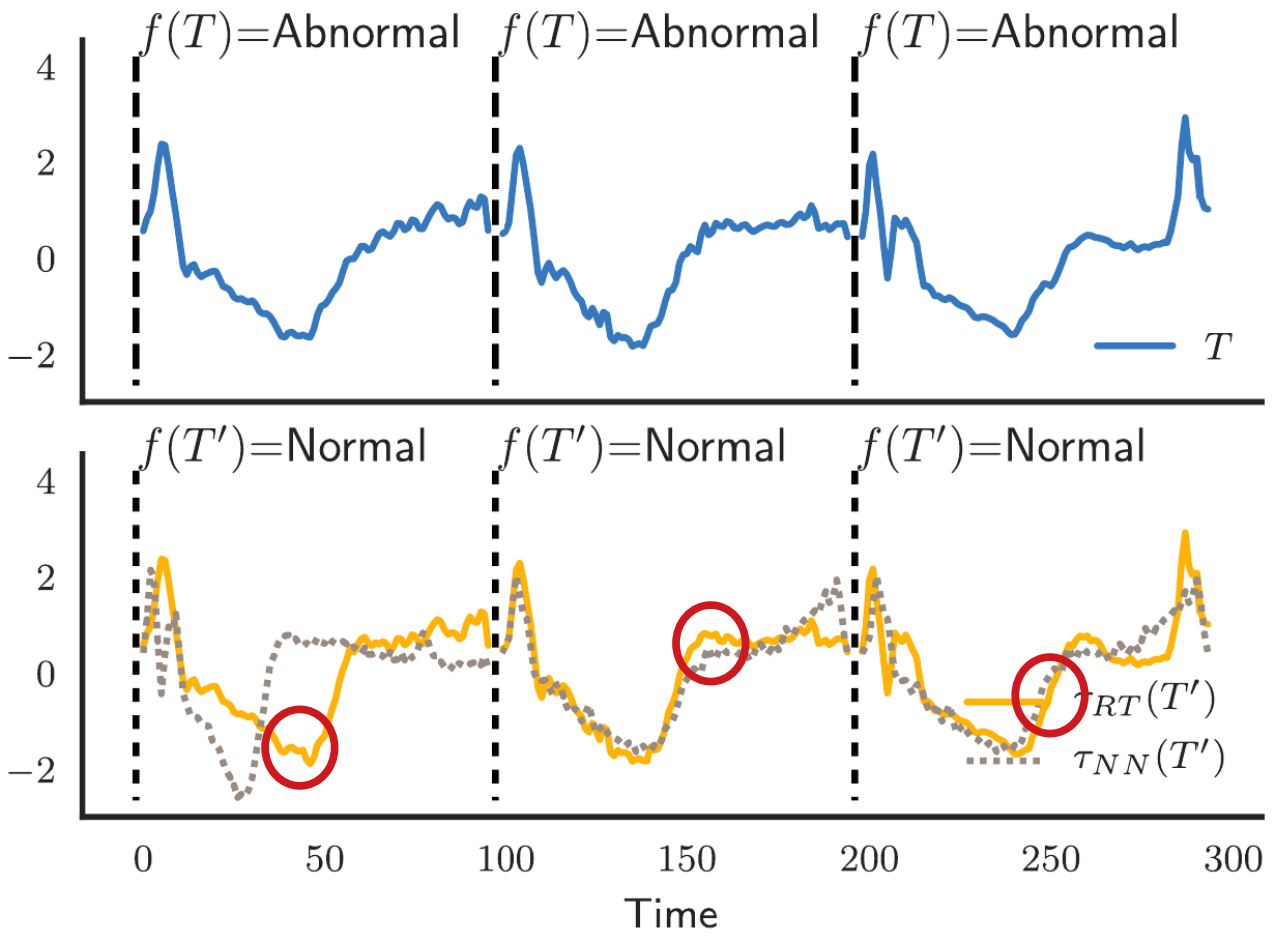
Reversible tweaking results in the **least costly** transformations

Irreversible tweaking results in the **most compact** transformations

The baseline is too naive

Dataset	Cost			Compactness			Accuracy	
	τ_{RT}	τ_{IRT}	τ_{NN}	τ_{RT}	τ_{IRT}	τ_{NN}	RSF	NN(1)
	7.3810	**7.3810**	26.6223	**0.5737**	**0.5737**	1.0000	**0.8750**	0.7500
	4.5071	4.5098	15.6695	**0.5048**	0.5169	1.0000	**1.0000**	0.6250
	1.1447	1.1846	1.9178	0.3824	**0.1809**	1.0000	**1.0000**	**1.0000**
	2.2197	2.5132	22.4809	0.4123	**0.4044**	1.0000	**0.7000**	0.4900
	0.9314	1.1150	1.1704	0.5917	**0.4466**	0.9999	**0.7886**	0.7143
	2.2725	3.1455	30.0943	**0.7449**	0.7577	1.0000	**0.7826**	0.6630
	1.8730	1.9080	4.1428	0.7976	**0.7686**	1.0000	**0.8750**	0.9500
ECGFiveDays	**1.9722**	2.0158	4.2143	0.5215	**0.4913**	1.0000	**1.0000**	0.9944
GunPoint	**1.9787**	1.9942	3.6975	0.4712	**0.4460**	0.9998	**1.0000**	0.9250
	2.1744	2.2187	7.8253	0.6791	**0.6621**	0.9999	**0.8605**	0.7907
	1.2492	**1.2488**	3.5817	0.4563	**0.4060**	0.9999	**0.5000**	0.3846
	1.1791	1.2645	1.3088	0.7262	**0.6397**	0.9998	**0.9726**	0.9589
	3.2741	3.9266	18.9703	0.7470	**0.7071**	1.0000	**0.6667**	0.6667
	0.6685	0.9877	0.6791	0.6182	**0.4493**	0.9999	**0.8258**	0.7753
	2.4413	2.5313	6.0249	0.5602	**0.4834**	1.0000	**0.9685**	0.9213
	0.6979	0.9568	0.7574	0.6186	**0.5116**	0.9998	**0.8421**	0.7782
ProximalPhalanxOutlineCorrect	**0.5895**	1.0056	0.5326	0.6552	**0.4121**	0.9997	**0.8315**	0.8090
SonyAIBORobotSurface1	1.7384	**1.7260**	4.7213	0.4429	**0.4394**	1.0000	0.9919	**1.0000**
SonyAIBORobotSurface2	1.8601	**1.8566**	5.6126	0.4133	**0.3584**	1.0000	0.9796	**0.9949**
	1.2082	1.3628	1.2802	0.6644	**0.5464**	0.9999	0.9695	**0.9797**
	3.1200	3.1436	14.7768	0.3871	**0.3718**	1.0000	**0.9259**	0.7407
	5.4407	5.8238	17.8733	0.6173	**0.5705**	1.0000	**0.9697**	0.7879
TwoLeadECG	**0.9112**	1.0671	1.3517	0.4966	**0.4028**	0.9999	**1.0000**	0.9957
Wafer	**3.0135**	3.1419	8.6207	0.7152	**0.6676**	0.9999	0.9958	**0.9979**
Wine	**0.5052**	0.9301	0.1708	0.7529	**0.3452**	0.9996	**1.0000**	**1.0000**
WormsTwoClass	**5.7723**	7.2023	28.7383	0.4416	**0.4219**	1.0000	**0.8269**	0.7308
Avg.	**2.3132**	**2.5329**	**8.9552**	**0.5733**	**0.4942**	**0.9999**	**0.8924**	**0.8240**

Transformation - example



Conclusions

- A new time series problem with two solutions using the random shapelet forest algorithm
- The proposed algorithms outperform an (admittedly naive) baseline approach in terms of both transformation cost and compactness
- Two simple optimization strategies are introduced and are shown to reduce the computational cost
- We exemplified the usefulness of the proposed algorithms

Ongoing challenges

- Generalize the baseline *k*-nearest neighbor transformation algorithm
- Evaluate the *helpfulness* of the transformations suggested by the algorithm(s) using human evaluators
- Explore alternative feature transformations, e.g., FFT, DWT, autocorrelation
- Explore different application domains

Isak Karlsson, Jonathan Rebane, Panagiotis Papapetrou, and Aristides Gionis, "**Locally and globally explainable time series tweaking**". In Knowledge and Information Systems (KAIS) - to appear

Isak Karlsson, Jonathan Rebane, Panagiotis Papapetrou, and Aristides Gionis, "**Explainable time series tweaking via irreversible and reversible temporal transformations**". In the IEEE International Conference on Data Mining (**ICDM**), 207-216, 2018

Peek into the Black Box

$$y = f(x)$$

- Black box classifier: the form of ***f*** is impossible to interpret
- Even if we can understand the parameters of ***f***, we may still not understand how the classifier uses the data

- Assume we don't know what is in the black box (form of the classifier)
- But…we can test the classifier with data of our choice

$$y^* = f(x^*)$$

y	y*	*A*	*B*	*C*	*D*
1	***1***	1	0	1	*1*
1	***1***	1	0	1	*1*
1	***1***	0	1	1	*0*
1	***1***	0	1	1	*1*
1	***1***	0	0	1	*1*
1	***1***	0	0	1	*0*
1	***1***	1	1	1	*1*
1	***1***	1	1	1	*0*
1	***1***	1	0	0	*0*
1	***1***	1	0	0	*1*
1	***1***	0	1	0	*0*
1	***1***	0	1	0	*0*
0	***0***	1	1	0	*1*
0	***0***	1	1	0	*0*
0	***0***	0	0	0	*0*
0	***0***	0	0	0	*1*

$$\text{fidelity} = \#(y = y^*)/N$$

$$y^* = f(x^*)$$

y	y*	*A*	*B*	*C*	*D*
1	***1***	1	0	1	*1*
1	***1***	1	0	1	*1*
1	***1***	0	1	1	*0*
1	***1***	0	1	1	*1*
1	***1***	0	0	1	*1*
1	***1***	0	0	1	*0*
1	***1***	1	1	1	*1*
1	***1***	1	1	1	*0*
1	***1***	1	0	0	*0*
1	***1***	1	0	0	*1*
1	***1***	0	1	0	*0*
1	***1***	0	1	0	*0*
0	***0***	1	1	0	*1*
0	***0***	1	1	0	*0*
0	***0***	0	0	0	*0*
0	***0***	0	0	0	*1*

select feature D

$$\text{fidelity} = \#(y = y^*)/N = 1$$

$$y^* = f(x^*)$$

y	y*	*A*	*B*	*C*	*D*
1	***1***	1	0	1	*0*
1	***1***	1	0	1	*0*
1	***1***	0	1	1	*1*
1	***1***	0	1	1	*0*
1	***1***	0	0	1	*1*
1	***1***	0	0	1	*1*
1	***1***	1	1	1	*0*
1	***1***	1	1	1	*1*
1	***1***	1	0	0	*0*
1	***1***	1	0	0	*1*
1	***1***	0	1	0	*1*
1	***1***	0	1	0	*0*
0	***0***	1	1	0	*0*
0	***0***	1	1	0	*0*
0	***0***	0	0	0	*1*
0	***0***	0	0	0	*0*

randomization 1

$$\text{fidelity} = \#(y = y^*)/N = 1$$

$$y^* = f(x^*)$$

y	y*	*A*	*B*	*C*	*D*
1	***1***	1	0	1	*1*
1	***1***	1	0	1	*1*
1	***1***	0	1	1	*1*
1	***1***	0	1	1	*0*
1	***1***	0	0	1	*0*
1	***1***	0	0	1	*1*
1	***1***	1	1	1	*0*
1	***1***	1	1	1	*0*
1	***1***	1	0	0	*1*
1	***1***	1	0	0	*1*
1	***1***	0	1	0	*0*
1	***1***	0	1	0	*1*
0	***0***	1	1	0	*0*
0	***0***	1	1	0	*0*
0	***0***	0	0	0	*1*
0	***0***	0	0	0	*0*

randomization 1

$$\text{fidelity} = \#(y = y^*)/N = 1$$

Observe:

- **idelity is stable**
- attribute *D* is neither used nor needed by the classifier

$$y^* = f(x^*)$$

y	y*	*A*	*B*	*C*	*D*
1	***1***	1	0	1	*1*
1	***1***	1	0	1	*1*
1	***1***	0	1	1	*1*
1	***1***	0	1	1	*0*
1	***1***	0	0	1	*0*
1	***1***	0	0	1	*1*
1	***1***	1	1	1	*0*
1	***1***	1	1	1	*0*
1	***1***	1	0	0	*1*
1	***1***	1	0	0	*1*
1	***1***	0	1	0	*0*
1	***1***	0	1	0	*1*
0	***0***	1	1	0	*0*
0	***0***	1	1	0	*0*
0	***0***	0	0	0	*1*
0	***0***	0	0	0	*0*

try attribute *C*…

- ✓ permute C
- ✓ permute C within class

$$y^* = f(x^*)$$

y	y*	*A*	*B*	*C*	*D*
1	***1***	1	0	*0*	1
1	***1***	1	0	*1*	0
1	***1***	0	1	*0*	1
1	***1***	0	1	*1*	0
1	***0***	0	0	*0*	1
1	***0***	0	0	*0*	0
1	***1***	1	1	*1*	1
1	***0***	1	1	*0*	0
1	***1***	1	0	*0*	1
1	***1***	1	0	*1*	0
1	***1***	0	1	*1*	1
1	***1***	0	1	*0*	0
0	***0***	1	1	*0*	1
0	***1***	1	1	*1*	0
0	***1***	0	0	*1*	1
0	***1***	0	0	*1*	0

simple randomization

$$\text{fidelity} = \#(y = y^*)/N = 0.63$$

Observe:

- **fidelity drops!**
- attribute *C* is used and needed by the classifier

$$y^* = f(x^*)$$

y	y*	*A*	*B*	*C*	*D*
1	***1***	1	0	*1*	1
1	***1***	1	0	*1*	0
1	***1***	0	1	*0*	1
1	***1***	0	1	*0*	0
1	***1***	0	0	*1*	1
1	***1***	0	0	*1*	0
1	***0***	1	1	*0*	1
1	***1***	1	1	*1*	0
1	***1***	1	0	*0*	1
1	***1***	1	0	*0*	0
1	***1***	0	1	*1*	1
1	***1***	0	1	*0*	0
0	***0***	1	1	*0*	1
0	***0***	1	1	*0*	0
0	***0***	0	0	*0*	1
0	***0***	0	0	*0*	0

within-class randomization

$$\text{fidelity} = \#(y = y^*)/N = 0.94$$

If an attribute is permuted within-class, and the performance of the classifier does not change, then the attribute is independent of the other attributes [Ojala and Garriga 2010]

y	y*	A	B	C	D
1	1	1	0	1	1
1	1	0	1	1	0
1	1	1	0	1	1
1	1	0	0	1	0
1	1	1	0	1	1
1	1	0	1	1	0
1	1	1	1	1	1
1	1	1	1	1	0
1	0	0	0	0	1
1	0	0	0	0	0
1	1	0	1	0	1
1	1	1	1	0	0
0	1	0	1	0	1
0	0	0	0	0	0
0	0	1	1	0	1
0	1	1	0	0	0

try two independent within-class randomizations

$$\text{fidelity} = \#(y = y^*)/N = 0.75$$

y	y*	*A*	*B*	*C*	*D*
1	***1***	*0*	*1*	1	1
1	***1***	*0*	*1*	1	0
1	***1***	*0*	*1*	1	1
1	***1***	*1*	*0*	1	0
1	***1***	*0*	*0*	1	1
1	***1***	*1*	*0*	1	0
1	***1***	*1*	*1*	1	1
1	***1***	*1*	*1*	1	0
1	***1***	*1*	*0*	0	1
1	***1***	*1*	*0*	0	0
1	***1***	*0*	*1*	0	1
1	***0***	*0*	*0*	0	0
0	***0***	*0*	*0*	0	1
0	***0***	*1*	*1*	0	0
0	***0***	*0*	*0*	0	1
0	***0***	*1*	*1*	0	0

try two joint within-class randomizations

$$\text{fidelity} = \#(y = y^*)/N = 0.94$$

Attributes A and B are both important and they must occur together as a group

Finally…

- ✓ Attribute D is neither used nor needed by the classifier
- ✓ Attribute C is used and needed by the classifier
- ✓ Attribute C is independent of all other attributes
- ✓ Attributes A and B are both important and they must occur together as a group

Grouping of attributes

$$\{\{A, B\}, \{C\}\}$$

$$\text{Class} = (A \oplus B) \vee C$$

Finally…

y	A	B	C	D
1	0	1	1	1
1	0	1	1	1
1	0	1	0	0
1	1	0	0	1
1	0	0	1	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0
1	1	0	0	0
1	1	0	0	1
1	0	1	1	0
1	0	0	0	0
0	0	0	0	1
0	1	1	0	0
0	0	0	0	0
0	1	1	0	1

Grouping of attributes

$$\{\{A, B\}, \{C\}\}$$

$$\text{Class} = (A \oplus B) \vee C$$

The grouping {{*A*, *B*}, {*C*}} means that

A and B randomized together within-class

C is randomized within-class

D is fully randomized

Problem formulations

Optimal k-grouping of attributes

Given a dataset, a classifier, and a constant k, find a grouping of attributes of size k, such that the fidelity is maximized!

$$\{\{A, B\}, \{C\}, \{D\}\}$$

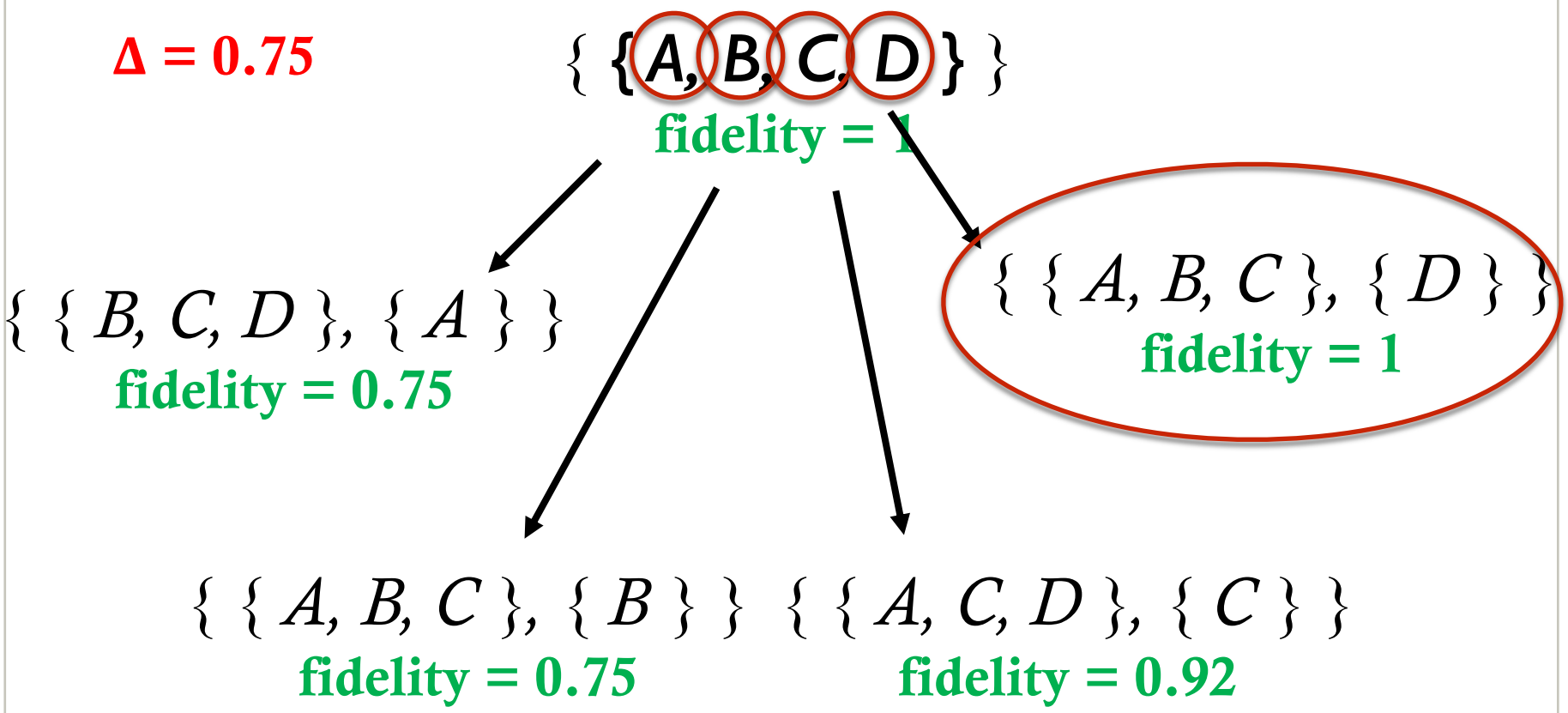
Optimal pruning of singleton unused attributes

$$\{\{A, B\}, \{C\}\}$$

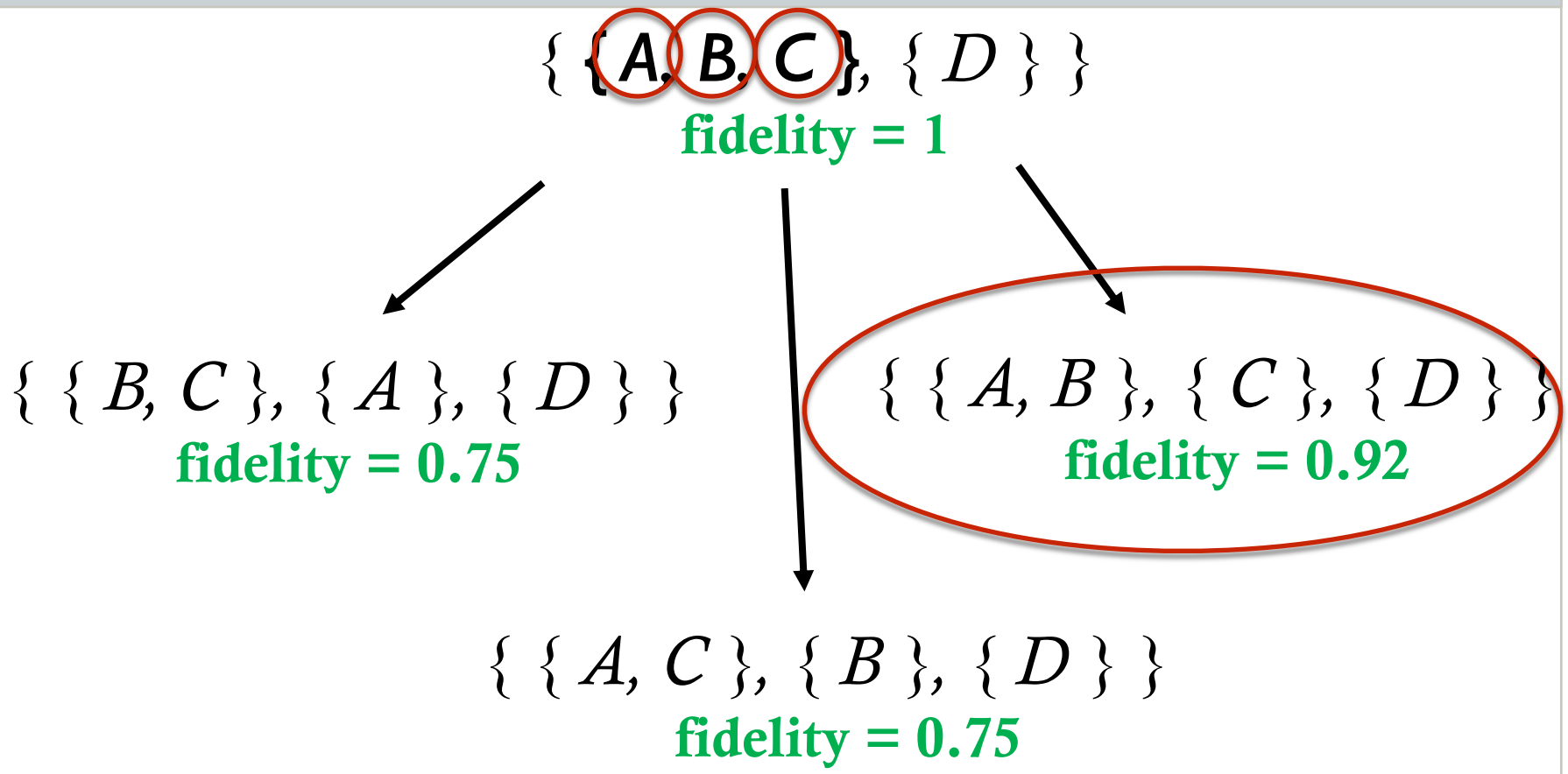
The GoldenEye algorithm

- Finds a grouping of attributes
- **Greedy** iterative **top-down algorithm**
  - define a fidelity baseline value Δ
  - start removing attributes until fidelity drops below Δ → a grouping is defined

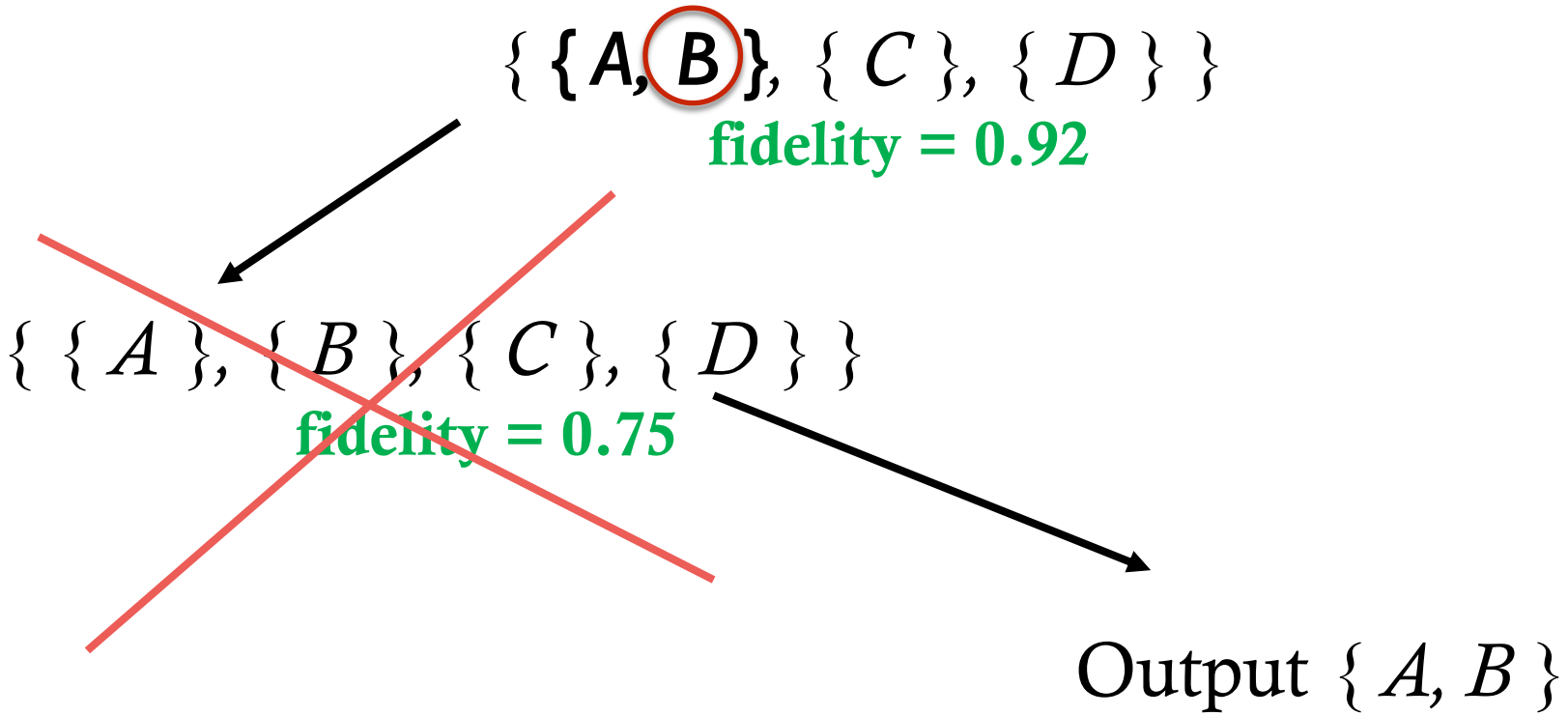
The GoldenEye algorithm



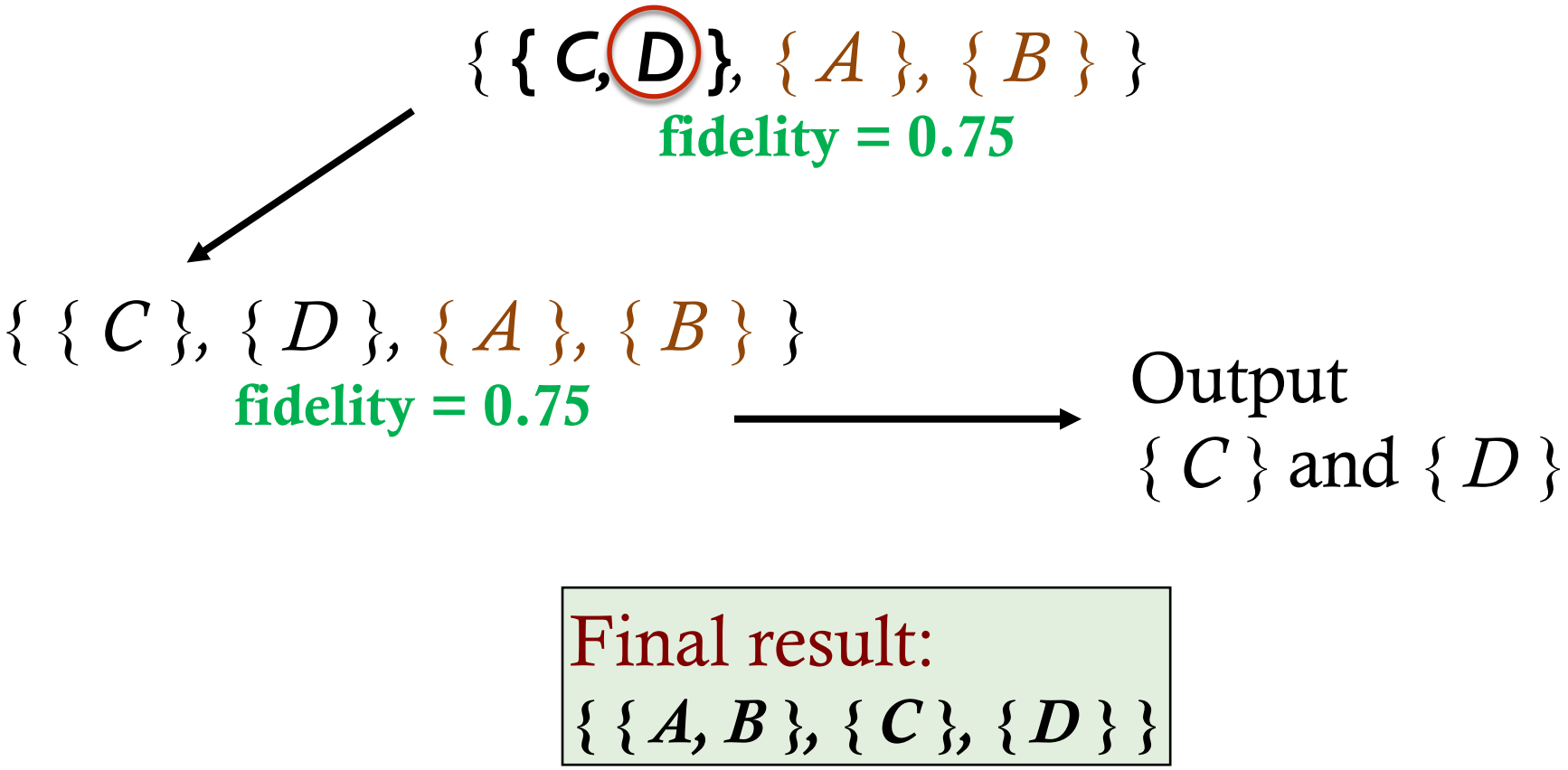
The GoldenEye algorithm



The GoldenEye algorithm



The GoldenEye algorithm



Final result:
{ { A, B }, { C }, { D } }

The GoldenEye algorithm

Finally, unnecessary singletons are pruned

Randomizing *D* fully does not reduce fidelity, hence singleton *D* can be pruned

$$\{\{A, B\}, \{C\}\}$$

Randomizing C fully reduces fidelity too much, hence singleton C *cannot* be pruned

Final output: $\{\{A, B\}, \{C\}\}$

The GoldenEye algorithm

- Finds a grouping of attributes
- Greedy iterative top-down algorithm
- GoldenEye can find the optimal solution, if *monotonicity* holds (breaking groups decreases fidelity)

The GoldenEye algorithm

- Efficient implementation using random sampling and permutations
- Running time:
  - constant wrt the number of data items
  - quadratic wrt the number of attributes

- 26 data sets (synthetic and UCI)
- 15 commonly used classifiers
- Implementation in R (package freely available)*

* https://bitbucket.org/aheneliu/goldeneye/

Limitations and Goldeneye++

- It is not enough to understand the parameters of the classifier
- The structure of data affects classification results
- **Example:** Naive Bayes binary classifier with 2 binary attributes benefits from correlations!

- **Fidelity** can be too crude, for example, when the classes are imbalanced
- **Goldeneye++:** Fidelity replaced with more sensitive measures, e.g., **correlation** between class membership probabilities for original and randomized data
  - ✓ allows for detecting changes in the classifier performance even when output labels do not change, i.e., when only the class probabilities change

Groupings of 9 ADEs

		GoldenEye				GoldenEye++			
dataset	N	groups	attrs	singl	pruned	groups	attrs	singl	pruned
G44.4	19	0	0	19	0	3	7	12	1
G62.0	57						8	49	25
I42.7	43						9	34	25
I95.2	65						9	56	27
T78.2	12						8	4	0
T78.3	12						8	4	0
T78.4	12						12	0	0
T88.6	49						9	40	29
T88.7	49	1	9	40	28	1	9	40	12

Four Datasets (trivial groupings)

- drug-induced headache (G44.4)
- anaphylactic shock (T78.2)
- angioneurotic oedema (T78.3)
- allergy (T78.4)

ADEs: a better look into the black box

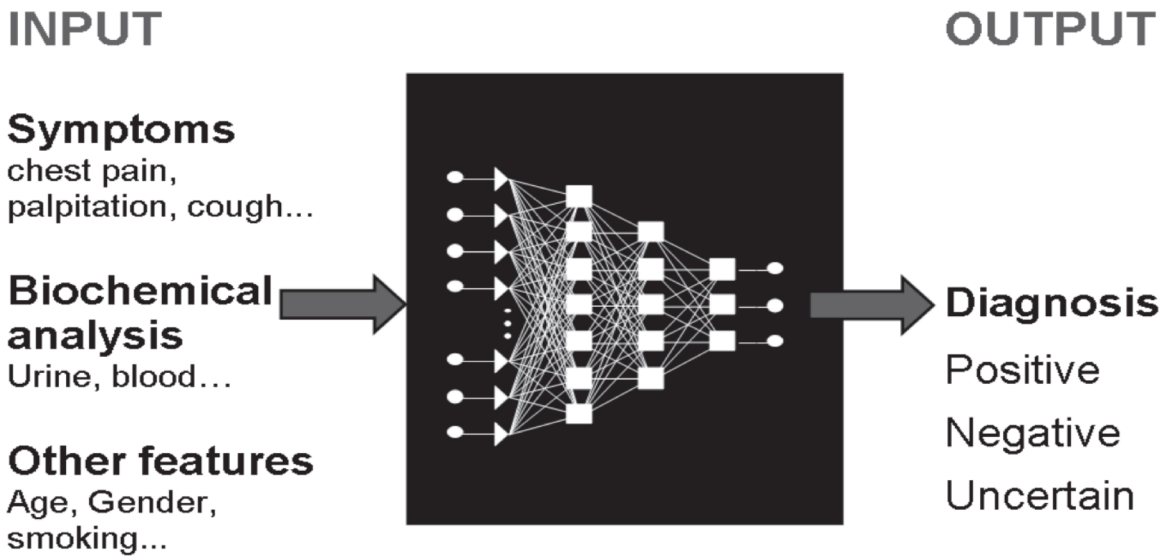
ADE	Patient group	Prior Drugs	Drugs treating
I42.7	Cancer, chemo-therapy	A03FA01, A06AB08, A06AD11, A06AD65, N02AA01	B01AA03, C07AB07, C07AB02, N05BA04
G62.0	Cancer, chemo-therapy	B01AC06, B03BA01, B05BA03, B05BB01, C10AA01, C10AA01, H02AB01, N02BE01	*none*
T88.6	Cancer, chemo-therapy, bacterial infection	A06AD65, N02AA01	B01AB01, B05BA03, B05BB01, B05BB02, C01CA24, R03AC02
T88.7	Cancer, chemo-therapy	A02BC01, A03FA01, H02AB01, H02AB06, N02AA01, N02BE01, N05CF01	R06AA04, R06AX13
I95.2	Older, Angina Pectoris	B05BA03, B05XA06, C01DA02, C03CA01, C10AA01, N02BE01, N05CF01	*none*

Goldeneye: Concluding remarks

- A method based on randomization to find out how a classifier exploits the data for decision making
- It is not enough to just to understand the classifier, the structure of the data matters too!
- Groupings are useful for data exploration
- Could be used to understand and improve classifiers

Deep Learning architectures

- A variety of **deep learning architectures** have been developed for the goal of predictive modelling in regards to accurately detecting diagnoses of interest in medical records



ADE prediction (Yes/No)

- **Main task:**

  predict the presence or absence of an ADE in a patient's next visit given EHR data entries from all previous visits

Adverse drug event ICD-10 codes	
D611	Drug-induced aplastic anaemia
D642	Drug-induced secondary sideroblastic anaemia
D695	Secondary thrombocytopenia
E273	Drug-induced adrenocortical insufficiency
G620	Drug-induced polyneuropathy
I952	Drug-induced hypotension
L270	Drug-induced generalized skin eruption
L271	Drug-induced localized skin eruption
M804	Drug-induced osteoporosis with pathological fracture
M814	Drug-induced osteoporosis
O355	Maternal care for (suspected) damage to fetus by drugs
R502	Drug-induced fever
T782	Adverse effects: anaphylactic shock
T783	Adverse effects: angioneurotic edema
T784	Adverse effects: allergy
T801	Vascular complications following infusion, transfusion and therapeutic injection
T886	Drug-induced anaphylactic shock

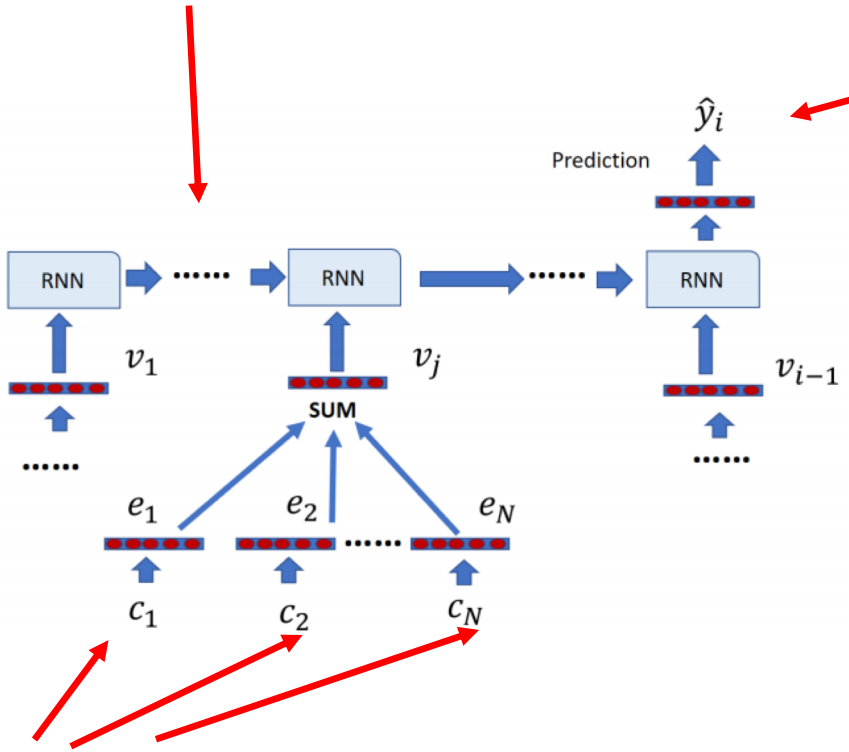
Main goals

- Empirically evaluate which code-level interpretable deep learning architecture provides the best performance for ADE prediction
- Examine which data sources (diagnoses, medications, lab tests) best aid in ADE predictive performance and medical interpretability
- Determine the extent in which code-level attention mechanisms contribute to interpretability for ADE predictions

Methods (Vanilla RNN)

Pass info from one visit to the next within network

Predict ADE yes/no in the future



Input: medical codes *c* for each medical visit *v* to train the network across patients

Limitations of Vanilla RNN

- Standard seq2seq models are normally composed of an encoder-decoder architecture
- **Encoder:** processes the input sequence and summarizes the information into a context vector of fixed length
- This representation is expected to be a good summary of the entire input sequence
- **Decoder:** initialized with the context vector and uses it to generate the transformed output

Limitations of Vanilla RNN

- Structrural assumption:
  - fixed-length context vector
- Why?
  - inability of remembering longer sequences
  - earlier parts of the sequence are forgotten once the entire sequence is processed
- The attention mechanism concept was born to resolve this problem

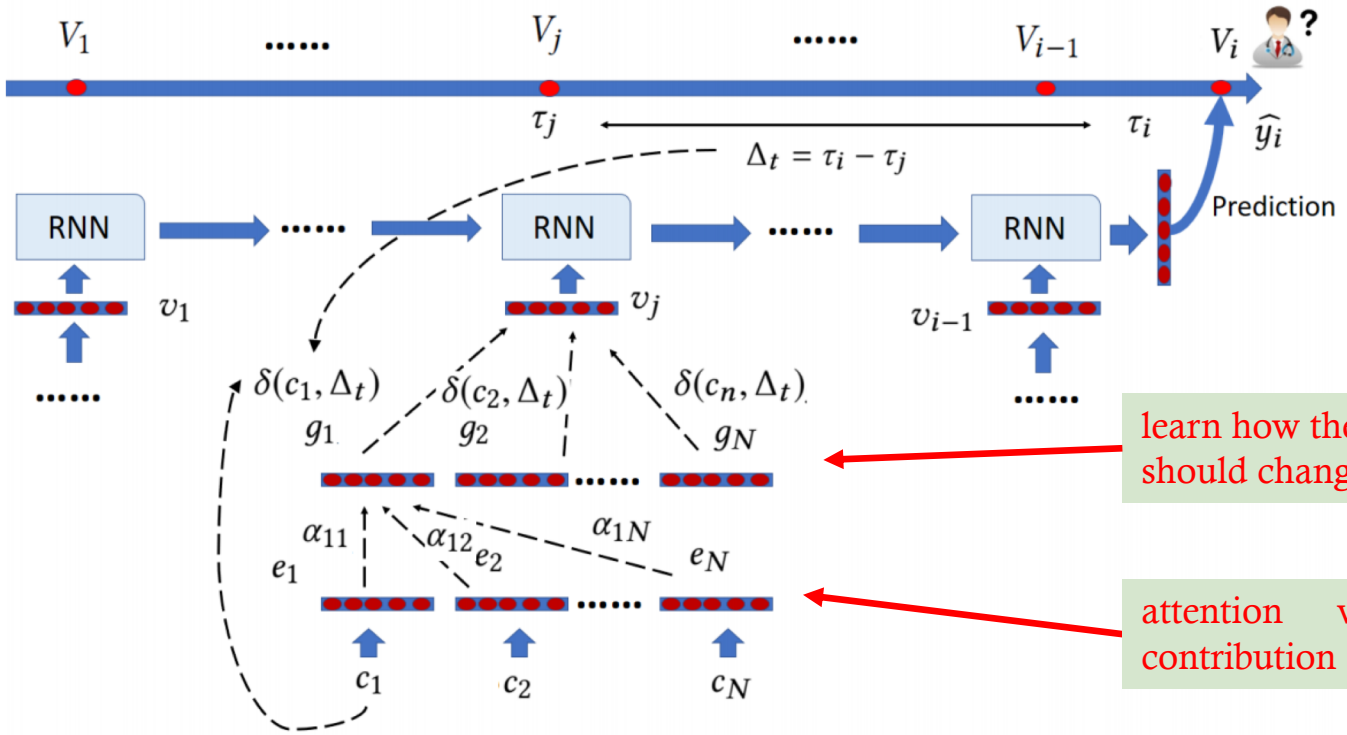
Inability to handle long sequences

- Traditional Seq2Seq:
  - discard all intermediate states of the encoder
  - use only its final states (vector) to initialize the decoder
- This works well for smaller sequences
- As the length of the sequence increases, a single vector becomes a bottleneck and it gets very difficult to summarize long sequences into a single vector
- **Attention mechanism:** keep these intermediate encoder states and utilize all of them in order to construct the context vectors required by the decoder to generate the output sequence

Medical ”attention”

- We may want decoder to focus more on visits 1 and 3, while paying less attention to the remaining visits of the patient
- Solution:
  - Train a feed forward neural network
  - learn to identify relevant encoder states
  - generate a high score for the visits for which attention is to be paid while low score for the visits which are to be ignored

Methods (Timeline, Bai et al.)



learn how the code contributions should change over time

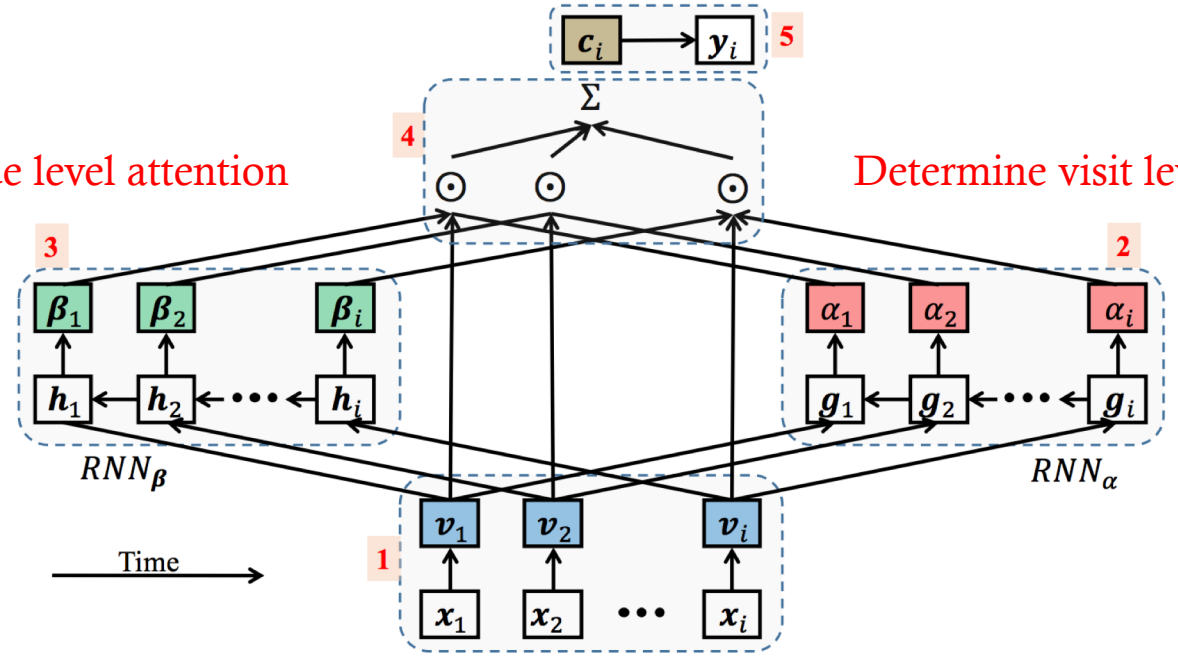
attention values for code contribution

$$\phi(c_x) = \sum_{j=1}^{N} \delta(c_j, \Delta_t)\alpha_{j,x}$$

Methods (RETAIN, Choi et al.)

Determine code level attention

Determine visit level attention



$$\omega(y_i, x_{j,k}) = \alpha_j W(\beta_j \odot W_{emb}[:, k])x_{j,k}$$

Experimental Setup

- RETAIN and Timeline:
  - proven to be competitive state-of-the-art architectures which permit thorough interpretability down to the code-level
  - trained for ADE prediction using an original data source consisting of information for 1,4 million patients obtained from HealthBank at Stockholm University

Experimental Setup

- Non-ADE ICD-10 and ATC codes were reduced to higher level hierarchical categories by selecting only the first three characters
- Such categories correspond to main categories of ICD-10 codes and to therapeutic subgroups in the case of ATC codes
- # of ICD-10 categories: 1692
- # of ATC subgroups: 109
- Visits defined on a monthly basis
- Patients also needed at least three such visits to be included
- Two data sets: including or excluding medication data

Experimental Setup

- Experiments performed by randomly partitioning the data into training, validation, and test set of ratios 0.7, 0.1, and 0.2, respectively
- In order to accommodate for the massive class imbalance stemming from the relative rarity of ADEs, a balanced training set was formed in which the majority class was under-sampled

Results: AUC / F1

Dataset	Area under ROC		Micro F1-Score	
	RETAIN	Timeline	RETAIN	Timeline
Without medication	**0.765**	0.668	**0.789**	0.699
With medication	0.759	0.693	0.775	0.754

RETAIN was determined to be the best performing architecture under the conditions of using diagnoses data

Interpretability of RETAIN

Code	Description	Score
	Visit 1	
L50	Urticaria	0.214
	Visit 2	
R42	Dizziness and giddiness	0.034
A02†	Drugs for acid related disorders	0.049
	Visit 3	
L50	Urticaria	0.239
R06†	Antihistamines for systemic use	0.344
H02†	Corticosteroids for systemic use	0.321
	Visit 4	
L50	Urticaria	0.225
R06†	Antihistamines for systemic use	0.322
C01†	Cardiac therapy	0.205
H02†	Corticosteroids for systemic use	0.230
	Prediction	
T784	Adverse effects: allergy	0.891

Very high risk from given history

Interpretability of RETAIN

Code	Description	Score
	Visit 1	
Z35	Supervision of high-risk pregnancy	0.189
	Visit 2	
O99	Other maternal diseases classifiable elsewhere but complicating pregnancy, childbirth and the puerperium	0.416
	Visit 3	
O99	Other maternal diseases classifiable elsewhere but complicating pregnancy, childbirth and the puerperium	0.398
Z91	Personal risk factors, not elsewhere classified	0.217
	Visit 4	
Z36	Encounter for antenatal screening of mother	0.0312
	Prediction	
O355	Maternal care for (suspected) damage to fetus by drugs	0.802

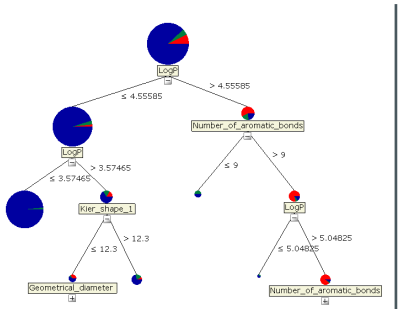
Very high risk from given history

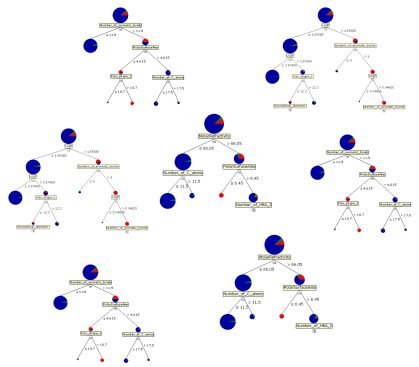
Conclusions/Outlook

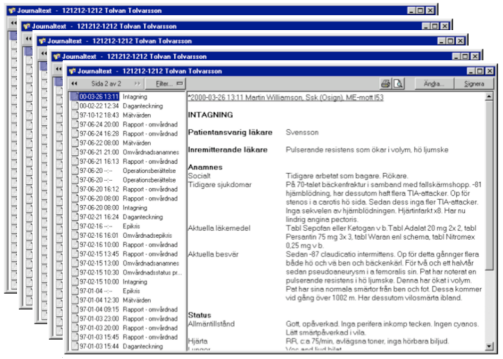
- The RETAIN model utilizing diagnosis data attained the best ADE predictive performance
- Extracted medical interpretations are insightful and logical, but require further medical validation
- Further performance improvements could likely be achieved from additional data sources, hyperparameter tuning, and architecture improvements

Predictive modeling for ADE detection

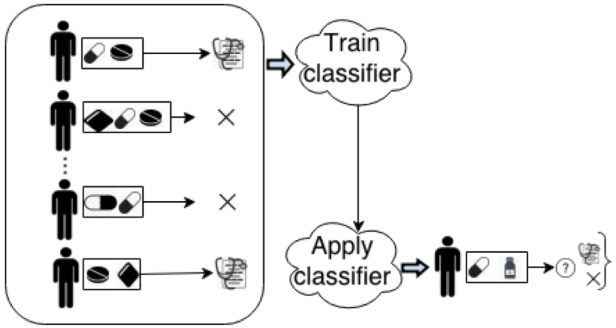
- **Predictive data mining**
  - Interpretable models
  - Ensemble methods
  - Prediction with confidence
  - Information fusion
  - Parallelization
- **Text mining**
  - NLP methods
  - Resource-lean methods





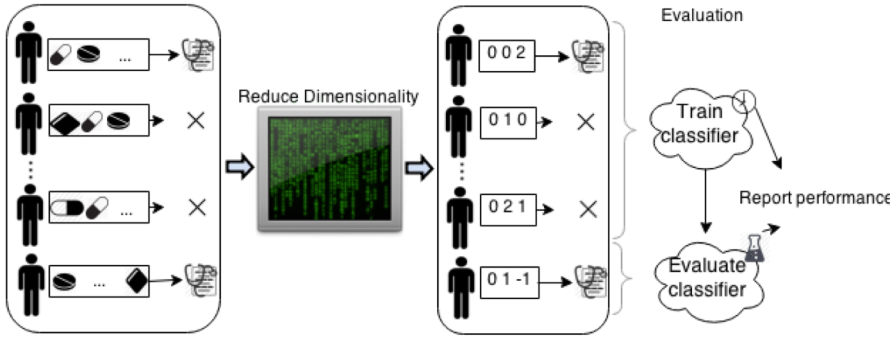


Random forests and random indexing



Random forests were shown capable of screening EHRs that are suspicious to be assigned with an ADE code [Karlsson2013, Karlsson2014]

High dimensionality could be a computational bottleneck



Feasibility of using **random indexing** to **reduce dimensionality** on EHR data

Computational cost of **random forest** is significantly reduced without sacrificing predictive performance

Other tasks predictive tasks

- **Clinical text mining**
  - Discover relationships between drugs and disorders in clinical notes
  - Provide resources for use in larger, hybrid systems
- **Conformal prediction**
  - Develop tools to enable ADE prediction with confidence
- **Parallel data mining**
  - Enable efficient, scalable and parallel solutions for training and evaluation of data and text mining algorithms on big data
  - Utilize cutting-edge parallel platforms such as GPUs and multi-core CPUs to achieve maximum efficiency

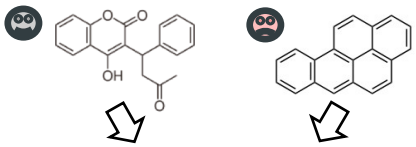
Multiple Data Sources

- Develop techniques and tools to support decision making and discovery of drug effects by analyzing patient records, drug registries, case safety reports and chemical compound data in the form of both structured and unstructured (free text) data
- Contribute with novel approaches to data mining and clinical text mining and develop a platform for large-scale analysis of massive, heterogeneous and continuously growing data sets

Electronic patient records

Karolinska University Hospital (TakeCare)

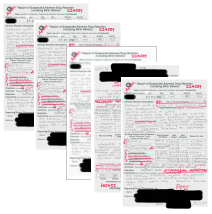
Chemical compound data



Name	Solubility	Frac. of rotatable bonds	Geom diam.	Log P	Zagreb group index	Topol diam.	No. C atoms	No. heavy bonds
methylpentane	good	0.40	3.46	2.44	19	4	6	5
methylcyclohexane	good	0	3.00	2.51	31	4	7	7
nonene	med.	0.75	6.93	3.53	28	8	9	8
hexadiene	good	0.60	4.36	2.14	16	5	6	5
butadiene	good	0.33	2.65	1.36	8	3	4	3
naphthalene	good	0	3.61	2.84	57	5	10	11
acenaphthylene	good	0	3.58	3.32	83	5	12	14
pyrene	poor	0	5.00	4.58	117	7	16	19
dimethylanthracene	poor	0	5.29	4.61	108	7	16	18
hexahydropyrene	med.	0	5.00	3.82	117	7	16	19
triphenylene	poor	0	5.00	5.15	126	7	18	21
benzo(e)pyrene	poor	0	5.29	5.64	152	7	20	24

Pharmaceutical companies

Individual case safety reports



WHO Collaborating Centre for International Drug Monitoring (VigiBase)

Towards actionable models

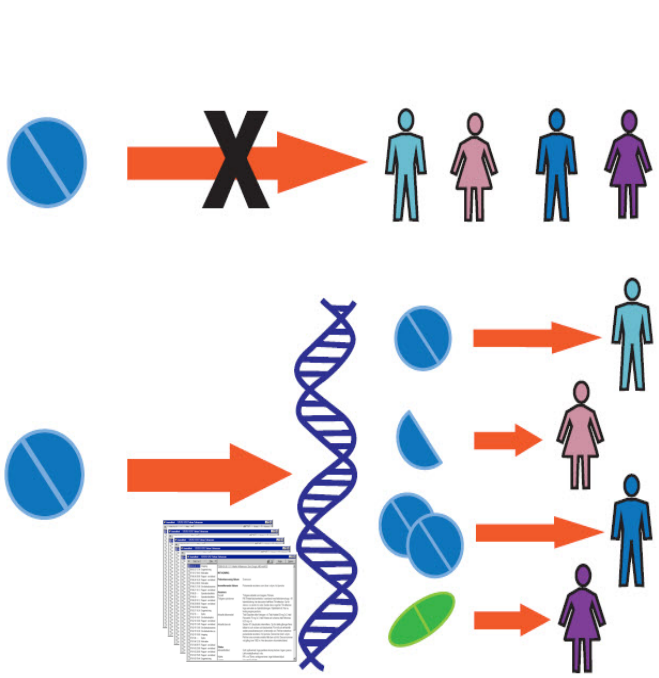
- Trade-offs between interpretability + accuracy
- Ability to understand the predictions + act to prevent undesirable outcomes without compromising predictive performance [Valdes2016]

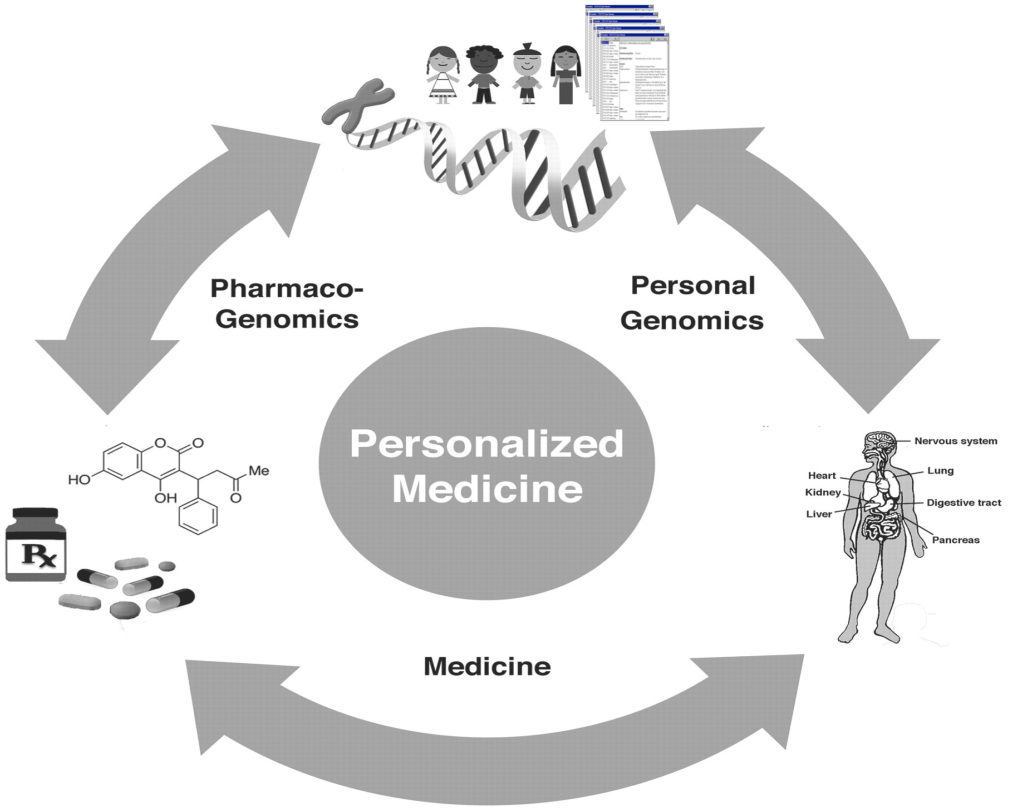
black box classifier

The patient will die from HF in **2 days**!

Now what? Please tell me **why**?

Towards personalized medicine





Closing remarks

- Many methods are out there
- Novel methods and tools are being developed as we speak
- Challenges:
  - EHRs are rich
  - The "temporality" factor is understudied
  - Especially for particular problems, such as ADE detection it is crucial and can make a difference
  - Availability is always an issue
  - Missing values: EHRs tend to be super-sparse! *missing* or *normal* ?

Closing remarks

- Security and privacy issues
- Hard to convince public authorities to make the data more available
- Hard to convince doctors to adopt new "black box" models
- Cloud solutions are in many cases unacceptable
- Many players/systems are used by practitioners
- Need for a unified cross-border database of medical records
- Many initiatives towards this objective: the NIASC* project

* https://www.nordforsk.org/en/programmes-and-projects/projects/nordic-information-for-action-escience-center-niasc

Thanks to…

Isak Karlsson

Jon Rebane

Maria Bampa

Aristides Gionis

Hans E. Persson

Henrik Boström

Hercules Dalianis

We are hiring!

- 1 research assistant in data science (short-term)
  - explainable and ethical machine learning
  - healthcare application domain
  - Deadline: August 31
- 1 associate professor in data science
  - Deadline: October 31

Thank you ☺

Bibliography

- Gunter TD, Terry NP, The Emergence of National Electronic Health Record Architectures in the United States and Australia: Models, Costs, and Questions. J Med Internet Res 2005;7(1):e3
- H.Dalianis, M. Duneld and S. Velupilai, The Stockholm EPR Corpus – Characteristics and Some Initial Findings, Swedish Language Technology Conference, (SLTC), 2009
- Skeppstedt, M., Kvist, M., Dalianis, H., Nilsson, G.H. Automatic recognition of disorders, findings, pharmaceuticals and body structures from clinical text: An annotation and machine learning study. *Journal of Biomedical Informatics*, 2014
- Iyad Batal, Dmitriy Fradkin, James Harrison, Fabian Moerchen, and Milos Hauskrecht. Mining recent temporal patterns for event detection in multivariate time series data. In ACM SIGKDD, 2012
- Henriksson, A., Kvist, M., Hassel, M., Dalianis, H. Exploration of Adverse Drug Reactions in Semantic Vector Space Models of Clinical Text. In Proc. of ICML Workshop on Machine Learning for Clinical Data Analysis, 2012
- Karlsson, I., Zhao, J., Asker, L. & Boström, H. *Predicting adverse drug events by analyzing electronic patient records.* Proc. of the 14th conference on Artificial Intelligence in Medicine (AIME), Murcia, Spain, 2013

Bibliography

- Karlsson, I., and Zhao, J. Dimensionality reduction with random indexing: an application on adverse drug events detection, Submitted to CBMS, 2014
- Zhao, J., Karlsson, I., Asker, L. & Boström, H. Applying methods for signal detection in spontaneous reports to electronic patient records. KDD 2013 Workshop on Data Mining for Health care (DMH), Chicago, USA, 2013
- Asker, L., Boström, H., Karlsson, I., Panagiotis, P., & Zhao, J. Mining candidates for adverse drug interactions in electronic patient records, Submitted to PETRA, 2014
- Henriksson, A., Conway, M., Duneld, M., Chapman, W. Identifying Synonymy between SNOMED Clinical Terms of Varying Length Using Distributional Analysis of Electronic Health Records. In Proc. of Annual Symposium of the American Medical Informatics Association (AMIA), 2013
- Tran T, Luo W, Phung D, Gupta S, Rana S, Kennedy RL, Larkins A, Venkatesh S, "A framework for feature extraction from hospital medical data with applications in risk prediction", BMC Bioinformatics, 2014

Bibliography

- Klimov, D., Shknevsky, A. & Shahar, Y., Exploration of patterns predicting renal damage in diabetes type II patients using a visual temporal analysis laboratory. Journal of the American Medical Informatics Association, 22(2), pp.275-289, 2015
- Moskovitch, R. & Shahar, Y. Classification of multivariate time series via temporal abstraction and time intervals mining. Knowledge and Information Systems, 45(1), pp.35-74, 2015
- Moskovitch, R. & Shahar, Y., Classification-driven temporal discretization of multivariate time series. Data Mining and Knowledge Discovery, 29(4), pp.871-913, 2015
- S. Wu, Y. Chen, Mining Non-ambiguous Temporal Patterns for Inter-val-Based Events, IEEE Transactions on Knowledge and Data Engineering, 19 (6), 2007
- Sacchi, L. et al., Data mining with Temporal Abstractions: learning rules from time series. Data Mining and Knowledge Discovery, 15(2), pp.217-247, 2007
- C. Yi-Cheng, J. Ji-Chiang, P. Wen-Chih, L. Suh-Yin, An Efficient Algorithm for Mining Time Interval-based Patterns in Large Databases, CIKM, 2010