ECG signal classification with liner laws

Péter Pósfay, Antal Jakovác, Kurbucz T. Marcell

Wigner FK

Péter Kovács

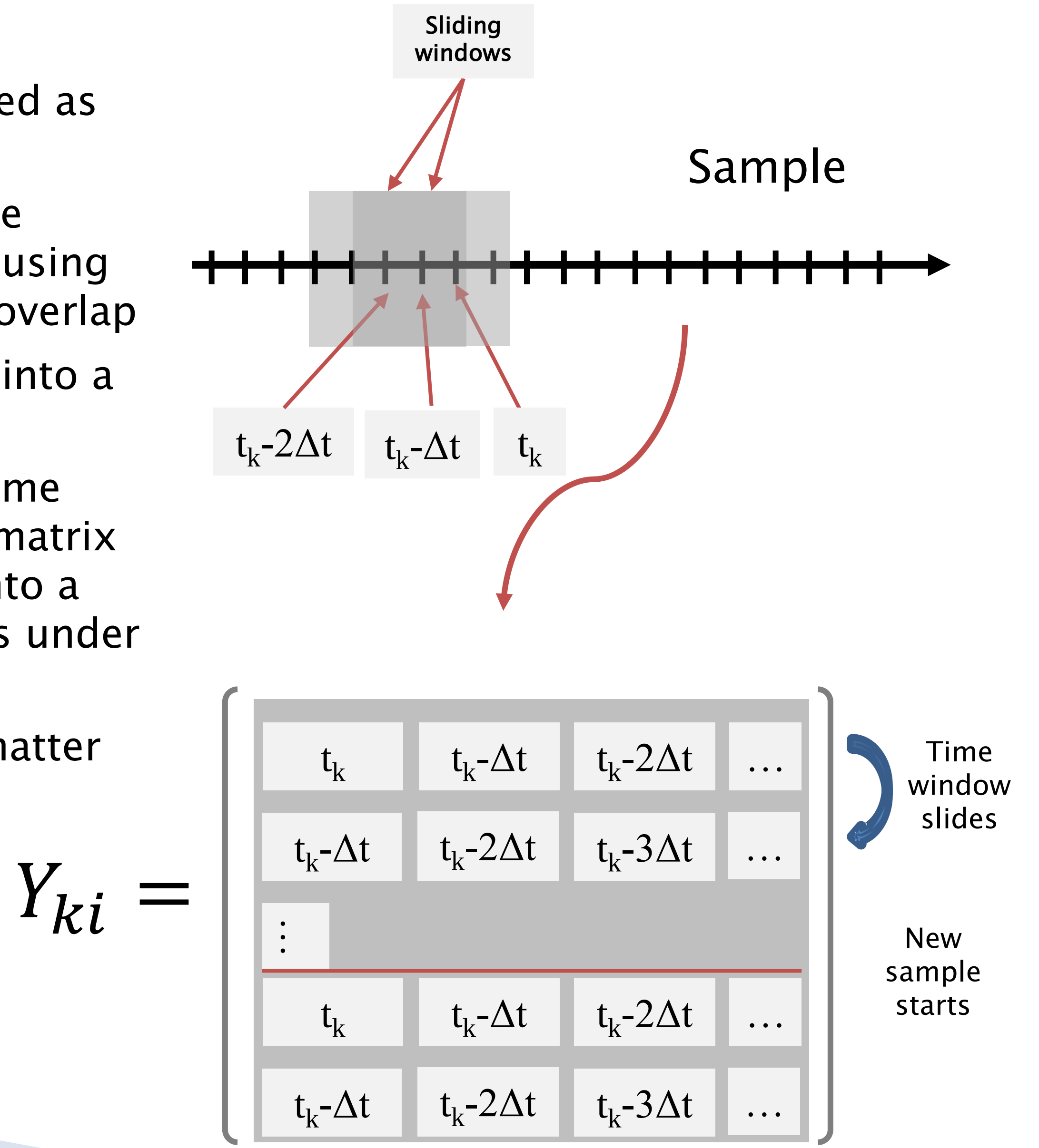
ELTE

Linear laws for time series

Time series representation

- The signal samples are represented as time series -> $y(t)$
- **Time embedding**: The samples are divided into shorter sub samples using a sliding window with maximum overlap
- The “sub samples” are organized into a **learning set matrix.**
- For multiple samples of shorter time series we create the learning set matrix individually and later join them into a large matrix by adding more rows under each other.
- Order of the elements does not matter in this matrix.

$$\boxed{Y_{ki} = Y_i^k = y(t_k - i\Delta t)}$$



Definition of linear laws

Consider a mapping which brings all the sub samples into 0:

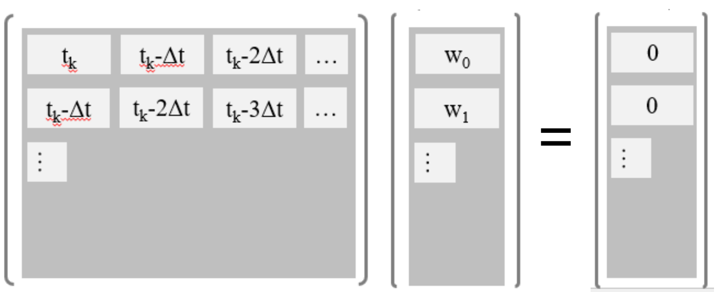
$$\mathcal{F}:\ \mathbb{V}^{n+1} \to \mathbb{R}, \quad \mathcal{F}\left(Y^{(k)}\right) = 0, \quad \forall\, k$$

Linear laws: fix the form of the mapping and consider only linear relations:

$$\boxed{\mathcal{F}(Y^k) = \sum_{i=0}^{n} Y_{ki} w_i \equiv (Yw)_k = 0, \qquad \forall\, k}$$

Linear law

- **Laws**: represent a relation which is true for the whole learning set
- Analogy to physical **conservation laws**: they describe a quantity which is “conserved” for the whole learning set (here it remains 0)



Determining linear laws

Starting from the definition:

$$(Yw)_k = 0 \quad \Longleftrightarrow \quad \|Yw\| = 0$$

In quadratic norm:

$$\|Yw\|^2 = \frac{1}{K}(Yw)^T(Yw) = w^T C w = 0 \qquad C = \frac{1}{K} Y^T Y$$

To avoid the trivial w=0 solution we need to require |w|=1. This leads to a constrained minimalization problem:

$$\chi^2(\lambda) = w^T C w - \lambda w^T w = \min.$$

The solution to this is the eigenvalue equation:

$$\boxed{C w^{(\lambda)} = \lambda w^{(\lambda)}}$$

The n+1 components of the eigenvectors are the linear laws.

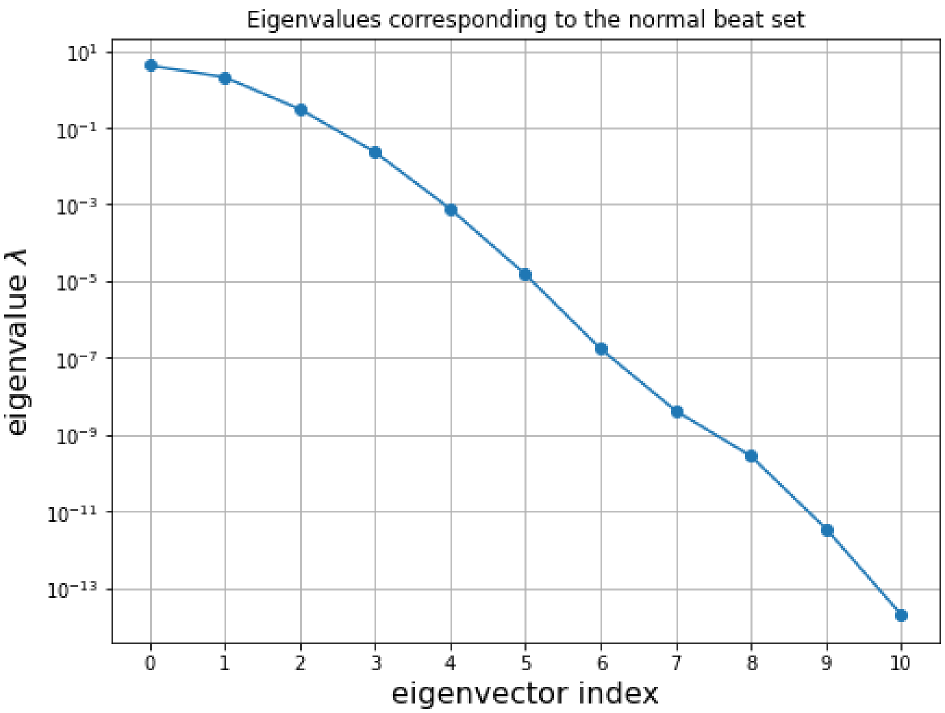
Finding linear laws is an eigenvalue problem

Calculating the linear laws for the ECG signals

- Solving the eigenvalue equation corresponding to the learning set yields linear laws

$$\boxed{Cw^{(\lambda)} = \lambda w^{(\lambda)}}$$

- This yields as many linear laws as many independent eigenvectors.
- Which one of the solutions should be chosen?



Selecting the best solution

The linear laws are true at a given precision. In practice instead of zero they map the learning samples to small numbers:

$$\mathcal{F}(Y^k) = \sum_{i=0}^{n} Y_{ki} w_i \equiv \xi_k$$

Substituting this into the quadratic norm form we get:

$$\|Yw\|^2 = \frac{1}{K} \sum_{k=0}^{K} \xi_k^2 = \langle \xi^2 \rangle$$

So instead of zero as norm we get the variance of the error on the sample. This norm can also be expressed using the eigenvalues:

$$\|Yw\|^2 = \frac{1}{K} (Yw)^T (Yw) = w^T C w \Longleftarrow C w^{(\lambda)} = \lambda w^{(\lambda)}$$

$$\Downarrow$$

The variance is the eigenvalue. $\longrightarrow$

$$\|Yw\|^2 = \lambda w^T w = \boxed{\lambda = \langle \xi^2 \rangle}$$

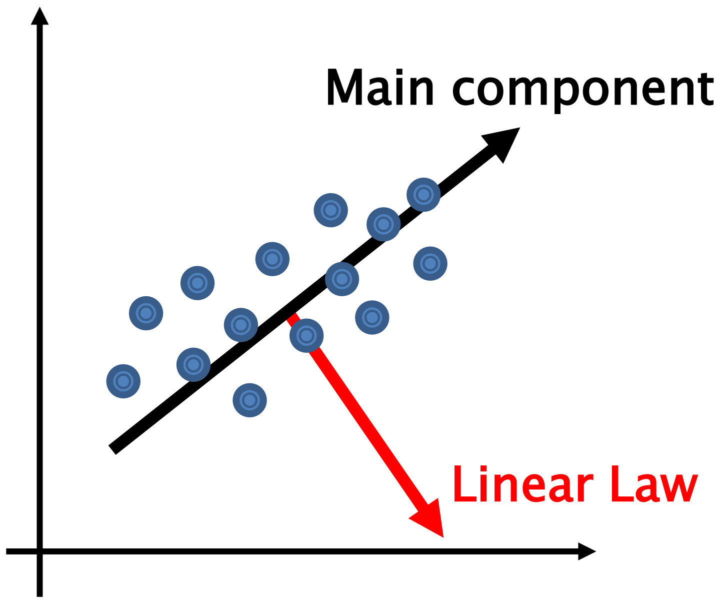
The best linear law is the eigenvector corresponding to the smallest eigenvalue, because this one has the highest precision on average.

Intuition behind linear laws

- **PCA**: The PCA method select the direction in which the data changes the most.

⇕

- **Linear laws**: Select the direction in which the data appears to be constant.
- The linear law is a **direction** in the multidimensional space.
- This direction is **“as orthogonal as possible”** to the data in a sense that it is the normal vector of the hyperplane in which the data can be found. (considering precision)
- This also means that the linear laws are generated for sets that are collecting things which are considered similar for the problem. They correspond to classes.



Linear laws ~ Anti-PCA

Generating Linear features (LLT)

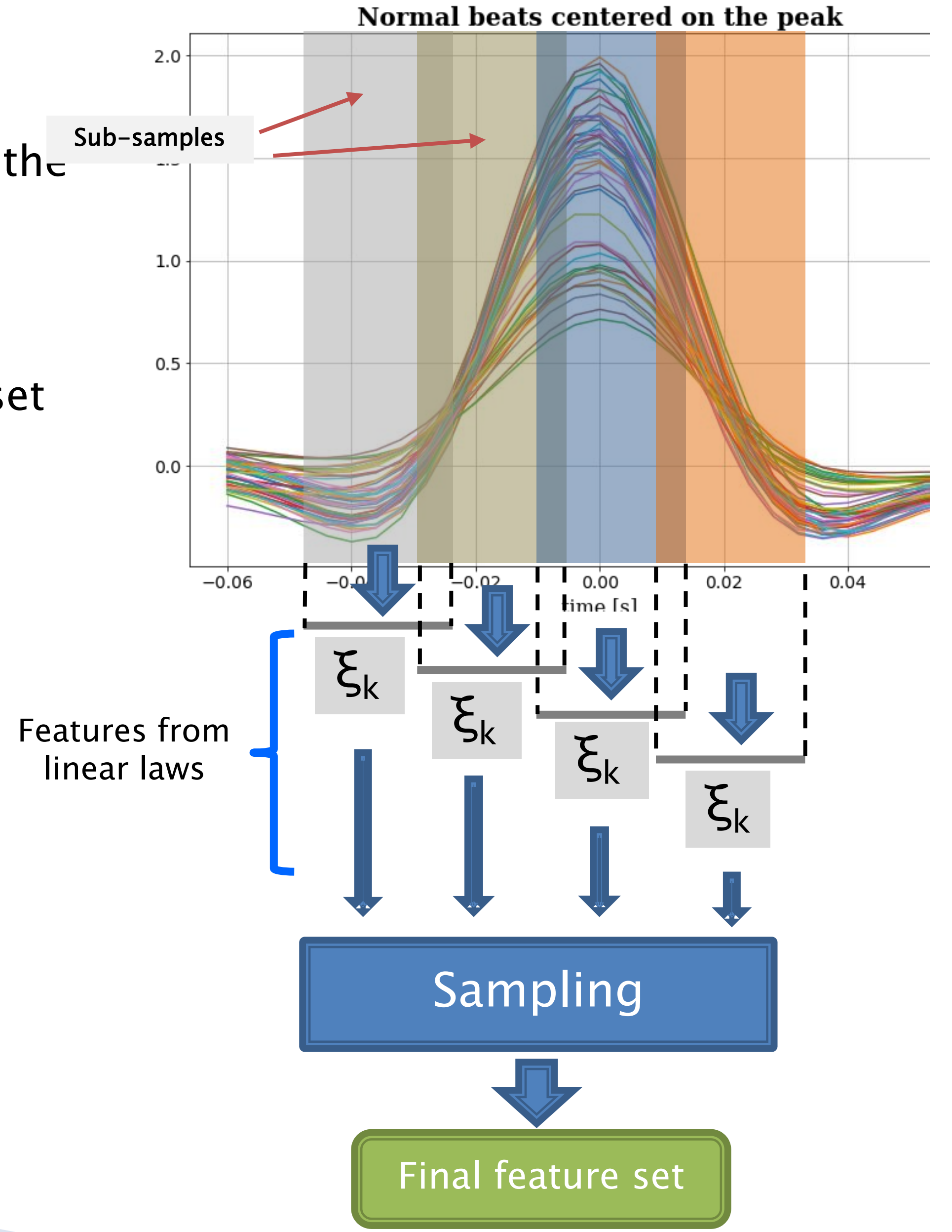
- The sample is **time embedded** with a window length equal to the length of the linear law
- A new Y_{ki} matrix is created
- The linear laws corresponding to the classes in the classification problem set are applied on the Y matrix.
- The results are downsampled than concatenated
- **Feature vector**: features generated by each linear law.

$$\boxed{\mathcal{F}(Y^k) = \sum_{i=0}^{n} Y_{ki} w_i \equiv \xi_k}$$

sample

Linear law

Features

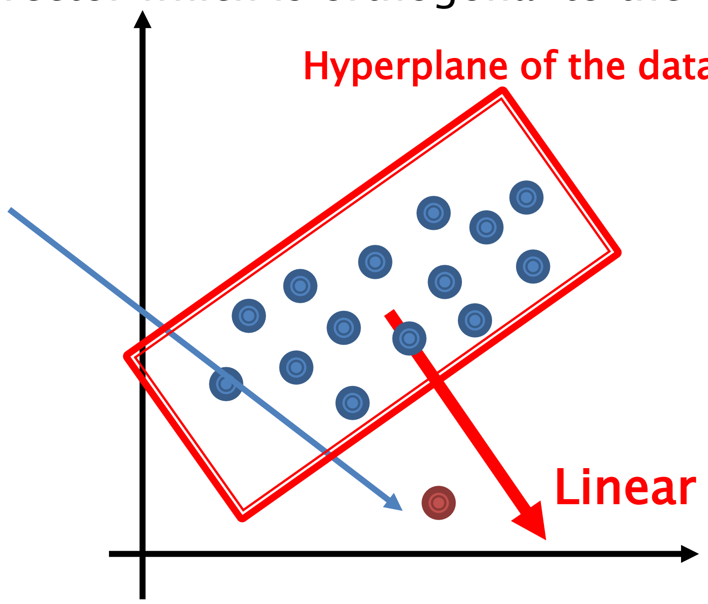


Intuition behind LLT

1. The features measure how **"similar"** a sub-sample to the learning set.
   - **Intuition**: scalar product measures how "large" is a vector in a given direction.
   - Direction = linear law, which represents the learning set
   - Feature = sample • linear law, we expect zero if sample is from LL class
2. LLT is like choosing a **coordinate system** which fit the problem better
   - Spherical problem -> spherical coordinates
     On the surface of the sphere r = const for everything ~ linear law
     variables are eliminated the problem become simpler
   - By finding the linear law we have a vector which is orthogonal to the hypersurface of the data

Different sample: outside of the hyperplane characterized by Linear law

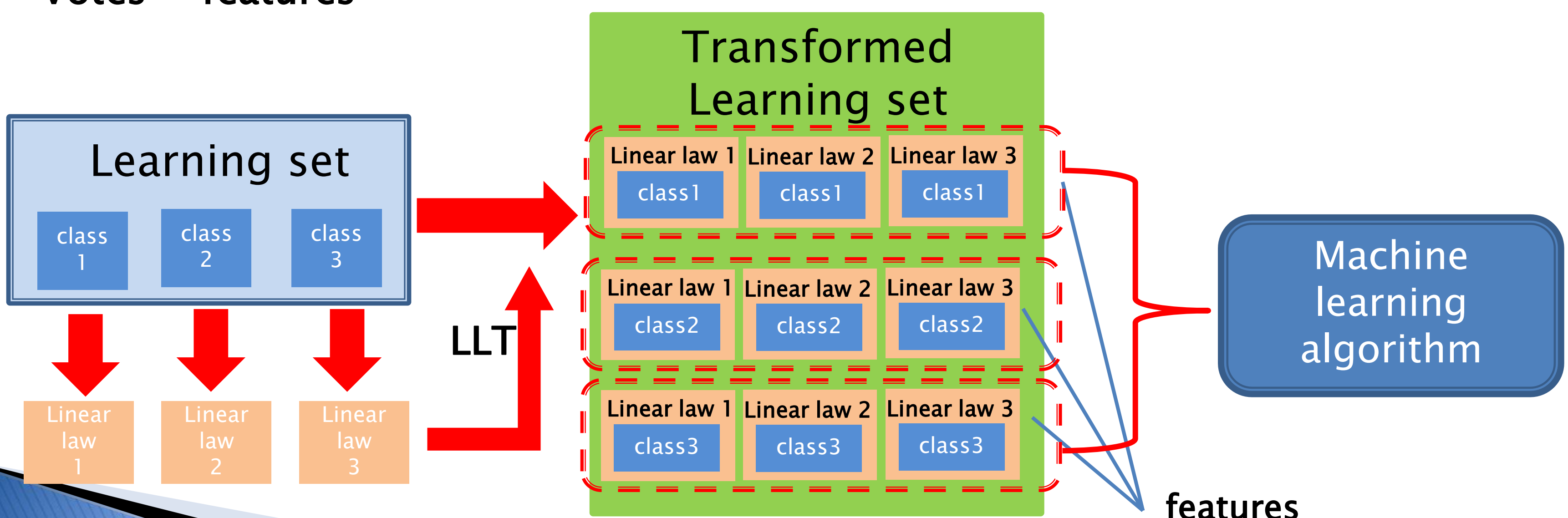
Features have large value



Classification with LLT

LLT aids classification by providing a quantity which is **constant over a given class**

- **Linear laws represent the classes** which they are derived from in an operator form.
- When applied on data Linear laws generate features, which are small for similar data and large for different ones.
- The generated features are **similarity measures** for each class:
  Applying linear laws derived from classes to a sample yields features which measure how similar the sample to the defining class of each linear law.
- Linear laws are "voting" how similar the sample to each class.
- **Votes = features**



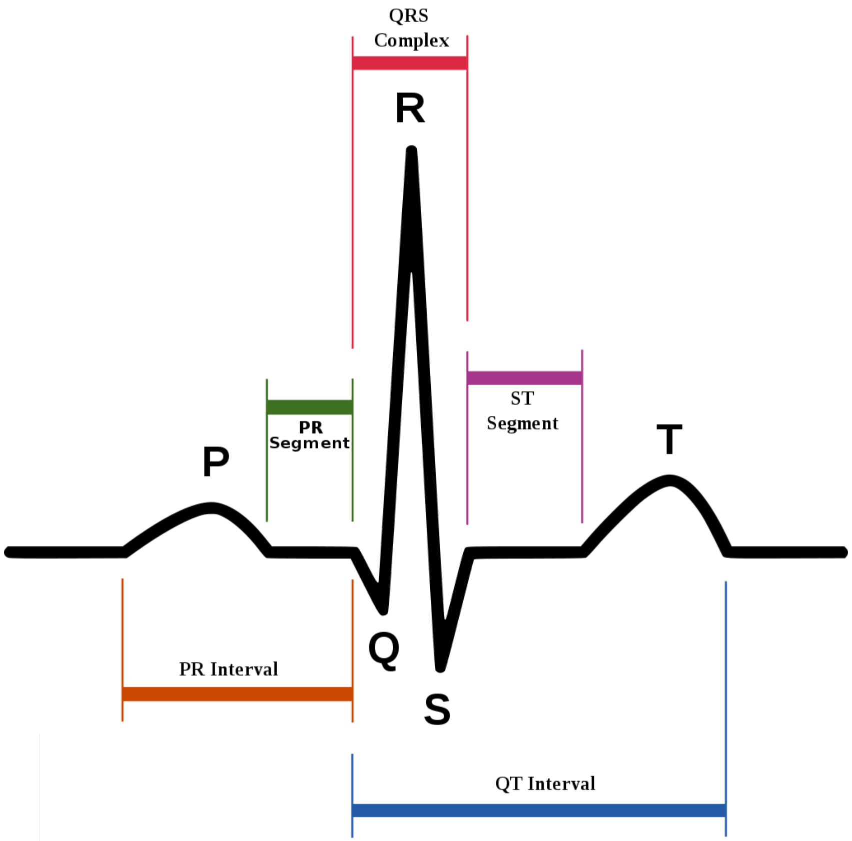
Application of LLT for ECG signal classification

Anatomy of ECG signals

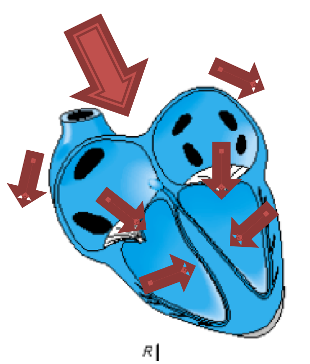
ECG: average (net) polarization of the hearth as function of time.

One cycle:

1. Atrial depolarization -> P
2. Septal depolarization -> Q
3. Left ventricle is behind in time, so the left and right can not cancel each other properly -> R, S
4. Repolarization of the ventricles -> T

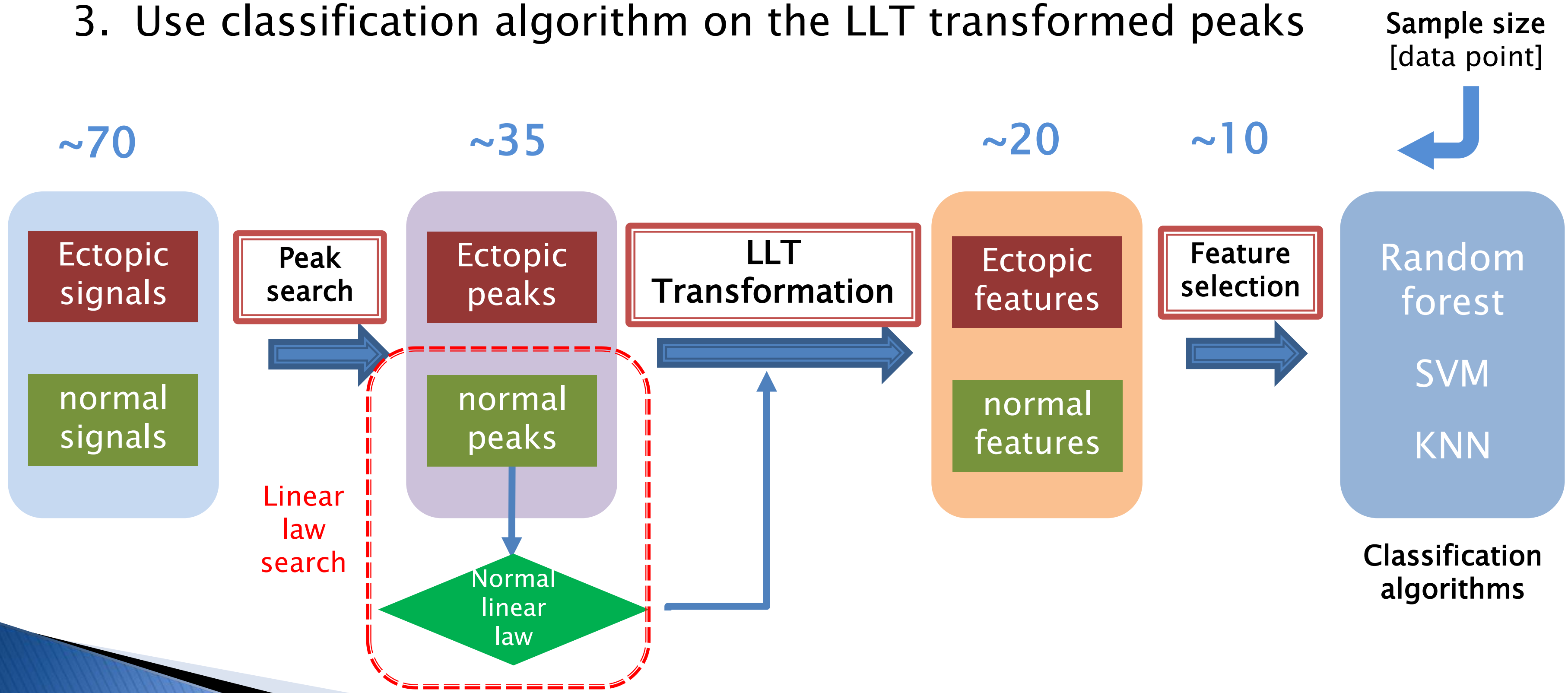


The polarization wave

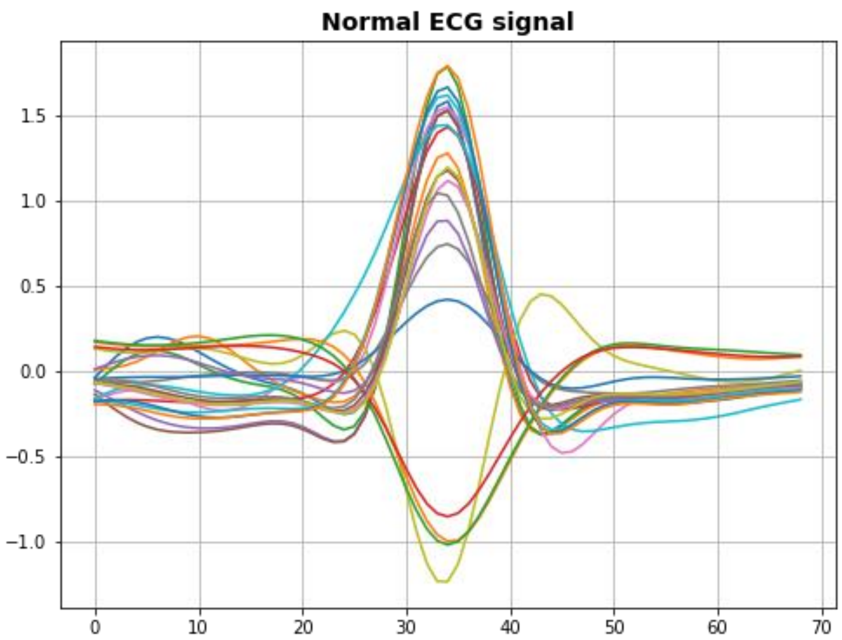


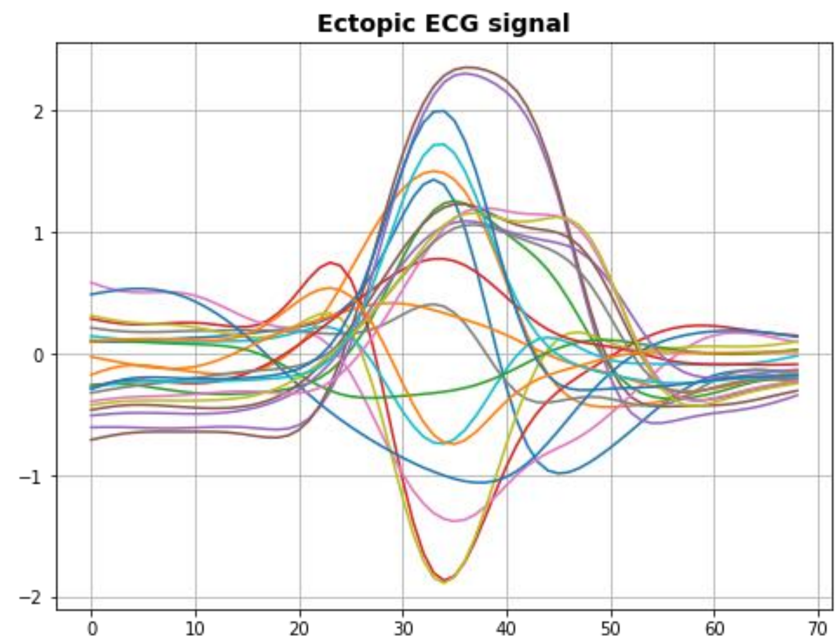
Categorization strategy

1. Focus on the QRS complex in healthy signals – cut the peaks
2. Use the Linear law for healthy QRS complexes for LLT for both ectopic and normal peaks.
   - This is different than it was explained above. It works because we have only 2 classes
3. Use classification algorithm on the LLT transformed peaks



Look at the data





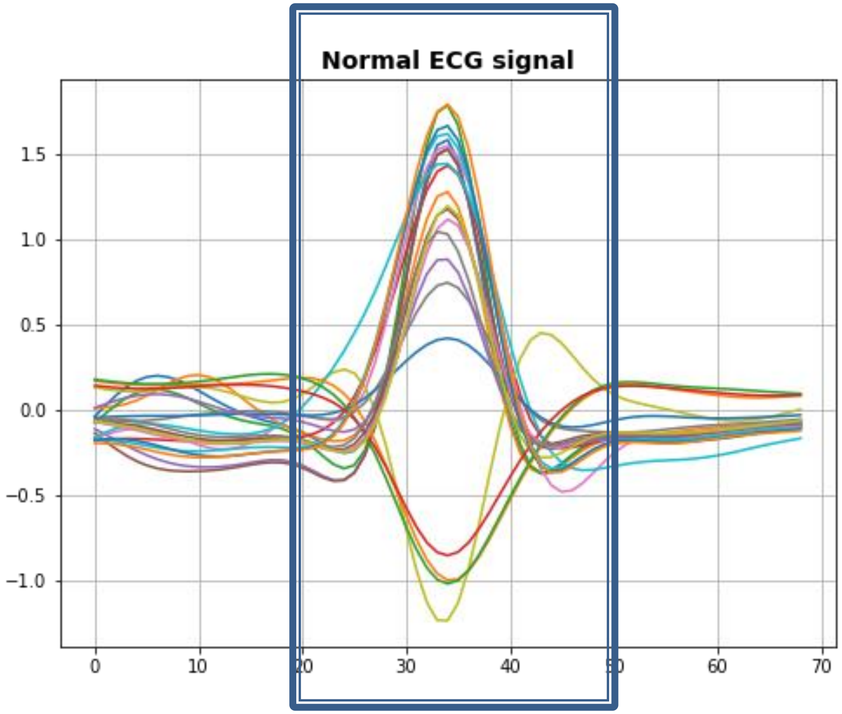
Balanced dataset of around 8000 signals

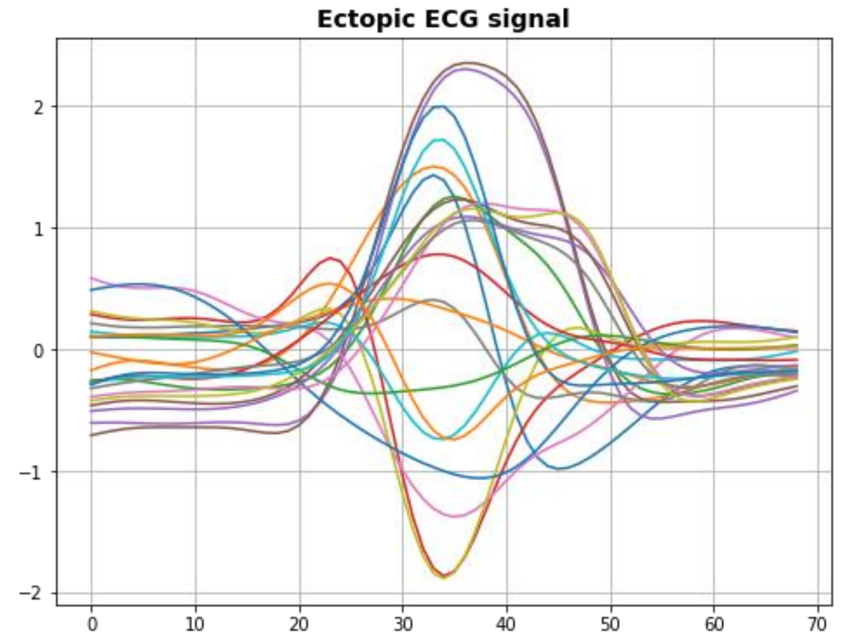
- 30% is used for training
- **70% is used for testing !**

YES, really !

Database was provided by Péter Kovács
https://numanal.inf.elte.hu/~kovi/

Look at the data





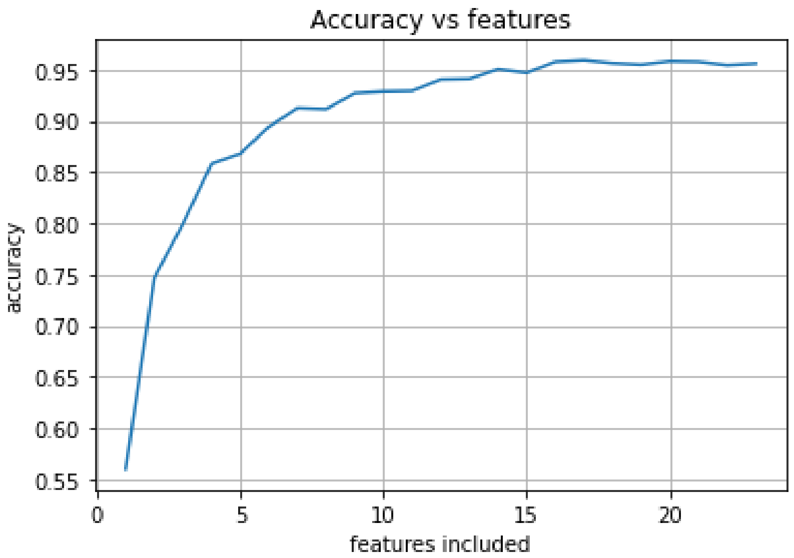
- 34 point long window were used to cut the peaks
- Meta parameter: fitted on healthy signals
- Peaks are standardized:
    - Mean = 0
    - Max = 1

- Because of the irregularity of the signals there are **artifacts** at the peak search
    - No peak found -> instant rejection
    - Multiple peaks found in one sample
- **Take the more strict route**: check performance only on samples where there are peaks and consider multiple peak samples too

Random forest

LLT results in 24 features which is used to fit a random forest

- Number of estimators = 30
- Max depth = 15
- **accuracy: 95.4%** avg
  (normal: 95.2%, ectopic: 95.6 %)
- Including the 10 most important features results in 93% accuracy



To increase accuracy further we need more features (larger forest does not work)

- Use **linear law for ectopic** set and use that to transform the full dataset again and create pairs:
- [Normal LLT(sample), ectopic LLT(sample)] => 48 features
- **Accuracy 96.4 %**
  (normal: 95.3%, ectopic: 97.5 %)

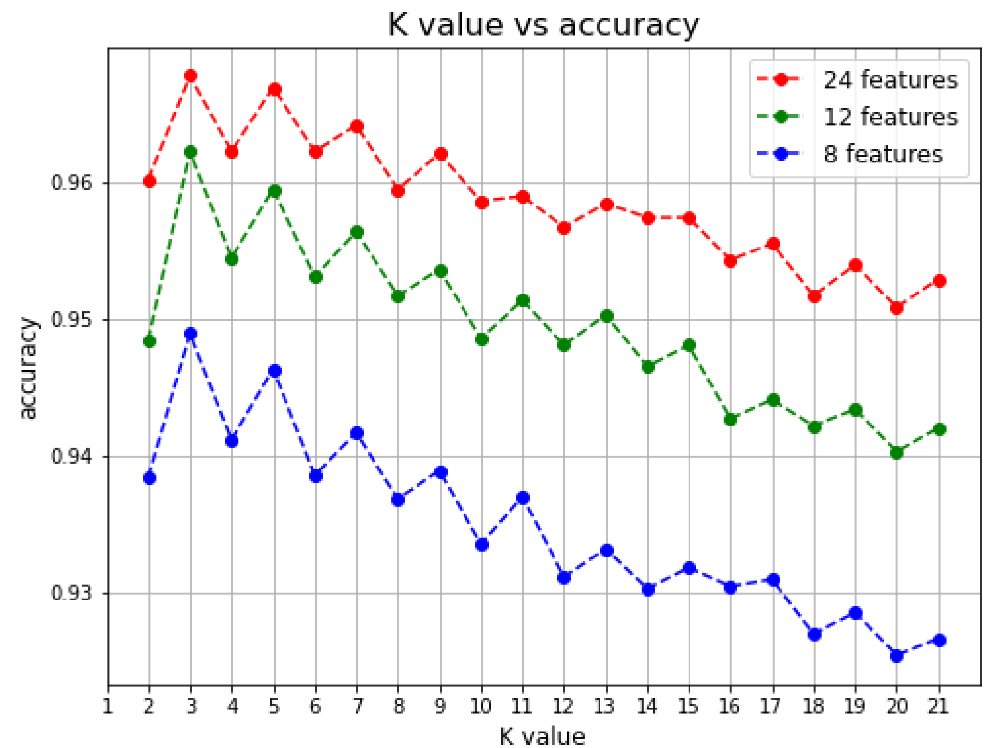
Other Classifiers

K nearest neighbours

- Smaller Ks are more stable
- accuracy: 95-96.7% avg
  (normal: 98%, ectopic: 95.4 %)
- The features can be downsampled to get similarly good results

Support Vector Machine

- Nonlinear Kernel was used
- accuracy: 94% avg
  (normal: 94.4%, ectopic: 93.8 %)



Thank you!

Auxiliary slides

Recursions and linear laws

If a linear law is found w_i are all determined. The law means:

$$\mathcal{F}(Y^k) = \sum_{i=0}^{n} Y_{ki} w_i \equiv (Yw)_k = 0, \xrightarrow[\text{given k}]{\text{For a}} \sum_{i=0}^{n} y(t_k - i\Delta t) w_i = 0$$

This can be rewritten as a recursion:

$$y(t_k) = -\frac{1}{w_0} \sum_{i=1}^{n} y(t_k - i\Delta t) w_i \xrightarrow[t_k = k\,\Delta t]{} y_k = -\frac{1}{w_0} \sum_{i=1}^{n} y_{k-i} w_i = \sum_{i=1}^{n} y_{k-i} \hat{w}_i$$

Since this relation is true for all k this n-length recursion can generate the dataset.

- If data reconstruction is the purpose sampling becomes important because the recursion is not exact, the linear law is true at a given precision
- Searching for linear laws is equivalent to finding recursive representations of data