# Feature Selection
# Interpretable Models

Nataliya Sokolovska

SPLEX, BIM, UPMC

Statistiques pour la classification et fouille de données en génomique

## Outline

- Recall: Problem of sequence labeling
- Recall: Graphical models for output prediction
- Necessity to perform model selection
- Penalization terms including the $L_1$ norm
- Optimization of a graphical model penalized by the $L_1$ penalty term
- What do we get on real-world problems

## Problem of Sequence Labeling: formalizations

Given N independent labelled sequences $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$, where

- $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_{T_i}^{(i)})$ denotes an input sequence
- $\mathbf{y}^{(i)} = (y_1^{(i)}, \ldots, y_{T_i}^{(i)})$ is an output sequence
- $T_i$ is a length of sequences $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$

## Problem of Sequence Labeling: formalizations

Given N independent labelled sequences $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$, where

- $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_{T_i}^{(i)})$ denotes an input sequence
- $\mathbf{y}^{(i)} = (y_1^{(i)}, \ldots, y_{T_i}^{(i)})$ is an output sequence
- $T_i$ is a length of sequences $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$

The aim is to minimize the negated conditional maximum likelihood

$$\ell(\mathcal{D}; \theta) = -\sum_{i=1}^{N} \log p_\theta(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) + \rho_2 \|\theta\|^2$$

with respect to the parameter $\theta$.

## Model of Conditional Random Fields

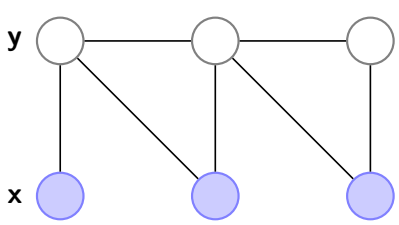
Conditional Random Fields (*Lafferty, McCallum, Pereira, 2001*) are based on the discriminative probabilistic model

$$p_\theta(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) = \frac{1}{Z_\theta(\mathbf{x}^{(i)})} \exp\left\{\sum_{t=1}^{T_i}\sum_{k=1}^{K} \theta_k f_k(y_{t-1}^{(i)}, y_t^{(i)}, x_t^{(i)})\right\},$$

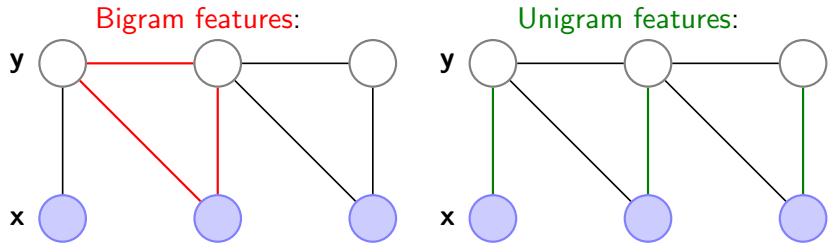
- $\{f_k\}_{1 \le k \le K}$ is an arbitrary set of feature functions
- $\{\theta_k\}_{1 \le k \le K}$ are real-valued parameters, associated with the feature functions
- the normalization factor

$$Z_\theta(\mathbf{x}^{(i)}) = \sum_{(y',y) \in \mathcal{Y}^2} \exp\left\{\sum_{t=1}^{T_i}\sum_{k=1}^{K} \theta_k f_k(y_{t-1}^{(i)}, y_t^{(i)}, x_t^{(i)})\right\}.$$

## Conditional Random Fields: Graphical Model



## Feature Functions




$$\sum_{k=1}^{K} \theta_k f_k(y_{t-1}, y_t, x_t) = \sum_{X \in \mathcal{X}} \left( \sum_{y \in Y,\, x \in X} \mu_{y,x} \mathbb{1}\{y_t = y, x_t = x\} + \sum_{(y',y) \in Y^2,\, x \in X} \lambda_{y',y,x} \mathbb{1}\{y_{t-1} = y', y_t = y, x_t = x\} \right).$$

We get $|X| \cdot |Y| + |X| \cdot |Y|^2$ to estimate.

## Feature Selection

What is Feature Selection for Classification?

- Given a set of predictors (features) and a target (class) variable
- Find minimum set of features that achieves maximum classification performance (for a given set of classifiers and classification performance metrics)

## Why Feature Selection?

- May improve performance of classification algorithm
- Classification algorithm may not scale up to the size of the full feature set either in sample or time
- Allows better understand the domain
- Cheaper to collect a reduced set of predictors
- Safer to collect a reduced set of predictors

## Three Classes of Feature Selection Approaches

1. Filter methods
   - Rely on the general characteristics of data and evaluate features without involving any learning algorithm
2. Wrapper methods
   - Require a predetermined learning algorithm and use its performance as evaluation criterion to select features (heuristic search, hill climbing, genetic algorithms)
3. Embedded models
   - Incorporate variable selection as a part of the training process and feature relevance is obtained analytically from the objective of the learning model

## Filter Methods

- Advantages
  - Fast, scalable, independent of the classifier, models feature dependencies in a multivariate case
- Disadvantages
  - Ignores feature dependencies (univariate case), ignores interaction with the classifier
- Some methods
  - $\chi^2$, $t$-test, Euclidean distance, Information gain, Gain ration, Markov blanket, correlation-based feature selection

## Wrapper Methods

- Deterministic
  - Advantages
    - Simple, interact with the classifier, models feature dependencies, less comp. intensive than randomized methods
  - Disadvantages
    - Risk of overfitting, more prone than randomized methods to getting stuck in a local optimum, classifier dependent
  - Some methods
    - Beam search, sequential forward selection
- Randomized
  - Advantages
    - Less prone to local optima, interacts with the classifier, models feature dependencies
  - Disadvantages
    - Computationally intensive, classifier dependent, risk of overfitting
  - Some methods
    - Genetic algorithm, simulated annealing

## Embedded Methods

- Advantages
  - Interacts with the classifier, better computational complexity than wrapper methods, models feature dependencies
- Disadvantages
  - Classifier dependent
- Some methods
  - Decision trees, LARS, Lasso

## Optimization of the CRF criterion

- The norm $L_2$ is added to avoid overfitting

$$\ell(\mathcal{D};\theta) = \ell(\mathcal{D};\theta) + \frac{\rho_2}{2}\|\theta\|_2^2.$$

- The CRF criterion is convex and differentiable
- First- and second-order numerical methods can be applied directly
  - Conjugate Gradient (Macopt of David MacKay)
  - Quasi-Newton L-BFGS (CRF++ of Tako Kudo)
  - Stochastic Gradient Descent (SGD for CRF of Léon Bottou)

## Some Approaches to Model Selection

- Heuristic methods
  - Eliminate dependencies a posteriori, e.g. those with values close to zero
  - Get rid of rare features a priori
  - Greedy approach to feature selection in CRF of *McCallum, 2003*
- Penalties including the $L_1$ norm
  - Applying the $L_1$ norm penalty instead of the $L_2$ norm:

$$\ell(\mathcal{D};\theta) = \ell(\mathcal{D};\theta) + \rho_1\|\theta\|_1$$

    Orthant-wise Limited Memory Quasi-Newton, *Galen Andrew, Jianfeng Gao, 2007*
  - Elastic Net: combine the $L_1$ and $L_2$ penalty terms

## Limitations of the $L_1$ Norm Penalty

- *Tibshirani 1996*: performance of $L_1$-penalized criterion (the least-squares) is sometimes dominated by the $L_2$-penalized criterion (e.g., in case of correlated parameters)
- *Zou and Hastie 2005*: in the case of correlated parameters $L_1$ norm tends to select one variable of a group of correlated variables

## Elastic Net

Elastic Net has been proposed by *Zou and Hastie, 2005* for the least squares and for logistic regression criteria.

$$\ell(\mathcal{D};\theta) = \ell(\mathcal{D};\theta) + P_{\rho_1,\rho_2}(\theta),$$

where

$$P_{\rho_1,\rho_2}(\theta) = \frac{1}{2}\rho_2\|\theta\|_2^2 + \rho_1\|\theta\|_1 = \sum_{j=1}^{p}\left(\frac{1}{2}\rho_2\theta_j^2 + \rho_1|\theta_j|\right),$$

where $p$ is the number of parameters in the model.

The criterion is not differentiable in zero.
Solution (*J. Friedman, T. Hastie, H. Höfling, R. Tibshirani, 2007*):
Minimize over one parameter at a time, keeping all others fixed.

## Analytical Solution in One-Dimensional Case

The quadratic approximation of the function $\ell(\mathcal{D};\theta)$ using Taylor series is

$$\ell(\mathcal{D};\theta) \approx \ell(\mathcal{D};\tilde{\theta}) + \frac{\partial\ell(\mathcal{D};\tilde{\theta})}{\partial\theta}(\theta-\tilde{\theta}) + \frac{1}{2}\frac{\partial^2\ell(\mathcal{D};\tilde{\theta})}{\partial\theta^2}(\theta-\tilde{\theta})^2 + \frac{1}{2}\rho_2\theta^2 + \rho_1|\theta|.$$

The update takes the form

$$\theta = \frac{S\left((\tilde{\theta}\frac{\partial^2\ell(\mathcal{D};\tilde{\theta})}{\partial\theta^2} - \frac{\partial\ell(\mathcal{D};\tilde{\theta})}{\partial\theta}), \rho_1\right)}{\frac{\partial^2\ell(\mathcal{D};\tilde{\theta})}{\partial\theta^2} + \rho_2},$$

where

$$S(a,\rho_1) \equiv \sigma(a)(|a|-\rho_1)_+ = \begin{cases} a-\rho_1, a \geq 0, \rho_1 \leq |a|, \\ a+\rho_1, a \leq 0, \rho_1 \leq |a|, \\ 0, \rho_1 \geq |a|. \end{cases}$$

## CRF Criterion and its Gradient

- Negated log-likelihood:

$$\ell(\mathcal{D};\theta) = \sum_{i=1}^{N} \Bigg( \underbrace{\log \sum_{(y',y)\in\mathcal{Y}^2} \exp\left\{ \sum_{t=1}^{T_i} \sum_{k=1}^{K} \theta_k f_k(y_{t-1}^{(i)}, y_t^{(i)}, x_t^{(i)}) \right\}}_{\log Z_\theta(\mathbf{x}^{(i)})} - \sum_{t=1}^{T_i} \sum_{k=1}^{K} \theta_k f_k(y_{t-1}^{(i)}, y_t^{(i)}, x_t^{(i)}) \Bigg)$$

- Partial derivatives of $\log Z_\theta(\mathbf{x}^{(i)})$

$$\frac{\partial \log Z_\theta(\mathbf{x}^{(i)})}{\partial \theta_k} = \sum_{t=1}^{T_i} \sum_{(y',y)\in\mathcal{Y}^2} f_k(y,y',x_t^{(i)}) \underbrace{\frac{\exp \theta_k f_k(y,y',x_t^{(i)})}{\sum_{(y',y)\in\mathcal{Y}^2} \exp \theta_k f_k(y,y',x_t^{(i)})}}_{p_\theta(y_{t-1}=y', y_t=y|\mathbf{x}^{(i)})}$$

## Computation of the Gradient

Partial first derivatives of the CRF criterion

$$\frac{\partial \ell(\theta)}{\partial \theta_k} = \underbrace{\sum_{i=1}^{N} \sum_{t=1}^{T_i} \sum_{(y',y)\in\mathcal{Y}^2} f_k(y,y',x_t^{(i)}) p_\theta(y_{t-1}=y', y_t=y|\mathbf{x}^{(i)})}_{\text{Model expectation of the feature vector}} - \underbrace{\sum_{i=1}^{N} \sum_{t=1}^{T_i} f_k(y_{t-1}^{(i)}, y_t^{(i)}, x_t^{(i)})}_{\text{Empirical average of the feature vector}}$$

The gradient is computed using Dynamic Programming.
Complexity of the Forward-Backward Algorithm for a sequence $\mathbf{x}^{(i)}$ is $O(T_i|Y|^2)$.

## Coordinate-Wise Descent for Elastic Net Penalized CRF

- ✓ Quadratic approximation of the CRF criterion.
- ✓☹ Minimization over one parameter at a time.
- ☹ Hessian matrix needed.

⇓

- Approximate the Hessian matrix.
- Block somehow the variables and perform blockwise descent.

## Hessian Matrix and its Approximation

- Diagonal elements of the Hessian

$$\frac{\partial^2 \ell(\theta)}{\partial \theta_k^2} = \sum_{i=1}^{N} \left\{ \mathrm{E}_{p_\theta(\mathbf{y}|\mathbf{x}^{(i)})} \left( \sum_{t=1}^{T_i} f_k(y_{t-1}, y_t, x_t^{(i)}) \right)^2 - \left( \mathrm{E}_{p_\theta(\mathbf{y}|\mathbf{x}^{(i)})} \sum_{t=1}^{T_i} f_k(y_{t-1}, y_t, x_t^{(i)}) \right)^2 \right\}.$$

- The approximation assumes that, given $\mathbf{x}^{(i)}$, $f_k(y_{t-1}, y_t, x_t^{(i)})$ and $f_k(y_{s-1}, y_s, x_s^{(i)})$ are uncorrelated when $s \neq t$

$$\frac{\partial^2 \ell(D;\theta)}{\partial \theta_k^2} \approx \sum_{i=1}^{N} \sum_{t=1}^{T_i} \left\{ \mathrm{E}_{p_\theta(\mathbf{y}|\mathbf{x}^{(i)})} f_k(y_{t-1}, y_t, x_t^{(i)}) - \left( \mathrm{E}_{p_\theta(\mathbf{y}|\mathbf{x}^{(i)})} f_k(y_{t-1}, y_t, x_t^{(i)}) \right)^2 \right\}.$$

## Hessian Approximation

- The approximation of the diagonal terms is exact, if the feature $f_k$ is observed once in a sequence (typical for the NER application).
- In the NER data set and NetTalk corpus, the values of the off-diagonal terms are small (in comparison to the diagonal ones) and can be neglected.

## Block Approximation

- Coordinate-wise update can not be used even for moderate size applications of CRF.
- We investigate the use of blockwise updating schemes, which update several parameters simultaneously trying to share as much computations as possible.
- It is natural to group to update simultaneously the set of all parameters (unigram and bigram) that correspond to the same value of $x$.

## Blockwise Coordinate Descent

- Block parameters $\mu_{y,x}$ and $\lambda_{y',y,x}$ that correspond to the same $x$
- Forward-Backward over sequences which contain the symbol $x$

**Input:** observations and labels, $\rho_1$ and $\rho_2$
**Output:** $\theta$
Initialize $\theta$
**while** until convergence **do**
  **for** $x \in X$ **do**
    **for** sequences which contain $x$ **do**
      $\{\partial \ell(\mathcal{D};\theta)/\partial \mu_{y,x}\ ;\ \partial^2 \ell(\mathcal{D};\theta)/\partial \mu^2_{y,x}\}_{y\in Y}$
      $\{\partial \ell(\mathcal{D};\theta)/\partial \lambda_{y',y,x}\ ;\ \partial^2 \ell(\mathcal{D};\theta)/\partial \lambda^2_{y',y,x}\}_{(y',y)\in Y^2}$
      Update $\{\mu_{y,x}\}_{y\in Y}$ and $\{\lambda_{y',y,x}\}_{(y',y)\in Y^2}$
    **end for**
  **end for**
**end while**

## Forward-Backward Recursions

- To compute the gradient we use the Forward-Backward recursions.
- Complexity for one sequence $\mathbf{x}^{(i)}$ is $O(T_i|Y|^2)$.

Standard approach: $|Y|^2$ (for one $x$)

$$\begin{cases} \alpha_1(y) = \exp(\mu_{y,x_1} + \lambda_{y_0,y,x_1}), \\ \alpha_{t+1}(y) = \sum_{y'} \alpha_t(y') \exp(\mu_{y,x_{t+1}} + \lambda_{y',y,x_{t+1}}). \end{cases}$$

$$\begin{cases} \beta_{T_1}(y) = 1, \\ \beta_t(y') = \sum_{y} \beta_{t+1}(y) \exp(\mu_{y,x_{t+1}} + \lambda_{y',y,x_{t+1}}). \end{cases}$$

$$Z_\theta(\mathbf{x}^{(i)}) = \sum_y \alpha_{T_i}(y)$$

$$\Downarrow$$

$$p_\theta(y_{t-1} = y', y_t = y, x_t^{(i)}) = \frac{\alpha_{t-1}(y') \exp(\mu_{y,x_t} + \lambda_{y',y,x_t}) \beta_t(y)}{Z_\theta(\mathbf{x}^{(i)})}$$

## Sparse Forward-Backward

If matrices of bigram features are sparse, there are $r(x) \ll |Y|^2$ non-zero values (for one $x$):

$$M_{t+1}(y',y) = \exp(\lambda_{y',y,x_{t+1}}) - 1$$

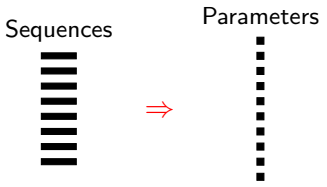
$$\alpha_{t+1}(y) = \exp(\mu_{y,x_{t+1}}) \left( \sum_{y'} \alpha_t(y') + \sum_{y'} \alpha_t(y') M_{t+1}(y',y) \right)$$

$$\beta_t(y') = \sum_y \beta_{t+1}(y) \exp(\mu_{y,x_{t+1}}) + \sum_y M_{t+1}(y',y) \beta_{t+1}(y) \exp(\mu_{y,x_{t+1}})$$

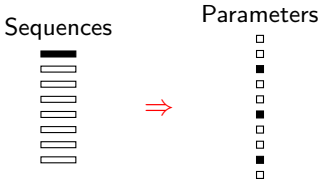
$r(x)$ multiplications instead of $|Y|^2$.

## Brief Comparison of Optimization Approaches

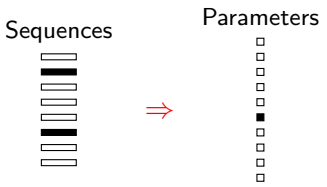
- Orthant-wise Limited Quasi Newton



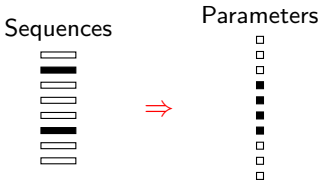

- Stochastic Gradient Descent




- Coordinate-wise Descent




- Sparse Blockwise Descent




## Application: Named Entity Recognition

Predict a sequence of labels given a sequence (or several aligned sequences) of observations.

- Named-Entity Recognition Task (CoNLL 2003). Predict a sequence of labels given 3 aligned sequences of observations.

| Word | Part of Speech | Syntactic Tag | Label |
|---|---|---|---|
| Slovenia | NNP | I-NP | I-LOC |
| and | CC | I-NP | O |
| Poland | NNP | I-NP | I-LOC |
| target | NN | I-NP | O |
| EU | NNP | I-INTJ | I-ORG |
| , | , | O | O |
| NATO | NNP | I-NP | I-ORG |
| membership | NN | I-NP | O |
| . | . | O | O |

Complexity of the model: $|X| \cdot |Y| + |X| \cdot |Y|^2 \approx 1\ 600\ 000$

## Feature functions for Named Entity Recognition

$$\sum_{k=1}^{K} \theta_k f_k(y_{t-1}, y_t, x_t) = \sum_{y \in Y, x \in X} \mu_{y,x} \mathbb{1}\{y_t = y, x_t = x\} + \sum_{(y',y) \in Y^2, x \in X} \lambda_{y',y,x} \mathbb{1}\{y_{t-1} = y', y_t = y, x_t = x\}.$$

- Unigram $\mu_{y,x}$ features

$$f(\text{I-ORG}, \text{NNP}) = \begin{cases} 1, & \text{if } y = \text{I-ORG}, x_{t,\text{POS}} = \text{NNP}, \\ 0, & \text{otherwise.} \end{cases}$$

- Bigram $\lambda_{y',y,x}$ features

$$f(\text{I-LOC}, \text{O}, \text{and}) = \begin{cases} 1, & \text{if } y' = \text{I-LOC}, y = \text{O}, x_{t,\text{word}} = \text{and}, \\ 0, & \text{otherwise.} \end{cases}$$

## Application: Phonetization task (NetTalk Corpus)

Phonetization task: 20 000 English words and their transcriptions

$$X = \{\text{letters}\},\ |X| = 26,$$
$$Y = \{\text{phonemes}\},\ |Y| = 53.$$

Ex. apple - [' æ p l]

We get 75 000 parameters to estimate.

## Feature Functions: NetTalk

- Unigram template

$$f(y = \text{æ}, x_t = a) = \begin{cases} 1, & \text{if } y = \text{æ}, x_t = a, \\ 0, & \text{otherwise.} \end{cases}$$

- Bigram template

$$f(y' = \text{æ}, y = p, x_{t,} = a) = \begin{cases} 1, & \text{if } y' = \text{æ}, y = p, x_{t,} = p, \\ 0, & \text{otherwise.} \end{cases}$$

We get 75 000 parameters to estimate. Do we need all of them?

## Computational Efficiency of Sparse Forward-Backward

Nettalk Data Set

- $|Y| = 53$
- $|X| = 26$

| Algorithm | Time/error(%) |
|---|---|
| SBCD | 70/14.2 |
| OWL-QN | 165/14.2 |
| L-BFGS | 302/14.1 |
| SGD | 17/19.1 |

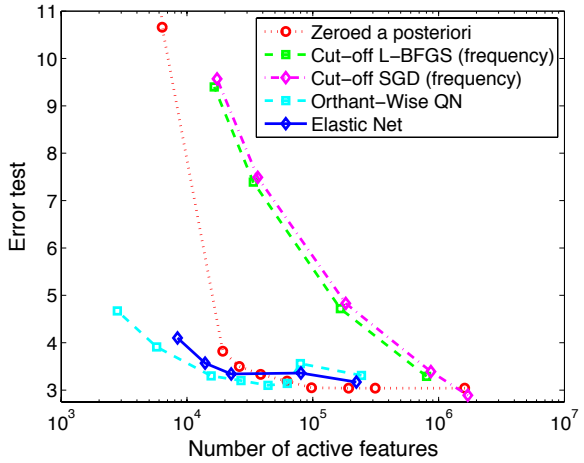
NER Data Set

- $|Y| = 8$
- $|X_1| = 30290$, $|X_2| = 44$, $|X_3| = 18$

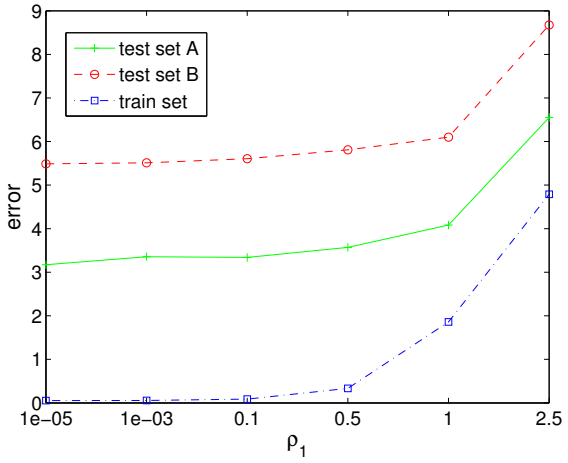
| Algorithm | Time (error $\approx 3\%$) |
|---|---|
| SBCD | 42 |
| OWL-QN | 5 |
| L-BFGS | 25 |
| SGD | 4 |

- Experiments on Intel Pentium 4, 3GHz, 2 G RAM (implementation in C by T. Lavergne, LIMSI, Paris XI)
- The Sparse Forward-Backward is efficient for problems with $|Y|$ large, $|X|$ small.

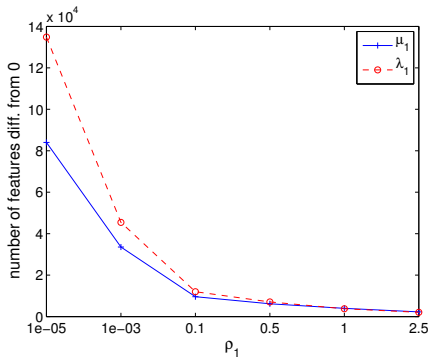
# Performance of Model Selection Methods (NER task)

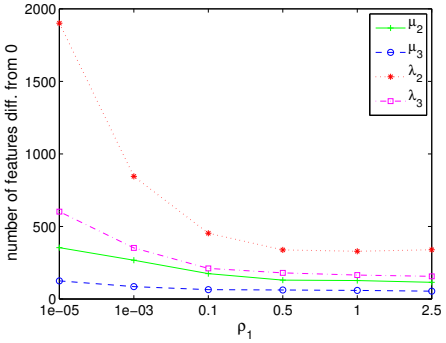



# Results (NER task): Train and Test Errors



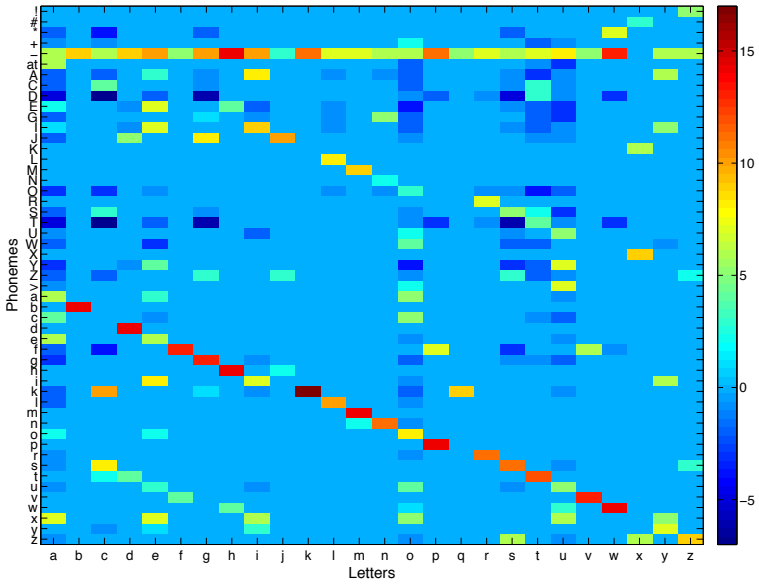

# Results (NER task): Number of Active Features



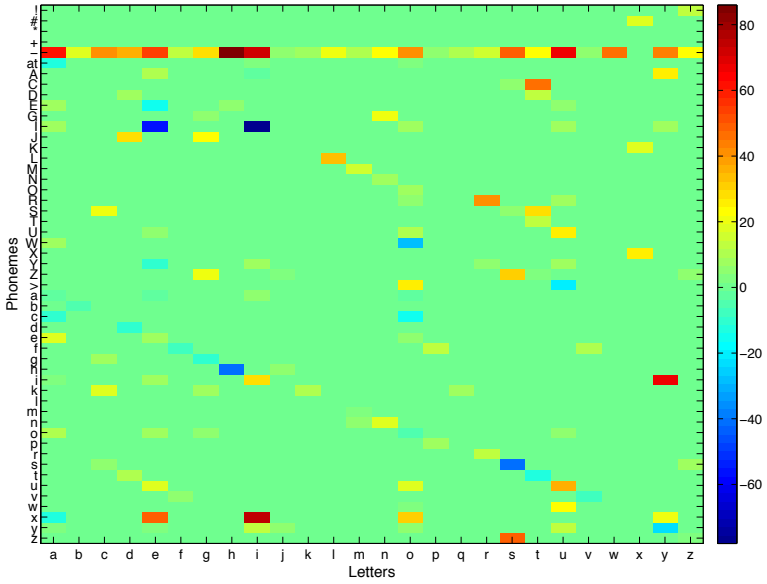




$\rho_1 = 0 \Rightarrow 1\,611\,832$ parameters,
$\rho_1 = 0.1 \Rightarrow 25\,090$ parameters.

## NetTalk: Values of Unigram Parameters




## NetTalk: Values of Bigram Parameters: $\sum_{y'} \lambda_{y',y,x}$




## Stability Issues

- Results are not reproducible!
- Moreover, different runs of the same algorithm would select a different sets of features
- Sometimes these sets of selected features are even not overlapping
- Feature selection is highly unstable
- A simple $t$-test seems to be the most stable feature selection method (see A.-C. Laure et al. *The Influence of Feature Selection Methods on Accuracy, Stability and Interpretability of Molecular Signatures*, 2011)
- Ideas: try to reach some stability on a functional level, and not on the level of separate features

## Some Words on Dimensionality Reduction

Dimensionality reduction is crucial not only for the computational issues but also for data visualization in a two- or three-dimensional space.

- Principal Component Analysis (PCA) is a linear approach to map high-dimensional data into its low-dimensional representation. PCA chooses the coordinates which maximize the variance in the data, and, therefore, the principal components explain most of the variance.
- Kernel PCA was developed to suite for nonlinear data, and, being a kernel method, it maps the data into a higher dimensional space before applying PCA.

## Some Words on Dimensionality Reduction Cont'd

- Isomap is a non-linear method which constructs a neighborhood graph weighted by shortest distances between nearest neighbors. The low-dimensional space is constructed by minimization of pairwise distances between all nodes of the graph.
- Laplacian Eigenmaps is a local approach. It builds a graph where the edges are weighted by values from the Gaussian kernel function, and the weighted distances between the nodes are minimized. The Laplacian eigenmaps incorporate cluster assumption, and enforce natural clusters in the data.

How to take the underlying data structure into consideration?

# Part 2: Interpretable Clinical Models

- **Motivation**
  - **Simple** and **interpretable** models
- **A scoring system**
  - **sparse linear** model
  - based on **simple arithmetic operations**
  - has **few significant digits** (ideally integers)
  - can be **explained by human experts**
  - to be learned purely from data

## Example: the DiaRem (Diabetes Prediction) Score

| Variable | Thresholds | Score |
|---|---|---|
| Age | <40 | 0 |
| | 40–49 | 1 |
| | 50 – 59 | 2 |
| | >60 | 3 |
| Glycated hemoglobin | <6.5 | 0 |
| | 6.5 – 6.9 | 2 |
| | 7 – 8.9 | 4 |
| | > 9 | 6 |
| Insuline | No | 0 |
| | Yes | 10 |
| Other drugs | No | 0 |
| | Yes | 3 |

Classify as Remission if sum of scores $< 7$
Classify as Non-remission if sum of scores $\geq 7$

*C. D. Still et al., Preoperative prediction of type 2 diabetes remission after Roux-en-Y gastric bypass surgery: a retrospective cohort study, 2013*

## The State-of-the-Art

**Medical Scores (widely used)**

- SAPS I, II, and III and APACHE I, II, III to assess intensive care units mortality risks
- $CHADS_2$ to assess the risk of stroke
- TIMI to estimate the risk of death of ischemic events

**None of the existing medical scores was learned directly from data without any human manipulation.**

## State-of-the-Art Cont'd

**Machine Learning point of view:**

- Problems are formulated and solved as linear integer tasks
  - *B. Ustun and C. Rudin. Supersparse linear integer models for optimized medical scoring systems. Machine Learning, 2015.*
- Bayesian optimisation is used to fit a model
  - *S. Ertekin and C. Rudin. A Bayesian approach to learning scoring systems. Big Data, 3(4), 2015.*
- Linear methods (regressions) using gradient-based optimisation, with rounded coefficients
  - *D. Golovin, D. Sculley, H. B. McMahan, and M. Young. Large-scale learning with less ram via randomization. In ICML, 2013.*

## Automated Score Construction

1. Identification of related clinical variables

| age | glycated hemoglobin | insuline | other drugs |
|---|---|---|---|

## Automated Score Construction

1. Identification of related clinical variables

| age | glycated hemoglobin | insuline | other drugs |
|---|---|---|---|

2. Meaningful thresholds for clinical variables

| age | | | | glycated hemoglobin | | | | insuline | | other drugs | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $<40$ | 40–49 | 50 – 59 | $>60$ | $<6.5$ | 6.5 – 6.9 | 7 – 8.9 | $> 9$ | yes | no | yes | no |

## Automated Score Construction

1. Identification of related clinical variables

| age | glycated hemoglobin | insuline | other drugs |
|---|---|---|---|

2. Meaningful thresholds for clinical variables

| age | | | | glycated hemoglobin | | | | insuline | | other drugs | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $<40$ | 40–49 | 50 – 59 | $>60$ | $<6.5$ | 6.5 – 6.9 | 7 – 8.9 | $> 9$ | yes | no | yes | no |

3. Optimization of weights for sub-groups of the variables

| age | | | | glycated hemoglobin | | | | insuline | | other drugs | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $<40$ | 40–49 | 50 – 59 | $>60$ | $<6.5$ | 6.5 – 6.9 | 7 – 8.9 | $> 9$ | yes | no | yes | no |
| 0 | 1 | 2 | 3 | 0 | 2 | 4 | 6 | 10 | 0 | 3 | 0 |

## Automated Score Construction

1. Identification of related clinical variables

| age | glycated hemoglobin | insuline | other drugs |
|---|---|---|---|

2. Meaningful thresholds for clinical variables

| age | | | | glycated hemoglobin | | | | insuline | | other drugs | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $<40$ | 40–49 | 50 – 59 | $>60$ | $<6.5$ | 6.5 – 6.9 | 7 – 8.9 | $> 9$ | yes | no | yes | no |

3. Optimization of weights for sub-groups of the variables

| age | | | | glycated hemoglobin | | | | insuline | | other drugs | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $<40$ | 40–49 | 50 – 59 | $>60$ | $<6.5$ | 6.5 – 6.9 | 7 – 8.9 | $> 9$ | yes | no | yes | no |
| 0 | 1 | 2 | 3 | 0 | 2 | 4 | 6 | 10 | 0 | 3 | 0 |

4. Find an optimal separator between two classes

Classify as `Remission` if sum of scores $< 7$
Classify as `Non-remission` if sum of scores $\geq 7$

## An Approach

- **Simultaneously** do: **binning** (a supervised discretization) and the **score learning** for the bins.
- The **Fused Lasso** (*R. Tibshirani et al., 2015*) shrinks similar variables to each other creating bins, and ordering them.
- In our approach: the **Fused Lasso creates categories and estimates the corresponding weights.**

## The Linear Formulation

We minimise the hinge loss

$$\sum_{i=1}^{N} \ell(y_i, \theta \cdot \bar{x}_i + b) + \lambda \sum_{j=1}^{\bar{d}-1} |\theta_j - \theta_{j+1}|. \tag{1}$$

If we re-write the task as an optimisation problem, we obtain:

$$\min \Big( \sum_{i=1}^{N} \xi_i + \sum_{j=1}^{\bar{d}} \eta_j \Big), \text{ such that} \tag{2}$$

$$\text{for all } i, \; y_i(\theta \cdot \bar{x}_i + b) \geq 1 - \xi_i, \tag{3}$$

$$\text{for all } j, \; -\lambda\eta_j \leq \theta_j - \theta_{j+1} \leq \lambda\eta_j, \tag{4}$$

$$\xi_i \geq 0, \theta_i \in \mathbb{N} \text{ for all } i, \tag{5}$$

and we get $\bar{d} + 1 + N + (\bar{d} - 1)$ variables
$\theta_1, \ldots, \theta_{\bar{d}}, b, \xi_1, \ldots, \xi_N, \eta_1, \ldots, \eta_{\bar{d}-1}$.

## The **Algorithm**: a Linear SVM Penalized by Fused Lasso for Score Learning

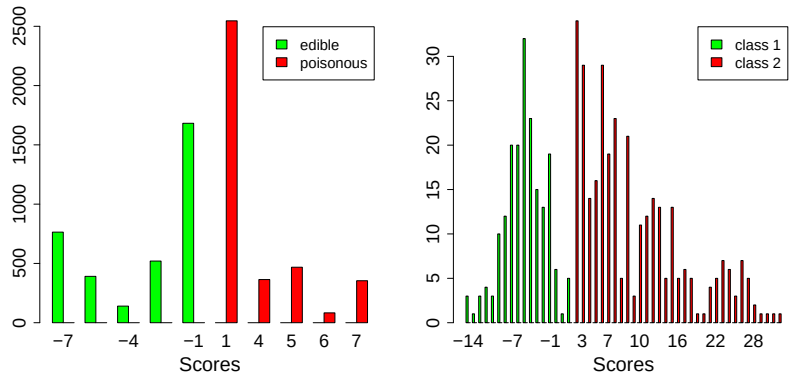
**Input:** a continuous matrix $X$ ($N \times d$), class vector $Y$
**Output:** weights associated with each (observed) value in $X$

**for** $j \in \{1, \ldots, d\}$ **do**
Reformulate $X^j$ as a matrix $\bar{X}$ using one-hot-encoding

Solve discrete L1-SVM with integrity constraints on $\theta$ and fused-lasso penalty using $\bar{X}$ and $Y$

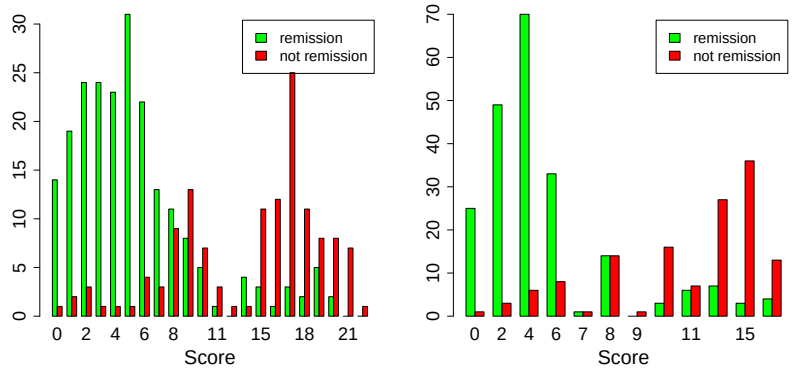
From the resulting $\theta$, build a binning of the values of $X^j$, such that two contiguous values associated with equal weights are in the same bin
**end for**

## The obtained **scores**: Mushrooms and Breast Cancer Data Sets




Distributions of the scores on the Mushrooms data (on the left), and on the Breast cancer data (on the right). On the horizontal axis: all possible scores in data sets. On the vertical axis: the number of observations with the corresponding score. The classes are quite well separated; the optimal separator value is 0.

## Prediction of the Diabetes Remission




Distributions of patients according to the diabetes remission scores. On the left: scores obtained with the DiaRem score, on the right: a distribution based on the learned scoring system.