Structured Output Prediction and Feature Selection

Nataliya Sokolovska

Sorbonne University
Paris, France

Master 2 in Statistics
January, 28, 2020

Outline

Feature Selection and Model Selection

Structured Output Prediction

Stability and Feature Selection

Some Words on Dimensionality Reduction

Feature Selection and Model Selection

Structured Output Prediction

Stability and Feature Selection

Some Words on Dimensionality Reduction

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

Feature selection: why?

- Training time (of classical algorithms) increases exponentially with number of features
- Models have increased risk of overfitting while increasing number of features

Three Classes of Feature Selection Approaches

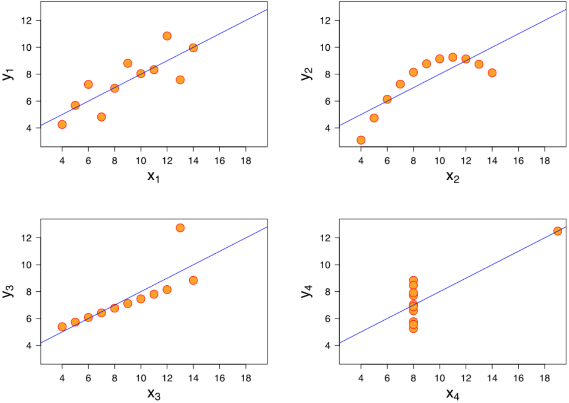
1. Filter methods
   - Rely on the general characteristics of data and evaluate features without involving any learning algorithm
2. Wrapper methods
   - Require a predetermined learning algorithm and use its performance as evaluation criterion to select features (heuristic search, hill climbing, genetic 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
  - χ^2, *t*-test, Euclidean distance, Information gain, Gain ration, Markov blanket, correlation-based feature selection

Filter Methods: correlation

Anscombe's quartet:



Mean (x), sample variance (x), mean (y), sample variance (y), correlation between x and y (0.8), linear regression line are equal.

Filter Methods: Mutual Information

Mutual Information measures the dependence of one variable to another. In case of 2 variables:

- X and Y are independent (no information about Y can be obtained from X and visa versa): mutual information $= 0$.
- If X is a deterministic function of Y: we can determine X from Y and Y from X with mutual information $= 1$.
- We can rank our features (by mutual information), and select the first k features.
- Advantage of MI over statistical tests: MI copes well with the non-linear relationship between observation and target variable.

Filter Methods: Variance Threshold

- Remove features with variation below a certain threshold
- The idea behind: if a feature does not vary much within itself, it has very little predictive power
- Can be a drawback: the method does not consider the relationship of features with the class
- Unsupervised method

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

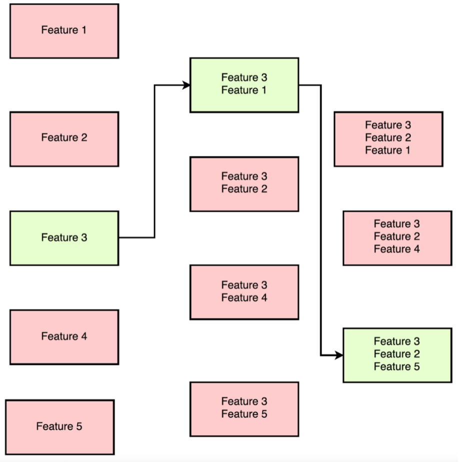
Wrapper methods: Forward Search

Allows to search for the best feature (with respect to models performance), and then add feature one after other:

- 1st iteration: k models are created with individual features, and the best is selected
- 2nd iteration: $k - 1$ models are created with each feature and the previously selected feature
- the procedure is repeated till an optimal subset of m features is selected

Wrapper methods: Forward Search

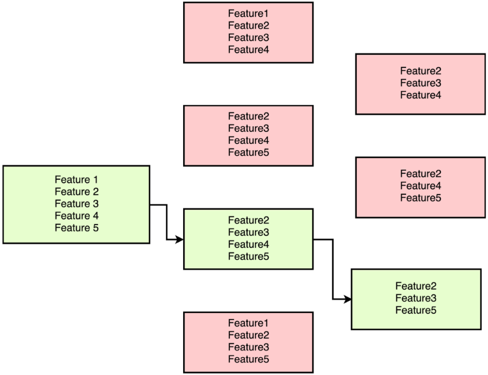
Allows to search for the best feature (with respect to models performance), and then add feature one after other:



(from `https://towardsdatascience.com/why-how-and-when-to-apply-feature-selection-e9c69adfabf2`)

Wrapper methods: recursive feature elimination

Eliminates the worst performing features from a model, and keeps the best features

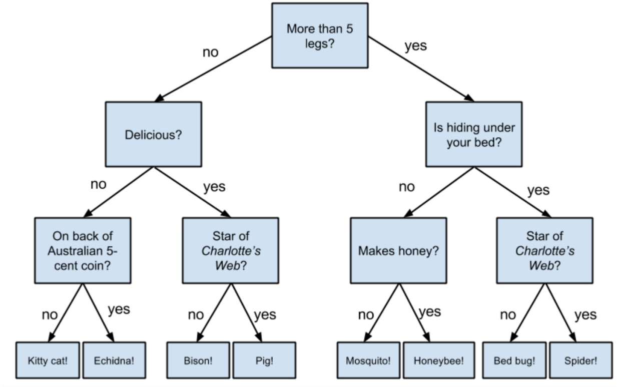


Wrapper methods are computationally expensive!

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

Embedded methods: Tree-based methods



- Calculates feature importance
- Best performing features are close to root

Embedded methods: Lasso

The lasso estimate is defined:

$$(\hat{\alpha}, \hat{\beta}) = \arg\min \sum_{i=1}^{N} (y_i - \alpha - \sum_j \beta_j x_{ij})^2, \tag{1}$$

$$\text{subject to } \sum_j |\beta_j| \leq t, \tag{2}$$

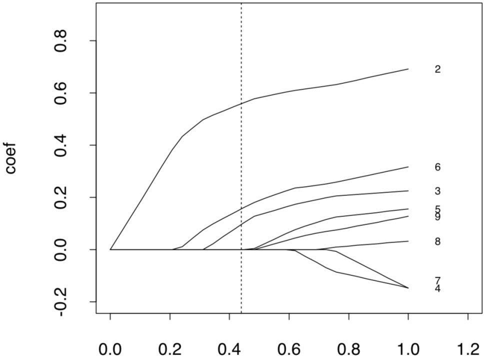
where t is a tuning parameter.
The function with the L_1 penalty term:

$$(\hat{\alpha}, \hat{\beta}) = \arg\min \sum_{i=1}^{N} (y_i - \alpha - \sum_j \beta_j x_{ij})^2 - \lambda \sum_j |\beta_j|, \tag{3}$$

λ is a parameter to fix.

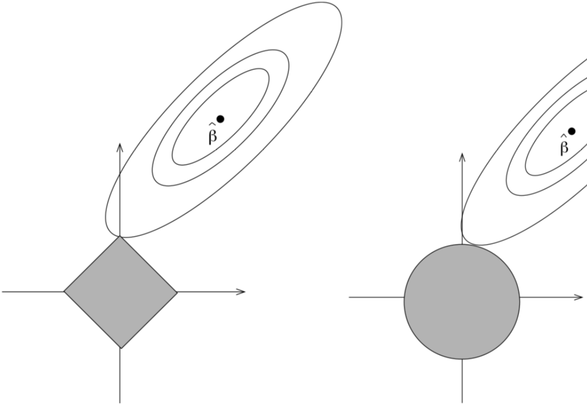
from R. Tibshirani, *Regression shrinkage and selection via the lasso*, 1996

Path



from R. Tibshirani, *Regression shrinkage and selection via the lasso*, 1996

Intuition behind the L_1 penalty term



On the left: the L_1 penalty, on the right: the L_2 penalty.

from R. Tibshirani, *Regression shrinkage and selection via the lasso*, 1996

Logistic regression penalised by the L_1 norm

In case of a binary logistic regression:

$$L(D) = -\sum_{i=1}^{N} \Big(y_i \log f(x_i\theta) + (1-y_i)\log(1-f(x_i\theta)) \Big) - \lambda \sum_j |\theta_j| \tag{4}$$

- S. M. Kim et al., *Logistic LASSO regression for the diagnosis of breast cancer using clinical demographic data and the BI-RADS lexicon for ultrasonography*, Ultrasonography, 2018
- S. Ryali et al., *Sparse logistic regression for whole-brain classification of fMRI data*, NeuroImage, 2010
- $\ldots, \ldots$

Optimisation (convex and differential function)

Some core methods in optimization:

- First-order methods
- Newton's method
- Dual method
- Interior-point methods

Coordinate-wise descent

Start with some initial guess x^0, and repeat for $t = 1, \dots, T$

$$\begin{aligned} x_1^t &\in \arg\min_{x_1} f(x_1, x_2^{t-1}, x_3^{t-1}, \dots, x_p^{t-1}) && (5) \\ x_2^t &\in \arg\min_{x_2} f(x_1^{t-1}, x_2, x_3^{t-1}, \dots, x_p^{t-1}) && (6) \\ &\qquad\qquad\qquad\qquad\qquad\qquad \dots\dots && (7) \\ x_p^t &\in \arg\min_{x_p} f(x_1^{t-1}, x_2^{t-1}, x_3^{t-1}, \dots, x_p) && (8) \end{aligned}$$

- We fix values of all coordinates except for x_i^t.
- Order of cycle through coordinates is arbitrary, can use any permutation of $\{1, 2, \dots, p\}$
- Individual coordinates can be replaced with blocks of coordinates

Coordinate-wise descent

$$f(\theta) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \sum_{j=1}^{p} x_{ij}\theta_j)^2 + \lambda \sum_{j=1}^{p} |\theta_j|, \tag{9}$$

Coordinate-wise descent

$$f(\theta) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \sum_{j=1}^{p} x_{ij}\theta_j)^2 + \lambda \sum_{j=1}^{p} |\theta_j|, \tag{9}$$

With multiple features that are uncorrelated, the lasso solutions are soft-thesholded versions of the individual least squares:

$$f(\tilde{\theta}) = \sum_{i=1}^{N}(y_i - \sum_{k \neq j} x_{ik}\tilde{\theta}_k - x_{ij}\theta_j)^2 + \lambda \sum_{k \neq j} |\tilde{\theta}_j| + \lambda|\theta_j| \tag{10}$$

Lasso: coordinate wise descent

$$f(\tilde{\theta}) = \sum_{i=1}^{N}(y_i - \sum_{k\neq j} x_{ik}\tilde{\theta}_k - x_{ij}\theta_j)^2 + \lambda \sum_{k\neq j} |\tilde{\theta}_j| + \lambda|\theta_j| \tag{11}$$

$$\tilde{\theta}_j(\lambda) = S\Big(\sum_{i=1}^{N} x_{ij}(y_i - \tilde{y}_i^j), \lambda\Big), \tag{12}$$

$$\tilde{y}_i^j = \sum_{k\neq j} x_{ik}\tilde{\theta}_k(\lambda), \text{partial residual on } j\text{th variable}, \tag{13}$$

$$\tilde{\theta}^{\text{lasso}}(\lambda) = S(\tilde{\theta}, \lambda) = \text{sign}(\tilde{\theta})(|\tilde{\theta}| - \lambda) = \tag{14}$$

$$\begin{cases} \tilde{\theta} - \lambda, \ \text{if } \tilde{\theta} > 0, \ \text{and } \lambda < |\tilde{\theta}|, \\ \tilde{\theta} + \lambda, \ \text{if } \tilde{\theta} < 0, \ \text{and } \lambda < |\tilde{\theta}|, \\ 0, if\, \lambda \geq |\tilde{\theta}|. \end{cases} \tag{15}$$

Graphical Lasso

- Given a data matrix $X \in R^{n \times p}$, where rows are i.i.d. observations from $\mathcal{N}(0, \Sigma)$, and Σ is unknown
- The goal is to estimate Σ
- $\Sigma_{ij}^{-1} = 0$: X_i and X_j conditionally independent given X_k, $k \neq i, j$
- If p is large, and the above is true for many i, j, then Σ^{-1} is sparse
- Graphical lasso criterion (Banerjee et al., 2007, Friedman et al., 2007):

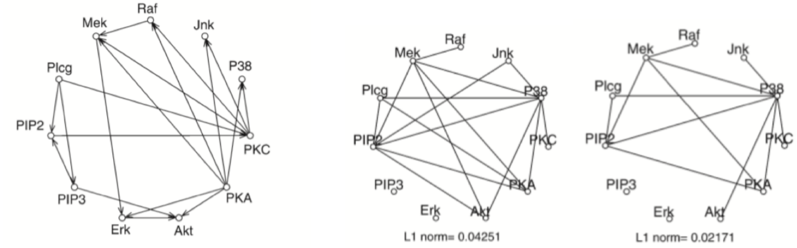
$$\min_{\Theta \in R^{p \times p}} - \log \det \Theta + tr(S\Theta) + \lambda \|\Theta\|_1 \tag{16}$$

- Minimizer Θ^* is an estimate for Σ^{-1}
- $S = X^T X / n$ is the empirical covariance matrix

from `https://www.cs.cmu.edu/~ggordon/10725-F12/slides/25-coord-desc.pdf`

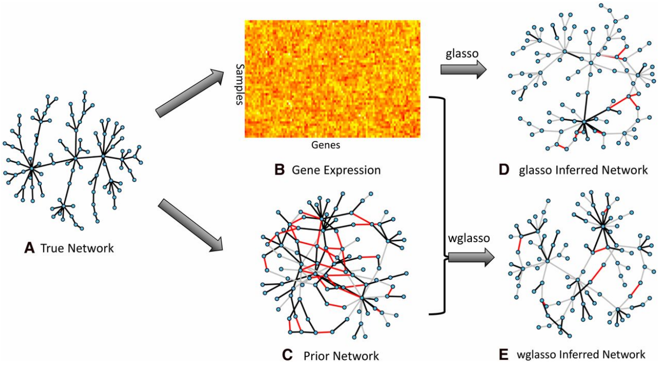
Graphical Lasso

Example from Friedman et al. (2007), cell-signaling network:



On the left: true network, on the right: graphical Lasso estimate

Graphical Lasso



Gene Network Reconstruction by Integration of Prior Biological Knowledge by Yupeng Li and Scott A. Jackson, 2015

Feature Selection and Model Selection

Structured Output Prediction

Stability and Feature Selection

Some Words on Dimensionality Reduction

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 θ.

Model of Conditional Random Fields

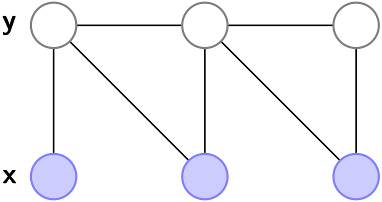
Conditional Random Fields (*Lafferty, McCallum, Pereira, 2001*) are based on the discriminative probabilistic model

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

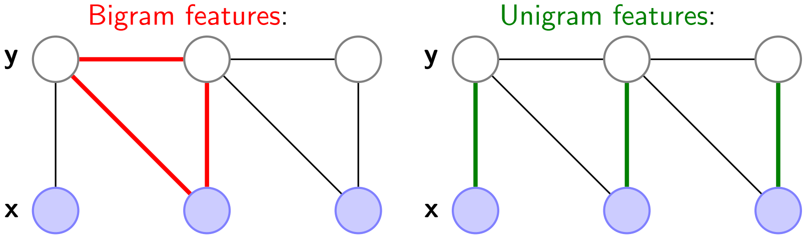
- $\{f_k\}_{1 \le k \le K}$ is an arbitrary set of feature functions
- $\{\theta_k\}_{1 \le k \le K}$ are real-valued parameters, associated with the feature functions
- the normalization factor

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

Conditional Random Fields: Graphical Model



Feature Functions



Bigram features:

Unigram features:

$$\sum_{k=1}^{K} \theta_k f_k(y_{t-1}, y_t, x_t) = \sum_{X \in \mathcal{X}} \left(\sum_{y \in Y, x \in X} \mu_{y,x} \mathbb{1}\{y_t = y, x_t = x\} \right.$$

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

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

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.

$$\Downarrow$$

- 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
            $\{\partial\ell(\mathcal{D};\theta)/\partial\mu_{y,x}\ ;\ \partial^2\ell(\mathcal{D};\theta)/\partial\mu_{y,x}^2\}_{y\in Y}$
            $\{\partial\ell(\mathcal{D};\theta)/\partial\lambda_{y',y,x}\ ;\ \partial^2\ell(\mathcal{D};\theta)/\partial\lambda_{y',y,x}^2\}_{(y',y)\in Y^2}$
            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):

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

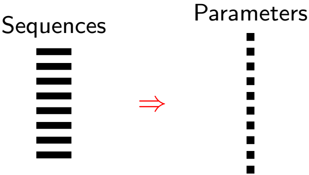
$$\alpha_{t+1}(y) = \exp(\mu_{y,x_{t+1}}) \left(\sum_{y'} \alpha_t(y') + \sum_{y'} \alpha_t(y') M_{t+1}(y', y) \right)$$

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

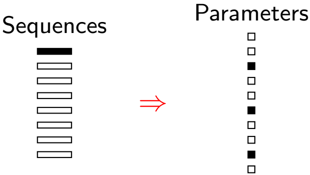
$r(x)$ multiplications instead of $|Y|^2$.

Brief Comparison of Optimization Approaches

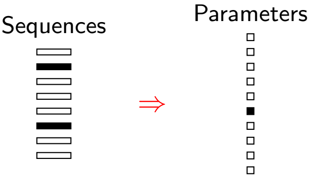
- Orthant-wise Limited Quasi Newton



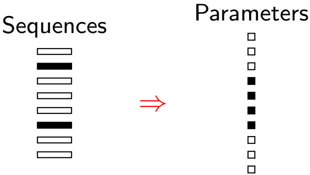
- Stochastic Gradient Descent



- 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

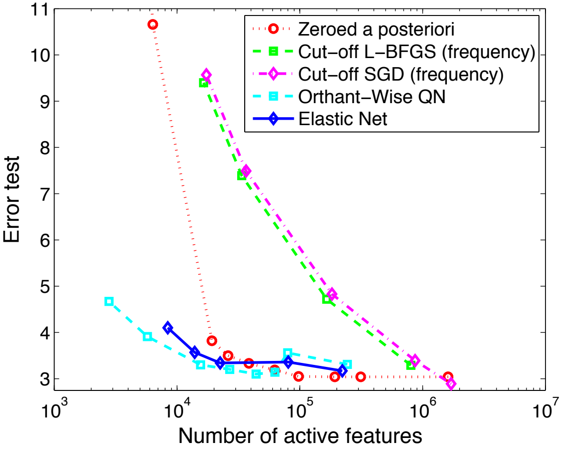
NER Data Set

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

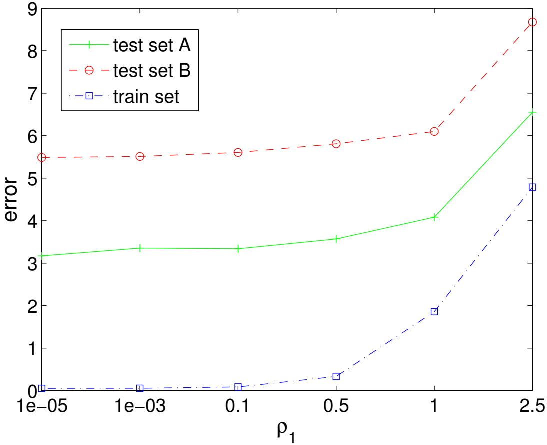
Algorithm	Time (error $\approx 3\%$)
SBCD	42
OWL-QN	5
L-BFGS	25
SGD	4

- 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.

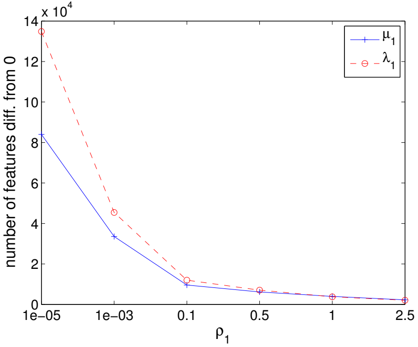
Performance of Model Selection Methods (NER task)

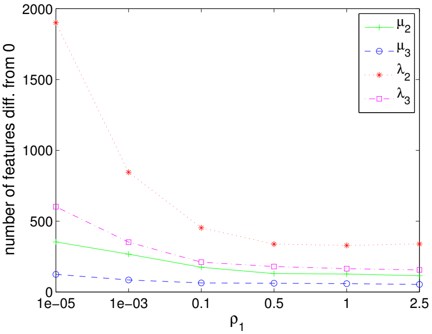


Results (NER task): Train and Test Errors



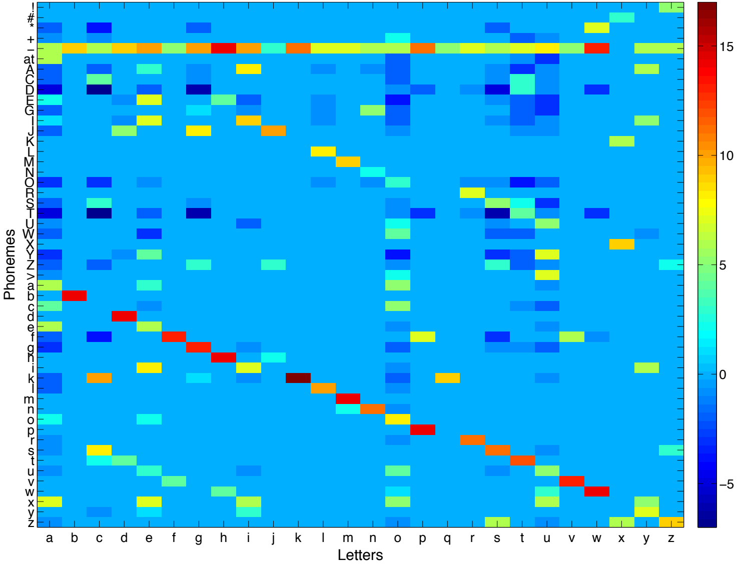
Results (NER task): Number of Active Features



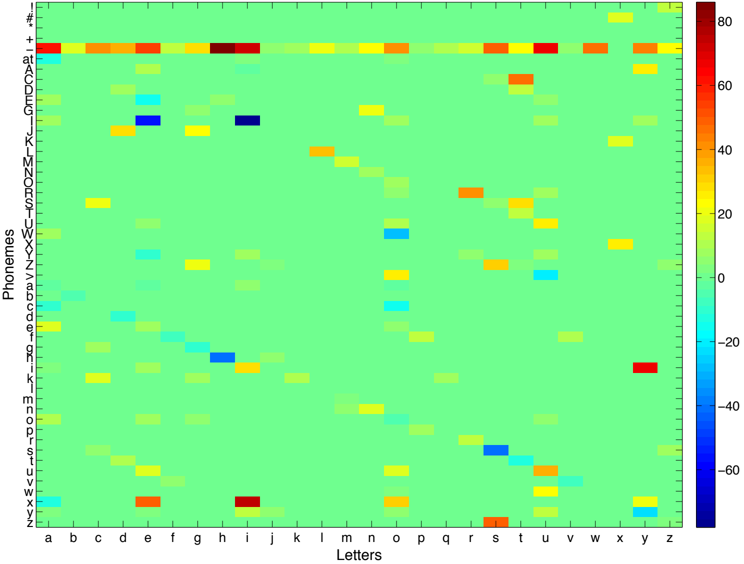


$\rho_1 = 0 \Rightarrow 1\,611\,832$ parameters,
$\rho_1 = 0.1 \Rightarrow 25\,090$ parameters.

NetTalk: Values of Unigram Parameters



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



A large margin approach for structured output prediction

B. Taskar et al., Learning Structured Prediction Models: A Large Margin Approach, ICML 2005 Max-margin estimation.

- Develop a method for finding parameters **w** such that

$$\arg\max_{\mathbf{y}\in Y} \mathbf{w}^T f(\mathbf{x}, \mathbf{y}) \approx \hat{\mathbf{y}}$$

Max margin for Structured Output

The optimal solution $\hat{\mathbf{y}}$ with respect to $\mathbf{w}$

$$\min \frac{1}{2}\|\mathbf{w}\|_2^2$$

such that

$$\mathbf{w}^T\mathbf{f}(\hat{\mathbf{y}},\mathbf{x}) \geq \mathbf{w}^T\mathbf{f}(\mathbf{y},\mathbf{x}) + l(\mathbf{y}),$$

where $l(\mathbf{y}) = l(\hat{\mathbf{y}},\mathbf{y})$.

- One can interpret

  $$\frac{1}{\|\mathbf{w}\|}\mathbf{w}^T(\mathbf{f}(\hat{\mathbf{y}},\mathbf{x}) - \mathbf{f}(\mathbf{y},\mathbf{x}))$$

  as the margin.
- The constraints enforce $\mathbf{f}(\hat{\mathbf{y}},\mathbf{x}) - \mathbf{f}(\mathbf{y},\mathbf{x}) \geq l$, so minimizing $\|\mathbf{w}\|$ maximized the smallest such margin, scaled by the loss l.
- Formulate as a convex optimization problem

Feature Selection and Model Selection

Structured Output Prediction

Stability and Feature Selection

Some Words on Dimensionality Reduction

Stability Issues

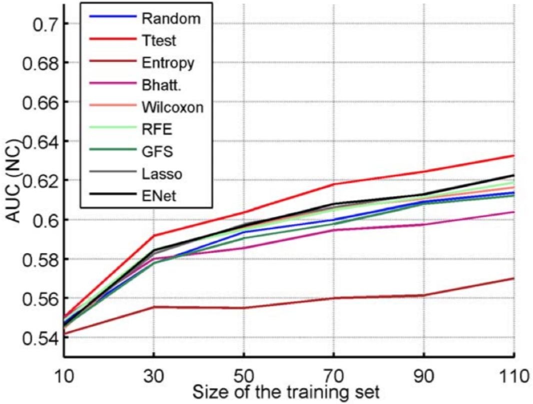
- Results are not reproducible!
- Moreover, different runs of the same algorithm would select a different sets of features
- Sometimes these sets of selected features are even not overlapping
- Feature selection is highly unstable
- A simple *t*-test seems to be the most stable feature selection method (see A.-C. Haury 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

Stability measures

S_i are sets of genes. The similarity between two sets S_i and S_j can be expressed as the number of genes that are present in both sets, $|S_i \cap S_j|$, normalized to be in $[0;1]$.

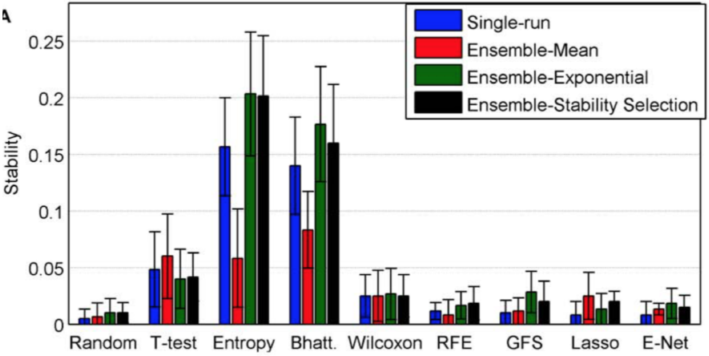
- Observation: the biological functions captured by different gene sets can be similar, despite a little degree of overlapping between these sets
- To exploit the gene sets in functional terms: gene annotations from the GO (Gene Ontology)
- GO provides a set of controlled vocabularies describing gene products based on their functions in the cell
- For each gene set S_i extract the list of molecular functions

Stability measures cont'd



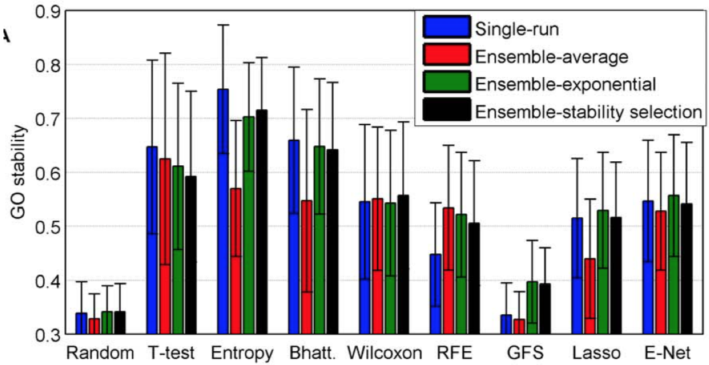
(from *A.-C. Haury et al., 2011*)

Stability on genes level



(from *A.-C. Haury et al., 2011*)

GO Stability



(from *A.-C. Haury et al., 2011*)

Feature Selection and Model Selection

Structured Output Prediction

Stability and Feature Selection

Some Words on Dimensionality Reduction

Some Words on Dimensionality Reduction

Dimensionality reduction $\neq$ feature selection!

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?