VARIATIONAL SEQUENCE LABELING

R. Flamary, S. Canu, A. Rakotomamonjy, J.L. Rose

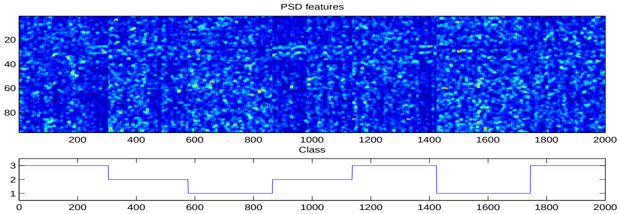
LITIS EA 4108, INSA-Université de Rouen
76800 Saint Etienne du Rouvray, France

Wednesday September 2, 2009

Sequence labeling (1)

Definition

To obtain a label for each sample of the signal while taking into account the sequentiality of the samples.



Example

Multi-class mental state decoding in BCI

- Subject thinking about the movement of his right arm, left arm or his feet
- PSD features along time

Sequence labeling (2)

Existing methods

- Hidden Markov Models [CMR05] ,Conditional Random Fields [LMP01]
- Structural SVM [TTHA05]
- Maximum Margin Markov Networks [TGK04]
- Structured Learning Ensemble[NG07]

Applications

- Automatic Speech Recognition
- Brain Computer Interfaces

Sequence labeling (3)

Structured Learning Ensemble[NG07]

Find the optimal sequence $\mathbf{y}^* \in \{1, 2, \ldots, N_c\}^T$ using results from M sequence labeling methods $(\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^M)$.

$$\mathbf{y}^* = \arg\min_{\mathbf{y}} \mathcal{L}(\mathbf{y}, \mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^M) \tag{1}$$

with $\mathcal{L}$ a loss function that takes into account the label provided by each method and all label transitions.

Our contribution

- Use scores instead of discrete labels (similar to soft decision[MZ06]).
- Express this problem in a variational framework (sum of functionals).
- Simple criterions proposed as functional
- Propose a general approximate algorithm to solve the problem

Variational approach

Variational framework

We cast the problem as a weighted sum of functionals:

$$\min_{\mathbf{y}} \sum_{i=1}^{N_f} \lambda_i J_i(\mathbf{y}, X, \mathbf{y}^{tr}, X^{tr}) \tag{2}$$

with each functional $J_i \in \mathbb{R}$ is balanced by $\lambda_i \in \mathbb{R}^+$, $X \in \mathbb{R}^{T \times d}$ feature matrix and $(\mathbf{y}^{tr}, X^{tr})$ is the training set.

Key ideas

- Each functional:criterion to optimize (Data, *a priori*)
- Straightforward to fuse several methods, to add prior information
- Focus on the variation of the functionals

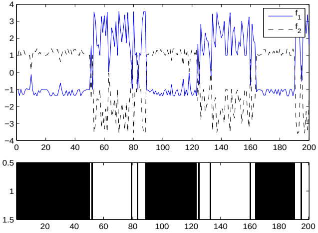
→ Need to express existing methods as a sum of functionals.

Labeling functional

- Functional corresponding to a supervised learning
- Needs functions f_n returning a class n membership score:

$$f_n = \arg\min_f \ \mathcal{L}_n(\mathbf{y}^{tr}, f(X^{tr})) + \lambda\Omega(f)$$

- If used alone, leads to winner takes all strategy



Labeling functional

$$J_{class}(\mathbf{y}, X) = -\sum_{i=1}^{T} f_{\mathbf{y}_i}(X_i) \quad (3)$$

By minimizing this functional, we choose for each sample the class with the maximum score

Other functionals

- *a priori* concerning the length of regions (large)
- Widely used in signal and image processing

Total Variation functional

$$J_{TV}(\mathbf{y}) = \sum_{i=1}^{T-1} \|\mathbf{y}_{i+1} - \mathbf{y}_i\|_0 \qquad (4)$$

where $||.||_0$ is the ℓ_0 norm.

Other functionals

- J_{edge} to add information from change detection methods
- J_{MM} to add Markov Model prior information

Discussion

Our algorithm

- Based on the Region Growing algorithm widely used in image processing
- Can handle any sum of functionals, even with non-differentiable ones
- We focus on the variation of the functionals and not in their value.

Variation of functionals

- Variation of J_{class} for changing the class of the ith sample from c_1 to c_2 is:

$$\Delta J_{class}(X, i, c_1, c_2) = f_{c_1}(X_i) - f_{c_2}(X_i)$$

- Variation of J_{TV} for changing the class of the ith sample from c_1 to c_2 is:

$$\Delta J_{TV}(\mathbf{y}, i, c_1, c_2) = \|c_2 - \mathbf{y}_{i-1}\|_0 + \|\mathbf{y}_{i+1} - c_2\|_0 - \|c_1 - \mathbf{y}_{i-1}\|_0 - \|\mathbf{y}_{i+1} - c_1\|_0$$

Algorithm (VSLA)

Initialization of $\mathbf{y}^0$ (0)

Edge moving (1)

For all edges:

- Compute ΔJ for moving edge to left or right
- Move edge if $\Delta J < 0$

Region switching (2)

For all regions:

- Compute ΔJ for switching regions to every other classes
- Change region if $\Delta J < 0$

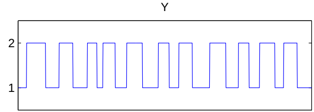
Repeat (1) and (2)

until no minimization is possible

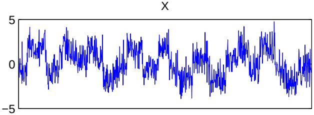
Example of the algorithm:

- 1-dimensional 2-class problem
- J_{class} is used with f_n svm classification functions.
- $\lambda_{class} = 1$.
- J_{TV} is used with $\lambda_{TV} = 5$

Training sequence:



Training signal:



Algorithm Example (0)

Initialization of $\mathbf{y}^0$ (0)

Edge moving (1)

For all edges:

- Compute ΔJ for moving edge to left or right
- Move edge if $\Delta J < 0$

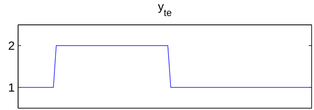
Region switching (2)

For all regions:

- Compute ΔJ for switching regions to every other classes
- Change region if $\Delta J < 0$

Repeat (1) and (2)

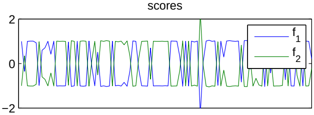
until no minimization is possible



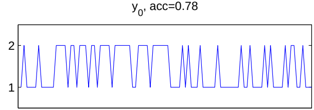
Initialization is done by solving a simple version of J:

$$\mathbf{y}^0 = \arg\min_{\mathbf{y}} J_{class}(\mathbf{y}, X)$$

with the scores f_n:



leading to this initialization:



Algorithm Example (1)

Initialization of $\mathbf{y}^0$ (0)

Edge moving (1)

For all edges:

- Compute ΔJ for moving edge to left or right
- Move edge if $\Delta J < 0$

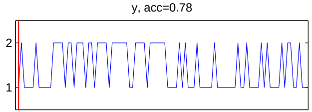
Region switching (2)

For all regions:

- Compute ΔJ for switching regions to every other classes
- Change region if $\Delta J < 0$

Repeat (1) and (2)

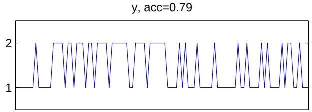
until no minimization is possible



Edge number 1:

movement	left	none	right
ΔJ_{class}	1.99	0	0.68
ΔJ_{TV}	0	0	-2
ΔJ	1.99	0	-9.31

$\Rightarrow$Edge moved to the right:



Algorithm Example (1)

Initialization of $\mathbf{y}^0$ (0)

Edge moving (1)

For all edges:

- Compute ΔJ for moving edge to left or right
- Move edge if $\Delta J < 0$

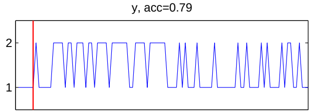
Region switching (2)

For all regions:

- Compute ΔJ for switching regions to every other classes
- Change region if $\Delta J < 0$

Repeat (1) and (2)

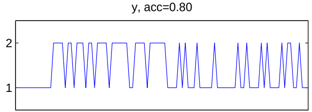
until no minimization is possible



Edge number 2:

movement	left	none	right
ΔJ_{class}	1.86	0	1.95
ΔJ_{TV}	0	0	-2
ΔJ	1.86	0	-8.04

$\Rightarrow$Edge moved to the right:



Algorithm Example (1)

Initialization of $\mathbf{y}^0$ (0)

Edge moving (1)

For all edges:

- Compute ΔJ for moving edge to left or right
- Move edge if $\Delta J < 0$

Region switching (2)

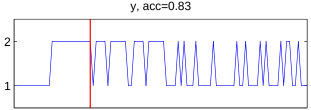
For all regions:

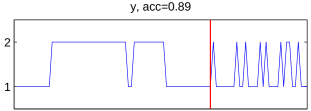
- Compute ΔJ for switching regions to every other classes
- Change region if $\Delta J < 0$

Repeat (1) and (2)

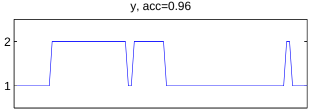
until no minimization is possible

Every edge in the current $\mathbf{y}$ is tested once:





which leads to this $\mathbf{y}$ at the end of (1):



Algorithm Example (2)

Initialization of $\mathbf{y}^0$ (0)

Edge moving (1)

For all edges:

- Compute ΔJ for moving edge to left or right
- Move edge if $\Delta J < 0$

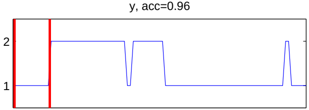
Region switching (2)

For all regions:

- Compute ΔJ for switching regions to every other classes
- Change region if $\Delta J < 0$

Repeat (1) and (2)

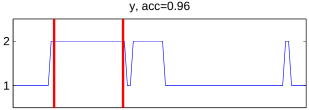
until no minimization is possible



Region 1:

switch to	1	2
ΔJ_{class}	0	12.7
ΔJ_{TV}	0	-1
ΔJ	0	7.2

$\Rightarrow$ Region not switched



Same for Region 2

Algorithm Example (2)

Initialization of $\mathbf{y}^0$ (0)

Edge moving (1)

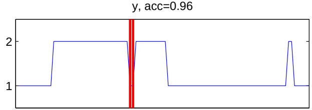
For all edges:
- Compute ΔJ for moving edge to left or right
- Move edge if $\Delta J < 0$

Region switching (2)

For all regions:
- Compute ΔJ for switching regions to every other classes
- Change region if $\Delta J < 0$

Repeat (1) and (2)

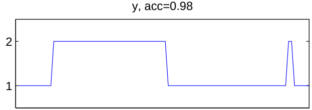
until no minimization is possible



Region 3:

switch to	1	2
ΔJ_{class}	0	4.02
ΔJ_{TV}	0	-2
ΔJ	0	-5.97

$\Rightarrow$ Region 3 switched to class 2



Algorithm Example (2)

Initialization of $\mathbf{y}^0$ (0)

Edge moving (1)

For all edges:
- Compute ΔJ for moving edge to left or right
- Move edge if $\Delta J < 0$

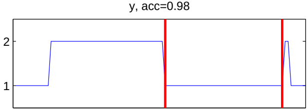
Region switching (2)

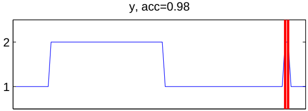
For all regions:
- Compute ΔJ for switching regions to every other classes
- Change region if $\Delta J < 0$

Repeat (1) and (2)

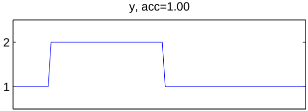
until no minimization is possible

Every region in the current $\mathbf{y}$ is tested once

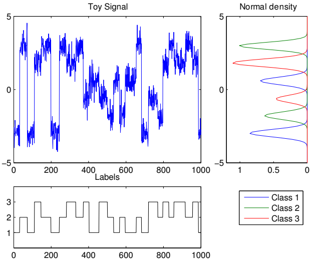




which leads to this $\mathbf{y}$ at the end of (2):



Toy Dataset



J_{class}	SVM	MG	KRR
$\varnothing$	0.7111	0.7393	0.7343
$+J_{TV}$	0.8677	0.9311	0.9155
$+J_{MM}$	0.8138	0.9005	0.8775

Toy Problem

- 1-Dimensional noisy signal
- Non linear (2 different values possible per class)
- SVM, MG, KRR classification methods for scores of J_{class}

BCI Dataset

Functionals	Sub. 1	Sub. 2	Sub. 3
J_{class}	0.7392	0.6262	0.4931
$\ldots + J_{TV}$	**0.9843**	**0.8531**	**0.5932**
$\ldots + J_{MM}$	0.9783	0.7955	0.4455
BCI III Res.	0.9598	0.7949	0.6743

Dataset

- BCI Competition III Dataset: 3 classes, 3 sessions training, 1 session test
- λ selected by validation on the third training session
- Classification scores obtained by linear regression with channel selection [Rak09].

Conclusion

Conclusion

- General framework for combining several sequence labeling criterions
- Easy integrating of prior knowledge
- Algorithm proposed based on Region Growing
- Promising results on a real life example

Future works

- Express other sequence labeling methods (Structural SVM, CRF) in the variational framework and fuse them
- Comparison of VSLA with other methods/fusion methods

Bibliography

[CMR05] O. Cappé, E. Moulines, and T. Rydén.
Inference in Hidden Markov Models.
Springer, 2005.

[LMP01] J. Lafferty, A. McCallum, and F. Pereira.
Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
In *Proc. 18th International Conf. on Machine Learning*, pages 282–289, 2001.

[MZ06] Robert H. Morelos-Zaragoza.
The Art of Error Correcting Coding.
Wiley, 2006.

[NG07] N. Nguyen and Y. Guo.
Comparisons of sequence labeling algorithms and extensions.
In *Proc. 24th international Conf. on Machine learning*, pages 681–688. ACM, 2007.

[Rak09] A. Rakotomamonjy.
Algorithms for multiple basis pursuit denoising.
In *Workshop on Sparse Approximation*, 2009.

[TGK04] B. Taskar, C. Guestrin, and D. Koller.
Max-margin markov networks.
In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[TTHA05] I. Tsochantaridis, J. Thorsten, T. Hofmann, and Y. Altun.
Large margin methods for structured and interdependent output variables.
In *Journal Of Machine Learning Research*, volume 6, pages 1453–1484, Cambridge, MA, USA, 2005. MIT Press.