Doctoral progress report

Vilen Jumutc
Promoter: prof. Johan A.K. Suykens







The second oral presentation for Supervisory Committee
August 2015

Outline

1. **PhD thesis motivation & breakdown**
2. **Research results & publications**
   - Supervised, Unsupervised and Semi-Supervised SVMs
     - Multi-Class Supervised Novelty Detection
     - Bilinear Semi-Supervised Kernel Spectral Clustering
   - Stochastic learning
     - Fixed-Size Pegasos with Pinball Loss
     - Weighted Coordinate-Wise Pegasos
     - Reweighted Regularized Dual Averaging
   - Flexible software design for Machine Learning libraries
     - SALSA.jl package
3. **Completed doctoral courses & seminars**
4. **Assisted courses**
5. **Publications**
6. **References**

Title: Flexible software design and kernel-based learning in advanced data-driven modelling

Motivation

Currently there are many software packages available for the end user but the majority of such solutions are merely intended for the use of machine learning practitioners and out-of-field scientists. We aim at combining novel kernel-based methods with the advanced software design for elaborating scalable, robust and user-friendly black-box Machine Learning libraries.

Tentative PhD thesis breakdown

1. Unsupervised and Semi-Supervised SVMs.
2. Stochastic learning for linear and nonlinear SVMs.
3. Flexible software design for Machine Learning libraries.

Multi-Class Supervised Novelty Detection [Jumutc and Suykens, 2014a]

1. Multi-Class Supervised Novelty Detection (MC-SND) [Jumutc and Suykens, 2014a] is designed for finding outliers in the presence of several classes.
2. MC-SND estimates the support of the all target classes (distributions) while trying to keep necessary discrimination between them.
3. MC-SND supports the i.i.d. assumption and density estimation for all target classes (distributions) separately.
4. MC-SND doesn't try to find outliers in the existing pool of training samples.

A small spice of theory

Primal form

$$\min_{w_i \in \mathcal{F}; \xi_i \in \mathbb{R}^n; \rho_i \in \mathbb{R}} \quad \frac{\gamma}{2} \sum_{i=1}^{n_c} \|w_i\|^2 + \sum_{i,j=1}^{n_c} \langle w_i, w_j \rangle + C \sum_{i=1}^{n} \sum_{j=1}^{n_c} \xi_{ij} - \sum_{i=1}^{n_c} \rho_i \tag{1}$$

$$\text{s.t.} \quad y_{ij}(\langle w_j, \Phi(x_i) \rangle) \geq \rho_j - \xi_{ij}, \quad i \in \overline{1,n}, \quad j \in \overline{1,n_c}$$
$$\xi_{ij} \geq 0, \quad i \in \overline{1,n}, \quad j \in \overline{1,n_c} \tag{2}$$

Dual form

$$\max_{\alpha_i} \mathcal{L}_D(\alpha_i) = \frac{1}{\mu} \sum_{i}^{n_c} \lambda_i^T K \alpha_i, \tag{3}$$

$$\text{s.t.} \quad C \geq \alpha_{ij} \geq 0, \ \forall i \in \overline{1,n}, \ \forall j \in \overline{1,n_c}$$
$$\sum_{i=1}^{n} \alpha_{ij} y_{ij} = -1, \ \forall j \in \overline{1,n_c}$$
$$\lambda_i = (\gamma + n - 2)(\alpha_i \circ y_i) - \sum_{j=1, j \neq i}^{n_c} (\alpha_j \circ y_j), \forall i$$
$$\mu = (\eta + 1)(\gamma - 1). \tag{4}$$

Real-life example of MC-SND usage

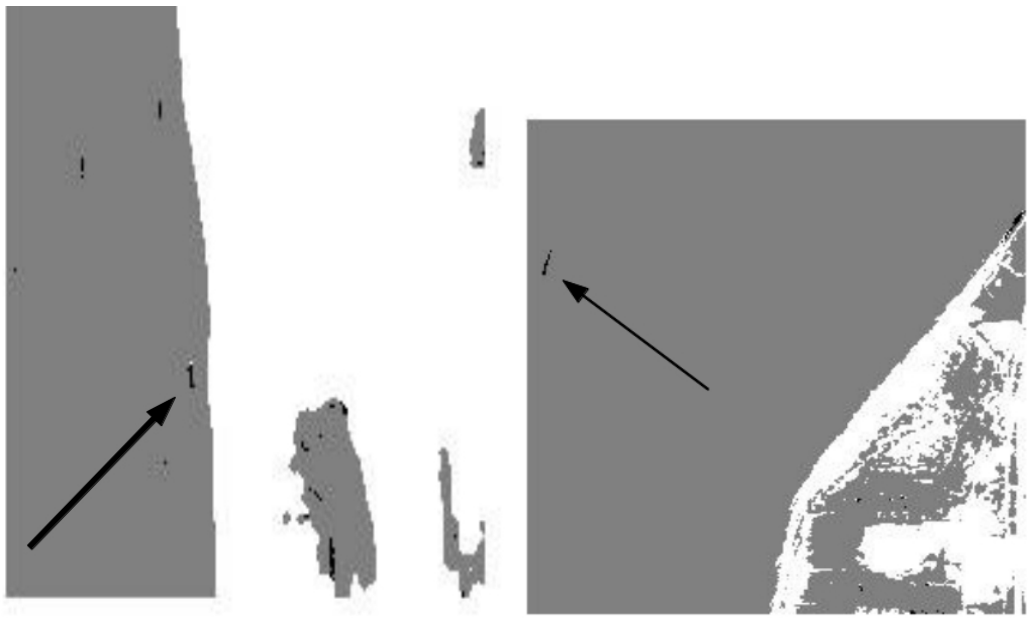


Figure: AVIRIS training image after preprocessing (left) and test image after evaluation by the MC-SND algorithm (right) with pointed out outliers.

Bilinear Semi-Supervised Kernel Spectral Clustering [Jumutc and Suykens, 2014b]

1. Bilinear formulation to Semi-Supervised Kernel Spectral Clustering (SS-KSC) [Jumutc and Suykens, 2014b] is designed to obtain better cluster estimates when only few labels are available.
2. Bilinear SS-KSC can serve both objective: classification and clustering.
3. Bilinear SS-KSC is similar to Multi-Class SND but it doesn't take into account density estimation. It includes variance maximization term and is similar to Spectral Clustering techniques.

A small spice of theory

Our primal formulation for a simple binary semi-supervised classification problem have an additional bilinear term $\langle w_1, w_2 \rangle$ between classifiers and separate manifold regularization for each individual classifier:

Optimization problem

$$\min_{w_1, w_2 \in \mathbb{R}^d; e_1, e_2 \in \mathbb{R}^m} \quad \frac{\gamma_1}{2}(\|w_1\|^2 + \|w_2\|^2) + \langle w_1, w_2 \rangle$$
$$+ \frac{\gamma_2}{2} \sum_{i=1}^{l} [(e_{1i} - y_i)^2 + (e_{2i} + y_i)^2] \tag{5}$$
$$- \frac{\gamma_3}{2} e_1^T V e_1 - \frac{\gamma_4}{2} e_2^T V e_2$$

$$\text{s.t.} \quad \Phi w_1 = e_1, \quad \Phi w_2 = e_2. \tag{6}$$

Toy dataset example for Bilinear SS-KSC

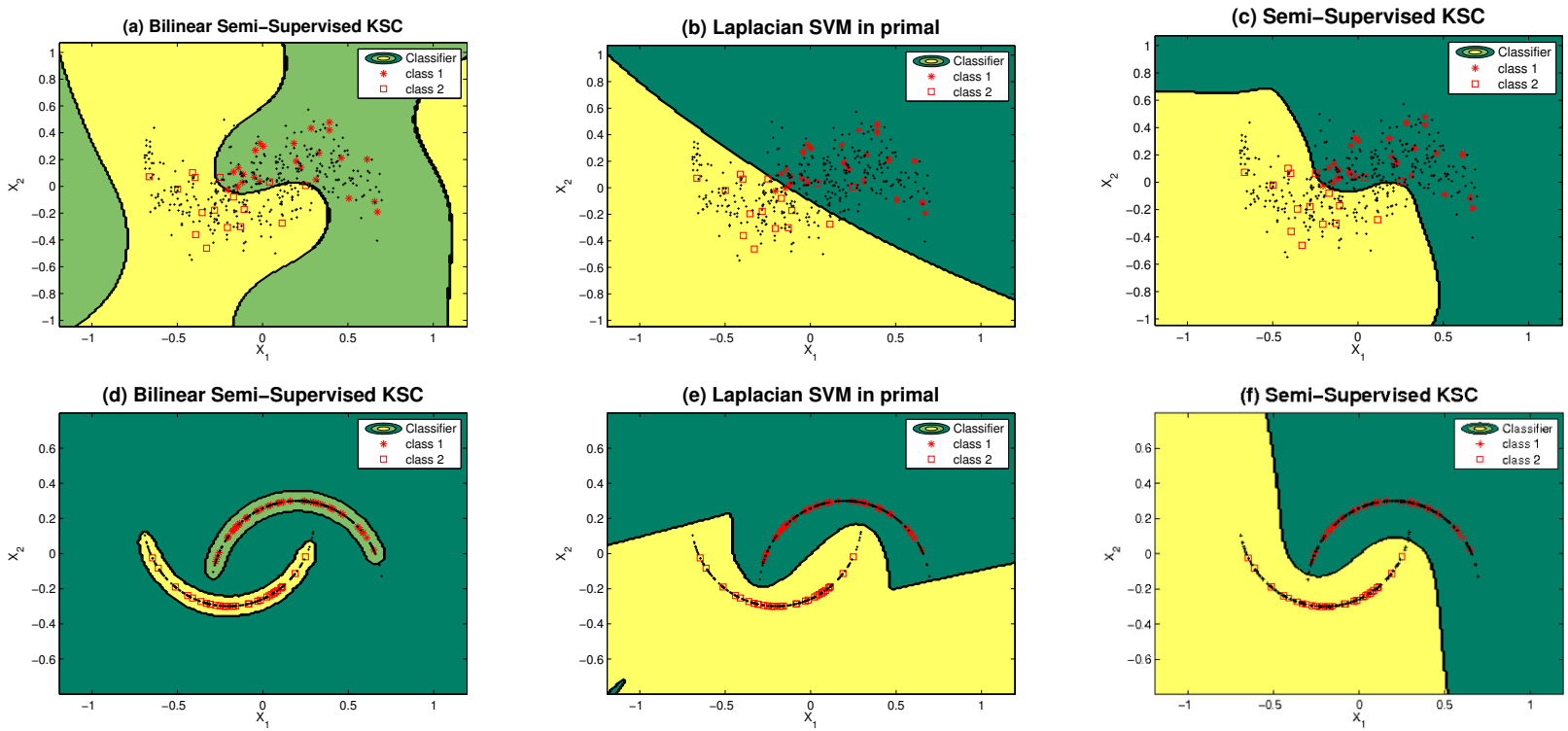


Figure: Different approaches applied to the "half-moons" problem. With small black dots we denote unlabeled data points. Bigger red stars and squares represent labeled samples from two classes.

Stochastic programming

- By stochastic programming [Nemirovski, 2009] we assume the following unconstrained optimization problem

$$\min_{x \in X} \{f(x) = \mathbb{E}[F(x, \xi)]\}. \tag{7}$$

Here $X \in \mathbb{R}^n$ is a nonempty bounded closed convex set, ξ is a random vector whose probability distribution P is supported on set $\Xi \in \mathbb{R}^d$ and $F : X \times \Xi \to \mathbb{R}$.

Pegasos

- Pegasos [S-Shwartz et al., 2007] has become a widely acknowledged algorithm for learning linear SVMs. It utilizes strongly convex optimization objective and hinge loss which replaces linear constraints.
- As a result we benefit from the faster convergence rates and can directly apply stochastic approaches via instantaneous optimization objective

$$f(w; \mathcal{A}_t) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{|\mathcal{A}_t|} \sum_{(x,y)\in\mathcal{A}_t} \mathbb{L}(w; (x, y)), \tag{8}$$

where $\mathcal{A}_t$ is our current data at evaluation step t and $\mathbb{L}(w; (x, y)) = \max\{0, 1 - y\langle w, x\rangle\}$ stands for the hinge loss.

Algorithm 1: Pegasos with pinball loss

Data: $\mathcal{S}, \lambda, \tau, T, k, \epsilon$

Select w_1 randomly s.t. $\|w^{(1)}\| \leq 1/\sqrt{\lambda}$;

for $t = 1 \rightarrow T$ **do**

  Set $\eta_t = \frac{1}{\lambda t}$

  Select $\mathcal{A}_t \subseteq \mathcal{S}$, where $|\mathcal{A}_t| = k$;

  $\rho = \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{A}_t} (y - \langle w_t, x\rangle)$;

  $\mathcal{A}_t^+ = \{(x,y) \in \mathcal{A}_t : y(\langle w_t, x\rangle + \rho) < 1\}$;

  $\mathcal{A}_t^- = \{(x,y) \in \mathcal{A}_t : y(\langle w_t, x\rangle + \rho) > 1\}$;

  $w_{t+\frac{1}{2}} = w_t - \eta_t(\lambda w_t - \frac{1}{k}\left[\sum_{(x,y)\in\mathcal{A}_t^+} yx - \sum_{(x,y)\in\mathcal{A}_t^-} \tau yx\right])$;

  $w_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\|w_{t+\frac{1}{2}}\|}\right\} w_{t+\frac{1}{2}}$;

  if $\|w_{t+1} - w_t\| \leq \epsilon$ **then**

    return $(w_{t+1}, \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} (y - \langle w_t, x\rangle))$;

  end

end

return $(w_{T+1}, \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} (y - \langle w_{T+1}, x\rangle))$;

Fixed-Size approach

- Algorithm 1 operates only in the primal space. To handle this restriction we go for the Fixed-Size approach [Suykens et al., 2002].
- Entropy based criterion is used to select m prototype vectors and construct $m \times m$ RBF kernel matrix K.
- Nyström approximation [Williams and Seeger, 2001] gives an expression for the entries of the approximated feature map $\hat{\Phi}(x) : \mathbb{R}^d \rightarrow \mathbb{R}^m$ with $\hat{\Phi}(x) = (\hat{\Phi}_1(x), \ldots, \hat{\Phi}_m(x))^T$ and

$$\hat{\Phi}_i(x) = \frac{1}{\sqrt{\lambda_{i,m}}} \sum_{t=1}^{m} u_{ti,m} k(x_t, x), \tag{9}$$

where $\lambda_{i,m}$ and $u_{i,m}$ denote the i-th eigenvalue and the i-th eigenvector of K and $k(x, y)$ denotes the RBF function.

Algorithm 2: Complete procedure [Jumutc and Suykens, 2013a]

Data : training data $\mathcal{S}$ with $|\mathcal{S}| = n$, labeling Y, parameters $\lambda, \tau, T, k, \epsilon, m$
Return: mapping $\hat{\Phi}(x), \forall x \in \mathcal{S}$, SVM model given by w and ρ
begin
  $\mathcal{S}_r \leftarrow$ `FindActiveSet(`$\mathcal{S}, m$`)`;
  $\hat{\Phi}(x) \leftarrow$ `ComputeNystromApprox(`$\mathcal{S}_r$`)`;
  $X \leftarrow [\hat{\Phi}(x_1)^T, \ldots, \hat{\Phi}(x_n)^T]$;
  $[w, \rho] \leftarrow$ `PegasosPBL(`$X, Y, \lambda, \tau, T, k, \epsilon$`)`;
end

Weighted Coordinate-Wise Pegasos [Jumutc and Suykens, 2013b]

- Treating every dimension of the problem equally might not be always beneficial and reasonable in terms of **convergence** and **generalization error**.
- To obtain our **weighted coordinate-wise** formulation for the Pegasos algorithm we concentrate on a new instantaneous optimization objective:

$$f_{wcw}(w;\mathcal{A}_t) = \frac{1}{2}w^T\Lambda w + \frac{1}{|\mathcal{A}_t|}\sum_{(x,y)\in\mathcal{A}_t}\mathbb{L}(w;(x,y)), \qquad (10)$$

where Λ stands for the diagonal matrix with entries corresponding to coordinate-wise λ_i regularization hyperparameters.

Algorithm 3: Weighted Coordinate-Wise Pagasos

Data: $\mathcal{S}, \Lambda, T, k, \epsilon$

Set $\lambda_{min} = \min_i \Lambda_{ii}$

Select w_1 randomly s.t. $\|w^{(1)}\| \leq 1/\sqrt{\lambda_{min}}$

for $t = 1 \rightarrow T$ **do**

  Set $\eta_t = \frac{1}{t}\Lambda^{-1}$

  Select $\mathcal{A}_t \subseteq \mathcal{S}$, where $|\mathcal{A}_t| = k$

  $\rho = \frac{1}{|\mathcal{S}|}\sum_{(x,y)\in\mathcal{S}}(y - \langle w_t, x\rangle)$

  $\mathcal{A}_t^+ = \{(x,y) \in \mathcal{A}_t : y(\langle w_t, x\rangle + \rho) < 1\}$

  $w_{t+\frac{1}{2}} = w_t - \eta_t(\Lambda w_t - \frac{1}{k}\sum_{(x,y)\in\mathcal{A}_t^+} yx)$

  $w_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda_{min}}}{\|w_{t+\frac{1}{2}}\|}\right\} w_{t+\frac{1}{2}}$

  if $\|w_{t+1} - w_t\| \leq \epsilon$ **then**

    return $(w_{t+1}, \frac{1}{|\mathcal{S}|}\sum_{(x,y)\in\mathcal{S}}(y - \langle w_t, x\rangle))$

  end

end

return $(w_{T+1}, \frac{1}{|\mathcal{S}|}\sum_{(x,y)\in\mathcal{S}}(y - \langle w_t, x\rangle))$

Performance of the Weighted Coordinate-Wise Pagasos

Table: Test errors for larger-scale datasets

Dataset	Pegasos	Pegasos$_{wcw}$
Magic	0.30550	**0.23607**
Shuttle	0.21450	**0.08549**
Red Wine	**0.26924**	0.27747
White Wine	0.32443	**0.30541**
Covertype	0.36466	**0.32807**
Pen Digits (1 vs all)	0.10432	**0.08984**
Pen Digits (2 vs all)	0.08448	**0.05133**
Pen Digits (5 vs all)	0.09609	**0.06830**
Pen Digits (6 vs all)	0.05835	**0.02896**

(a) $k = 10\%$ of $|\mathcal{S}|$ (partially stochastic)

Dataset	Pegasos	Pegasos$_{wcw}$
Magic	0.32288	**0.27743**
Shuttle	0.21751	**0.08598**
Red Wine	0.29552	**0.29224**
White Wine	0.32807	**0.30599**
Covertype	0.36474	**0.34962**
Pen Digits (1 vs all)	0.10409	**0.09255**
Pen Digits (2 vs all)	0.08437	**0.05317**
Pen Digits (5 vs all)	0.09635	**0.07059**
Pen Digits (6 vs all)	0.05863	**0.02847**

(b) $k = 1$ (fully stochastic)

Regularized Dual Averaging

- In the stochastic Regularized Dual Averaging approach developed by Xiao [Xiao, 2010] one approximates the loss function $f(w)$ by using a finite set of independent observations $\mathcal{S} = \{\xi_t\}_{1 \leq t \leq T}$. Under this setting one minimizes the following optimization objective:

$$\min_w \frac{1}{T} \sum_{t=1}^{T} f(w, \xi_t) + \psi(w), \tag{11}$$

where $\psi(w)$ represents a regularization term. Every observation is given as a pair of input-output variables $\xi = (x, y)$. In the above setting one deals with a simple classification model $\hat{y}_t = \text{sign}(\langle w, x_t \rangle)$ and calculates the corresponding loss $f(w, \xi_t)$ accordingly.

Reweighted Regularized Dual Averaging

- For promoting sparsity we define an iterate-dependent regularization $\psi_t(w) \triangleq \lambda\|\Theta_t w\|$ which in the limit ($t \to \infty$) applies an approximation to the l_0-norm penalty. At every iteration t we will be solving a separate convex instantaneous optimization problem conditioned on a combination of the diagonal reweighting matrices Θ_t. By using a *simple dual averaging* scheme [Nesterov, 2009] we can solve our problem effectively by the following sequence of iterates w_{t+1}:

$$w_{t+1} = \arg\min_w \{\frac{1}{t}\sum_{\tau=1}^{t}(\langle g_\tau, w\rangle + \psi_\tau(w)) + \frac{\beta_t}{t}h(w)\}. \qquad (12)$$

Generalization performance for Reweighted Stochastic schemes

Table: Generalization performance

Dataset	Generalization (test) errors in % for UCI datasets											
	l_1-RDA$_{re}$ [Jumutc and Suykens, 2014c]		l_2-RDA$_{re}$ [Jumutc and Suykens, 2014d]		l_1-RDA$_{ada}$ [Duchi, 2011]		l_1-RDA [Xiao, 2010]		Pegasos [S-Shwartz et al., 2007]		Pegasos$_{drop}$	
Pen Digits	**6.3**	(±2.1)	7.9	(±2.5)	6.9	(±2.3)	9.2	(±13)	6.3	(±1.9)	**6.1**	(±2.2)
Opt Digits	**3.9**	(±1.7)	4.8	(±2.1)	4.0	(±1.9)	4.4	(±1.9)	**3.4**	(±1.2)	5.3	(±6.4)
Shuttle	**5.3**	(±2.4)	6.9	(±2.7)	5.8	(±2.1)	5.6	(±2.0)	5.3	(±1.7)	**4.7**	(±1.4)
Spambase	11.0	(±3.0)	11.5	(±2.0)	**10.8**	(±1.7)	12.6	(±13)	10.0	(±1.7)	**9.4**	(±1.6)
Magic	22.7	(±2.4)	**22.2**	(±1.3)	22.4	(±1.7)	22.6	(±2.0)	**22.2**	(±1.1)	25.3	(±2.8)
Covertype	28.3	(±1.8)	27.0	(±1.4)	**25.3**	(±1.1)	26.6	(±2.6)	**27.6**	(±1.0)	28.2	(±2.6)
CNAE-9	2.0	(±1.4)	3.6	(±3.7)	**1.9**	(±1.4)	2.3	(±1.8)	1.2	(±1.1)	**0.9**	(±0.9)
Semeion	**8.9**	(±2.6)	13.3	(±18)	10.0	(±3.0)	11.6	(±13)	5.6	(±1.9)	**5.3**	(±1.8)
CT slices	**5.6**	(±1.4)	8.9	(±4.0)	8.4	(±2.8)	8.0	(±1.9)	**5.0**	(±0.7)	5.2	(±1.0)
URI	4.4	(±1.7)	5.2	(±3.0)	**4.0**	(±1.0)	4.8	(±2.5)	**4.3**	(±1.8)	8.4	(±6.0)

Algorithm 4: Stochastic Reweighted l_1-Regularized Dual Averaging [Jumutc and Suykens, 2014c]

Data: $\mathcal{S}, \lambda > 0, \gamma > 0, \rho \geq 0, \epsilon > 0, T > 1, k \geq 1, \varepsilon > 0$

Set $w_1 = 0, \hat{g}_0 = 0, \Theta_1 = diag([1, \ldots, 1])$

for $t = 1 \rightarrow T$ **do**

  Draw a sample $\mathcal{A}_t \subseteq \mathcal{S}$ of size k

  Calculate $g_t \in \partial f_t(w_t; \mathcal{A}_t)$

  Compute the dual average $\hat{g}_t = \frac{t-1}{t}\hat{g}_{t-1} + \frac{1}{t}g_t$

  Compute the next iterate w_{t+1} by

$$w_{t+1}^{(i)} = \begin{cases} 0, & \text{if } |\hat{g}_t^{(i)}| \leq \eta_t^{(i)} \\ -\frac{\sqrt{t}}{\gamma}(\hat{g}_t^{(i)} - \eta_t^{(i)}\text{sign}(\hat{g}_t^{(i)})), & \text{otherwise} \end{cases}$$

  Re-calculate the next Θ by $\Theta_{t+1}^{(ii)} = 1/(|w_{t+1}^{(i)}| + \epsilon)$

  if $\|w_{t+1} - w_t\| \leq \varepsilon$ **then**

    return w_{t+1}

  end

end

return w_{T+1}

Algorithm 5: Stochastic Reweighted l_2-Regularized Dual Averaging [Jumutc and Suykens, 2014d]

Data: $\mathcal{S}, \lambda > 0, k \geq 1, \epsilon > 0, \varepsilon > 0, \delta > 0$

Set $w_1 = 0, \hat{g}_0 = 0, \Theta_0 = diag([1, \ldots, 1])$

for $t = 1 \rightarrow T$ **do**

  Draw a sample $\mathcal{A}_t \subseteq \mathcal{S}$ of size k

  Calculate $g_t \in \partial f(w_t, \mathcal{A}_t)$

  Compute the dual average $\hat{g}_t = \frac{t-1}{t}\hat{g}_{t-1} + \frac{1}{t}g_t$

  Compute the next iterate $w_{t+1}^{(i)} = -\hat{g}_t^{(i)} / (\lambda + \frac{1}{t}\sum_{\tau=1}^{t} \Theta_\tau^{(ii)})$

  Recalculate the next Θ by $\Theta_{t+1}^{(ii)} = 1/((w_{t+1}^{(i)})^2 + \epsilon)$

  if $\|w_{t+1} - w_t\| \leq \delta$ **then**

    `Sparsify(`w_{t+1}, ε`)`

  end

end

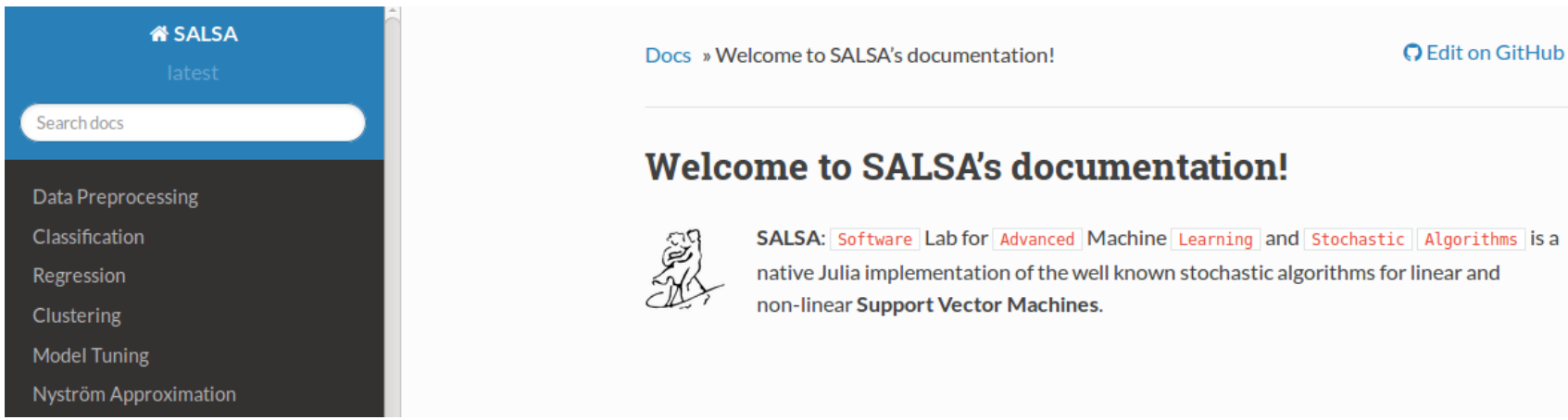
return `Sparsify(`w_{T+1}, ε`)`

SALSA: Software Lab for Advanced Machine Learning and Stochastic Algorithms in julia

Knowledge agnostic usage

```
using MAT, SALSA

# Load Ripley data
data = matread(joinpath(Pkg.dir("SALSA"),"data","ripley.mat"))

# Train and cross-validate Pegasos algorithm (default) on training data
# and evaluate it on the test data provided as the last function argument
model = salsa(data["X"], data["Y"], data["Xt"]);

# Compute accuracy in %
@printf "Accuracy: %.2f%%\n" mean(model.output.Ytest .== data["Yt"])*100

# Or use map_predict function and map data beforehand by the extracted mean/std (default)
@printf "Accuracy: %.2f%%\n" mean(map_predict(model, data["Xt"]) .== data["Yt"])*100
```

SALSA: documentation I



SALSA: documentation II

salsa(*X*, *Y* [, *Xtest*])

Create a linear classification model:

$$\hat{y} = \text{sign}(\langle x, w \rangle + b)$$

based on data given in `X` and labeling specified in `Y`. Optionally evaluate it on `Xtest`. Data should be given in the row-wise format (one sample per row). The classification model is embedded into returned `model` as `model.output`. The choise of different algorithms, loss functions and modes will be explained further in this chapter.

```
using SALSA, MAT, Base.Test

ripley = matread(joinpath(Pkg.dir("SALSA"),"data","ripley.mat"))
model = salsa(ripley["X"],ripley["Y"],ripley["Xt"]) # --> SALSAModel(...)
@test_approx_eq_eps mean(ripley["Yt"] .== model.output.Ytest) 0.89 0.01
```

Completed doctoral courses & seminars

Completed courses and seminars

1. Robust Statistics (B-KUL-G0B16A, @ KU Leuven)
2. Derivative-Free Optimization: Basic Principles and State of the Art (@ UCL)
3. Graphics in R (@ UAntwerpen)
4. Supervising exercise sessions at KU Leuven (seminar)
5. How to supervise a master thesis (seminar)

Assisted courses (2013-2015)

Assisted courses @ KU Leuven

1. H00H3a Support Vector Machines: Methods and Applications: Exercises

Publications (2012-2015)

- Jumutc V., Suykens J.A.K., "Reweighted Stochastic Learning", Neurocomputing, 1st revision (minor).
- Jumutc V., Suykens J.A.K., "Regularized and Sparse Stochastic K-Means for Distributed Large-Scale Clustering", IEEE BigData (BigData 2015), Submitted.
- Jumutc V., Suykens J.A.K., "Multi-Class Supervised Novelty Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 12, Dec. 2014, pp. 2510 - 2523.
- Mall R., Jumutc V., Langone R., Suykens J.A.K., "Representative Subsets For Big Data Learning using kNN graphs", in Proc. of the IEEE BigData (BigData 2014), Washington DC, U.S.A., Oct. 2014, pp. 37-42.
- Jumutc V., Suykens J.A.K., "New Bilinear Formulation to Semi-Supervised Classification Based on Kernel Spectral Clustering", in Proc. of the 2014 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2014), Orlando, Florida, Dec. 2014, pp. 41-47.
- Jumutc V., Suykens J.A.K., "Reweighted l2-Regularized Dual Averaging Approach for Highly Sparse Stochastic Learning", in Proc. of the 11th International Symposium on Neural Networks (ISNN 2014), Hong-Kong Makao, People's Republic of China , Nov. 2014, pp. 232-242.
- Jumutc V., Suykens J.A.K., "Reweighted l1 Dual Averaging Approach for Sparse Stochastic Learning", in Proc. of the 22th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2014), Bruges, Belgium, Apr. 2014, pp. 1 - 6.
- Jumutc V., Suykens J., "Weighted Coordinate-Wise Pegasos", in Proc. of the 5th International Conference on Pattern Recognition and Machine Intelligence (PREMI 2013), Kolkata, India, Dec. 2013, pp. 262-269.
- Jumutc V., Huang X., Suykens J.A.K., "Fixed-Size Pegasos for Hinge and Pinball Loss SVM", in Proc. of the 2013 International Joint Conference on Neural Networks (IJCNN 2013), Dallas, USA, Aug. 2013, pp. 1122-1128.
- Jumutc V., Suykens J.A.K., "Supervised Novelty Detection", in Proc. of the IEEE Symposium Series on Computational Intelligence (SSCI 2013), Singapore, Singapore, Apr. 2013, pp. 143 - 149.
- Jumutc V., Suykens J.A.K., "SALSA: Software Lab for Advanced Machine Learning and Stochastic Algorithms in `julia`", *J. Mach. Learn. Res.* (software section), to be submitted.

References I



V. Jumutc and J. A. K. Suykens

Multi-Class Supervised Novelty Detection.
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 12, Dec. 2014, pp. 2510–2523.



V. Jumutc and J. A. K. Suykens.

New Bilinear Formulation to Semi-Supervised Classification Based on Kernel Spectral Clustering.
in *Proc. of the 2014 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2014)*, Orlando, Florida, Dec. 2014, pp. 41–47.



V. Jumutc and J. A. K. Suykens

Fixed-Size Pegasos for Hinge and Pinball Loss SVM.
in *Proc. of the 2013 International Joint Conference on Neural Networks (IJCNN 2013)*, Dallas, USA, Aug. 2013, pp. 1122–1128.



V. Jumutc and J. A. K. Suykens

Weighted Coordinate-Wise Pegasos.
in *Proc. of the 5th International Conference on Pattern Recognition and Machine Intelligence (PREMI 2013)*, Kolkata, India, Dec. 2013, pp. 262–269.



A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro.

Robust stochastic approximation approach to stochastic programming.
SIAM J. on Optimization, 19(4):1574–1609, January 2009.



S. Shalev-Shwartz, Y. Singer and N. Srebro.

Pegasos: Primal Estimated sub-GrAdient SOlver for SVM.
In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 807–814, New York, NY, USA, 2007.

References II



C. Williams and M. Seeger.

Using the Nyström method to speed up kernel machines.
In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.



J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle.

Least Squares Support Vector Machines.
World Scientific, Singapore, 2002.



L. Xiao.

Dual averaging methods for regularized stochastic learning and online optimization.
J. Mach. Learn. Res., 11:2543–2596, Dec. 2010.



Y. Nesterov.

Primal-dual subgradient methods for convex problems.
Mathematical Programming, 120(1):221–259, 2009.



V. Jumutc and J. A. K. Suykens.

Reweighted l_2-regularized dual averaging approach for highly sparse stochastic learning.
In Proceedings of ISNN 2014, Hong Kong and Macao, China, Nov 28 – Dec 1, 2014, pp. 232–242.



V. Jumutc and J. A. K. Suykens.

Reweighted l_1 dual averaging approach for sparse stochastic learning.
In Proceedings of ESANN 2014, Bruges, Belgium, April 23 – 25, 2014.



J. Duchi, E. Hazan, Y. Singer.

Adaptive subgradient methods for online learning and stochastic optimization.
J. Mach. Learn. Res. 12 (2011) 2121–2159.

Thank you for attention!