SVM online learning

...

Salvatore Frandina
salvatore.frandina@gmail.com

Department of Information Engineering, Siena, Italy

Siena, August 19, 2012

Summary

An overview of the presentation:

- Online learning problem
- State of art of the online learning algorithms for SVM
- Remarks

Classification of the learning problem

- The learning problem can be divided in three categories:
  - Batch or exact methods
  - Online or approximate methods
  - Semi-online (semi-batch) methods
- **Batch methods**: the algorithm has a fixed collection of examples and uses them to construct an hypothesis which is used also for classification without further modification [G. Cauwenberghs(2000), F. Orabona(2010)]
- **Online methods**: the algorithm continually modifies its hypothesis during its use; each time it receives a pattern, predicts its classification, finds out the correct classification, and possibly updates its hypothesis accordingly [G. Cauwenberghs(2000), F. Orabona(2010)]
- **Semi-online (semi-batch) methods**: an algorithm of this kind tries to combine the advantages and to reduce the drawbacks of the previous approaches

Introduction of learning problem I

- In **batch learning** a training set i.e. a collection of (sample, label) pairs drawn from an unknown probability distribution, is given in advance: the task is to find a function such that its sign best determines the label of any future sample drawn from the same distribution [N. Cristianini(2000)]
- In **online learning**, samples and labels are available in time, so that no knowledge of the training set can be assumed a priori. The function must be built incrementally every time a new sample is available. This operation is called a round or trial [N. Cristianini(2000)]
- NOTE: in general is possible to convert a batch algorithm in online algorithm and viceversa [Yi Li(1999)]
- NOTE: for batch algorithms updating the model often means re-training from scratch [G. Cauwenberghs(2000), F. Orabona(2010)]

Introduction of learning problem II

- **Many real-life machine learning problems are instrisecally online rather than batch learning problems**. In fact, the data is often collected continuously in time as well the concepts to be learned may also evolve in time
- NOTE: in online learning there is no distinction between a training and a testing phase: learning proceeds sequentially and the knowledge is continuously exploited and updated [G. Cauwenberghs(2000), F. Orabona(2010)]
- NOTE: the application of batch learning to vision doesn't make much sense since many computer vision problems are intrinsically sequential [M. Muneeb Ullah(2009), A. Pronobis(2010)]

Online learning versus batch learning algorithms

- **Online learning algorithms**:
  - * advantages: incremental learning, scalable complexity and bounded requirement (time and memory)
    [G. Cauwenberghs(2000), F. Orabona(2010)]
  - * drawbacks: convergence to suboptimal solution and consequently loss of classification accuracy [G. Cauwenberghs(2000), F. Orabona(2010)]
- **Batch learning algorithms**:
  - * advantages: optimal solution therefore optimal classification performance, well known bound of number of mistakes [S. Shalev-Shwartz(2006)]
  - * drawbacks: inefficient management of the computational resources in term of time and memory, sometimes unusable
    [G. Cauwenberghs(2000), F. Orabona(2010)]

Online learning problem and SVM

- **The online scenario** e.g. an autonomous agent, **is difficult task for SVMs** since the **size of an SVM solution grows linearly with the number of training samples** taken into account [G. Cauwenberghs(2000), F. Orabona(2010)]; moreover the training procedure of an SVM requires solving a quadratic programming problem in a number of coefficient equal to number of training samples
- Any real system has access to finite resources, therefore in necessary a strategy to limit the number of data points i.e. a trade-off of the accuracy cannot be avoided
- **The goals of a online SVM algorithm** are:
    * quasi optimal solution: approximately converges to the batch SVM solution
    * limited resources and high speed (real time): limited computational power both for training and test phases

Learning algorithms for SVM

- **The standard SVM algorithm is originally thought to be used in batch setting**; to extend it to the online setting have been proposed two different main approaches:
    - **Online algorithms**: exact methods [G. Cauwenberghs(2000)] or approximate methods [J. Kivinen(2001), A. Bordes(2005)] that incrementally update the solution
    - **Semi-online (semi-batch) algorithms**: the batch algorithm is adapted to examine one sample at the time and produce a new approximate solution [N. Syed(1999), C. Domeniconi(2001)]
- NOTE: in both cases the continuous flow of training samples of the online setting will cause an explosion of the number of support vectors [N. Cristianini(2000)]
- NOTE: the SVM is a batch algorithm, while the Perceptron is an online algorithm. In batch settings, the SVM is typically slower than the Perceptron algorithm, but generalizes better while in the online setting the Perceptron algorithm is more suitable [Yi Li(1999)]

Online algorithms

- **The solution of the problem can be obtained by training the SVM incrementally on new data** in two main ways:
    - * exact solution: maintaining the Kuhn-Tucker (KT) conditions on all seen data, while updating the solution [G. Cauwenberghs(2000), D. Tax(2003)]
    - * approximate solution: optimizing via stochastic gradient descent within a feature space of Reproducing Kernel Hilbert Space (RKHS) [J. Kivinen(2001)] or through an online SMO-like algorithm [A. Bordes(2005)]

Online algorithms: incremental decremental technique

- The most important approach of this category is the **incremental decremental SVM** [G. Cauwenberghs(2000)], it constructs the solution recursively one point at a time
- **The key idea** is to maintain the Kuhn-Tucker (KT) conditions on all previously seen data, while "adiabatically" adding new data point to the solution i.e. the support vectors coefficients change value during each incremental step to keep the KT conditions satisfied
- NOTE: the incremental procedure is reversible, and decremental "unlearning" of each training sample produces a leave-one-out estimate of generalization performance on the training set
    - **advantages**: it allows to construct the exact solution in online fashion, moreover the learning procedure is reversible and has a geometric interpretation
    - **drawbacks**: it is no efficient since it needs to store the support vectors and to build recursively a matrix related to the Gram's matrix

Summary

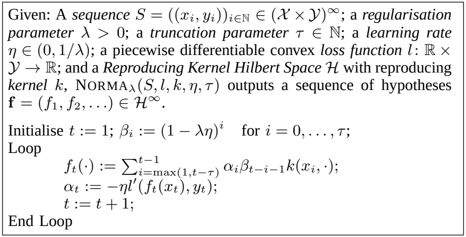
Online learning problem

State of art of the online learning algorithms for SVM

Remarks

Online algorithms: stochastic gradient descent on RKHS I

- The unique remarkable approach of this category is the *Naive Online R_{reg} Minimisation Algorithm* (NORMA) [J. Kivinen(2001)]
- **The key idea** is to perform the classical stochastic gradient descent on the soft margin loss function i.e. within the features space in a RKHS: it uses the instantaneous regularized risk that is the stochastic approximation of the standard regularized risk of batch algorithm $R_{inst,\lambda}(f) = R_{reg,\lambda}(f) = R_{emp}(f) + \frac{\lambda}{2}||f||_H^2$



Given: A *sequence* $S = ((x_i, y_i))_{i\in\mathbb{N}} \in (\mathcal{X}\times\mathcal{Y})^\infty$; a *regularisation parameter* $\lambda > 0$; a *truncation parameter* $\tau \in \mathbb{N}$; a *learning rate* $\eta \in (0, 1/\lambda)$; a piecewise differentiable convex *loss function* $l\colon \mathbb{R}\times\mathcal{Y}\to\mathbb{R}$; and a *Reproducing Kernel Hilbert Space* $\mathcal{H}$ with reproducing *kernel* k, $\text{NORMA}_\lambda(S, l, k, \eta, \tau)$ outputs a sequence of hypotheses $\mathbf{f} = (f_1, f_2, \ldots) \in \mathcal{H}^\infty$.

```
Initialise $t := 1$; $\beta_i := (1 - \lambda\eta)^i$   for $i = 0, \ldots, \tau$;
Loop
    $f_t(\cdot) := \sum_{i=\max(1,t-\tau)}^{t-1} \alpha_i\beta_{t-i-1}k(x_i, \cdot)$;
    $\alpha_t := -\eta l'(f_t(x_t), y_t)$;
    $t := t + 1$;
End Loop
```

Figure: $NORMA_\lambda$ with constant learning rate η and truncation approximation

Online algorithms: stochastic gradient descent on RKHS II

- **The regularization parameter λ allows to realize the forgetting mechanism** for old examples and consequently to control the storage requirements
- It is simple and efficient algorithm for classification, regression and novelty detection. The authors [J. Kivinen(2001)] provide a theoretical proof of the convergences rates and error bounds: the average instantaneous risk converges towards the minimum regularised risk at rate $O(m^{-\frac{1}{2}})$ where m is the number of examples
- NOTE: among the large margin classifiers, the NORMA is quite similar to *Approximate Large Margin Algorithm* (ALMA) [Gentile(2001)]
  - **advantages**: it is based on the theory of RKHS, it is simple and efficient
  - **drawbacks**: the bounds provided by theoretical analysis are not close, the performance analysis is not exhaustive

Summary

Online learning problem

State of art of the online learning algorithms for SVM

Remarks

Online algorithms: LASVM

- **The LASVM algorithm** [A. Bordes(2005)] is based on the traditional soft margin SVM formulation: it is an online kernel classifier similar to *Sequential Minimization Optimization* (SMO) [Platt(1999)] and therefore converges to the solution of the SVM problem
- The algorithm operates alternating two phases:
  - Process step: the algorithm processes a new example and try to add it to the set of support vectors
  - Reprocess step: the algorithm tries to reduce the number of support vectors in the current model
- NOTE: the authors show that active example selection can improve the training speed and the accuracy using only a fraction of the training example labels
  - **advantages**: it is efficient and flexible, it achieves the performance of batch SVM
  - **drawbacks**: it requires an initialization step to build the initial model

Semi-online algorithms

- The approximate solution of the problem can be obtained by **training the SVM incrementally on new data and discarding all previous data except their support vectors**
- NOTE: in general, these methods achieve classification performances equivalent to batch SVM, but the number of support vectors tends to grow proportionally to the number of incremental steps
  [A. Pronobis(2010)]

Semi-online algorithms: fixed-partition technique I

- In the **fixed-partition technique** [N. Syed(1999)], the training data set is partitioned in batches of fixed size and then the SVM is incrementally trained on them preserving the support vectors between the steps
- The **key idea** is related on the ability of SVM to represent a set of data through the corresponding support vectors; the experimental results show that **the model obtained is statistically equivalent to the model obtained using all the data together**

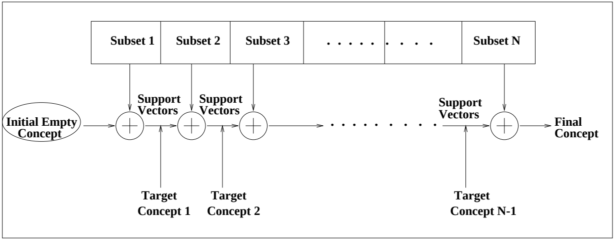


Figure: The incremental training procedure

Semi-online algorithms: fixed-partition technique II

- NOTE: on each new batches, the existing support vectors are updated to generate the new model of the classifier
- NOTE: the method is similar to chunking decomposition technique [N. Cristianini(2000)]
    - **advantages**: it is simple and efficient, use the standard SVM algorithm with few modifications
    - **drawbacks**: it needs to store the support vectors, it allows to construct only the approximate (exact) solution, it requires an initialization step to build the initial model in the online setting

Semi-online algorithms: error-driven (exceeding-margin) technique

- The **error-driven (or exceeding-margin) technique** [C. Domeniconi(2001)] is opposed to the fixed-partition method
- This method makes a filtering on the new data at each incremental step: at given step the model of the classifier classifies the new data (or check if it exceed the margin). If the data is misclassified (or exceed the margin) it is kept, otherwise it is discarded. The support vectors of the model together with the misclassified points (or exceeded margin points), are used as training data to obtain the new model of the classifier
    - **advantages**: it is simple and efficient, use the standard SVM algorithm with few modifications
    - **drawbacks**: it needs to store the support vectors, allows to construct only the approximate (exact) solution, it requires an initialization step to build the initial model in the online setting

Semi-online algorithms: memory-controlled technique I

- NOTE: both previous techniques, after an initialization step on initial training data, can be used in the online setting where the data may arrive continuously
- NOTE: common problem to both incremental technique is that in principle there is **no limitation to the memory growth**
- In literature there are several approaches that use different heuristics to reduce memory requirements during incremental process:
  - Random forget
  - Least significant support vector forget
  - Oldest support vector forget
  - Linear dependence forget
- All the previous methods use a fixed amount of memory to training the SVM classifier; the threshold for the amount memory can be specified by a parameter of the algorithm and allows also to reduce redundant data

Semi-online algorithms: memory-controlled technique II

- **Random forget** [M. Muneeb Ullah(2009)]: it discards in randomly way a support vectors of the current model only if the performance of the classifier does not decay
- **Least significant support vector forget** [D. Tax(2003)]: this discard the least relevant support vector i.e. the support vector with small value of the weight
- **Oldest support vector forget** [C. Domeniconi(2001)]: this removes the oldest support vector of the current model; although this is useful for applications in which the distribution of the data changes over the time, at the same time it has the same result as if the learning has been done only on the last recent data
- **Linear dependence forget** [A. Pronobis(2010), F. Orabona(2010)]: is a simplification method that preserves only the support vectors which cannot be expressed as linear combination of others in the feature space. The excessive accumulation of the support vectors is controlled by a parameters that realize a trade-off between the classification accuracy and memory requirement

Semi-online algorithms: memory-controlled technique III

- **The previous methods simplify the solution after each update and this can be be extremely onerous**
- Another idea is the *On-line Independent SVM* (OISVM) [F. Orabona(2010)] that suggests to optimize the SVM explicitly selecting as basis vectors only independent vectors in the feature space. Hence, it directly builds the solution with a small number of basis vectors
- NOTE:
  - the process of reducing the number of support vectors can be viewed as sliding window on the training data [C. Domeniconi(2001)]; it removes potential support vectors that cannot be recovered at a later stage
  - in a given problem no more points than the number of support vectors are needed, but the number is not known in advance
  - the reduction rate of support vectors grows with the dimension of the training set while the classification rate decreases monotonically with number of support vectors [A. Pronobis(2010)]

Other approaches I

- SVM combines the kernel trick with the large margin idea [N. Cristianini(2000)]
- In literature there are **several online learning algorithms that exploit the large margin idea** without a direct relationship with the SVM:
  - *Relaxed Online Maximum Margin Algorithm* (ROMMA) [Yi Li(1999)]: repeatedly chooses the hyperplane that classifies previously seen examples correctly with the maximum margin
  - *Margin Infused Relaxed Algorithm* (MIRA) [K. Crammer(2001)]: it maximizes the margin between the correct prediction and other possible predictions. It works with binary and multi classification separable problem: in the case of binary problem it very similar to the Perceptron algorithm [K. Crammer(2001)]
  - *Passive-Aggressive* (PA) algorithm [K. Crammer(2006)]: since the margin of an example is proportional to the distance between the instance and the hyperplane, the algorithm uses the margin to modify the current classifier
  - SMO algorithm [Platt(1999)]: it is a method for solving convex quadratic optimization problem and it based on the idea to iterativley solve subproblems of size two; though it wasn't developed as an online algorithm, it is strictly related with the Perceptron algorithm [J. Kivinen(2001)]

Other approaches II

- NOTE: some approaches use **the mistake bound model** for analyzing the algorithms [K. Crammer(2001)]. On each round the algorithms get a new example and make a prediction of it. Then they receive the correct label and update their predication rule in case they made a prediction error. The goal of the algorithms is to minimize the number of mistakes they made compared to the minimal number of errors that a batch algorithms can achieve

Online algorithms for linear predictors I

- In literature [J. Kivinen(1997)], there are two important family of online algorithms for linear predictors:
  - (Stochastic) *Gradient Descent* (GD): it uses additive updating of the weights
  - *Exponentiated Gradient* (EG): it uses multiplicative updating of the weights
- The **GD is related to the Perceptron algorithm** while the **EG is related to the Winnow algorithm** [J. Kivinen(1997)] and it is analogous to the Weighted Majority Algorithm [J. Kivinen(1997)]
- NOTE: the algorithms are studied in the case of linear predictors but they can be used in the more general context of SVM and kernel methods

Online algorithms for linear predictors II

- The general update rule after receiving the t-th instance x_t and making its prediction $\hat{y}_t = w_t \cdot x_t$ is:
  - **GD** $w_{t+1,i} = w_{t,i} - \eta L^{'}_{y_t}(\hat{y}_t)x_{t,i}$: it derives from the gradient of the distance $d(w,s)$ e.g. Euclidean distance $||w-s||_2$
  - **EG** $w_{t+1,i} = \frac{w_{t,i}r_{t,i}}{\sum_{j=1}^{N} w_{t,j}r_{t,j}}$ with $r_{t,i} = exp(-\eta L^{'}_{y_t}(\hat{y}_t)x_{t,i})$: it derives from the gradient of the relative entropy (Kullback-Leibler divergence) $d_{re}(w,s) = \sum_{i=1}^{N} w_i ln \frac{w_i}{s_i}$
- The GD performs better when the input is consistent i.e. few or no errors and almost all the input variables are relevant while EG is better in error prone learning environments; moreover it needs of few variables for prediction [J. Kivinen(1997)]
- NOTE: the EG can be used only with positive weights $w_{t,i}$ and under the normalization constraints $\sum_{j=1}^{N} w_{t,j} = 1$. There exists an extension EG^{+-} that works with positive and negative weights

Online algorithms for linear predictors III

- In the scenario of learning with large dataset, [Bottou(2010)] highlights the **common points and the differences between the gradient descent algorithms and the stochastic gradient descent algorithms**
- The performance are analysed in term of complexity and accuracy for several algorithms including SVM: the obtained results show that although the stochastic gradient descent algorithms are worst optimization algorithms w.r.t. the gradient descent algorithms, they need less time to reach a predefined level of accuracy
- NOTE: with a large dataset (or in online setting) when the limiting factor is the complexity rather than the number of examples, the stochastic gradient descent algorithms perform asympotically better

Available algorithms implementation

- The incremental decremental SVM is implemented via MATLAB package [Diehl(2006)]
- The most of the previous algorithms (Perceptron, Passive-Aggresive, ALMA, NORMA) are available via DOGMA MATLAB toolbox [Orabona(2009)]
- An implementation of the NORMA algorithm is also available in the Kernlab R package (C++/MATLAB) [ker(2012)]
- The implementation of LASVM (C/C++) is obtainable from [Bottou(2009)]

Remarks I

- The online learning is an open research problem; especially for SVMs
- There is **need of strong theoretical bounds on number of mistakes** to measure the quality of the online learning algorithms (experiments are not sufficient)
- In the online learning framework is very important to handle:
  - * learning from experience (manage old knowledge)
  - * learning form continuous flow of incoming data (never ending learning)
- In the online learning framework is fundamental to get:
  - * quasi optimal solution (optimal performance)
  - * real time operations (limited resources and high speed)

Remarks II

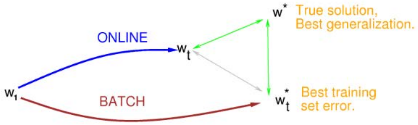


Figure: Online learning versus batch learning algorithms

- In the online setting, the stochastic gradient descent algorithms directly optimize the expected risk instead the empirical risk, since the examples are randomly drawn form the ground truth distribution
- **The online algorithms are in general worse optimization algorithms than the batch algorithms, but the performance depends on the distance between the found solution and the actual solution**

Remarks III

- **The choice of the best method is problem** (and available resources) **dependent**
- Some algorithms require an initialization step to work in online setting (LASVM, memory-controlled technique)
- Some algorithms (incremental decremental SVM, NORMA) are data dependent: they may produce different model if they receive two sequences of examples with the same examples in different order

References I

The kernlab R-package, 2012.
URL http://www.cran.r-project.org/.

J. Weston L. Bottou A. Bordes, S. Ertekin.
Fast Kernel Classifiers with Online and Active Learning.
Journal of Machine Learning Research (JMLR), 2005.

B. Caputo A. Pronobis, J. Luo.
The More you Learn, the Less you Store: Memory-Controlled Incremental SVM for Visual Place Recognition.
Image and Vision Computing (IMAVIS), 2010.

L. Bottou.
LASVM, 2009.
URL http://leon.bottou.org/projects/lasvm.

L. Bottou.
Large-Scale Machine Learning with Stochastic Gradient Descent.
Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT), 2010.

References II

D. Gunopulos C. Domeniconi.
Incremental Support Vector Machine Construction.
In Proceedings of the International Conference on Data Mining (ICDM), 2001.

P. Laskov D. Tax.
Online SVM learning: from classification to data description and back.
IEEE XIII Workshop on Neural Networks for Signal Processing - NNSP, 2003.

C. Diehl.
Incremental SVM learning, 2006.
URL http://www.cpdiehl.org/code.html.

B. Caputo Jie Luo-G. Sandini F. Orabona, C. Castellini.
On-line Independent Support Vector Machines.
Pattern Recognition - PR, 2010.

T. Poggio G. Cauwenberghs.
Incremental and Decremental Support Vector Machine Learning.
Neural Information Processing Systems - NIPS, 2000.

References III

C. Gentile.
A new Approximate Maximal Margin Classification Algorithm.
Journal of Machine Learning Research - JMLR, 2001.

M. Warmuth J. Kivinen.
Exponentiated Gradient versus Gradient Descent for Linear Predictors.
Information and Computation/information and Control - IANDC, 1997.

R. Williamson J. Kivinen, A. Smola.
Online Learning with Kernels.
Neural Information Processing Systems - NIPS, 2001.

J. Keshet S. Shalev-Shwartz Y. Singer K. Crammer, O. Dekel.
Online Passive-Aggressive Algorithms.
Journal of Machine Learning Research - JMLR, 2006.

Y. Singer K. Crammer.
Ultraconservative Online Algorithms for Multiclass Problems.
Computational Learning Theory - COLT, 2001.

References IV

B. Caputo M. Muneeb Ullah, F. Orabona.
You Live, you Learn, you Forget: Continuous Learning of Visual Places with a Forgetting Mechanism.
International Conference on Intelligent Robots and Systems - IROS, 2009.

J. Taylor N. Cristianini.
An introduction to Support Vector Machines and other Kernel-based Learning Methods.
Cambridge University Press, 2000.

L. Kah K. Sung N. Syed, S. Huan.
Incremental Learning with Support Vector Machines.
1999.

F. Orabona.
DOGMA: a MATLAB toolbox for online learning, 2009.
URL `http://dogma.sourceforge.net`.

References V

J. Platt.
Fast training of Support Vector Machines using Sequential Minimal Optimization.
Advances in Kernel Methods - Support Vector Learning, 1999.

Y. Singer S. Shalev-Shwartz.
Online Learning meets Optimization in the Dual.
Proceedings of the Nineteenth Annual Conference on Computational Learning Theory, 2006.

P. Long Yi Li.
The Relaxed Online Maximum Margin Algorithm.
Neural Information Processing Systems - NIPS, 1999.