



A Machine Learning (Theory) Perspective on Computer Vision

Peter Auer

Montanuniversität Leoben



Outline

- What I am doing and how computer vision approached me (in 2002).
- Some modern machine learning algorithms used in computer vision, and their development:
 - Boosting
 - Support Vector Machines
- Concluding remarks

My background

- COLT 1993
 - Conference on Learning Theory
 - „On-Line Learning of Rectangles in Noisy Environments“
- FOCS 1995
 - Symp. Foundations of Computer Science
 - „Gambling in a Rigged Casino: The Adversarial Multi-Arm Bandit Problem“
 - with N. Cesa-Bianchi, Y. Freund, R. Schapire
- ICML, NIPS, STOC, ...

A computer vision project

- EU-Project LAVA, 2002
 - “Learning for adaptable visual assistants”
 - XRCE: Ch. Dance, R. Mohr
 - IRIA Grenoble: C. Schmid, B. Triggs
 - RHUL: J. Shawe-Taylor
 - IDIAP: S. Bengio



LAVA Proposal

- Vision (goals)
 - Recognition of generic objects and events
 - Attention Mechanisms
 - Base line and high-level descriptors
- Learning (means)
 - Statistical Analysis
 - Kernels and models and features
 - Online Learning



Online learning

- Online Information Setting
 - An input is received, a prediction is made, and then feedback is acquired.
 - Goal: To make good predictions, in respect to a (large) set of fixed predictors.
- Online Computation Setting
 - The amount of computation per new example – to update the learned information – is constant (or small).
 - Goal: To be fast computationally.
- (Near) real-time learning?

Learning for vision around 2002

- Viola, Jones, CVPR 2001:
 - Rapid object detection using a boosted cascade of simple features. ([Boosting](#))
- Agarwal, Roth, ECCV 2002:
 - Learning a Sparse Representation for Object Detection. ([Winnow](#))
- Fergus, Perona, Zisserman, CVPR 2003:
 - Object class recognition by unsupervised scale-invariant learning. ([EM-type algorithm](#))
- Wallraven, Caputo, Graf, ICCV 2003:
 - Recognition with local features: the kernel recipe. ([SVM](#))



Our contribution in LAVA

- Opelt, Fussenegger, Pinz, Auer, ECCV 2004:
 - Weak hypotheses and boosting for generic object detection and recognition.



Image classification as a learning problem

Image classification as a learning problem

- ▶ Images are represented as vectors $x = (x_1, \dots, x_n) \in X \subset \mathbf{R}^n$.
- ▶ Given
 - ▶ training images $x^{(1)}, \dots, x^{(m)} \in X$
 - ▶ with their classifications $y^{(1)}, \dots, y^{(m)} \in Y = \{-1, +1\}$,a classifier $H : X \rightarrow Y$ is learned.
- ▶ We consider linear classifiers H_w , $w \in \mathbf{R}^n$,

$$H_w(x) = \begin{cases} +1 & \text{if } w \cdot x \geq 0 \\ -1 & \text{if } w \cdot x < 0 \end{cases}$$

$$(w \cdot x = \sum_{i=1}^n w_i x_i).$$

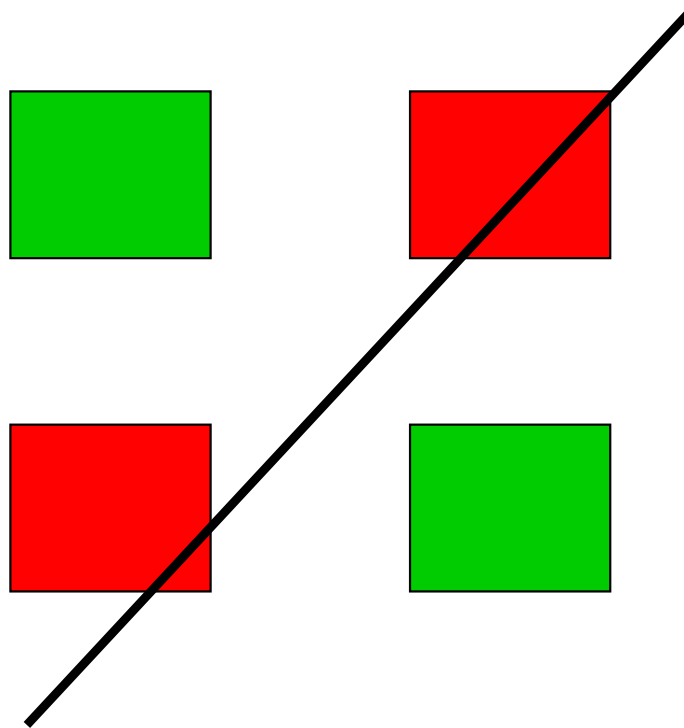
The Perceptron algorithm (Rosenblatt, 1958)

The Perceptron algorithm maintains a weight vector $w^{(t)}$ as its current classifier.

- ▶ Initialization $w^{(1)} = \mathbf{0}$.
- ▶ Predict $\hat{y}^{(t)} = \begin{cases} +1 & \text{if } w^{(t)} \cdot x^{(t)} \geq 0 \\ -1 & \text{if } w^{(t)} \cdot x^{(t)} < 0 \end{cases}$
- ▶ If $\hat{y}^{(t)} = y^{(t)}$ then $w^{(t+1)} = w^{(t)}$,
else $w^{(t+1)} = w^{(t)} + \eta y^{(t)} x^{(t)}$.
(η is the learning rate.)

- ▶ The Perceptron was abandoned in 1969, when Minsky and Papert showed that Perceptrons are not able to learn some simple functions.
- ▶ Revived only in the 1980's when neural networks became popular.

Perceptron cannot learn XOR



- No single line can separate the green from the red boxes.

- ▶ Extending the feature space (or using kernels) prevents the problem:
- ▶ Since XOR is a quadratic function, use $(1, x_1, x_2, x_1^2, x_2^2, x_1x_2)$ instead of (x_1, x_2) .
- ▶ For $x_1, x_2 \in \{+1, -1\}$,

$$x_1 \text{ XOR } x_2 = x_1x_2.$$

Winnow (Littlestone 1987)

- ▶ Works like the Perceptron algorithm except for the update of the weights:

$$w_i^{(t+1)} = w_i^{(t)} * \exp\left(\eta y^{(t)} x_i^{(t)}\right)$$

for some $\eta > 0$. ($w^{(1)} = \mathbf{1}$.)

- ▶ Observe the multiplicative update of the weights and $\log w_i^{(t+1)} = \log w_i^{(t)} + \eta y^{(t)} x_i^{(t)}$.

- ▶ Very related work:
The Weighted Majority Algorithm (Littlestone, Warmuth)

Comparison of the Perceptron algorithm and Winnow

- ▶ Perceptron and Winnow scale differently in respect to relevant, used, and irrelevant attributes:

all attributes	n
relevant attributes	k
used attributes	d

	# training ex.
Perceptron	\sqrt{dk}
Winnow	$k \log n$

Adaboost (Freund, Schapire, 1995)

AdaBoost maintains weights $v_t^{(s)}$ on the training examples $(x^{(s)}, y^{(s)})$ over time t :

- ▶ Initialize weights $v_0^{(s)} = 1$.
- ▶ For $t = 1, 2, \dots$
 - ▶ Select coordinate i_t with maximal correlation with the labels, $\left| \sum_s v_t^{(s)} y^{(s)} x_i^{(s)} \right|$, as weak hypothesis.
 - ▶ Choose α_t which minimizes $\sum_s v_t^{(s)} \exp \left\{ -\alpha_t y^{(s)} x_{i_t}^{(s)} \right\}$.
 - ▶ Update $v_{t+1}^{(s)} = v_t^{(s)} \exp \left\{ -\alpha_t y^{(s)} x_{i_t}^{(s)} \right\}$.
- ▶ For $x = (x_1, \dots, x_n)$ predict $\text{sign} \left(\sum_t \alpha_t x_{i_t} \right)$.

History of Boosting (1)

- Rob Schapire:
The strength of weak learnability, 1990.
 - Showed that classifiers which are only 51% correct, can be combined into a 99% correct classifier.
 - Rather a theoretical result, since the algorithm was complicated and not practical.
 - I know people who thought that this was not an interesting result.

History of Boosting (2)

- Yoav Freund:
Boosting a weak learning algorithm by majority, 1995.
 - Improved boosting algorithm, but still complicated and theoretical.
 - Only logarithmically many examples are forwarded to the weak learner!

History of Boosting (3)

- Y. Freund and R. Schapire:
A decision-theoretic generalization of on-line learning and an application to boosting, 1995.
 - Very simple boosting algorithm, easy to implement.
 - Theoretically less interesting.
 - Performs very well in practice.
- Won the Gödel price in 2003 and the Kanellakis price in 2004. (Both are prestigious prices in Theoretical Computer Science.)
- Since then many variants of Boosting (mainly to improve error robustness):
 - BrownBoost, Soft margin boosting, LPBoost.



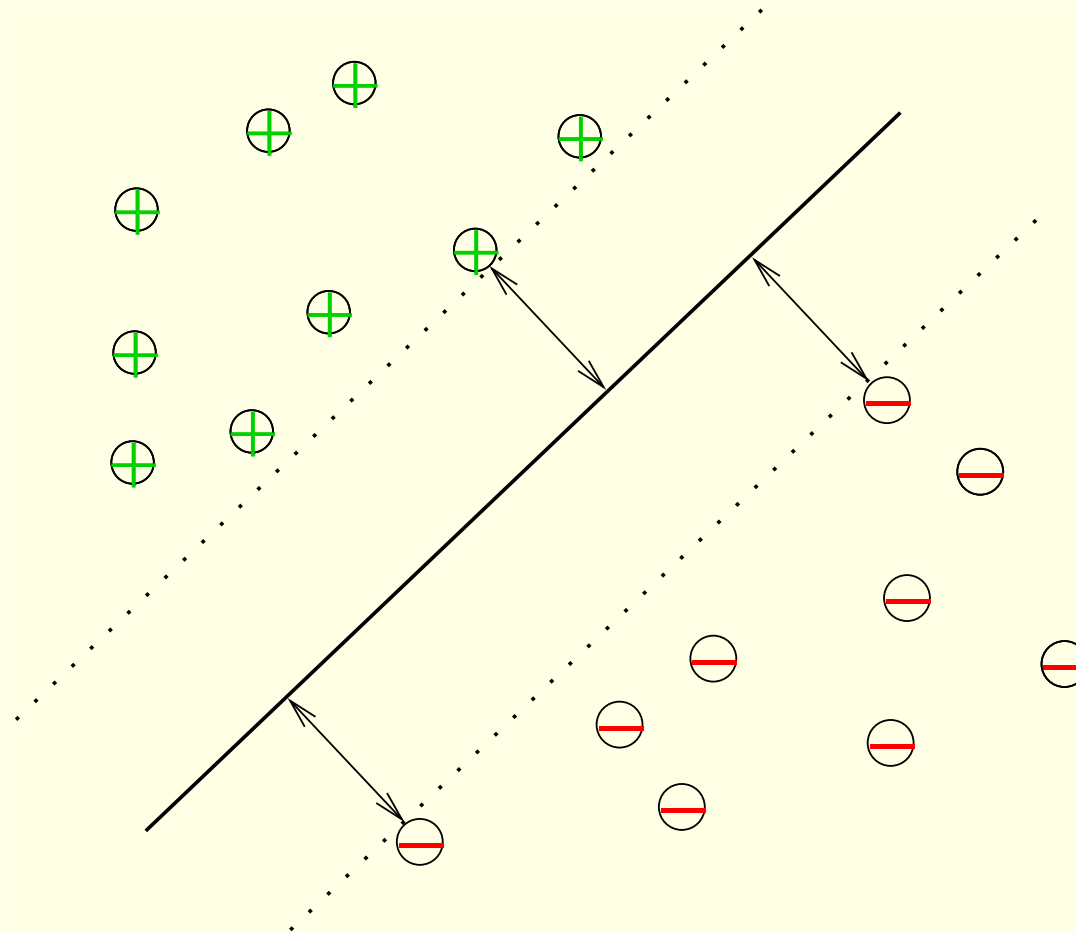
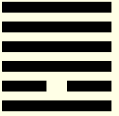
Support Vector Machines (SVMs)

- In its vanilla version also learns a linear classifier.
- It maximizes distance between the decision boundary and the nearest training points.
 - Formulates learning as a well-behaved optimization problem.
- Invented by Vladimir Vapnik (1979, Russian paper).
 - Translated in 1982.
 - No practical applications, since it required **linear separability**.

Practical SVMs

- Vapnik:
 - The Nature of Statistical Learning Theory, 1995.
 - Statistical Learning Theory, 1998.
- Shawe-Taylor, Cristianini:
Support Vector Machines, 2000.
- Soft margin SVMs:
 - Tolerate incorrectly labeled training examples (by using slack variables).
- Non-linear classification using the “kernel trick”.

Support Vector Machines (SVMs)



The kernel trick (1)

- ▶ Recall the perceptron update,

$$w^{(t+1)} = w^{(t)} + \eta y^{(t)} x^{(t)} = \eta \sum_{\tau=1}^t y^{(\tau)} x^{(\tau)},$$

and classification,

$$\hat{y} = \text{sign} \left(w^{(t+1)} \cdot x \right) = \text{sign} \left(\sum_{\tau=1}^t y^{(\tau)} \left(x^{(\tau)} \cdot x \right) \right).$$

- ▶ A kernel function generalizes the inner product,

$$\hat{y} = \text{sign} \left(\sum_{\tau=1}^t y^{(\tau)} K \left(x^{(\tau)}, x \right) \right).$$

The kernel trick (2)

- ▶ The inner product $x^{(\tau)} \cdot x$ is a measure of similarity:
 $x^{(\tau)} \cdot x$ is maximal if $x^{(\tau)} = x$.
- ▶ The kernel function is a similarity measure in feature space,
 $K(x^{(\tau)}, x) = \Phi(x^{(\tau)}) \cdot \Phi(x)$.
- ▶ Kernel functions can be designed to capture the relevant similarities of the domain.
- ▶ Aizerman, Braverman, Rozonoer:
Theoretical foundations of the potential function method in pattern recognition learning, 1964.



Where are we going?

- New learning algorithms?
- Better image descriptors!
- Probably they need to be learned.
- Probably they need to be hierarchical.
- We need (to use) more data.



Final remark on algorithm evaluation and benchmarks

- Computer vision is in the state of machine learning 10 years ago (at least for object classification).
- Benchmark datasets start to become available, e.g. PASCAL VOC.