# BRUTE FORCE AND INTELLIGENT METHODS OF LEARNING

**Vladimir Vapnik**

**Columbia University, New York**
**Facebook AI Research, New York**

# PART 1
# BASIC LINE OF REASONING

**Problem of pattern recognition can be formulated as the basic problem of Information Theory:**

***Using $\ell$ bits of information that contains training data find in a given set of functions the desired one.***

# BASIC SETTING OF LEARNING PROBLEM



**Learning is defined as the problem of estimating a function from a given collection of functions based on finite number of observations** $(y_i, x_i),\ i = 1, ..., \ell$

**To extract the conceptual part of the problem let us simplify the setting: Consider set of *indicator functions* containing *finite number* $N$ of elements.** ( The generalization to infinite number of real valued functions is given by the VC theory is just technical result.)

# SHANNON'S IDEA OF INFORMATION



**THE PROBLEM:**

**How many questions that require answers YES or NO one has to ask Query to find the desired object among $N$ objects.**

**THE ANSWER:** $\ell = \log_2 N = \frac{\ln N}{\ln 2}$.

**One has to split $N$ objects into 2 equal subsets and ask Query is the desired object in the first subset. After Query's answer one removes subset which does not contains desired object, splits remaining part into two subsets and ask the same question.**

**After $\log_2 N$ answers of Query one find the desired object ($\log_2 N$ answers defines $\log_2 N$ bits of information).**

***This is the smallest number of questions that is guarantee to find the desired object among $N$ objects.***

# INTERPRETATION FOR PATTERN RECOGNITION



**THE QUESTION:**

**How many examples $(x_i, y_i), x \in X,\ y \subset \{0,1\}$ is required to find among $N$ indicator functions ($f(x) \subset \{0,1\}$) the desired one.**

**THE ANSWER:** $\ell = \log_2 N = \frac{\ln N}{\ln 2}$.

**One has to find vector $x_i$ for which half of functions take value $1$ and half take value $0$ and ask Query which value takes the desired indicator function for vector $x_i$. Then remove the half of functions that disagree with Query and do it $\log_2 N$ times.**

# STATISTICAL PATTERN RECOGNITION MODEL-1



**Since it is difficult to find vector $x$ that splits $N$ functions into two equal parts, consider the model where training examples**

$$x_1, ..., x_\ell$$

**are (iid) generated according to some unknown generator $P(x)$.**

**THE QUESTION:**

**How many training examples is required to find among $N$ indicator functions the one that with probability $1-\eta$ is $\varepsilon$-close to the desired function.**

**THE ANSWER:** $\ell = \frac{\ln N - \ln \eta}{\varepsilon}, \quad 0 < \varepsilon < 1$

**THIS BOUND CANNOT BE IMPROVED**

# PATTERN RECOGNITION MODEL-2

**Let among $N$ functions there is NO function that does not commit errors.**

**THE QUESTION:**

**How many training examples is required to find among $N$ indicator functions the one that with probability $1-\eta$ is $\varepsilon$-close to the *best function in the set.***

**THE ANSWER:** $\ell = \dfrac{\ln N - \ln \eta}{\varepsilon^2} \quad 0 < \varepsilon < 1$

**THIS BOUND CANNOT BE IMPROVED**

# THREE CATEGORY OF NUMBERS



**Let us distinguish three category of integers:**

1. **ORDINARY numbers $\ell$. Say $1 \le \ell \le 1,000,000$.**
   ***The number of objects we deal with during life time***
2. **BIG numbers $\mathcal{B}$. Say $\mathcal{B} = 2^{\ell}$.**
   ***The number of objects from which we can choose the desired one (using $\ell = \log_2 \mathcal{B}$ bits of Query information obtained during life time).***
3. **HUGE numbers $\mathcal{H}$. Say, $\mathcal{H} = 2^{\mathcal{B}} = 2^{2^{\ell}}$.**

***Generally speaking, using Ordinary number of observations one can NOT choose the desired function from a set with Huge numbers $\mathcal{H}$ of elements.***

# BRUTE FORCE AND INTELLIGENT LEARNING



**In classical machine learning models Teacher supplies any training vector $x$ with one bit of information $y$, generated according to some (unknown) conditional probability function $P(y|x)$. The goal is to find the desired function using order of $\log_2 N$ examples.**

***The corresponding methods we call BRUTE FORCE methods.***

**The new approach considers an Intelligent Teacher which supplies any vector $x$ with more than one bit of information $(x^*, y)$ using some (unknown) Intelligence Generator $P(x^*, y|x)$. The goal is to find the desired function using less than $\log_2 N$ examples.**

***The corresponding methods we call INTELLIGENT methods and vector $x^*$ we call the PRIVILEGED information.***

# GENERALIZATION FOR INFINITE SET: VC DIMENSION



**Let $N^F(x_1, ..., x_\ell)$ be the number of different separations of set $x_1, ..., x_\ell$ into two classes using indicator functions from the set $F$.**

**We call the *Growth-function of set* $F$ the function**

$$G(\ell) = \sup_{x_1,...,x_\ell} \log_2 N(x_1, ..., x_\ell)$$

**THEOREM (VC 1968) Growth function either linear function**

$$G(\ell) = \ell$$

**or bounded by the logarithmic function**

$$G(\ell) \leq h \ln \ell,$$

**where $h$ is the smallest $\ell$ for which $G(h) \neq h$**

**VALUE $h$ IS CALLED VC DIMENSION OF THE SET $F$.**

# PATTERN RECOGNITION (VC MODEL)



**Consider an infinite set of functions $f(x, \alpha), \alpha \in \Lambda$ which VC dimension is $VC_{dim}$.**

**QUESTION:**

**How many examples one has to see to find among these functions the function that with probability $1 - \eta$ is $\varepsilon$-close to the *best function in the set.*.**

**ANSWER:**

**$\ell \approx \frac{VC_{dim} - \ln \eta}{\varepsilon}$ if in the set of functions there exist one that does not commit errors on training set.**

**$\ell \approx \frac{VC_{dim} - \ln \eta}{\varepsilon^2}$ if in the set of functions there are NO function that does not commit errors on training set.**

**IN THESE BOUNDS $VC_{dim}$ REPLACES $\ln N$.**

# INTELLIGENT AND ADVANCED METHODS OF LEARNING



**Bounds obtained by VC theory are:**

$$\ell = O^*\left(\frac{VC_{dim}}{\varepsilon^k}\right), \quad (k = 1 \text{ or } k = 2),\ \varepsilon < 1.$$

**To improve learning rate one must either:**
**1. To improve statistical part of method increasing the denominator (obtaining $k^* < k$ in the bound) or**
**2. To decrease the numerator of bound (obtaining $h < VC_{dim}$ in the bound).**

**Methods designed to decrees the numerator is called**
***Intelligent Methods of Learning***
**Methods designed to increase the denominator is called**
***Advanced Statistical Methods of Learning***

# PART 2
# ELEMENTS OF VC THEORY

**VC theory analyze Brute Force methods of learning. It provides:**

1. **The necessary and sufficient conditions of consistency of learning processes.**
2. **The bound on convergence rate of learning processes.**
3. **Universally consistent principle of learning (Structural Risk Minimization (SRM) principle).**
4. **Efficient algorithms realizing SRM principle (SVM)**

# RISK MINIMIZATION PROBLEM



**THE PROBLEM: Given iid observation of pairs**

$$(x_1, y_1), ..., (x_\ell, y_\ell), \quad x \in X \subset R^n, \ y \subset \{0, 1\}$$

**generated according to unknown probability measure $P(x, y)$, find in the given set of functions $f(x, \alpha), \ \alpha \in \Lambda$, the one that minimizes the expected loss**

$$E(\alpha) = \int \rho(y, f(x, \alpha)) dP(x, y), \qquad \alpha \in \Lambda.$$

**THE SOLUTION: Approximate the function that minimizes expected loss $f(x, \alpha_0)$ with the function that minimizes empirical loss:**

$$E_\ell(\alpha) = \frac{1}{\ell} \sum_{i=1}^{\ell} \rho(y_i, f(x_i, \alpha)).$$

**THE THEORY: Answers the question: When this works?**

# VC-THEORY OF EMPIRICAL RISK MINIMIZATION



**THEOREM 1. For consistency of ERM principle convergence**

$$\lim_{\ell\to\infty} P\{\sup_{\alpha\in\Lambda} |E(\alpha) - E_\ell(\alpha)| \geq \varepsilon\} = 0, \quad \forall\varepsilon > 0.$$

**(*the uniform law of large numbers*) is necessary and sufficient.**

**THEOREM 2. For existence of uniform convergence independently of probability measure $P(x, y)$, it is necessary and sufficient that set $r(x, \alpha), \alpha \in \Lambda$ has a finite VC-dimension (to be explained).**

**THEOREM 3. For existence of uniform convergence given probability measure $P(x, y)$, it is necessary and sufficient that set $r(x, \alpha), \alpha \in \Lambda$ has a VC-entropy (to be explained) which ratio to the number of observations converges to zero.**

**THEOREM 4. If ratio of VC-entropy to the number of observations converges to $\mu > 0$, then there exist $X^* \subset X$, $P(X^*) = \mu$ such that almost any $\ell$ examples from $X^*$ can be shattered.**

# VC BOUNDS

**THEOREM 5. Let VC dimension of the set of indicator functions $f(x,\alpha), \alpha \in \Lambda$ be $h$. Then with probability $1-\eta$ simultaneously for all rules $f(x,\alpha), \alpha \in \Lambda$ the inequality**

$$E(\alpha) \le E_\ell(\alpha) + C\sqrt{\frac{h\ln\ell - \ln\eta}{\ell}}, \quad \forall\alpha \in \Lambda$$

**holds true.**

**If among rules $f(x,\alpha), \alpha \in \Lambda$ there exist rules $f(x,\alpha), \alpha \in \Lambda^*$ such that $E_\ell(\alpha) = 0, \alpha \in \Lambda$, then the better bound**

$$E(\alpha) \le 0 + C\frac{h\ln\ell - \ln\eta}{\ell}, \quad \forall\alpha \in \Lambda^*$$

**holds true.**

**Both bounds are achievable.**

# STRUCTURAL RISK MINIMIZATION (SRM) PRINCIPLE.

**Let a structure $S$ on a set of rules $f(x,\alpha), \alpha \in \Lambda$ be defined**

$$\Lambda_1 \subset \Lambda_2 \subset ... \subset \Lambda_n \subset ...$$

**such that VC dimension of the rules of subset are**

$$h_1 < h_2 < ... < h_n < ...$$

**SRM PRINCIPLE: Given set of rules $f(x,\alpha), \alpha \in \Lambda$ and the structure $S$ on it find, for any fixed number $\ell$ of observations, the function $f(x,\alpha_\ell)$ that minimizes the r.h.s. of VC bound**

$$W(\alpha) = E_\ell(\alpha) + C\sqrt{\frac{h_k \ln \ell - \ln \eta}{\ell}}, \quad \forall \alpha \in \Lambda_k$$

**choosing both the element $k$ of the structure $S$ and the best rule belonging to this element $f(x,\alpha_\ell), \alpha \in \Lambda_k$.**

# CORE MECHANISM OF LEARNING: SRM PRINCIPLE



**Let $S_k$ be elements of the structure and**

$$\overline{S} = \overline{\cup_{k=1}^{\infty} S_k}$$

**be the closure of this structure. Let the minimum of risk be achieved on the element $f(x, \alpha_0)$ of the closure.**

**Consider the sequence of solutions of SRM method**

$$f(x, \alpha_1), f(x, \alpha_2), ..., f(x, \alpha_\ell), ...$$

**obtained using samples of different size $\ell$.**

**THEOREM 6. With increasing number of observations $\ell$, sequence $f(x, \alpha_\ell)$ converges uniformly to $f(x, \alpha_0)$ with probability one:**

$$P\{\sup_x |f(x, \alpha_0) - f(x, \alpha_\ell)| \longrightarrow_{\ell \to \infty} 0\} = 1.$$

# STRUCTURE IN HILBERT SPACES

**Let vectors $x_i \in R^n$ is mapped into Hilbert space $z_i \in Z$**

**THEOREM 8. Let**

$$||z||^2 \leq 1.$$

**The VC dimension of subset of hyperplanes**

$$(w, z) + b = 0$$

**for which**

$$(w, w) \leq \Delta$$

**is bounded as**

$$h \leq \Delta + 1.$$

# REALIZATION OF SRM PRINCIPLE: SVM



**Let vectors $x_i \in R^n$ is mapped into Hilbert space $z_i \in Z$. Consider element of the structure on hyperplanes in Hilbert space**

$$(w, z) + b = 0$$

**for which $(w, w) \leq \Delta$.**

**Choose the hyperplane that minimizes the empirical loss. That is, minimizes the functional**

$$R(\alpha) = \sum_{i=1}^{\ell} \xi_i,$$

**subject to the constraints**

$$y_i((w, z_i) + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \quad i = 1, ..., \ell,$$

$$(w, w) \leq \Delta.$$

***The SVM algorithm is the solution of this problem.***

# PART 3.

# TEACHER-STUDENT INTERACTION

**To accelerate speed of learning one includes in the learning process *Intelligent Teacher*.**

**There are two mechanisms of Teacher-Student interaction:**
**1) *Similarity Control* between training examples (2005 – 2009).**
**2) *Knowledge Transfer* from Teacher to Student (2015).**

# BACK TO SHANNON

**1. To choose the desired indicator function from $N$ functions one has to ask Query at least $\ell$ questions and get $\ell$ YES-NO (1,0) answers, where**

$$\ell = \log_2 N.$$

**2. To choose the desired indicator function asking *less* than $\log_2 N$ questions is possible only if Query provides $x_i$ with the answer more that has more one bit of information. That is if on question $x_i$ Query answering $(y_i, x_i^*)$ where , $y_i \subset \{0, 1\}$, $x_i^* \in \mathcal{X}^*$.**

**We call $x_i^*$ the *privileged information.***

# TWO LEARNING MODELS



**THE CLASSICAL MODEL: Given iid training pairs**

$$(x_1, y_1), ..., (x_\ell, y_\ell), \quad x_i \in X, \quad y_i \in \{0, 1\}, \quad i = 1, ..., \ell,$$

***where*** $x_i$ ***is generated by unknown*** $P(x)$ ***and*** $y_i$ ***by unknown*** $P(y_i|x_i)$***, find in a given set of functions*** $f(x, \alpha), \alpha \in \Lambda$ ***the one*** $y = f(x, \alpha_\ell)$ ***that minimizes the probability of error.***

**THE LUPI MODEL: Given iid training triplets**

$$(x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y_\ell), \quad x_i \in X, \quad x_i^* \in X^*, \quad y_i \in \{0, 1\},$$

***where*** $x$ ***generated by unknown*** $P(x)$ ***and*** $y_i, x_i^*$ ***by unknown*** $P(y_i, x_i^*|x_i)$***, find in a given set of functions*** $f(x, \alpha), \alpha \in \Lambda$ ***the function*** $y = f(x, \alpha_\ell)$ ***that minimizes the probability of error.***

# SVM FOR SEPARABLE CASE

**Generalization 1: Large margin.**

**Minimize the functional**

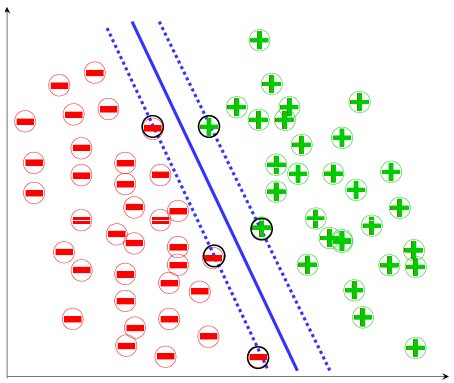
$$R = (w, w)$$

**subject to the constraints**

$$y_i[(w, z_i) + b] \geq 1, \quad i = 1, ..., \ell.$$

**The solution** $(w_\ell, b_\ell)$ **with probability** $1 - \eta$ **has the bound**

$$P_{test} \leq O^* \left( \frac{VCdim - \ln \eta}{\ell} \right).$$



# SVM FOR NON-SEPARABLE CASE

**Generalization 2: Nonseparable case.**

**Minimize the functional**

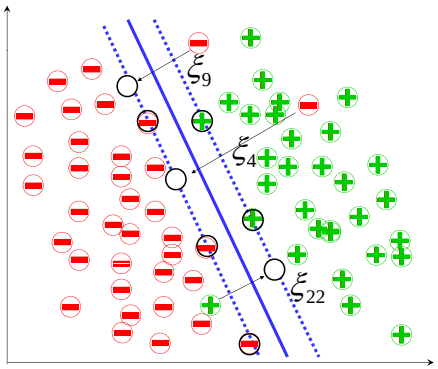
$$R(w,b) = (w,w) + C\sum_{i=1}^{\ell}\xi_i$$

**subject to constraints**

$$y_i[(w,z_i)+b] \geq 1-\xi_i, \quad \xi_i \geq 0, \quad i=1,...,\ell.$$

**The solution** $(w_\ell, b_\ell)$ **with probability** $1-\eta$ **has the bound**

$$P_{test} \leq \nu_{train} + O^*\left(\sqrt{\frac{VCdim - \ln\eta}{\ell}}\right).$$



# WHY WE HAVE SO BIG DIFFERENCE?



- **In the separable case, one estimates $n$ parameters of vector $w$ using $\ell$ examples.**
- **In the non-separable case, one estimates $n + \ell$ parameters ($n$ parameters of vector $w$ and $\ell$ parameters of slacks).**

- **Suppose that we know a set of functions $\xi(x, \delta) \geq 0, \ \delta \in \mathcal{D}$ with finite $VCdim$ (let $\delta$ be an m-dimensional vector) such that**

$$\xi = \xi(x) = \xi(x, \delta_0).$$

**In this setting to find optimal hyperplane in non-separable case one needs to estimate $n + m$ parameters using $\ell$ observations.**

**Can the rate of convergence in new setting be faster?**

# THE KEY OBSERVATION: ORACLE SVM

**Suppose we are given triplets**

$$(x_1, \xi_1^0, y_1), ..., (x_\ell, \xi_\ell^0, y_\ell),$$

**where** $\xi_i^0 = \xi^0(x_i), \quad i = 1, ..., \ell$ **are the slack values with respect to the best hyperplane. Then to find the approximation** $(w_{best}, b_{best})$ **we minimize the functional**

$$R(w, b) = (w, w)$$

**subject to constraints**

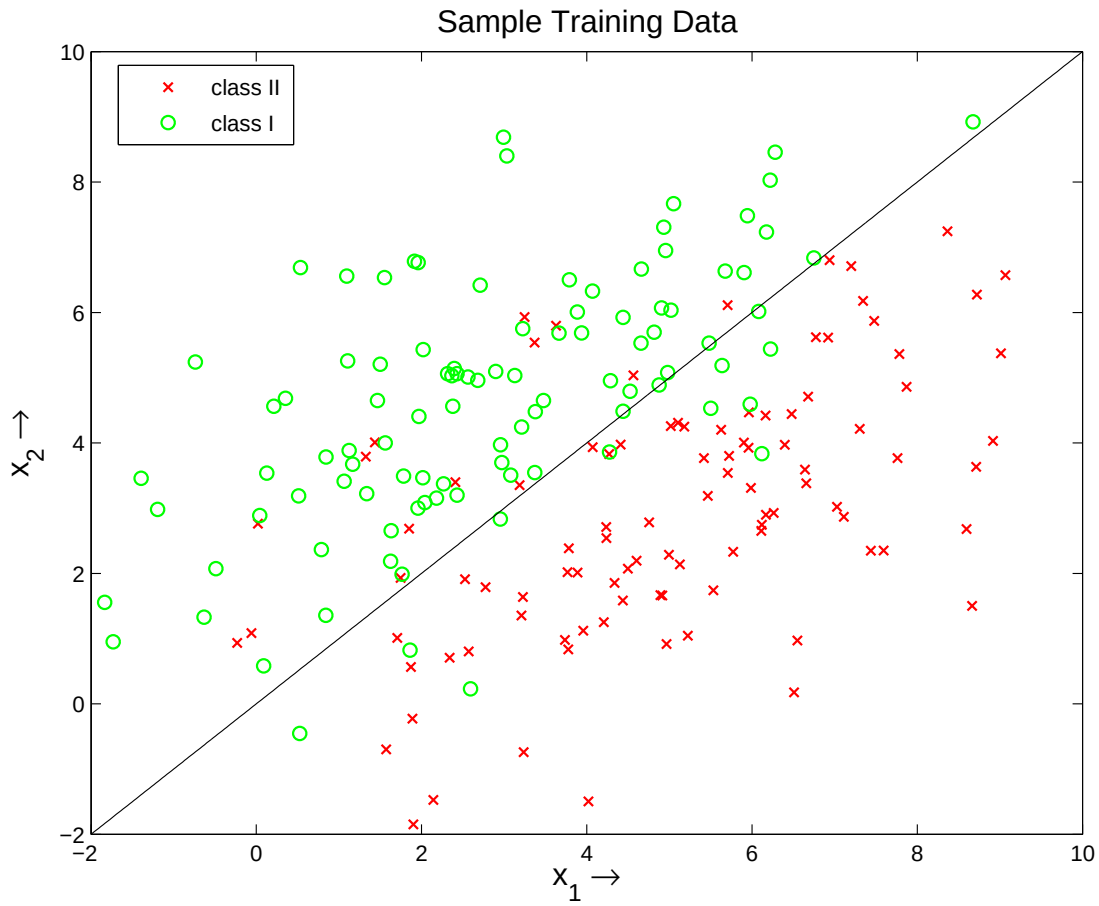
$$y_i[(w, x_i) + b] \geq r_i, \quad r_i = 1 - \xi^0(x_i), \quad i = 1, ..., \ell.$$

**Proposition 1. With probability** $1 - \eta$ **the bound holds**

$$P_{test} \leq \nu_{train} + O^* \left( \frac{VCdim - \ln \eta}{\ell} \right).$$

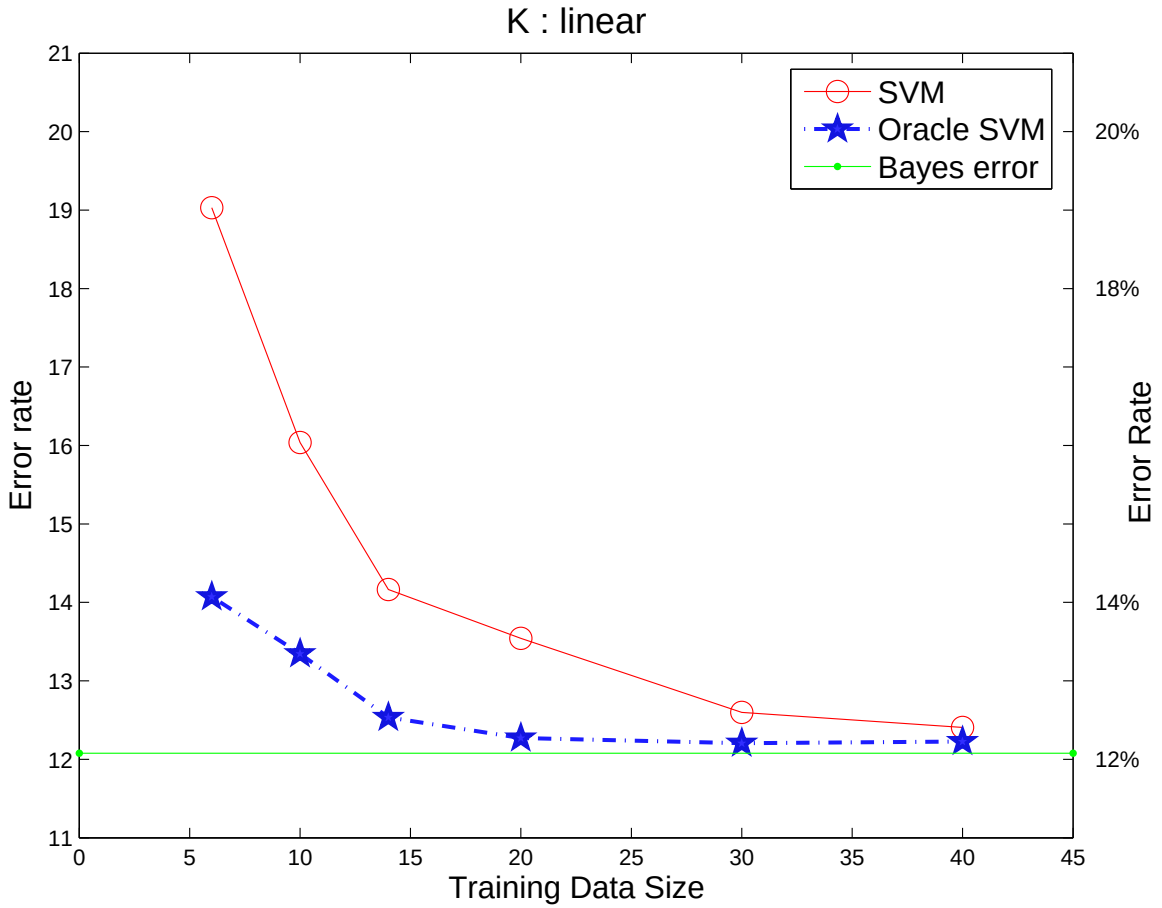
# ILLUSTRATION — I





# ILLUSTRATION —II





# WHAT CAN A REAL TEACHER DO



**Teacher does not know values of slacks. However he can:**

- **Supply students with a *correcting space* $X^*$ and a set of functions $\xi(x^*, \delta), \delta \in D$, with VC dimension $h^*$ which contains a function**

$$\xi_i = \xi(x_i^*, \delta_{best})$$

**that approximates the oracle slack function $\xi^0 = \xi^0(x^*)$ well.**

- **During training process teacher supplies students with triplets**

$$(x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y_\ell)$$

**in order to estimate simultaneously both the correcting (slack) function**

$$\xi = \xi(x^*, \delta_\ell)$$

**and the decision hyperplane (pair $(w_\ell, b_\ell)$).**

# IDEA OF SVM ALGORITHM

## Generalization of Perceptron 3: Mercer Kernels

1. **Transform the training pairs** $(x_1, y_1), ..., (x_\ell, y_\ell)$
   **into the training pairs** $(z_1, y_1), ..., (z_\ell, y_\ell)$

**by mapping vectors** $x \in X$ **into vectors** $z \in Z$**.**

2. **Find in** $Z$ **the hyperplane that minimizes the functional**

$$R(w, b) = (w, w) + C \sum_{i=1}^{\ell} \xi_i$$

**subject to the constraints**

$$y_i[(w, z_i) + b] \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, ..., \ell.$$

3. **Define the inner product in** $Z$ **space using Mercer Kernels**

$$(z_i, z_j) = K(x_i, x_j).$$

# DUAL SPACE SOLUTION FOR SVM



**The decision function has the form**

$$r(x,\alpha) = \mathbf{sgn}\left[\sum_{i=1}^{\ell} \alpha_i y_i K(x_i, x) + b\right], \tag{1}$$

**where** $\alpha_i \geq 0,\ i = 1, ..., \ell$ **are values which maximize the functional**

$$W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(x_i, x_j), \tag{2}$$

**subject to the constraints**

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, ..., \ell.$$

**Here kernel** $K(\cdot,\cdot)$ **is used for two different purposes:**
**1. In (1), to define a set of expansion-functions** $K(x_i, x)$**.**
**2. In (2), to define similarity between vectors** $x_i$ **and** $x_j$**.**

# IDEA OF SVM+ ALGORITHM



• **Transform the training triplets** $(x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y_\ell)$
**into the triplets** $(z_1, z_1^*, y_1), ..., (z_\ell, z_\ell^*, y_\ell)$
**by mapping** $x \in X$ **into** $z \in Z$ **and** $x^* \in X^*$ **into** $z^* \in Z^*$**.**

• **Define the slack-function in the form**

$$\xi_i = y_i[(w^*, z_i^*) + b^*]$$

**and find in space** $Z$ **the hyperplane that minimizes the functional**

$$W(w, b, w^*, b^*) = (w, w) + \gamma(w^*, w^*) + C\sum_{i=1}^{\ell}(y_i[(w^*, z_i^*) + b^*])_+,$$

**subject to the constraints**

$$y_i[(w, z_i) + b] \geq 1 - (y_i[(w^*, z_i^*) + b^*])_+, \quad i = 1, ..., \ell.$$

• **Use inner products in** $Z$ **and** $Z^*$ **spaces in the kernel form**

$$(z_i, z_j) = K(x_i, x_j), \quad (z_i^*, z_j^*) = K^*(x_i^*, x_j^*).$$

# DUAL SPACE SOLUTION FOR SVM+



**The decision function has the form**

$$r(x,\alpha) = \theta\left(\sum_{i=1}^{\ell} \alpha_i y_i K(x_i,x) + b\right),$$

**where** $\alpha_i,\ i = 1,...,\ell$ **are values that maximize the functional**

$$W(\alpha,\beta) =$$

$$\sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} \alpha_i\alpha_j y_i y_j K(x_i,x_j) - \frac{1}{2\gamma}\sum_{i,j=1}^{\ell}(\alpha_i-\beta_i)(\alpha_j-\beta_j)y_i y_j K^*(x_i^*,x_j^*),$$

**subject to the constraints**

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0, \qquad \sum_{i=1}^{\ell} y_i\beta_i = 0.$$

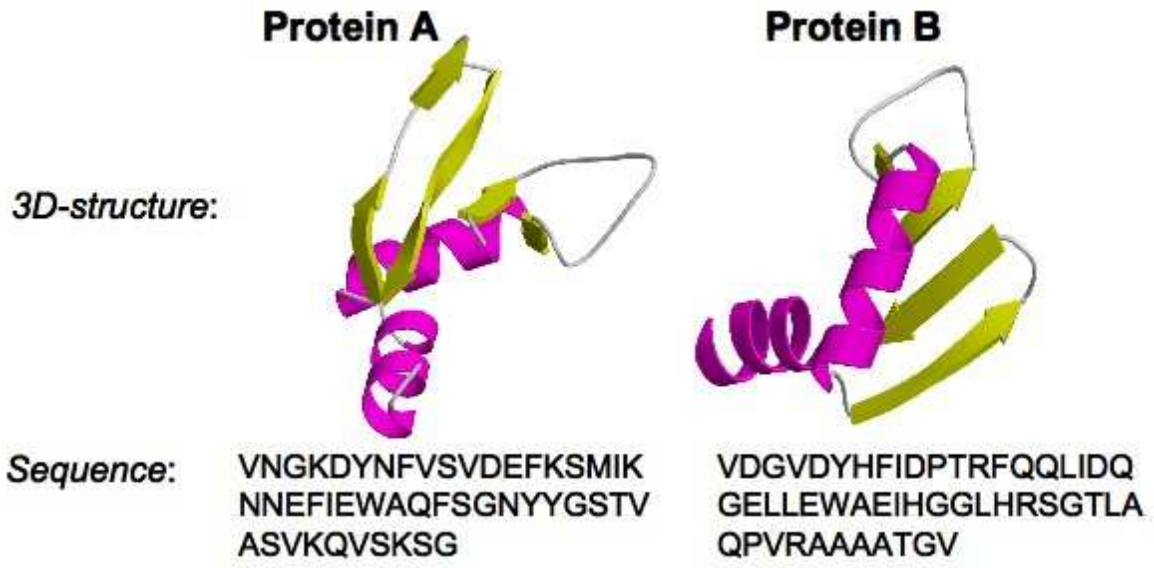
**and the constraints**

$$0 \le \alpha_i \le \beta_i, \qquad 0 \le \beta_i \le C.$$

# ADVANCED TECHNICAL MODEL AS PRIVILEGED INFORMATION

## Classification of proteins into families

**The problem is: Given amino-acid sequences of proteins construct a rule to classify families of proteins. The decision space $X$ is the space of amino-acid sequences. The privileged information space $X^*$ is the space of 3D structures of the proteins.**



# CLASSIFICATION OF PROTEINS (FRAGMENT)



| Protein superfamily pair | SVM | SVM+ | SVM (3D) | |
|---|---|---|---|---|
| a.26.1-vs-c.68.1 | 7.3 | 7.3 | 0 | |
| a.26.1-vs-g.17.1 | 16.4 | 14.3 | 0 | |
| a.118.1-vs-b.82.1 | 19.2 | 6.4 | 0 | ★ |
| a.118.1-vs-d.2.1 | 41.5 | 24.5 | 3.8 | |
| a.118.1-vs-d.14.1 | 13.1 | 13.1 | 2.2 | |
| a.118.1-vs-e.8.1 | 22.8 | 2.3 | 2.3 | ★ |
| b.1.18-vs-b.55.1 | 14.6 | 13.5 | 0 | |
| b.18.1-vs-b.55.1 | 31.5 | 15.1 | 0 | ★ |
| b.18.1-vs-c.55.1 | 36.2 | 36.2 | 0 | ★ |
| b.18.1-vs-c.55.3 | 38.1 | 36.6 | 0 | ★ |
| b.18.1-vs-d.92.1 | 25 | 11.8 | 0 | ★ |
| b.29.1-vs-b.30.5 | 16.9 | 16.9 | 3.6 | |
| b.29.1-vs-b.55.1 | 10 | 5.5 | 0 | |
| b.29.1-vs-b.80.1 | 8.3 | 5.9 | 0 | |
| b.29.1-vs-b.121.4 | 35.9 | 16.8 | 5.3 | ★ |

| Protein superfamily pair | SVM | SVM+ | SVM (3D) | |
|---|---|---|---|---|
| b.29.1-vs-c.47.1 | 10.2 | 3.2 | 0 | ★ |
| b.30.5-vs-b.80.1 | 43.3 | 6.7 | 0 | ★ |
| b.30.5-vs-b.55.1 | 25.5 | 14.6 | 0 | |
| b.55.1-vs-b.82.1 | 11.8 | 10.3 | 0 | |
| b.55.1-vs-d.14.1 | 20.9 | 19.4 | 0 | |
| b.55.1-vs-d.15.1 | 17.7 | 12.7 | 0 | |
| b.80.1-vs-b.82.1 | 4.7 | 4.7 | 0 | |
| b.82.1-vs-b.121.4 | 7.9 | 3.4 | 0 | ★ |
| b.121.4-vs-d.14.1 | 29.5 | 23.9 | 0 | |
| b.121.4-vs-d.92.1 | 15.3 | 9.2 | 0 | |
| c.36.1-vs-c.68.1 | 8.9 | 0 | 0 | ★ |
| c.36.1-vs-e.8.1 | 12.8 | 2.2 | 0 | ★ |
| c.47.1-vs-c.69.1 | 1.9 | 0.6 | 0 | ★ |
| c.52.1-vs-b.80.1 | 11.8 | 5.9 | 0 | |
| c.55.1-vs-c.55.3 | 45.1 | 28.2 | 22.5 | |

★ 3D structure is essential for classification; SVM+ does not improve classification of SVM

★ SVM+ provides significant improvement over SVM (several times)

# FUTURE EVENTS AS PRIVILEGED INFORMATION

**Time series prediction**

Given pairs

$$(x_1, y_1)..., (x_\ell, y_\ell),$$

find the rule

$$y_t = f(x_{t+\Delta}),$$

where

$$x_t = (x(t), ..., x(t-m)).$$

**For regression model of time series:**

$$y_t = x(t+\Delta).$$

**For classification model of time series:**

$$y_t = \begin{cases} 1, & \text{if } x(t+\Delta) > x(t), \\ -1, & \text{if } x(t+\Delta) \leq x(t). \end{cases}$$

# MACKEY-GLASS TIME SERIES PREDICTION



Let data be generated by the Mackey-Glass equation:

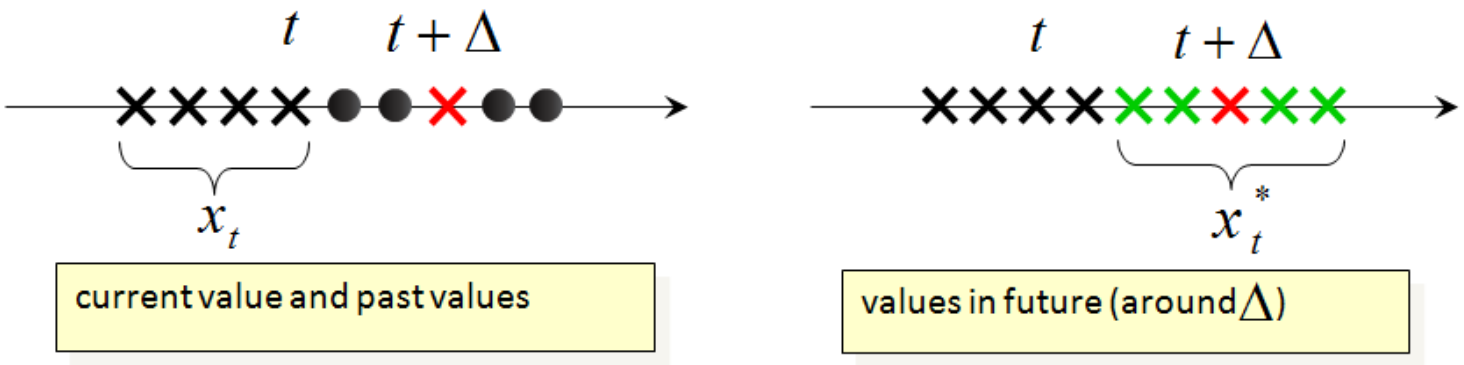
$$\frac{dx(t)}{dt} = -ax(t) + \frac{bx(t-\tau)}{1 + x^{10}(t-\tau)},$$

where $a, b$, and $\tau$ (delay) are parameters.

The training triplets $(x_1, , x_1^*, y_1), ...., (x_\ell, x_\ell^*, y_\ell)$ are defined as follows:

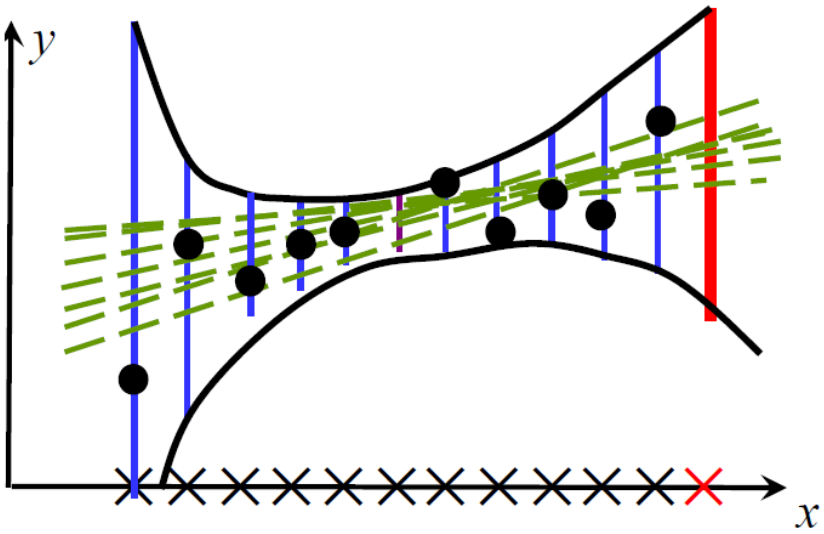
$$x_t = (x(t), x(t-1), x(t-2), x(t-3))$$

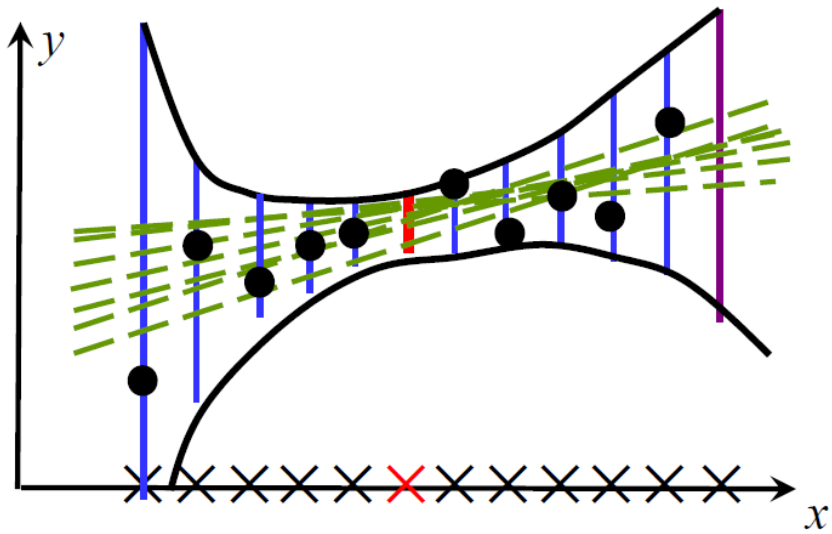
$$x_t^* = (x(t+\Delta-1), x(t+\Delta-2), x(t+\Delta+1), x(t+\Delta+2))$$



# INTERPOLATION AND EXTRAPOLATION





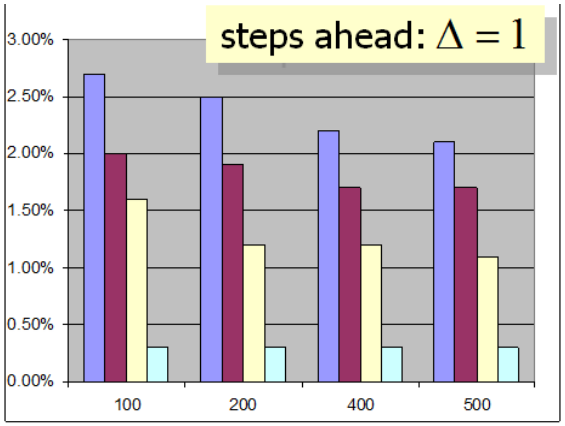



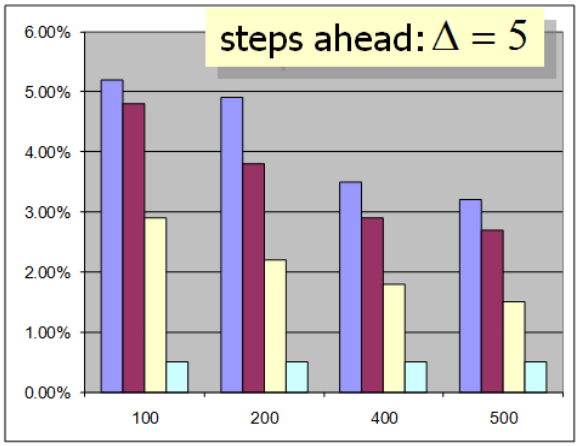

- Extrapolation of trends has to face large conditional variance

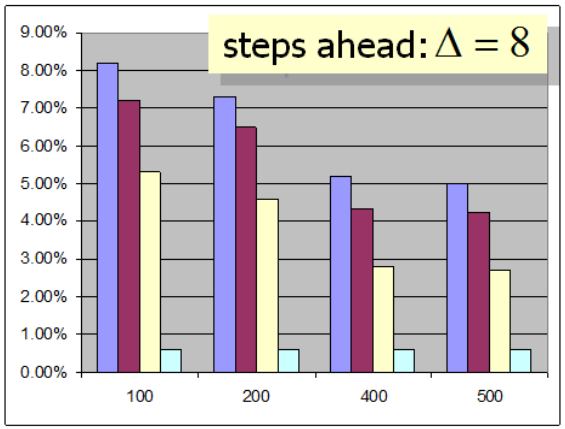
- Interpolation of trends faces small conditional variance

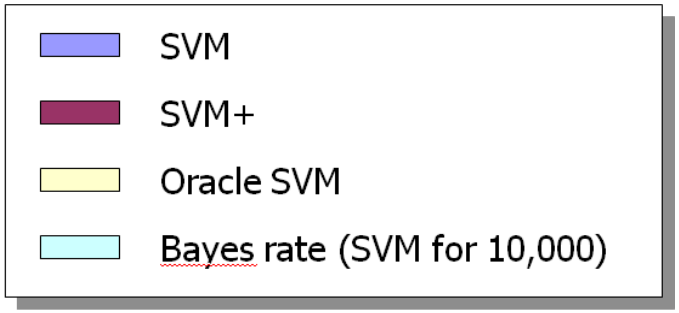
# ILLUSTRATION





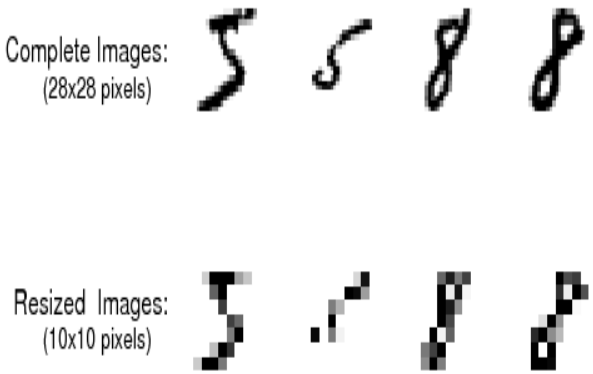



# HOLISTIC DESCRIPTION AS PRIVILEGED INFORMATION



**Classification of digit 5 and digit 8 from the NIST database.**




Given triplets $(x_i, x_i^*, y_i),\ i = 1, ..., \ell$ find the classification rule $y = f(x)$, where $x_i^*$ is the holistic description of the digit $x_i$.

# HOLISTIC DESCRIPTIONS



Straightforward, very active, hard, very masculine with rather clear intention. A sportsman or a warrior. Aggressive and ruthless, eager to dominate everybody, clever and accurate, more emotional than rational, very resolute. No compromise accepted. Strong individuality, egoistic. Honest. Hot, able to give much pain. Hard. Belongs to surface. Individual, no desire to be sociable. First moving second thinking. Will never give a second thought to whatever. Upward-seeking. 40 years old.

A young man is energetic and seriously absorbed in his career. He is not absolutely precise and accurate. He seems a bit aggressive mostly due to lack of sense of humor. He is too busy with himself to be open to the world. He has simple mind and evident plans connected with everyday needs. He feels good in familiar surroundings. Solid soil and earth are his native space. He is upward seeking but does not understand air.

# CODES FOR HOLISTIC DESCRIPTION

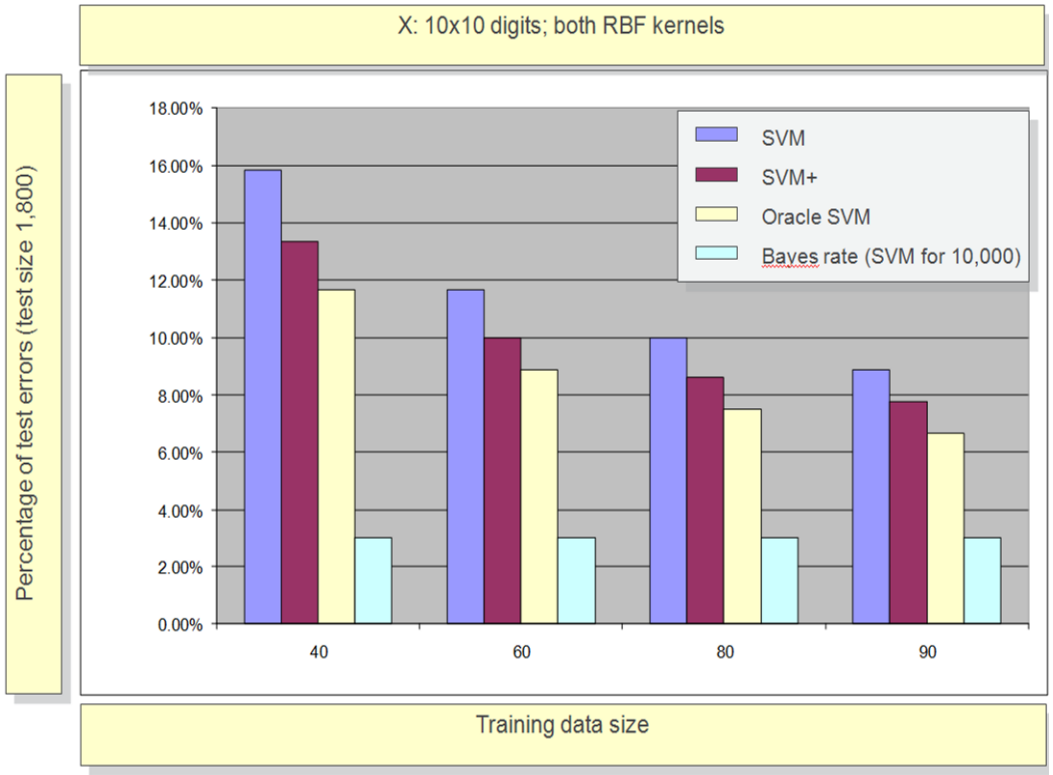


**1.Active** (0 — 5), **2.Passive** (0 — 5), **3.Feminine** (0 — 5),
**4.Masculine** (0 — 5), **5.Hard** (0 — 5), **6.Soft** (0 — 5),
**7.Occupancy** (0 — 3), **8.Strength** (0 — 3), **9.Hot** (0 — 3),
**10.Cold** (0 — 3), **11.Aggressive** (0 — 3), **12.Controlling** (0 — 3),
**13.Mysterious** (0 — 3), **14.Clear** (0 — 3), **15.Emotional** (0 — 3),
**16.Rational** (0 — 3), **17.Collective** (0 — 3), **18.Individual** (0 — 3),
**19.Serious** (0 — 3), **20.Light-minded** (0 —3), **21.Hidden** (0 — 3),
**22.Evident** (0 — 3), **23.Light** (0 — 3), **24.Dark** (0 — 3),
**25.Upward-seeking** (0 — 3), **26.Downward-seeking** (0 — 3),
**27.Water flowing** (0 — 3), **28.Solid earth** (0 — 3),
**29.Interior** (0 — 2), **30.Surface** (0 — 2), **31.Air** (0—3).

`http://ml.nec-labs.com/download/data/svm+/mnist.priviledged`

# RESULTS




X: 10x10 digits; both RBF kernels
Percentage of test errors (test size 1,800)
18.00%
16.00%
14.00%
12.00%
10.00%
8.00%
6.00%
4.00%
2.00%
0.00%
SVM
SVM+
Oracle SVM
Bayes rate (SVM for 10,000)
40
60
80
90
Training data size

# PART 4.

# TEACHER-STUDENT INTERACTION: KNOWLEDGE TRANSFER

**Knowledge transfer is the most important element of Teacher-Student interactions. According to Japanese proverb:**

***"Better than a thousand days of diligent study is one day with a great teacher."***

# KNOWLEDGE TRANSFER MECHANISM



**Given examples with privileged information**

$$(x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y_\ell), \quad x_i \in X,\ x_i^* \in X^*,$$

**consider two pattern recognition problems:**

1. **Using data,** $(x_1, y_1), ..., (x_\ell, y_\ell)$**, find a rule** $y = \mathbf{sgn}\{f_\ell(x)\}$
2. **Using data,** $(x_1^*, y_1), ..., (x_\ell^*, y_\ell)$**, find a rule** $y = \mathbf{sgn}\{f_\ell^*(x^*)\}$**.**

**Suppose that the best rule** $y = \mathbf{sgn}\{f_0^*(x^*)\}$ **in space** $X^*$ **can be as good as the best rule** $y = \mathbf{sgn}\{f_0(x)\}$ **in space** $X$**.**

**Can knowledge of a good rule in** $X^*$ **space**

$$f_\ell^*(x^*) = \sum_{i=1}^{\ell} y_i \alpha_i^* K^*(x_i^*, x^*) + b^*$$

**help to construct a good rule** $y = f_\ell(x)$ **in** $X$ **space?**

**WHY CAN IT HELP?**

# WHY KNOWLEDGE TRANSFER WORKS



**Suppose that our goal is to classify images $x_i$ of biopsy in pixel space $X$ into two classes: cancer and non-cancer.**

**Suppose that, along with images $x_i \in X$, we are given description of the images $x_i^* \in X^*$ (privileged information), reflecting the existing model of developing cancer:**

- ***Aggressive proliferation of A-cells into B-cells.***
- ***Absence of any dynamic in standard picture.***

**Suppose that in space $X^*$ there exist a rule which can separate training data not much worse than the best rule in space $X$.**

**Since space $X$ is universal (can be used for many problems) and space $X^*$ is created just for cancer model, the VC dimension of admissible set of functions in $X$ space has to be much bigger than VC dimension of admissible set of functions in $X^*$.**

**Therefore the rule constructed from examples in space $X^*$ will be more accurate than rule constructed in space $X$. That is why transfer the rule from space $X^*$ into space $X$ is useful.**

# KNOWLEDGE REPRESENTATION IN $X^*$ SPACE.

**Consider three elements of knowledge representation in $X^*$:**
**1. Fundamental elements of the knowledge in space $X^*$.**
**2. Main frames (fragments) of the knowledge in space $X^*$.**
**3. Structure of knowledge: combination of the frames in $X^*$.**

**1. We define as *fundamental elements* of knowledge in $X^*$ the support vectors $u_s^* \in X^*, \quad s = 1, ..., k$ of obtained (in $X^*$) function**

$$f_\ell^*(x^*) = \sum_{s=1}^{k} \beta_s^* K^*(u_s^*, x^*) + b.$$

**2. We define as *frames* the functions $K^*(u_s^*, x^*), \ s = 1, ..., k$.**
**3. We construct in $X$ the images $\phi_s(x)$ of the frames $K^*(u_s^*, x^*)$**
**3. We estimate the desired function $f_\ell(x)$ using expansion on *frame images***

$$f_\ell(x) = \sum_{s=1}^{k} \sigma_s \phi_s(x) + b.$$

# FRAMES AND THEIR IMAGES



**Let $u_1^*..., u_m^*$ be fundamental elements in $X^*$ and let**

$$K^*(u_1^*, x^*), ..., K^*(u_m^*, x^*)$$

**be frames of the rule in space $X^*$.**

**The images (in $X$) of the frames $K^*(u_k^*, x^*)$ (in $X^*$) are**

$$\phi_k(x) = \int K^*(u_k^*, x^*) p(x^*|x) dx^* \quad k = 1, .., m.$$

**To find images $\phi_1(x), ..., \phi_m(x)$ of the frames $K^*(u_1^*, x^*), ..., K^*(u_m^*, x^*)$ using triplets $(x_i, x_i^*, y_i)$, $i = 1, ..., \ell$ one has to:**

***Find*** $m$ ***regression functions*** $\phi_k(x)$, $k = 1, ..., m$ ***given data***

$$(x_1, z_1^k), ..., (x_\ell, z_\ell^k), \quad k = 1, ..., m,$$

**where**

$$z_i^k = K^*(u_k^*, x_i^*), \quad i = 1, ..., \ell, \ k = 1, ...m.$$

# SVM BASED ALGORITHM FOR LUPI



**THE PROBLEM: Find the desired approximation, given triplets**

$$(x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y_\ell) \qquad (1)$$

**THE ALGORITHM:**

**1) Using SVM find the classification rule in $X^*$ space**

$$s^* = \sum_{i=1}^{\ell} y_i \alpha_i K^*(x_i^*, x^*) + b^* \qquad (2)$$

**2) Using RSVM find in space $X$ the images $\phi_1(x), ..., \phi_k(x)$ of the frames $K^*(x_1^*, x^*), ..., K^*(x_k^*, x^*)$ transforming $X$ space as $\mathcal{X} = \mathcal{F}X$.**

**3) Using triplets (1), transformed vectors $\mathcal{F}x_i$, and classification rule (2), construct the triplets**

$$(\mathcal{F}x_1, x_1^*, s_1^*), ..., (\mathcal{F}x_\ell, x_\ell^*, s_\ell^*) \qquad (3)$$

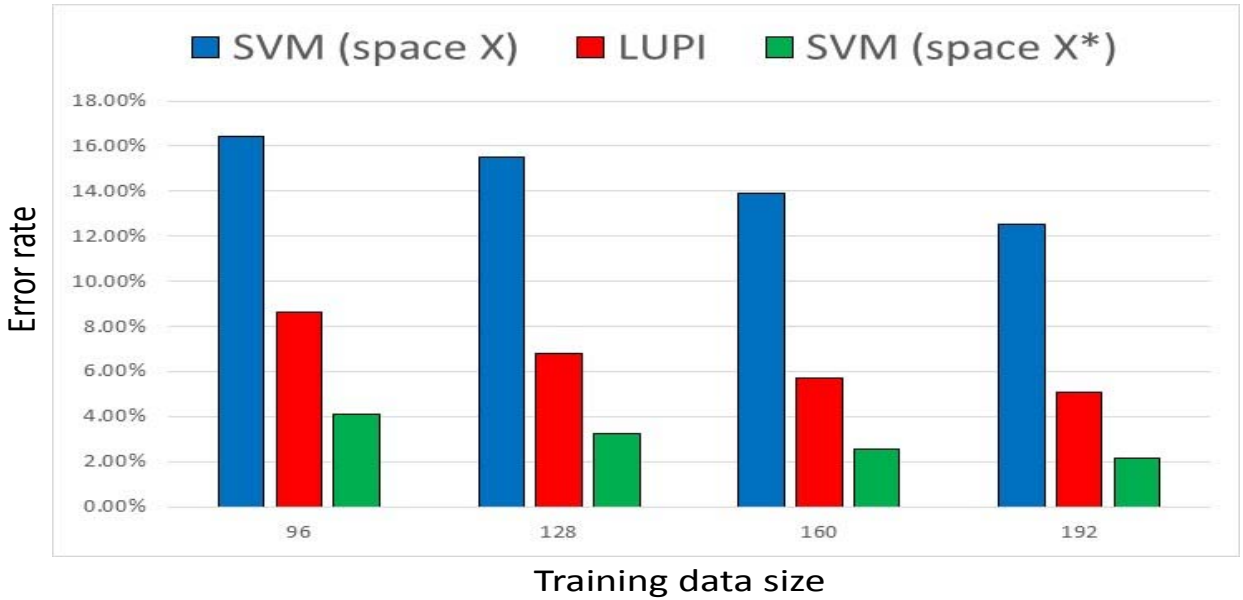
**4) Using SVM+ obtain the desired rule solving LUPI for (3)**

$$y = \sum_{i=1}^{\ell} c_i K(\mathcal{F}x_i, \mathcal{F}x) + b$$

# EXAMPLE OF KNOWLEDGE TRANSFER



**Knowledge transfer from space of high resolution images $X^*$ to space of low resolution images $X$.**



# INTELLIGENCE: REFLECTION IN PHILOSOPHY



**1. Both, old Greek philosophy and classical German philosophy use bilingual model of understanding of World: There exist *"World of Things"* $x \in X$ and *"World of Essences"* $x^* \in X^*$. World of Things is reflection of World of Essences.**

**2. Plato's *Cave Allegory* describes this model as follows:**

**People in the Cave are illuminated behind them. They can see on the wall only *"shadow"* $x$ of *reality* $x^*$. Relation of the reality $x^*$ to shadow $x$ is defined by (unknown) generator $P(x^*|x)$.**

**3. The difference between *Brute* force and *Intelligent* reasoning depends on information that is used for the inference: "shadow" information $x$ or also information about reality $x^*$. To get information $x^*$ one needs to have access to an generator of Intelligence $P(x^*|x)$.**

# INTELLIGENCE: REFLECTION IN MATHEMATICS



**BRUTE FORCE METHODS: using *Ordinary* number $\ell$ of examples $x_i, i = 1, ..., \ell$ chose the best function from a set of *Big* number ($\mathcal{B} \sim 2^{\ell}$) of functions.**

**INTELLIGENT METHODS: using *Ordinary* number $\ell$ of examples $x_i, i = 1, ..., \ell$ AND information $x_i^*$ given by (unknown) *Intelligence generator* $P(x^*|x)$ chose the best function from a set of *Huge* number ($\mathcal{H} >> \mathcal{B}$) of functions.**