Recent Advances on Generalization Bounds
Part II: Combinatorial Bounds

Konstantin Vorontsov

Computing Center RAS • Moscow Institute of Physics and Technology

4th International Conference on
Pattern Recognition and Machine Intelligence (PReMI'11)
Moscow, Russian Federation • June 27 – July 1, 2011

Contents

Learning with binary loss

$\mathbb{X}^L = \{x_1, \dots, x_L\}$ — a finite universe set of objects;
$A = \{a_1, \dots, a_D\}$ — a finite set of classifiers;

$I(a, x) = [\text{classifier } a \text{ makes an error on object } x]$ — binary loss;

Loss matrix of size $L \times D$, all columns are distinct:

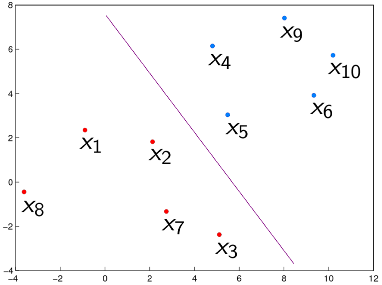
	a_1	a_2	a_3	a_4	a_5	a_6	$\cdots$	a_D	
x_1	1	1	0	**0**	0	1	$\cdots$	1	X — observable
$\dots$	0	0	0	**0**	1	1	$\cdots$	1	(training) sample
x_ℓ	0	0	1	**0**	0	0	$\cdots$	0	of size ℓ
$x_{\ell+1}$	0	0	0	**1**	1	1	$\cdots$	0	$\bar{X}$ — hidden
$\dots$	0	0	0	**1**	0	0	$\cdots$	1	(testing) sample
x_L	0	1	1	**1**	1	1	$\cdots$	0	od size $k = L - \ell$

$n(a)$ — *number of errors* of a classifier a on the set $\mathbb{X}^L$;

$n(a, X)$ — *number of errors* of a classifier a on a sample $X \subset \mathbb{X}^L$;

$\nu(a, X) = n(a, X)/|X|$ — *error rate* of a on a sample $X \subset \mathbb{X}^L$;

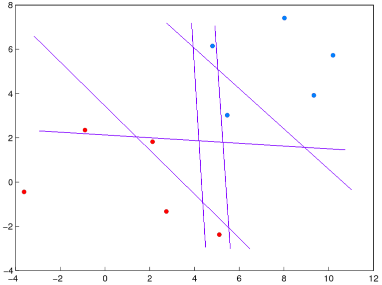
Example. The loss matrix for a set of linear classifiers



1 vector having no errors

	no errors
x_1	0
x_2	0
x_3	0
x_4	0
x_5	0
x_6	0
x_7	0
x_8	0
x_9	0
x_{10}	0

Example. The loss matrix for a set of linear classifiers

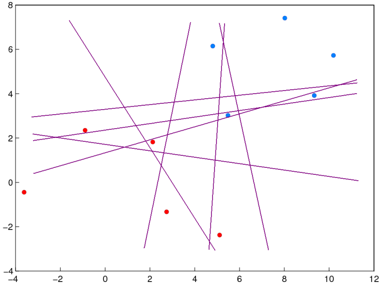


1 vector having no errors

5 vectors having 1 error

	no errors	1 error				
x_1	0	1	0	0	0	0
x_2	0	0	1	0	0	0
x_3	0	0	0	1	0	0
x_4	0	0	0	0	1	0
x_5	0	0	0	0	0	1
x_6	0	0	0	0	0	0
x_7	0	0	0	0	0	0
x_8	0	0	0	0	0	0
x_9	0	0	0	0	0	0
x_{10}	0	0	0	0	0	0

Example. The loss matrix for a set of linear classifiers



1 vector having no errors
5 vectors having 1 error
8 vectors having 2 errors

	no errors	1 error					2 errors								
x_1	0	1	0	0	0	0	1	0	0	0	0	1	1	0	...
x_2	0	0	1	0	0	0	1	1	0	0	0	0	0	0	...
x_3	0	0	0	1	0	0	0	1	1	0	0	0	0	1	...
x_4	0	0	0	0	1	0	0	0	1	1	0	0	0	0	...
x_5	0	0	0	0	0	1	0	0	0	1	1	1	0	0	...
x_6	0	0	0	0	0	0	0	0	0	0	1	0	1	0	...
x_7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	...
x_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
x_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
x_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

Probability of overfitting

Def. The *learning algorithm* $\mu\colon X \mapsto a$ takes a training sample $X \subset \mathbb{X}^L$ and returns a classifier $a \equiv \mu X \in A$.

Def. Algorithm μ *overfits* on a given partition $X \sqcup \bar{X} = \mathbb{X}^L$ if

$$\delta(\mu, X) \equiv \nu\bigl(\mu X, \bar{X}\bigr) - \nu\bigl(\mu X, X\bigr) \geqslant \varepsilon.$$

Def. ***Probability of overfitting***

$$Q_\varepsilon(\mu, \mathbb{X}^L) = \mathsf{P}\bigl[\delta(\mu, X) \geqslant \varepsilon\bigr].$$

Def. *Exact bound*: $Q_\varepsilon = \eta(\varepsilon)$.

Def. *Upper bound*: $Q_\varepsilon \leqslant \eta(\varepsilon)$.

Weak (permutational) probabilistic assumptions

Axiom

All partitions $\mathbb{X}^L = \{x_1, \dots, x_L\} = X \sqcup \bar{X}$ *are equiprobable, where*
X *— observable training sample of size* ℓ*;*
$\bar{X}$ *— hidden testing sample of size* $k = L - \ell$*;*

Probability is defined as a *fraction of partitions*:

$$Q_\varepsilon = \mathsf{P}\bigl[\delta(\mu, X) \geqslant \varepsilon\bigr] = \frac{1}{C_L^\ell} \sum_{\substack{X, \bar{X} \\ X \sqcup \bar{X} = \mathbb{X}^L}} \bigl[\delta(\mu, X) \geqslant \varepsilon\bigr].$$

Interpretation: Only *independence* of observations is postulated. Continuous measures, infinite sets, and limits $|X| \to \infty$ are illegal.

Nevertheless, tight generalization bounds can be obtained!

One-classifier bound (OC-bound)

Let $A = \{a\}$, $m = n(a)$. Obviously, $\mu X = a$ for all $X \subset \mathbb{X}^L$.

Definition

Hypergeometric distribution function:

$$\text{PDF:} \quad h_L^{\ell,m}(s) = \mathsf{P}\big[n(a,X) = s\big] = \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell};$$

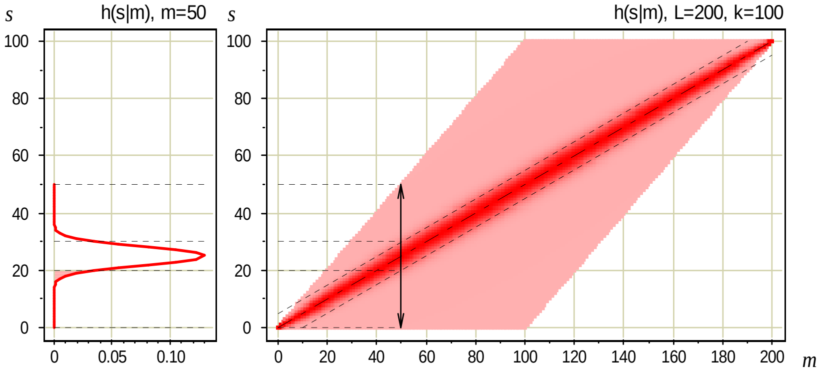
$$\text{CDF:} \quad H_L^{\ell,m}(z) = \mathsf{P}\big[n(a,X) \leqslant z\big] = \sum_{s=0}^{\lfloor z \rfloor} h_L^{\ell,m}(s).$$

Theorem (exact OC-bound)

For one-classifier set $A = \{a\}$, $m = n(a)$, and any $\varepsilon \in (0,1)$

$$Q_\varepsilon = H_L^{\ell,m}\big(s_m(\varepsilon)\big), \quad s_m(\varepsilon) = \tfrac{\ell}{L}(m - \varepsilon k).$$

Hypergeometric distribution, PDF $h_L^{\ell,m}(s) = C_m^s C_{L-m}^{\ell-s} / C_L^\ell$



Distribution is concentrated along diagonal $s \approx \frac{\ell}{L} m$, thus allowing to predict both $n(a) = m$ and $n(a, \bar{X}) = \frac{m-s}{k}$ from $n(a, X) = s$.

Law of Large Numbers: $\nu(a, X) \to \nu(a)$ with $\ell, k \to \infty$.

Vapnik-Chervonenkis bound (VC-bound), 1971

For any $\mathbb{X}^L$, A, μ, and $\varepsilon \in (0,1)$

$$Q_\varepsilon = \mathsf{P}\bigl[\nu(\mu X, \bar{X}) - \nu(\mu X, X) \geqslant \varepsilon\bigr] \leqslant$$

STEP 1: *uniform bound* makes the result independent on μ:

$$\leqslant \widetilde{Q}_\varepsilon = \mathsf{P} \max_{a \in A}\bigl[\nu(a, \bar{X}) - \nu(a, X) \geqslant \varepsilon\bigr] \leqslant$$

STEP 2: *union bound* (wich is usually higly overestimated):

$$\leqslant \mathsf{P} \sum_{a \in A}\bigl[\nu(a, \bar{X}) - \nu(a, X) \geqslant \varepsilon\bigr] =$$

exact one-classifier bound:

$$= \sum_{a \in A} H_L^{\ell, m}\bigl(s_m(\varepsilon)\bigr), \quad m = n(a).$$

OC-bound vs. VC-bound

The VC-bound [Vapnik and Chervonenkis, 1971] can be represented as a sum of OC-bounds over all classifiers $a \in A$:

Theorem (OC-bound)

$$Q_\varepsilon = H_L^{\ell,\,m}\left(s_m(\varepsilon)\right), \quad m = n(a).$$

Theorem (VC-bound)

$$Q_\varepsilon \leqslant \widetilde{Q}_\varepsilon \leqslant \sum_{a\in A} H_L^{\ell,\,m}\left(s_m(\varepsilon)\right), \quad m = n(a).$$

VC-bound is loose because of uniform bound and union bound, which discards the ***splitting*** and ***similarity*** properties of A.

Paradigms of COLT not using union bound

- Uniform convergence bounds [Vapnik, Chervonenkis, 1968]
- Theory of learnable (PAC-learning) [Valiant, 1982]
- Data-dependent bounds [Haussler, 1992]
- Concentration inequalities [Talagrand, 1995]
- Connected function classes [Sill, 1995]
- Similar classifiers VC bounds [Bax, 1997]
- Margin based bounds [Bartlett, 1998]
- Self-bounding learning algorithms [Freund, 1998]
- Rademacher complexity [Koltchinskii, 1998]
- Adaptive microchoice bounds [Langford, Blum, 2001]
- Algorithmic stability [Bousquet, Elisseeff, 2002]
- Algorithmic luckiness [Herbrich, Williamson, 2002]
- Shell bounds [Langford, 2002]
- PAC-Bayes bounds [McAllester, 1999; Langford, 2005]
- Splitting and connectivity bounds [Vorontsov, 2010]

Splitting and Connectivity graph

Define two binary relations on classifiers:
partial order $a \leqslant b$: $I(a,x) \leqslant I(b,x)$ for all $x \in \mathbb{X}^L$;
precedence $a \prec b$: $a \leqslant b$ and Hamming distance $\|b - a\| = 1$.

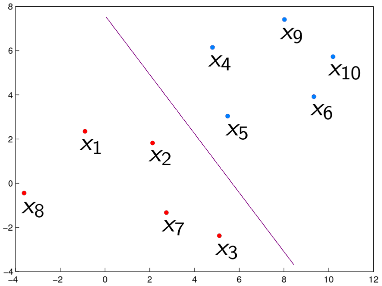
Definition (SC-graph)

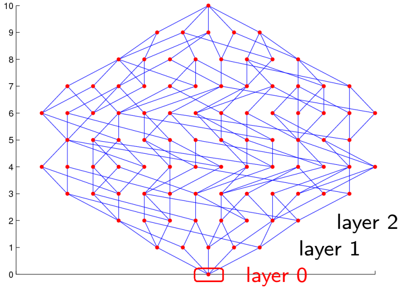
Splitting and Connectivity (SC-) graph $\langle A, E \rangle$:
A — a set of classifiers with distinct binary loss vectors;
$E = \bigl\{(a, b) \colon a \prec b\bigr\}$.

Properties of the SC-graph:

- each edge (a, b) is labeled by an object $x_{ab} \in \mathbb{X}^L$ such that $0 = I(a, x_{ab}) < I(b, x_{ab}) = 1$;
- multipartite graph with layers $A_m = \bigl\{a \in A \colon n(a) = m\bigr\}$, $m = 0, \dots, L+1$;

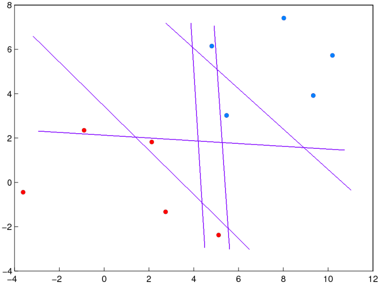
Example. Loss matrix and SC-graph for a set of linear classifiers

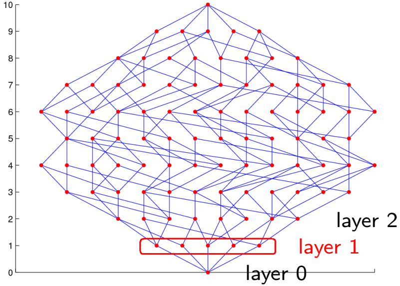




	layer 0
x_1	0
x_2	0
x_3	0
x_4	0
x_5	0
x_6	0
x_7	0
x_8	0
x_9	0
x_{10}	0

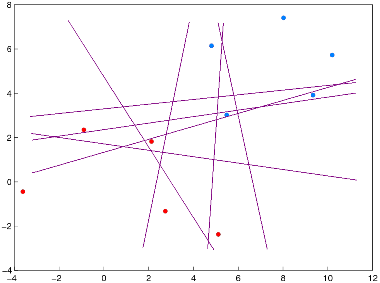
Example. Loss matrix and SC-graph for a set of linear classifiers

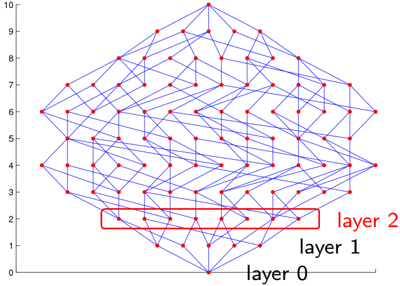




	layer 0	layer 1				
x_1	0	1	0	0	0	0
x_2	0	0	1	0	0	0
x_3	0	0	0	1	0	0
x_4	0	0	0	0	1	0
x_5	0	0	0	0	0	1
x_6	0	0	0	0	0	0
x_7	0	0	0	0	0	0
x_8	0	0	0	0	0	0
x_9	0	0	0	0	0	0
x_{10}	0	0	0	0	0	0

Example. Loss matrix and SC-graph for a set of linear classifiers





	layer 0	layer 1					layer 2								
x_1	0	1	0	0	0	0	1	0	0	0	0	1	1	0	...
x_2	0	0	1	0	0	0	1	1	0	0	0	0	0	0	...
x_3	0	0	0	1	0	0	0	1	1	0	0	0	0	1	...
x_4	0	0	0	0	1	0	0	0	1	1	0	0	0	0	...
x_5	0	0	0	0	0	1	0	0	0	1	1	1	0	0	...
x_6	0	0	0	0	0	0	0	0	0	0	1	0	1	0	...
x_7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	...
x_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
x_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
x_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

Connectivity and inferiority of a classifier

Def. *Connectivity* of a classifier $a \in A$

$p(a) = \#\{x_{ba} \in \mathbb{X}^L : b \prec a\}$ — low-connectivity.

$q(a) = \#\{x_{ab} \in \mathbb{X}^L : a \prec b\}$ — up-connectivity;

Def. *Inferiority* of a classifier $a \in A$

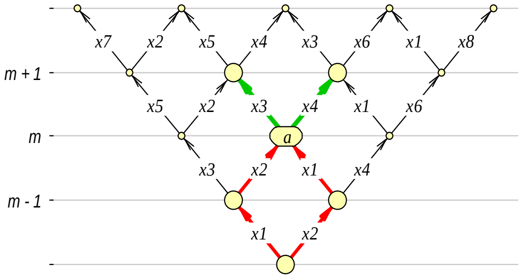
$r(a) = \#\{x_{cb} \in \mathbb{X}^L : c \prec b \leqslant a\} \in \{p(a), \dots, n(a)\}$.

Example:

$p(a) = \#\{x1, x2\} = 2,$

$q(a) = \#\{x3, x4\} = 2,$

$r(a) = \#\{x1, x2\} = 2.$



Uniform Connectivity (UC-) bound

Theorem (UC-bound)

For all $\mathbb{X}^L$, μ, A *and* $\varepsilon \in (0,1)$

$$\widetilde{Q}_\varepsilon \leqslant \sum_{a\in A} [p \leqslant k] \left(\frac{C_{L-q-p}^{\ell-q}}{C_L^\ell} \right) H_{L-q-p}^{\ell-q,\ m-p}\,(s_m(\varepsilon))$$

where $m = n(a)$, $q = q(a)$, $p = p(a)$.

1. UC-bound improves the VC-bound, even if $p(a) \equiv q(a) \equiv 0$:
   $\widetilde{Q}_\varepsilon \leqslant \sum_{a\in A} H_L^{\ell,\,m}\,(s_m(\varepsilon))$.
2. The contribution of $a \in A$ decreases exponentially by $p(a)$
   $\Rightarrow$ **connected sets are less subjected to overfitting**.
3. UC-bound relies on connectivity, but disregards splitting.

Pessimistic Empirical Risk Minimization

Definition (ERM)

Learning algorithm μ is Empirical Risk Minimization if

$$\mu X \in A(X), \qquad A(X) = \operatorname{Arg}\min_{a \in A} n(a, X);$$

A choice of a classifier a from $A(X)$ is ambiguous.
Pessimistic choice will result in modestly inflated upper bound.

Definition (pessimistic ERM)

Learning algorithm μ is pessimistic ERM if

$$\mu X = \arg\max_{a \in A(X)} n(a, \bar{X});$$

The Splitting and Connectivity (SC-) bound

Theorem (SC-bound)

For pessimistic ERM μ, any $\mathbb{X}^L$, A and $\varepsilon \in (0,1)$

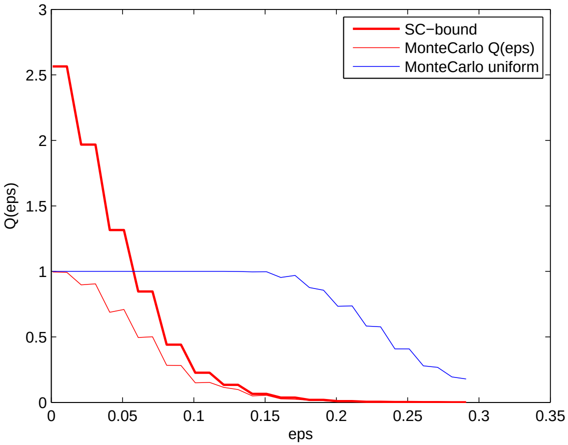
$$Q_\varepsilon \leqslant \sum_{a \in A} [r \leqslant k] \left(\frac{C_{L-q-r}^{\ell-q}}{C_L^\ell} \right) H_{L-q-r}^{\ell-q,\, m-r}\left(s_m(\varepsilon)\right),$$

where $m = n(a)$, $q = q(a)$, $r = r(a)$.

1. If $q(a) \equiv r(a) \equiv 0$ then SC-bound transforms to VC-bound:
   $Q_\varepsilon \leqslant \sum_{a \in A} H_L^{\ell,\, m}\left(s_m(\varepsilon)\right)$.
2. The contribution of $a \in A$ decreases exponentially by:
   $q(a) \Rightarrow$ **connected sets are less subjected to overfitting**;
   $r(a) \Rightarrow$ **only lower layers contribute significantly to Q_ε.**

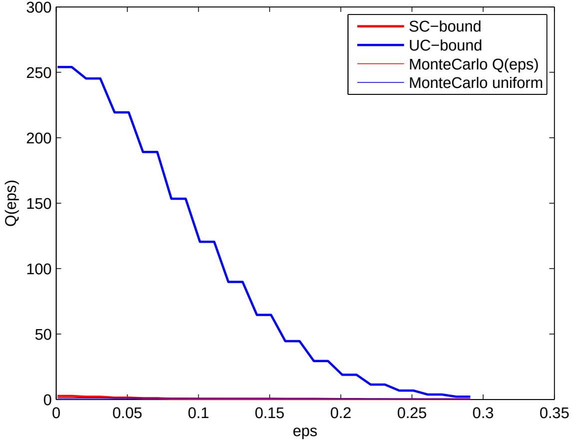
Experiment on model data: SC-bound vs. Monte Carlo estimate

Separable two-dimensional task, $L = 100$, two classes.



Experiment on model data: UC-bound vs. Monte Carlo estimate

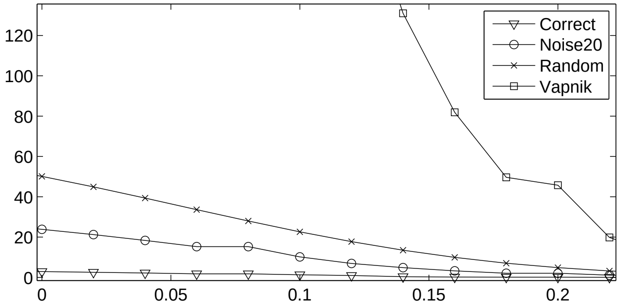
Separable two-dimensional task, $L = 100$, two classes.



Experiment on model data: SC-bounds vs. VC-bound

Two-dimensional task, $L = 100$, two classes.

Correct — 0% errors;
Noise20 — 20% errors;
Random — 50% errors;
Vapnik — data-independent VC-bound.

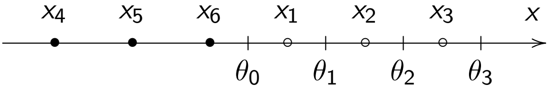


Monotone chain of classifiers

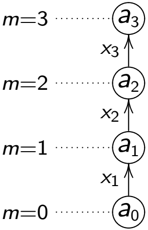
Def. *Monotone chain* of classifiers: $a_0 \prec a_1 \prec \cdots \prec a_D$.

Example: 1-dimensional threshold classifiers $a_j(x) = [x - \theta_j]$;

2 classes $\{\bullet, \circ\}$
6 objects



SC-graph:



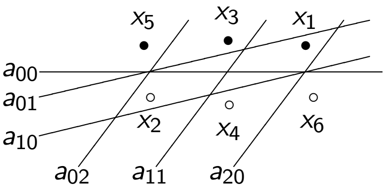
Loss matrix:

	a_0	a_1	a_2	a_3
x_1	0	1	1	1
x_2	0	0	1	1
x_3	0	0	0	1
x_4	0	0	0	0
x_5	0	0	0	0
x_6	0	0	0	0

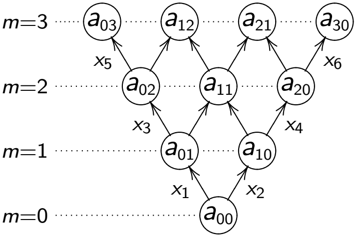
Two-dimensional monotone lattice of classifiers

Example:

2-dimensional linear classifiers,
2 classes $\{\bullet, \circ\}$,
6 objects



SC-graph:



Loss matrix:

	a_{00}	a_{01}	a_{10}	a_{02}	a_{11}	a_{20}	a_{03}	a_{12}	a_{21}	a_{30}
x_1	0	1	0	1	1	0	1	1	1	0
x_2	0	0	1	0	1	1	0	1	1	1
x_3	0	0	0	1	0	0	1	1	0	0
x_4	0	0	0	0	0	1	0	0	1	1
x_5	0	0	0	0	0	0	1	0	0	0
x_6	0	0	0	0	0	0	0	0	0	1

SC-bound is exact(!) for multidimensional(!) lattices of classifiers

Denote $\mathbf{d} = (d_1, \ldots, d_h)$ an h-dimensional index vector, $d_j = 0, 1, \ldots$
Denote $|\mathbf{d}| = d_1 + \ldots + d_h$.

Definition

Monotone h-dimensional lattice of classifiers of height D:

$$A = \left\{ a_{\mathbf{d}},\ |\mathbf{d}| \leqslant D \;\middle|\; \begin{array}{l} \mathbf{c} < \mathbf{d} \;\Rightarrow\; a_{\mathbf{c}} < a_{\mathbf{d}} \\ n(a_{\mathbf{d}}) = m_0 + |\mathbf{d}| \end{array} \right\}.$$

Theorem (exact SC-bound)

If A is monotone h-dimensional lattice of height D, $D \geqslant k$, and μ is pessimistic ERM then for any $\varepsilon \in (0,1)$

$$Q_\varepsilon = \sum_{t=0}^{k} C_{h+t-1}^{t} \frac{C_{L-h-t}^{\ell-h}}{C_L^\ell} H_{L-h-t}^{\ell-h,\ m_0}\left(s_{m_0+t}(\varepsilon)\right).$$

Sets of classifiers with known SC-bound

Model sets of classifiers with known exact SC-bound:

- monotone chains and multidimensional lattices;
- unimodal chains and multidimensional lattices;
- pencils of monotone chains;
- layers and intervals of boolean cube;
- hamming balls and their lower layers;
- some sparse subsets of multidimensional lattices;
- some sparse subsets of hamming balls;

Real sets of classifiers with known tight SC-bound:

- conjunction rules (see further);
- linear classifiers (under construction now).

Conclusions

- Combinatorial framework can give tight and sometimes exact generalization bounds.
- OC (one-classifier) bound is exact.
- UC (uniform connectivity) bound rely on *connectivity* but neglect *splitting*.
- SC (splitting and connectivity) bound is most tight and even *exact* for monotone chains and lattices of classifiers.
- SC-bound being applied to rule induction reduces testing error of classifiers by 1–2%.

Further: thee appendix slides about underlying combinatorial technique for SC-bounds.

Generating and inhibiting subsets of objects

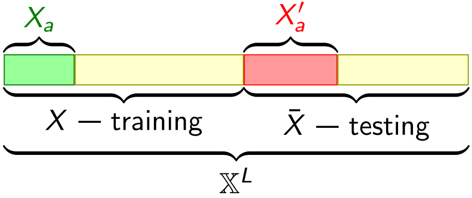
Conjecture

For any $a \in A$ generating set $X_a \subset \mathbb{X}^L$ and inhibiting set $X'_a \subset \mathbb{X}^L$ exist such that if classifier $a \in A$ is a result of learning then

all objects X_a lie in the training set and

all objects X'_a lie in the testing set:

$$[\mu X = a] \leqslant [X_a \subseteq X]\,[X'_a \subseteq \bar{X}].$$



Bounds based on generating and inhibiting subsets

Lemma (Probability of obtaining each of classifiers)

*If **Conjecture** is true then for any* μ, X, $a \in A$

$$\mathsf{P}\bigl[\mu X{=}a\bigr] \leqslant P_a = C_{L_a}^{\ell_a} / C_L^{\ell}.$$

where $L_a = L - |X_a| - |X'_a|$, $\quad \ell_a = \ell - |X_a|$.

Theorem (Probability of overfitting)

*If **Conjecture** is true then for any* $\mathbb{X}^L$, μ, A *and* $\varepsilon \in (0,1)$

$$Q_\varepsilon \leqslant \sum_{a \in A} P_a H_{L_a}^{\ell_a,\, m_a}\left(s_a(\varepsilon)\right),$$

where $m_a = n(a, \mathbb{X}^L) - n(a, X_a) - n(a, X'_a)$,

$$s_a(\varepsilon) = \tfrac{\ell}{L}\bigl(n(a, \mathbb{X}^L) - \varepsilon k\bigr) - n(a, X_a).$$

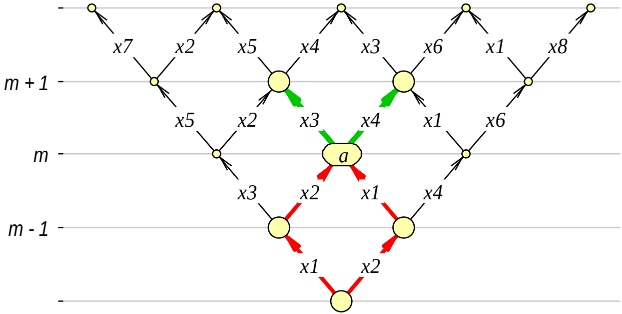
Correspondence between SC-graph and generating/inhibiting subsets

Upper connectivity of a classifier $a \in A$

$q(a) = |X_a|,\ \ X_a = \{x_{ab} \in \mathbb{X}^L\colon a \prec b\}$ — generating subset.

Inferiority of a classifier $a \in A$

$r(a) = |X'_a|,\ \ X'_a = \{x_{cb} \in \mathbb{X}^L\colon c \prec b \leqslant a\}$ — inhibiting subset.



Open problems

- Combinatorial framework can give tight and sometimes exact generalization bounds.
- OC (one-classifier) bound is exact.
- UC (uniform connectivity) bound rely on *connectivity* but neglect *splitting*.
- SC (splitting and connectivity) bound is most tight and even *exact* for monotone chains and lattices of classifiers.
- SC-bound being applied to rule induction reduces testing error of classifiers by 1–2%.

Further: thee appendix slides about underlying combinatorial technique for SC-bounds.

Classifier — weighted voting of conjunctive rules

Rule-based classifier (weighted voting of rules):

$$a(x) = \arg\max_{y\in Y} \sum_{r\in R_y} w_r r(x),$$

where Y — set of class labels,
R_y — set of rules that votes for the class y,
$r\colon X \to \{0,1\}$ — rule, and w_r — its weight.

Conjunctive rule:

$$r(x) = \bigwedge_{j\in J} \bigl[f_j(x) \leqslant \theta_j\bigr],$$

where $f_j(x)$ — real features, θ_j — thresholds, $j = 1,\dots,n$;
$J \subseteq \{1,\dots,n\}$ — subset of features, usually $|J| \lesssim 7$;

Rule evaluation heuristics

Intrinsically the rule learning is a two-criteria optimization problem:

$$N(r, X) = \frac{1}{|X|} \#\{x_i \in X \colon r(x_i) = 1,\ y_i \neq y\} \to \min_r;$$
$$P(r, X) = \frac{1}{|X|} \#\{x_i \in X \colon r(x_i) = 1,\ y_i = y\} \to \max_r;$$

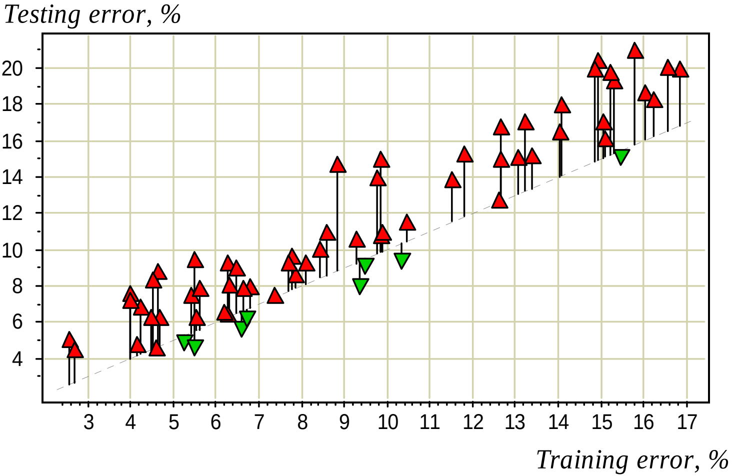
Practically one-criterion heuristics $H(P, N) \to \max_r$ are used:

- Information gain;
- Gini Index;
- Fisher exact test, χ^2 or ω^2 statistical tests, etc.

A common drawback of all these criteria:

Ignoring an overfitting that results from thresholds θ_j learning:
$N(r, \bar{X})$ will be greater than expected;
$P(r, \bar{X})$ will be less than expected.

Problem: rules are typically overfitted in real applications



Real task: predicting the result of atherosclerosis surgical treatment, $L = 98$.

SC-modification of rule evaluation heuristics

Problem:

Estimate $N(r, \bar{X})$ and $P(r, \bar{X})$ to select rules more carefully.

Solution:

1. Calculate data-dependent SC-bounds:

$$\mathsf{P}\big[N(r, \bar{X}) - N(r, X) \geqslant \varepsilon\big] \leqslant \eta_N(\varepsilon);$$
$$\mathsf{P}\big[P(r, X) - P(r, \bar{X}) \geqslant \varepsilon\big] \leqslant \eta_P(\varepsilon);$$

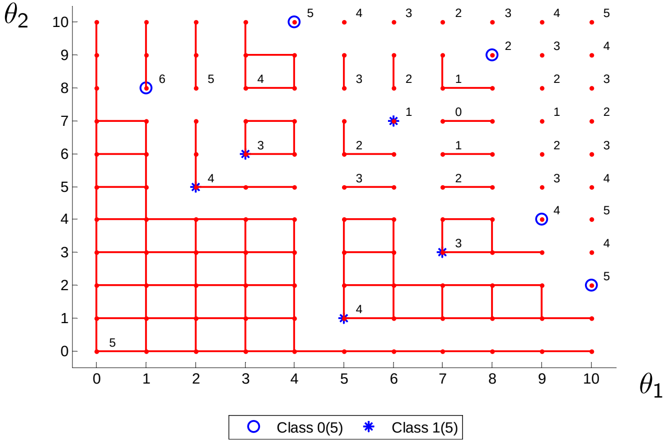
2. Invert SC-bounds: with probability at least $1 - \eta$

$$N(r, \bar{X}) \leqslant \hat{N}(r, \bar{X}) = N(r, X) + \varepsilon_N(\eta);$$
$$P(r, \bar{X}) \geqslant \hat{P}(r, \bar{X}) = P(r, X) - \varepsilon_P(\eta).$$

3. Substitute $\hat{P}$, $\hat{N}$ in a one-criterion heuristic: $H(\hat{P}, \hat{N}) \to \max\limits_r$.

Classes of equivalent rules: one point per rule

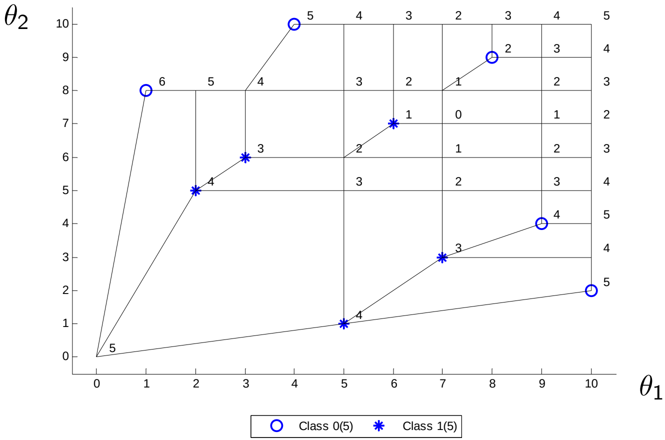
Example: separable 2-dimensional task, $L = 10$, two classes.
rules: $r(x) = \big[f_1(x) \leqslant \theta_1 \text{ and } f_2(x) \leqslant \theta_2\big]$.



Classes of equivalent rules: one point per class

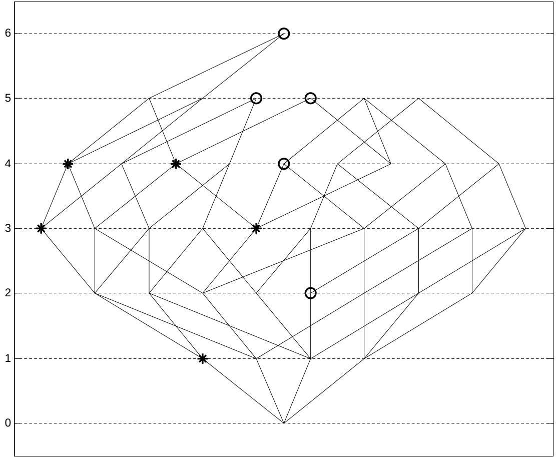
Example: the same classification task. One point per class.

rules: $r(x) = \big[f_1(x) \leqslant \theta_1 \text{ and } f_2(x) \leqslant \theta_2\big].$



Classes of equivalent rules: SC-graph

Example: SC-graph isomorphic to the graph at previous slide.



Experiment on real data sets

Data sets from UCI repository:

Task	Objects	Features
australian	690	14
echo cardiogram	74	10
heart disease	294	13
hepatitis	155	19
labor relations	40	16
liver	345	6

Learning algorithms:

- WV — weighted voting (boosting);
- DL — decision list;
- LR — logistic regression.

Testing method: 10-fold cross validation.

Experiment on real data sets. Results

	tasks					
Algorithm	austr	echo	heart	hepa	labor	liver
RIPPER-opt	15.5	2.97	19.7	20.7	18.0	32.7
RIPPER+opt	15.2	5.53	20.1	23.2	18.0	31.3
C4.5(Tree)	14.2	5.51	20.8	18.8	14.7	37.7
C4.5(Rules)	15.5	6.87	20.0	18.8	14.7	37.5
C5.0	14.0	4.30	21.8	20.1	18.4	31.9
SLIPPER	15.7	4.34	19.4	17.4	12.3	32.2
LR	14.8	4.30	19.9	18.8	14.2	32.0
WV	14.9	4.37	20.1	19.0	14.0	32.3
DL	15.1	4.51	20.5	19.5	14.7	35.8
WV+CS	14.1	3.2	19.3	18.1	13.4	30.2
DL+CS	14.4	3.6	19.5	18.6	13.6	32.3

Two top results are highlighted for each task.

Conclusions

- Combinatorial framework can give tight and sometimes exact generalization bounds.
- OC (one-classifier) bound is exact.
- UC (uniform connectivity) bound rely on *connectivity* but neglect *splitting*.
- SC (splitting and connectivity) bound is most tight and even *exact* for monotone chains and lattices of classifiers.
- SC-bound being applied to rule induction reduces testing error of classifiers by 1–2%.

Questions, please

Konstantin Vorontsov

vokov@forecsys.ru

http://www.ccas.ru/voron

www.MachineLearning.ru/wiki (in Russian):

- Участник:Vokov
- Слабая вероятностная аксиоматика
- Расслоение и сходство алгоритмов (виртуальный семинар)