Generalization bounds based on the splitting and connectivity properties of a set of classifiers

Konstantin Vorontsov, Andrey Ivahnenko, Denis Kochedykov, Pavel Botov, Ilya Reshetnyak, Eugene Sokolov

Computing Center RAS • MIPT • MSU, Moscow

10th International Conference
PATTERN RECOGNITION and IMAGE ANALYSIS: NEW INFORMATION TECHNOLOGIES
St. Petersburg, Russian Federation • December 5-12, 2010
1. **Combinatorial framework for generalization bounds**
   - Probability of overfitting
   - Weak (permutational) probabilistic assumptions
   - OC-bound and VC-bound

2. **Splitting and Connectivity (SC-) bounds**
   - SC-graph, UC-bound and SC-bound
   - SC-bound is exact for some model sets of classifiers
   - Proofs technique: generating and inhibiting subsets

3. **Application of SC-bound to rule induction**
   - SC-bound for conjunctive rules induction
   - SC-modification of rule evaluation heuristics
   - Experiments and conclusions
Learning with binary loss

\( X^L = \{x_1, \ldots, x_L\} \) — a finite universe set of objects;
\( A = \{a_1, \ldots, a_D\} \) — 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:

<table>
<thead>
<tr>
<th></th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
<th>( a_4 )</th>
<th>( a_5 )</th>
<th>( a_6 )</th>
<th>( \ldots )</th>
<th>( a_D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>( \ldots )</td>
<td>1</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>( \ldots )</td>
<td>1</td>
</tr>
<tr>
<td>( x_\ell )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( \ldots )</td>
<td>0</td>
</tr>
<tr>
<td>( x_{\ell+1} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( \ldots )</td>
<td>0</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>( \ldots )</td>
<td>1</td>
</tr>
<tr>
<td>( x_L )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( \ldots )</td>
<td>0</td>
</tr>
</tbody>
</table>

\( X \) — observable (training) sample of size \( \ell \);
\( \bar{X} \) — hidden (testing) sample of size \( k = L - \ell \);

\( n(a) \) — *number of errors* of a classifier \( a \) on the set \( X^L \);
\( n(a, X) \) — *number of errors* of a classifier \( a \) on a sample \( X \subset X^L \);
\( \nu(a, X) = n(a, X)/|X| \) — *error rate* of \( a \) on a sample \( X \subset X^L \);
Example. The loss matrix for a set of linear classifiers

<table>
<thead>
<tr>
<th>no errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

1 vector having no errors
Example. The loss matrix for a set of linear classifiers

<table>
<thead>
<tr>
<th></th>
<th>no errors</th>
<th>1 error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1 0 0 0 0 0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>0 1 0 0 0 0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
<td>0 0 1 0 0 0</td>
</tr>
<tr>
<td>$x_4$</td>
<td>0</td>
<td>0 0 0 1 0 0</td>
</tr>
<tr>
<td>$x_5$</td>
<td>0</td>
<td>0 0 0 0 0 1</td>
</tr>
<tr>
<td>$x_6$</td>
<td>0</td>
<td>0 0 0 0 0 0</td>
</tr>
<tr>
<td>$x_7$</td>
<td>0</td>
<td>0 0 0 0 0 0</td>
</tr>
<tr>
<td>$x_8$</td>
<td>0</td>
<td>0 0 0 0 0 0</td>
</tr>
<tr>
<td>$x_9$</td>
<td>0</td>
<td>0 0 0 0 0 0</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>0</td>
<td>0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
Combinatorial framework for generalization bounds
Splitting and Connectivity (SC-) bounds
Application of SC-bound to rule induction

Example. The loss matrix for a set of linear classifiers

<table>
<thead>
<tr>
<th>no errors</th>
<th>1 error</th>
<th>2 errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>0 1 0 0 0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
<td>0 0 1 0 0</td>
</tr>
<tr>
<td>$x_4$</td>
<td>0</td>
<td>0 0 0 1 0</td>
</tr>
<tr>
<td>$x_5$</td>
<td>0</td>
<td>0 0 0 0 1</td>
</tr>
<tr>
<td>$x_6$</td>
<td>0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>$x_7$</td>
<td>0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>$x_8$</td>
<td>0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>$x_9$</td>
<td>0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>0</td>
<td>0 0 0 0 0</td>
</tr>
</tbody>
</table>

1 vector having no errors
5 vectors having 1 error
8 vectors having 2 errors
Def. The *learning algorithm* $\mu : X \mapsto a$ takes a training sample $X \subset X^L$ and returns a classifier $a \equiv \mu X \in A$.

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

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

**Def. Probability of overfitting**

$$Q_\varepsilon(\mu, X^L) = P[\delta(\mu, X) \geq \varepsilon].$$

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

Def. *Upper bound*: $Q_\varepsilon \leq \eta(\varepsilon)$. 
Weak (permutational) probabilistic assumptions

Axiom

All partitions \(X^L = \{x_1, \ldots, x_L\} = X \sqcup \tilde{X}\) are equiprobable, where

\(X\) — observable training sample of size \(\ell\);

\(\tilde{X}\) — hidden testing sample of size \(k = L - \ell\);

Probability is defined as a fraction of partitions:

\[
Q_\varepsilon = P\left[\delta(\mu, X) \geq \varepsilon\right] = \frac{1}{C^L_\ell} \sum_{X, \tilde{X}} [\delta(\mu, X) \geq \varepsilon].
\]

Interpretation: Only independence of observations is postulated. Continuous measures, infinite sets, and limits \(|X| \rightarrow \infty\) are illegal.

Nevertheless, tight generalization bounds can be obtained!
Let $A = \{a\}$, $m = n(a)$. Obviously, $\mu_X = a$ for all $X \subset \mathbb{X}^L$.

**Definition**

Hypergeometric distribution function:

PDF: $h_{\ell, m}^L(s) = P\left[n(a, X) = s\right] = \frac{C_s^m C_{\ell-s}^{L-m}}{C_{\ell}^L}$;

CDF: $H_{\ell, m}^L(z) = P\left[n(a, X) \leq z\right] = \sum_{s=0}^{[z]} h_{\ell, m}^L(s)$.

**Theorem (exact OC-bound)**

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

$$Q_\varepsilon = H_{\ell, m}^L(s_m(\varepsilon)),$$

where $s_m(\varepsilon) = \frac{\ell}{L}(m-\varepsilon k)$. 
Hypergeometric distribution, PDF \( h_L^m(s) = C_m^s C_{L-m}^{\ell-s} / C_L^{\ell} \)

Distribution is concentrated along diagonal \( s \approx \frac{\ell}{L} m \), thus allowing to predict both \( n(a) = m \) and \( n(a, \bar{X}) = m - s \) 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 $X^L$, $A$, $\mu$, and $\varepsilon \in (0, 1)$

$$Q_\varepsilon = P\left[\nu(\mu X, \bar{X}) - \nu(\mu X, X) \geq \varepsilon\right] \leq$$

**STEP 1:** *uniform bound* makes the result independent on $\mu$:

$$\leq \tilde{Q}_\varepsilon = P \max_{a \in A} \left[\nu(a, \bar{X}) - \nu(a, X) \geq \varepsilon\right] \leq$$

**STEP 2:** *union bound* (which is usually highly overestimated):

$$\leq P \sum_{a \in A} \left[\nu(a, \bar{X}) - \nu(a, X) \geq \varepsilon\right] =$$

exact one-classifier bound:

$$= \sum_{a \in A} H_{L, m}^\ell \left(s_m(\varepsilon)\right), \quad m = n(a).$$
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)**

$$Q_{\varepsilon} = H_{L}^m (s_m(\varepsilon)), \quad m = n(a).$$

**Theorem (VC-bound)**

$$Q_{\varepsilon} \leq \tilde{Q}_{\varepsilon} \leq \sum_{a \in A} H_{L}^m (s_m(\varepsilon)), \quad m = n(a).$$

VC-bound is highly overestimated because of 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]
Define two binary relations on classifiers:

- **partial order** \( a \leq b \): \( I(a, x) \leq I(b, x) \) for all \( x \in X^L \);
- **precedence** \( a \prec b \): \( a \leq b \) 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;  
- \( E = \{(a, b) : a \prec b \} \).

**Properties of the SC-graph:**

- each edge \((a, b)\) is labeled by an object \( x_{ab} \in X^L \) such that \( 0 = I(a, x_{ab}) < I(b, x_{ab}) = 1 \);
- multipartite graph with layers \( A_m = \{a \in A : n(a) = m\} \), \( m = 0, \ldots, L + 1 \);
Combinatorial framework for generalization bounds
Splitting and Connectivity (SC-) bounds
Application of SC-bound to rule induction

SC-graph, UC-bound and SC-bound
SC-bound is exact for some model sets of classifiers
Proofs technique: generating and inhibiting subsets

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

<table>
<thead>
<tr>
<th>layer 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
</tr>
<tr>
<td>$x_2$</td>
</tr>
<tr>
<td>$x_3$</td>
</tr>
<tr>
<td>$x_4$</td>
</tr>
<tr>
<td>$x_5$</td>
</tr>
<tr>
<td>$x_6$</td>
</tr>
<tr>
<td>$x_7$</td>
</tr>
<tr>
<td>$x_8$</td>
</tr>
<tr>
<td>$x_9$</td>
</tr>
<tr>
<td>$x_{10}$</td>
</tr>
</tbody>
</table>
Example. Loss matrix and SC-graph for a set of linear classifiers

<table>
<thead>
<tr>
<th>Layer 0</th>
<th>Layer 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
</tr>
<tr>
<td>$x_4$</td>
<td>0</td>
</tr>
<tr>
<td>$x_5$</td>
<td>0</td>
</tr>
<tr>
<td>$x_6$</td>
<td>0</td>
</tr>
<tr>
<td>$x_7$</td>
<td>0</td>
</tr>
<tr>
<td>$x_8$</td>
<td>0</td>
</tr>
<tr>
<td>$x_9$</td>
<td>0</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>0</td>
</tr>
</tbody>
</table>
Example. Loss matrix and SC-graph for a set of linear classifiers

<table>
<thead>
<tr>
<th>layer 0</th>
<th>layer 1</th>
<th>layer 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_4$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_5$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_6$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_7$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_8$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_9$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

SC-graph, UC-bound and SC-bound
SC-bound is exact for some model sets of classifiers
Proofs technique: generating and inhibiting subsets
Def. *Connectivity* of a classifier $a \in A$

\[
p(a) = \# \{ x_{ba} \in X^L : b \prec a \} \quad \text{— low-connectivity.}
\]
\[
q(a) = \# \{ x_{ab} \in X^L : a \prec b \} \quad \text{— up-connectivity;}
\]

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

\[
r(a) = \# \{ x_{cb} \in X^L : c \prec b \preceq a \} \in \{ p(a), \ldots, n(a) \}.
\]

Example:

\[
p(a) = \# \{ x_1, x_2 \} = 2,
\]
\[
q(a) = \# \{ x_3, x_4 \} = 2,
\]
\[
r(a) = \# \{ x_1, x_2 \} = 2.
\]
Uniform Connectivity (UC-) bound

**Theorem (UC-bound)**

For all \( X^L, \mu, A \) and \( \varepsilon \in (0, 1) \)

\[
\tilde{Q}_\varepsilon \leq \sum_{a \in A} \left[ p \leq k \right] \left( \frac{C_{L-q-p}^L}{C_{L-q-p}^L} \right) H_{L-q-p}^{L-q-p}(s_{m}(\varepsilon))
\]

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

1. UC-bound improves the VC-bound, even if \( p(a) \equiv q(a) \equiv 0 \):

\[
\tilde{Q}_\varepsilon \leq \sum_{a \in A} H_{L}^{L,m}(s_{m}(\varepsilon)).
\]

2. The contribution of \( a \in A \) decreases exponentially by \( p(a) \)

\[ \Rightarrow \text{connected sets are less subjected to overfitting}. \]

3. UC-bound relies on connectivity, but disregards splitting.
Pessimistic Empirical Risk Minimization

**Definition (ERM)**

*Learning algorithm $\mu$ is Empirical Risk Minimization if*

$$\mu X \in A(X), \quad A(X) = \operatorname{Arg \min}_{a \in A} n(a, X);$$

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 $\mu$ is pessimistic ERM if*

$$\mu X = \operatorname{arg \max}_{a \in A(X)} n(a, \bar{X});$$
Theorem (SC-bound)

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

\[
Q_\varepsilon \leq \sum_{a \in A} \left[ r \leq k \right] \left( \frac{C^L_{L-q-r}}{C^L_{L}} \right) H^{L-q, m-r}_{L-q-r} (s_m(\varepsilon)),
\]

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:

\[
Q_\varepsilon \leq \sum_{a \in A} H^{L, m}_{L} (s_m(\varepsilon)).
\]

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 \( Q_\varepsilon \).
Separable two-dimensional task, $L = 100$, two classes.
Separable two-dimensional task, $L = 100$, two classes.
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:**

\[
\begin{array}{c}
m=3 \quad \cdots \quad a_3 \\
m=2 \quad \cdots \quad a_2 \\
m=1 \quad \cdots \quad a_1 \\
m=0 \quad \cdots \quad a_0
\end{array}
\]

\[
\begin{array}{c}
x_3 \\
x_2 \\
x_1
\end{array}
\]

**Loss matrix:**

\[
\begin{array}{c|cccc}
 & a_0 & a_1 & a_2 & a_3 \\
\hline
x_1 & 0 & 1 & 1 & 1 \\
x_2 & 0 & 0 & 1 & 1 \\
x_3 & 0 & 0 & 0 & 1 \\
x_4 & 0 & 0 & 0 & 0 \\
x_5 & 0 & 0 & 0 & 0 \\
x_6 & 0 & 0 & 0 & 0 \\
\end{array}
\]
Two-dimensional monotone lattice of classifiers

Example:

2-dimensional linear classifiers, 2 classes \{•, ○\}, 6 objects

SC-graph:

<table>
<thead>
<tr>
<th>m=3</th>
<th>a_{03}</th>
<th>a_{12}</th>
<th>a_{21}</th>
<th>a_{30}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>x_{5}</td>
<td></td>
<td>x_{6}</td>
</tr>
<tr>
<td>m=2</td>
<td>a_{02}</td>
<td>a_{11}</td>
<td>a_{20}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>x_{3}</td>
<td></td>
<td>x_{4}</td>
</tr>
<tr>
<td>m=1</td>
<td>a_{01}</td>
<td>a_{10}</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>x_{1}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m=0</td>
<td>a_{00}</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Loss matrix:

<table>
<thead>
<tr>
<th></th>
<th>a_{00}</th>
<th>a_{01}</th>
<th>a_{10}</th>
<th>a_{02}</th>
<th>a_{11}</th>
<th>a_{20}</th>
<th>a_{03}</th>
<th>a_{12}</th>
<th>a_{21}</th>
<th>a_{30}</th>
</tr>
</thead>
<tbody>
<tr>
<td>x_1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>x_2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x_3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>x_4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x_5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>x_6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
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 $D$*:

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

**Theorem (exact SC-bound)**

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

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

For any \( a \in A \) generating set \( X_a \subset X^L \) and inhibiting set \( X'_a \subset X^L \) 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:

\[
[\mu X = a] \leq [X_a \subseteq X][X'_a \subseteq \bar{X}].
\]
Lemma (Probability of obtaining each of classifiers)

*If Conjecture is true then for any \( \mu, X, a \in A \)*

\[
P[\mu X = a] \leq P_a = C_{L_a}^{\ell_a} / C_L.
\]

where \( L_a = L - |X_a| - |X_a'| \), \( \ell_a = \ell - |X_a| \).

Theorem (Probability of overfitting)

*If Conjecture is true then for any \( X^L, \mu, A \) and \( \varepsilon \in (0, 1) \)*

\[
Q_\varepsilon \leq \sum_{a \in A} P_a H_{L_a}^{\ell_a, m_a} (s_a(\varepsilon)),
\]

where \( m_a = n(a, X^L) - n(a, X_a) - n(a, X_a') \),

\[
s_a(\varepsilon) = \frac{\ell}{L} (n(a, X^L) - \varepsilon k) - n(a, X_a).
\]
Upper connectivity of a classifier \( a \in A \)
\[
q(a) = |X_a|, \quad X_a = \{x_{ab} \in \mathbb{X}^L : a < b\} \quad \text{— generating subset.}
\]

Inferiority of a classifier \( a \in A \)
\[
r(a) = |X'_a|, \quad X'_a = \{x_{cb} \in \mathbb{X}^L : c < b \leq a\} \quad \text{— inhibiting subset.}
\]
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 \(y\),
\(r: X \rightarrow \{0, 1\}\) — rule, and \(w_r\) — its weight.

Conjunctive rule:

\[
r(x) = \bigwedge_{j \in J} \left[ f_j(x) \leq \theta_j \right],
\]

where \(f_j(x)\) — real features, \(\theta_j\) — thresholds, \(j = 1, \ldots, n\);
\(J \subseteq \{1, \ldots, n\}\) — subset of features, usually \(|J| \lesssim 7\);
Intrinsically the rule learning is a two-criteria optimization problem:

\[ N(r, X) = \frac{1}{|X|} \# \{ x_i \in X : r(x_i) = 1, y_i \neq y \} \rightarrow \min_r \]

\[ P(r, X) = \frac{1}{|X|} \# \{ x_i \in X : r(x_i) = 1, y_i = y \} \rightarrow \max_r \]

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

- Information gain;
- Gini Index;
- Fisher exact test, \( \chi^2 \) or \( \omega^2 \) statistical tests, etc.

A common drawback of all these criteria:

Ignoring an overfitting that results from thresholds \( \theta_j \) 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

Training error, %

Testing error, %

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:
   \[
   P\left[ N(r, \bar{X}) - N(r, X) \geq \varepsilon \right] \leq \eta_N(\varepsilon); \\
   P\left[ P(r, X) - P(r, \bar{X}) \geq \varepsilon \right] \leq \eta_P(\varepsilon);
   \]
2. Invert SC-bounds: with probability at least $1 - \eta$
   \[
   N(r, \bar{X}) \leq \hat{N}(r, \bar{X}) = N(r, X) + \varepsilon_N(\eta); \\
   P(r, \bar{X}) \geq \hat{P}(r, \bar{X}) = P(r, X) - \varepsilon_P(\eta).
   \]
3. Substitute $\hat{P}, \hat{N}$ in a one-criterion heuristic: $H(\hat{P}, \hat{N}) \rightarrow \max_r$.
Classes of equivalent rules: one point per rule

Example: separable 2-dimensional task, $L = 10$, two classes.

rules: $r(x) = [f_1(x) \leq \theta_1$ and $f_2(x) \leq \theta_2]$. 

\[ \begin{align*} 
\theta_2 & \quad 10 \\
9 & \quad 8 \quad 7 \quad 6 \quad 5 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0 \\
6 & \quad 5 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0 \\
4 & \quad 3 \quad 2 \quad 1 \quad 0 \\
2 & \quad 1 \quad 0 \\
0 & \quad 5 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0 \\
\theta_1 & \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \\
\end{align*} \]

Class 0(5)  Class 1(5)
Classes of equivalent rules: one point per class

**Example:** the same classification task. **One point per class.**
rules: $r(x) = \left[ f_1(x) \leq \theta_1 \text{ and } f_2(x) \leq \theta_2 \right]$. 

![Diagram showing classes of equivalent rules](image-url)
Example: SC-graph isomorphic to the graph at previous slide.
Experiment on real data sets

Data sets from UCI repository:

<table>
<thead>
<tr>
<th>Task</th>
<th>Objects</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>australian</td>
<td>690</td>
<td>14</td>
</tr>
<tr>
<td>echo cardiogram</td>
<td>74</td>
<td>10</td>
</tr>
<tr>
<td>heart disease</td>
<td>294</td>
<td>13</td>
</tr>
<tr>
<td>hepatitis</td>
<td>155</td>
<td>19</td>
</tr>
<tr>
<td>labor relations</td>
<td>40</td>
<td>16</td>
</tr>
<tr>
<td>liver</td>
<td>345</td>
<td>6</td>
</tr>
</tbody>
</table>

Learning algorithms:

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

Testing method: 10-fold cross validation.
## Experiment on real data sets. Results

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

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


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