

Combinatorial theory of overfitting

Konstantin Vorontsov^{1,2,3} voron@forecsys.ru
Alexander Frey¹ sashafrey@gmail.com
Evgeny Sokolov² sokolov.evg@gmail.com

¹Moscow Institute of Physics and Technology

²Moscow State University

³Computing Center RAS

International Workshop on Statistical Learning (IWSL)

Moscow, Russia • June 26 – June 28, 2013

- 1 Combinatorial framework for generalization bounds**
 - Probability of overfitting
 - Links to Cross-Validation and Rademacher Complexity
 - Splitting-Connectivity bounds

- 2 Combinatorial theory of overfitting: overview**
 - Model sets for which exact bounds are known
 - Bound computation via Random Walks
 - Making bounds observable

- 3 Applications to learning algorithms design**
 - Ensembles of Conjunction Rules
 - Ensembles of low-dimensional Linear Classifiers
 - Comparing with state-of-art PAC-Bayesian bounds

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

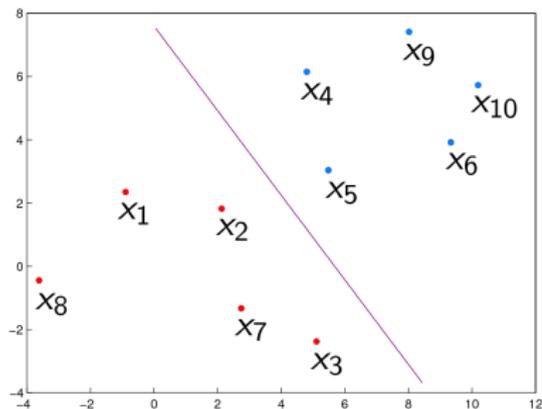
Loss matrix of size $L \times D$, all columns are distinct:

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

$a \mapsto (I(a, x_i))_{i=1}^L$ — binary loss vector of classifier a

$\nu(a, X) = \frac{1}{|X|} \sum_{x \in X} I(a, x)$ — error rate of a on a sample $X \subset \mathbb{X}^L$

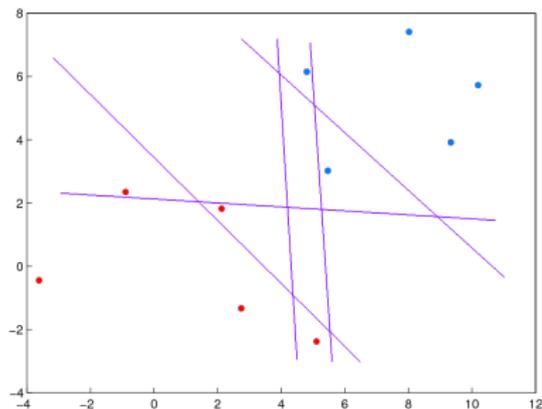
Example. The loss matrix for a set of linear classifiers



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

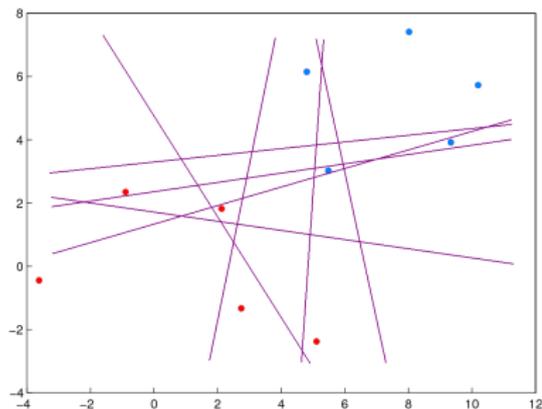
Example. The loss matrix for a set of linear classifiers



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 large overfitting

$\mu: X \mapsto a$ — learning algorithm

$\nu(\mu X, X)$ — training error rate

$\nu(\mu X, \bar{X})$ — testing error rate

$\delta(\mu, X) \equiv \nu(\mu X, \bar{X}) - \nu(\mu X, X)$ — overfitting of μ on X and \bar{X}

Axiom (weaken i.i.d. assumption)

\mathbb{X}^L is not random, all partitions $\mathbb{X}^L = X \sqcup \bar{X}$ are equiprobable,

X — observable training sample of a fixed size ℓ ,

\bar{X} — hidden testing sample of a fixed size k , $L = \ell + k$

Def. Probability of large overfitting

$$Q_\varepsilon(\mu, \mathbb{X}^L) = \mathbf{P}[\delta(\mu, X) \geq \varepsilon] = \frac{1}{C_L^\ell} \sum_{X \subset \mathbb{X}^L} [\delta(\mu, X) \geq \varepsilon]$$

Bounding problems

- Probability of large overfitting:

$$Q_\varepsilon(\mu, \mathbb{X}^L) = \mathbb{P}[\delta(\mu, \mathbf{X}) \geq \varepsilon] \leq ?$$

- Probability of large testing error:

$$R_\varepsilon(\mu, \mathbb{X}^L) = \mathbb{P}[\nu(\mu \mathbf{X}, \bar{\mathbf{X}}) \geq \varepsilon] \leq ?$$

- Expectation of OverFitting:

$$\text{EOF}(\mu, \mathbb{X}^L) = \mathbb{E} \delta(\mu, \mathbf{X}) \leq ?$$

- Expectation of testing error (Complete Cross-Validation):

$$\text{CCV}(\mu, \mathbb{X}^L) = \mathbb{E} \nu(\mu \mathbf{X}, \bar{\mathbf{X}}) \leq ?$$

Links to Cross-Validation

Taking expectation is equivalent to averaging over all partitions.
Expected testing error also called Complete Cross-Validation:

$$\text{CCV}(\mu, \mathbb{X}^L) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \nu(\mu X, \bar{X})$$

Usual cross-validation techniques (e.g. hold-out, t -fold, $q \times t$ -fold, partition sampling, etc.) can be viewed as empirical measurement of CCV by averaging over a subset of partitions.

Leave-One-Out is equivalent to CCV for the case $k = 1$.

:) Combinatorial functionals Q_ε , R_ε , CCV, EOF can be easily measured empirically by generating $\sim 10^3$ random partitions.

Links to Local Rademacher Complexity

Def. *Local Rademacher complexity* of the set A on \mathbb{X}^L

$$\mathcal{R}(A, \mathbb{X}^L) = \mathbb{E}_\sigma \sup_{a \in A} \frac{2}{L} \sum_{i=1}^L \sigma_i l(a, x_i), \quad \sigma_i = \begin{cases} +1, & \text{prob. } \frac{1}{2} \\ -1, & \text{prob. } \frac{1}{2} \end{cases}$$

$\sigma_1, \dots, \sigma_L$ — independent Rademacher random variables.

Expected overfitting is almost the same thing for the case $\ell = k$:

$$\text{EOF}(\mu, \mathbb{X}^L) = \mathbb{E} \sup_{a \in A} \frac{2}{L} \sum_{i=1}^L \sigma_i l(a, x_i), \quad \sigma_i = \begin{cases} +1, & x_i \in \bar{X} \\ -1, & x_i \in X \end{cases}$$

where μ is *overfitting maximization* (very unnatural learning!):

$$\mu X = \arg \max_{a \in A} \left(\nu(\mu X, \bar{X}) - \nu(\mu X, X) \right)$$

Links to usual SLT framework

Usual probabilistic assumptions:

\mathbb{X}^L is i.i.d. from probability space $\langle \mathcal{X}, \sigma, P \rangle$ on infinite \mathcal{X}

Transferring of combinatorial generalization bound to i.i.d. framework first used in (Vapnik and Chervonenkis, 1971):

- 1 Give a combinatorial bound on probability of large overfitting:

$$P_{X \sim \mathbb{X}^L} [\delta(\mu, X) \geq \varepsilon] = Q_\varepsilon(\mu, \mathbb{X}^L) \leq \eta(\varepsilon, \mathbb{X}^L)$$

- 2 Take expectation on \mathbb{X}^L :

$$\begin{aligned} P_{X \sim \mathcal{X}^l, \bar{X} \sim \mathcal{X}^k} [\nu(\mu X, \bar{X}) - \nu(\mu X, X) \geq \varepsilon] &= \\ &= E_{\mathbb{X}^L} Q_\varepsilon(\mu, \mathbb{X}^L) \leq E_{\mathbb{X}^L} \eta(\varepsilon, \mathbb{X}^L). \end{aligned}$$

Splitting-Connectivity graph (1-inclusion graph)

Define two binary relations on classifiers:

partial order $a \leq b$: $I(a, x) \leq I(b, x)$ for all $x \in \mathbb{X}^L$;

precedence $a \prec b$: $a \leq b$ and Hamming distance $\|b - a\| = 1$.

Definition (SC-graph)

Splitting and Connectivity (SC-) graph $\langle A, E \rangle$:

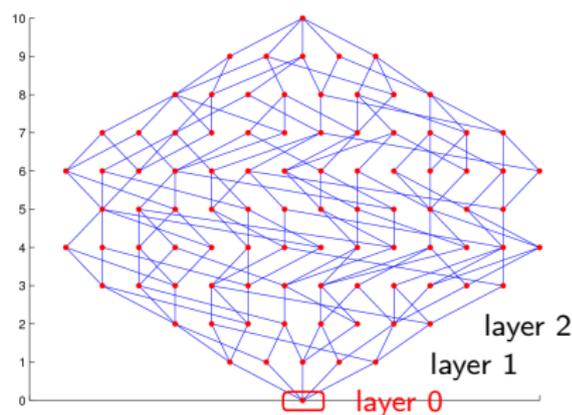
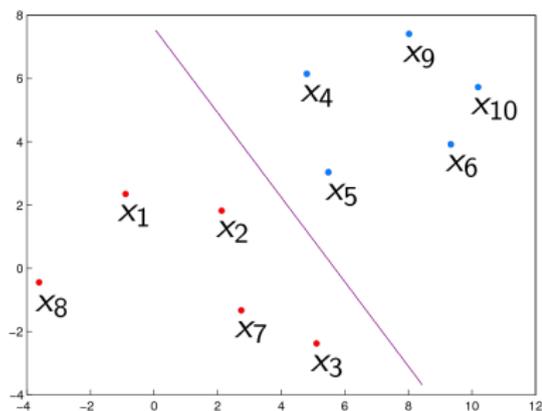
A — a set of classifiers with distinct binary loss vectors;

$E = \{(a, b) : a \prec b\}$.

Properties of the SC-graph:

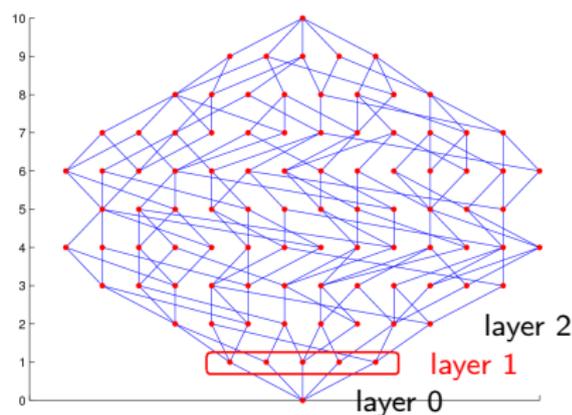
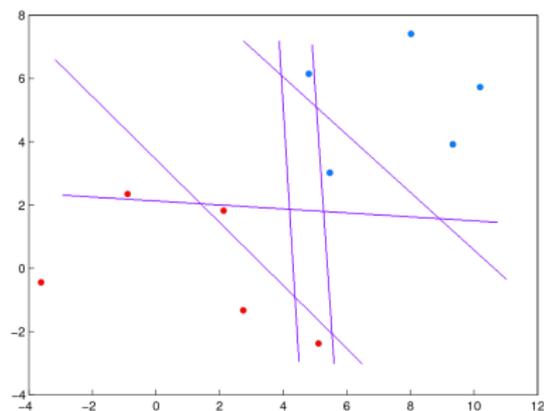
- each edge (a, b) is labeled by an object $x_{ab} \in \mathbb{X}^L$ such that $0 = I(a, x_{ab}) < I(b, x_{ab}) = 1$;
- multipartite graph with layers $A_m = \{a \in A : \nu(a, \mathbb{X}^L) = \frac{m}{L}\}$, $m = 0, \dots, L + 1$;

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



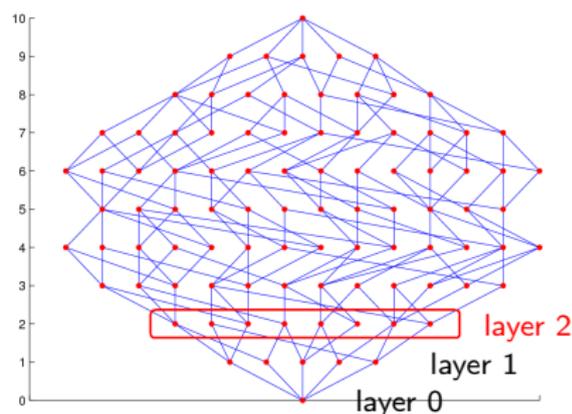
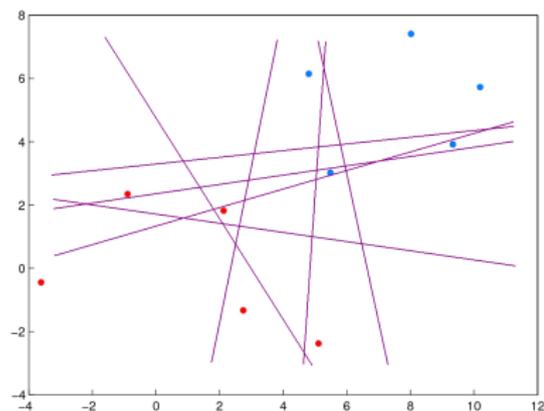
	layer 0
x_1	0
x_2	0
x_3	0
x_4	0
x_5	0
x_6	0
x_7	0
x_8	0
x_9	0
x_{10}	0

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	1	1	0	...
X ₂	0	0	1	0	0	0	1	1	0	0	0	0	0	...
X ₃	0	0	0	1	0	0	0	1	1	0	0	0	1	...
X ₄	0	0	0	0	1	0	0	0	1	1	0	0	0	...
X ₅	0	0	0	0	0	1	0	0	0	1	1	0	0	...
X ₆	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	1	...
X ₈	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	...
X ₁₀	0	0	0	0	0	0	0	0	0	0	0	0	0	...

Connectivity and inferiority of a classifier

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

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

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

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

$q(a) = \#\{x_{cb} \in \mathbb{X}^L : \exists b \ c \prec b \leq a\}$

Inferiority:

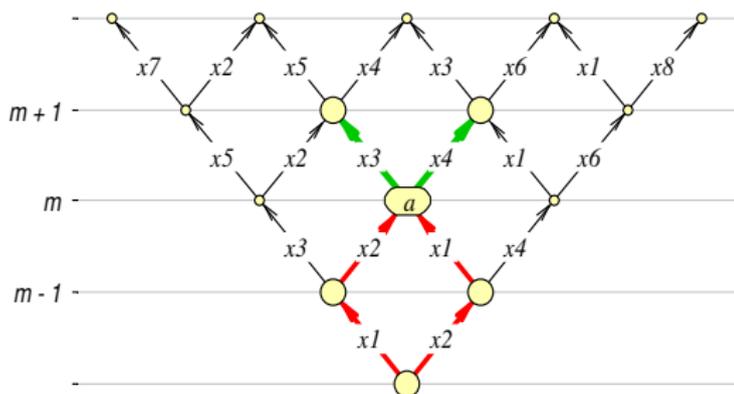
$d(a) \leq q(a) \leq m(a, \mathbb{X}^L)$

Example:

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

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

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



The Splitting-Connectivity (SC-) bound

Empirical Risk Minimization (ERM) — learning algorithm μ :

$$\mu X \in A(X), \quad A(X) = \text{Arg min}_{a \in A} \nu(a, X)$$

Theorem (SC-bound)

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

$$Q_\varepsilon \leq \sum_{a \in A} \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell} H_{L-u-q}^{\ell-u, m-q}(\varepsilon),$$

where $m = L\nu(a, \mathbb{X}^L)$, $u = u(a)$, $q = q(a)$,

$$H_L^{\ell, m}(\varepsilon) = \sum_{s=0}^{\lfloor (m-\varepsilon k)\ell/L \rfloor} \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell} - \text{hypergeometric tail function.}$$

The properties of the SC-bound

$$Q_\varepsilon \leq \sum_{a \in A} \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell} H_{L-u-q}^{\ell-u, m-q}(\varepsilon)$$

- 1 If $|A| = 1$ then SC-bound gives an exact estimate of testing error for a single classifier:

$$Q_\varepsilon = P[\nu(a, \bar{X}) - \nu(a, X) > \varepsilon] = H_L^{\ell, m}(\varepsilon) \stackrel{\ell=k}{\leq} \frac{3}{2} e^{-\varepsilon^2 \ell}$$

- 2 Substitution $u(a) \equiv q(a) \equiv 0$ transforms the SC-bound into Vapnik–Chervonenkis bound:

$$Q_\varepsilon \leq \sum_{a \in A} H_L^{\ell, m}(\varepsilon) \stackrel{\ell=k}{\leq} |A| \cdot \frac{3}{2} e^{-\varepsilon^2 \ell}$$

The properties of the SC-bound

$$Q_\varepsilon \leq \sum_{a \in A} \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell} H_{L-u-q}^{\ell-u, m-q}(\varepsilon)$$

- 4 The probability to get a classifier a as a result of learning:

$$P[\mu X = a] \leq \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell}$$

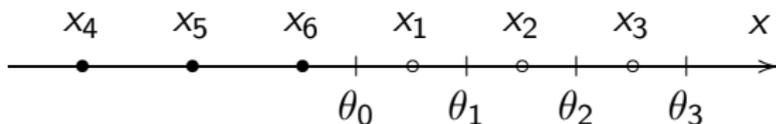
- 5 The contribution of $a \in A$ decreases exponentially by:
 $u(a) \Rightarrow$ **connected sets are less subjected to overfitting;**
 $q(a) \Rightarrow$ **only lower layers contribute significantly to Q_ε .**
- 6 The SC-bound is *exact* for some nontrivial sets of classifiers.

Monotone chain of classifiers

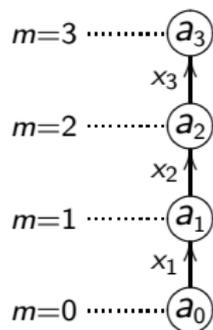
Def. *Monotone chain* of classifiers: $a_0 \prec a_1 \prec \dots \prec a_D$.

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

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



SC-graph:



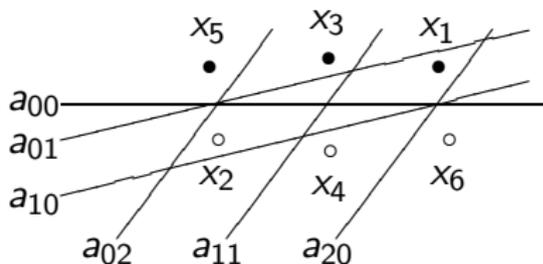
Loss matrix:

	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

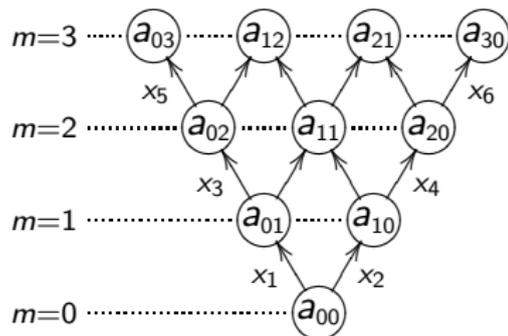
Two-dimensional monotone lattice of classifiers

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

Sets of classifiers with known combinatorial bounds

Model sets of classifiers with **exact** SC-bound:

- monotone and unimodal n -dimensional lattices (Botov, 2010)
- pencils of monotone chains (Frey, 2011)
- intervals in boolean cube and their slices (Vorontsov, 2009)
- Hamming balls in boolean cube and their slices (Frey, 2010)
- sparse subsets of lattices and Hamming balls (Frey, 2011)

Real sets of classifiers with **tight** computable SC-bound:

- conjunction rules (Ivahnenko, 2010)
- linear classifiers (Sokolov, 2012)
- decision stumps or arbitrary chains (Ishkina, 2013)

Real sets of classifiers with **exact** computable not-SC bound:

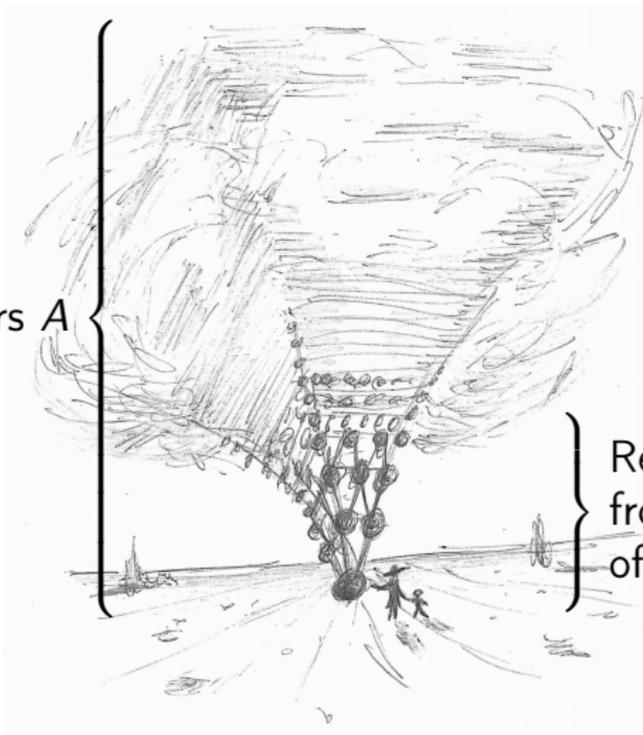
- k nearest neighbor classification (Vorontsov, 2004; Ivanov, 2009)
- isotonic separation (Vorontsov and Makhina, 2011; Guz, 2011)

Some theoretical and practical results for model sets

- 1 Necessary and sufficient conditions for the SC-bound exactness:
 - 1) a graph must have unique source and unique sink
 - 2) for any $a \in A$ with n outgoing edges a set A must contain an n -dimensional cube induced by these edges
(Zhivotovskiy, 2012)
- 2 Low-dimensional unimodal lattice can be used to approximate a set of rules in decision tree nodes.
This helps to find less overfitted node splits
(Botov, 2011)
- 3 Covering a set of classifiers A by slices of Hamming balls helps to obtain tighter generalization bound
(Frey, 2013)

Splitting gives an idea of effective SC-bound computation

All classifiers A



Really used classifiers
from lower layers
of the set A

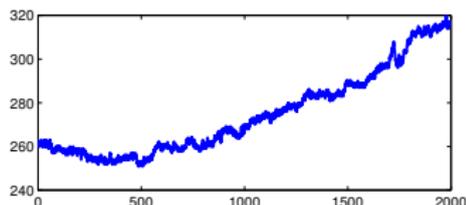
SC-bound computation via Random Walks

1. Learn a good classifier
2. Run a large number of short walks to get a subset $B \subset A$
3. Compute a partial sum $Q_\epsilon \approx \sum_{a \in B} \text{contribution}(a)$

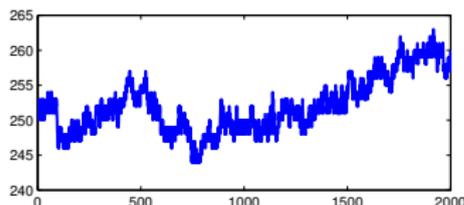
Special kind of Random Walks for multipartite graph:

- 1) based on Frontier sampling algorithm
- 2) do not permit to walk in higher layers of a graph
- 3) estimate contributions of layers separately

Simple random walk:



Random walk with gravitation:



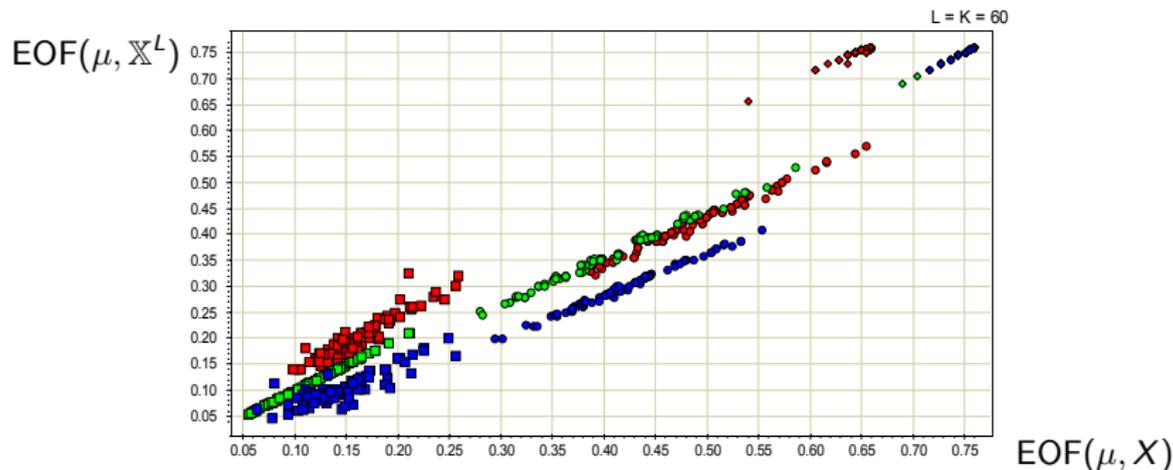
Open problem: making bounds observable

Seems to be a big problem:

SC-bound depends of a full loss matrix including a hidden sample \bar{X}

Really $\text{EOF}(\mu, X)$ is well concentrated near to $\text{EOF}(\mu, \mathbb{X}^L)$:

Experiments on model data, $L = 60$, testing sample size $K = 60$



Ensemble learning

2-class classification problem:

$(x_i, y_i)_{i=1}^L$ — training set, $x_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$

Ensemble — *weighted voting* of base weak classifiers $b_t(x)$:

$$a(x) = \text{sign} \sum_{t=1}^T w_t b_t(x)$$

The main idea:

to apply generalization bounds for a family of base classifiers

Our goals:

- 1) to reduce overfitting of base classifiers
- 2) to reduce T — the complexity of composition

Learning ensembles of Conjunction Rules

Conjunction rule is a simple well interpretable 1-class classifier:

$$r_y(x) = \bigwedge_{j \in J} [f_j(x) \lesseqgtr_j \theta_j],$$

where $f_j(x)$ — features

$J \subseteq \{1, \dots, n\}$ — a small subset of features

θ_j — thresholds

\lesseqgtr_j — one of the signs \leq or \geq

y — the class of the rule

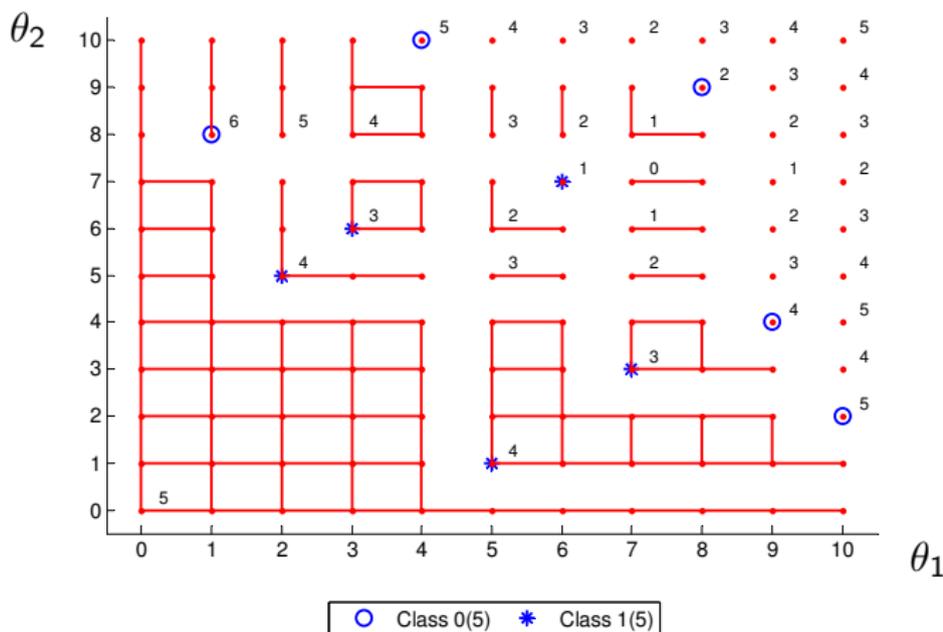
Weighted voting of rule sets R_y , $y \in Y$:

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

We use SC-bounds to reduce overfitting of rule learning

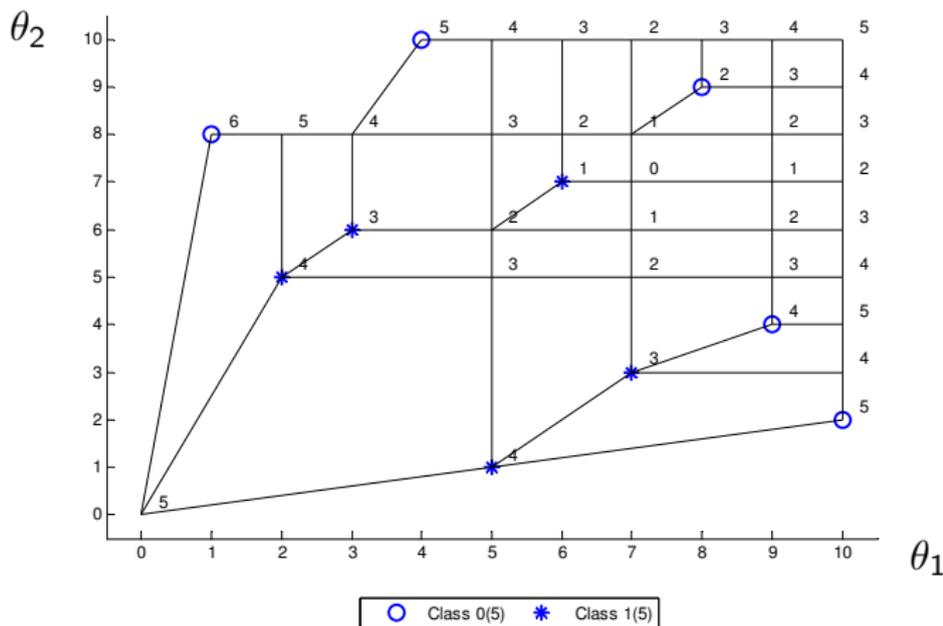
Classes of equivalent rules: one point per rule

Example: separable 2-dimensional task, $L = 10$, two classes.
 rules: $r(x) = [f_1(x) \leq \theta_1] \wedge [f_2(x) \leq \theta_2]$.



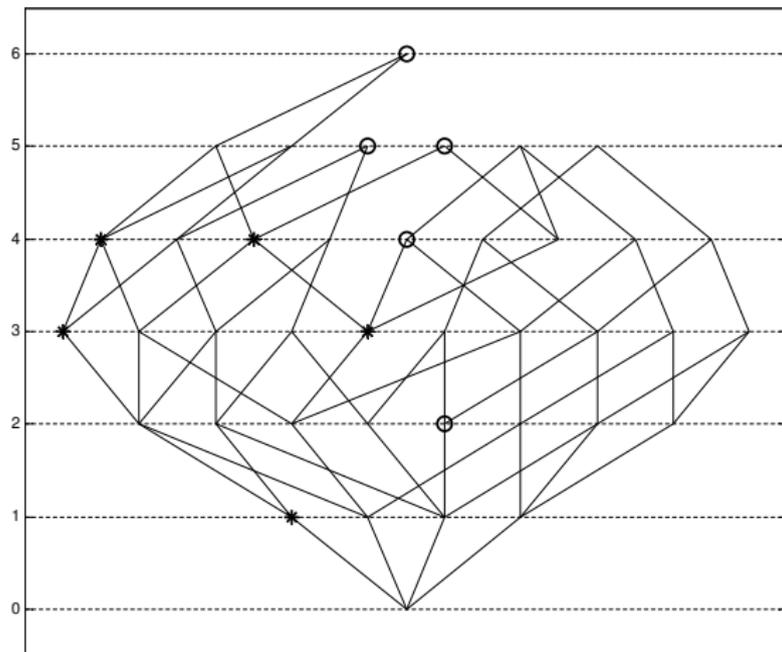
Classes of equivalent rules: one point per class

Example: the same classification task. **One point per class.**
 rules: $r(x) = [f_1(x) \leq \theta_1] \wedge [f_2(x) \leq \theta_2]$.



Classes of equivalent rules: SC-graph

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



Experiment on UCI 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
our WV	14.9	4.37	20.1	19.0	14.0	32.3
our WV + CS	14.1	3.2	19.3	18.1	13.4	30.2

Two top results are **highlighted** for each task.

Vorontsov K. V., Ivahnenko A. A. Tight Combinatorial Generalization Bounds for Threshold Conjunction Rules // LNCS. PReMI'11, 2011. Pp.66–73.

Liner classifiers and ensembles

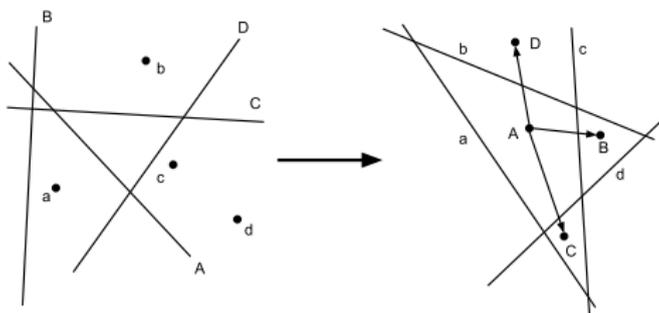
Linear classifier: $a(x) = \text{sign}\langle w, x \rangle$

Ensemble of low-dimensional linear classifiers

$$a(x) = \text{sign} \sum_{t=1}^T \text{th}\langle w_t, x \rangle$$

Random Walks for SC-bound computation

1) find all neighbor classifiers in the dual space:



2) lookup along random rays

Experiment 1: ensemble of liner classifiers

	statlog	waveform	wine	faults
ERM + MCCV	85,35	87,56	71,63	73,62
ERM + SC-bound	85,08	87,66	71,08	71,65
LR + MCCV	84,04	88,13	71,52	70,86
LR	80,77	87,34	71,49	71,09
PacBayes DD	82,13	87,17	64,68	67,67

The percentage of correct predictions on testing set (averaged over 5 partitions). Two top results for every task are shown in **bold**.

Feature selection criteria:

- ERM — learning by minimizing error rate from subset of classifiers sampled from random walks
- LR — learning by Logistic Regression
- MCCV — Monte-Carlo cross-validation
- DD — PAC-Bayes Dimension-Dependent bound (Jin, 2012)

Experiment 2: comparing bounds for Logistic Regression

All bounds are calculated from subset generated by random walk

- MC — Monte-Carlo bound (very slow)
- VC — Vapnik–Chervonenkis bound
- SC — Splitting-Connectivity bound
- DD — Dimension-Dependent PAC-Bayes bound (Jin, 2012)

UCI Task	Overfitting	MC	VC	SC	PAC DD
glass	0.029	0.078	0.211	0.140	0.738
liver	0.015	0.060	0.261	0.209	1.067
ionosphere	0.077	0.052	0.150	0.112	1.153
wdbc	0.055	0.032	0.071	0.043	0.705
australian	0.013	0.030	0.137	0.110	0.678
pima	0.007	0.028	0.159	0.127	0.749

Conclusions:

- 1) combinatorial bounds are much tighter than PAC-Bayes bounds
- 2) SC-bound initially proved for ERM fit well for Logistic Regression

Conclusions

Combinatorial framework

- gives tight (in some cases exact) generalization bounds
- that can be computed approximately from Random Walks
- bypass significantly state-of-art PAC-Bayesian bounds

Restrictions:

- binary loss
- computational costs
- low sample sizes, low dimensions

Further work:

- more effective approximations
- bigger sample sizes, bigger dimensions
- more applications

Konstantin Vorontsov
vokov@forecsys.ru

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

- Участник:Vokov
- Расслоение и сходство алгоритмов (виртуальный семинар)
- Теория надёжности обучения по прецедентам (курс лекций, К. В. Воронцов)