Combinatorial Theory of Overfitting

How Connectivity and Splitting Reduces the Local Complexity

Konstantin Vorontsov[1,2,3] voron@forecsys.ru
Alexander Frey[1] sashafrey@gmail.com
Evgeny Sokolov[2] sokolov.evg@gmail.com

[1]Moscow Institute of Physics and Technology
[2]Moscow State University
[3]Computing Center RAS

9th IFIP International Conference
Artificial Intelligence Applications and Innovations (AIAI 2013)
Measures of Complexity Symposium
Paphos, Cyprus • September 30 – October 2, 2013

Contents

The central problem of Statistical Learning

$X = \{x_1, \dots, x_\ell\}$ — a finite training set of objects,
A — a set of classifiers,

$a = \arg\min_{a \in A} Err(a, X)$ — the empirical risk minimization,
or, more commonly,

$a = \mu(X)$ — a *learning algorithm* μ trains a classifier a on a set X.

The Generalization Problem:

1. How to bound a testing error $Err(a, \bar{X})$, where $\bar{X} = \{x'_1, \dots, x'_k\}$ is an independent testing set?
2. How to design learning algorithms that generalize well, i.e. have a small testing error $Err(a, \bar{X})$ almost always?

The classical approach to Generalization Bounds

In classical approach one find the uniform convergence conditions:

$$\mathsf{P}_X\Bigl(\sup_{a\in A}\bigl|P(a)-Err(a,X)\bigr|\geq\varepsilon\Bigr)\leq \mathrm{GenBound}(\ell,k,A,\varepsilon)$$

where $P(a)=\mathsf{E}_X Err(a,X)$ [Vapnik, Chervonenkis, 1971].

The Problem:

- GenBound may be very loose: $\sim 10^5..10^{11}$ in realistic cases

To tackle the problem we

1. modify the functional at the left-side of the inequality
2. propose a combinatorial approach to get the right-side bound

Modifying the functional (step 1 from 4)

In classical approach one find the uniform convergence conditions:

$$\mathsf{P}_X\Big(\sup_{a\in A} \big|P(a) - Err(a,X)\big| \geq \varepsilon \Big) \leq \mathrm{GenBound}(\ell, k, A, \varepsilon)$$

In combinatorial approach instead of a probability of error $P(a)$ we bound a testing error:

$$\mathsf{P}_{X,\bar{X}}\Big(\sup_{a\in A} \big|Err(a,\bar{X}) - Err(a,X)\big| \geq \varepsilon \Big) \leq \mathrm{GenBound}(\ell, k, A, \varepsilon)$$

Motivation:

- we bound an empirically measurable quantity of *overfitting*:

$$\delta(a, X, \bar{X}) = Err(a, \bar{X}) - Err(a, X)$$

- we remove a redundant technical step of *symmetrization* that weakens the bound without adding a sense to the result

Modifying the functional (step 2 from 4)

In classical approach one find the uniform convergence conditions:

$$\mathsf{P}_X\Bigl(\sup_{a\in A}\bigl|P(a)-Err(a,X)\bigr|\geq\varepsilon\Bigr)\leq \mathrm{GenBound}(\ell,k,A,\varepsilon)$$

In combinatorial approach instead of supremum over A
we use a learning algorithm μ:

$$\mathsf{P}_{X,\bar{X}}\Bigl(\bigl|Err(\mu(X),\bar{X})-Err(\mu(X),X)\bigr|\geq\varepsilon\Bigr)\leq \mathrm{GenBound}(\ell,k,\mu,\varepsilon)$$

Motivation:

- we remove the most restrictive condition from the functional
- we discard classifiers irrelevant to a given learning task
- we take into account the learning algorithm μ

Modifying the functional (step 3 from 4)

In classical approach one find the uniform convergence conditions:

$$\mathsf{P}_X\Bigl(\sup_{a\in A}\bigl|P(a)-Err(a,X)\bigr|\geq\varepsilon\Bigr)\leq \text{GenBound}(\ell,k,A,\varepsilon)$$

In combinatorial approach instead of usual i.i.d. assumption we use a uniform distribution over all partitions $\mathbb{X}^L = X\sqcup\bar{X}$:

$$\frac{1}{C_L^\ell}\sum_{\substack{X\subset\mathbb{X}^L\\|X|=\ell}}\Bigl[\bigl|Err(\mu(X),\bar{X})-Err(\mu(X),X)\bigr|\geq\varepsilon\Bigr]\leq \text{GenBound}(\mathbb{X}^L,\mu,\varepsilon)$$

Motivation:

- we make both sides of the inequality data-dependent and empirically measurable
- we remove a redundant step of integration over object space

Modifying the functional (step 4 from 4)

In classical approach one find the uniform convergence conditions:

$$\mathsf{P}_X\Bigl(\sup_{a\in A}\bigl|P(a)-Err(a,X)\bigr|\geq\varepsilon\Bigr)\leq \mathrm{GenBound}(\ell,k,A,\varepsilon)$$

In combinatorial approach instead of two-side deviation
we remove $|\cdot|$ and estimate one-side deviation:

$$\mathsf{P}_{X\sim\mathbb{X}^L}\Bigl[Err(\mu(X),\bar{X})-Err(\mu(X),X)\geq\varepsilon\Bigr]\leq \mathrm{GenBound}(\mathbb{X}^L,\mu,\varepsilon)$$

Motivation:

- we discard a non-interesting case of negative overfitting

Finished: we defined *the probability of large overfitting*

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

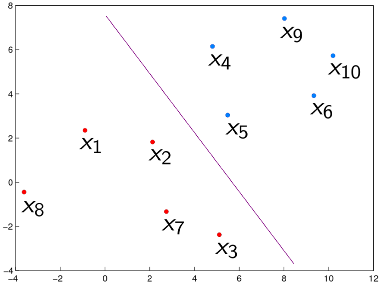
Error matrix of size $L \times D$, all columns are distinct:

	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 $k = L - \ell$

$a \mapsto \bigl(I(a, x_1), \dots, I(a, x_L)\bigr)$ — binary *error vector* of classifier a

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

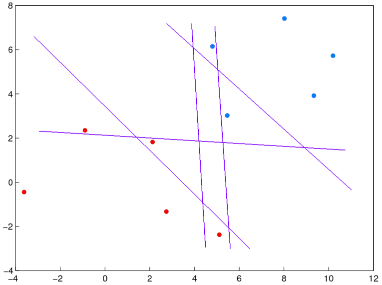
Example. The error matrix for a set of linear classifiers



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 error matrix for a set of linear classifiers

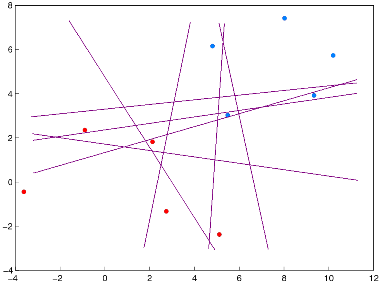


1 vector having no errors

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 error 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

$\mu\colon X \mapsto a$ — *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

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

Bounding problems

- Probability of large overfitting:

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

- Probability of large testing error:

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

- Expectation of OverFitting:

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

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

$$\mathrm{CCV}(\mu, \mathbb{X}^L) = \mathsf{E}\,\nu\bigl(\mu X, \bar{X}\bigr) \leq ?$$

Links to Cross-Validation

Expected testing error also called Complete Cross-Validation (taking expectation is equivalent to averaging over all partitions):

$$\mathrm{CCV}(\mu, \mathbb{X}^L) = \mathsf{E}\,\nu(\mu X, \bar{X}) = \frac{1}{C_L^\ell} \sum_{X \subset \mathbb{X}^L} \nu(\mu X, \bar{X})$$

Usual cross-validation techniques (e.g. hold-out, t-fold, $q \times t$-fold, partition sampling, etc.) can be viewed as empirical measurements of CCV by averaging over a representative 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 $\mathbb{X}^L$

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

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

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

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

if we set μ to *overfitting maximization* (very unnatural learning!):

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

Links to usual SLT framework

Usual probabilistic assumptions:
$\mathbb{X}^L$ is i.i.d. from probability space $\langle \mathscr{X}, \sigma, \mathsf{P} \rangle$ on infinite $\mathscr{X}$

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:

$$\mathsf{P}_{X \sim \mathbb{X}^L} \bigl[\delta(\mu, X) \geq \varepsilon \bigr] = Q_\varepsilon(\mu, \mathbb{X}^L) \leq \eta(\varepsilon, \mathbb{X}^L)$$

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

$$\mathsf{P}_{\substack{X \sim \mathscr{X}^\ell \\ \bar{X} \sim \mathscr{X}^k}} \bigl[\delta(\mu, X) \geq \varepsilon \bigr] = \mathsf{E}_{\mathbb{X}^L}\, Q_\varepsilon(\mu, \mathbb{X}^L) \leq \mathsf{E}_{\mathbb{X}^L}\, \eta(\varepsilon, \mathbb{X}^L).$$

(No) Links to Transductive Learning

In both cases data are partitioned on two subsets, but
(training $\sqcup$ testing) $\neq$ (labeled $\sqcup$ unlabeled)

In transductive learning:

- the aim is to get a semi-supervised data clustering,
- labels for the second subset are unknown,
- learning algorithm uses both labeled and unlabeled data.

In our combinatorial approach:

- the aim is to get generalization bounds,
- labels for both training and testing subsets are known,
- learning algorithm can not use the testing set.

Vapnik-Chervonenkis bound

Theorem

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

$$Q_\varepsilon(\mu, \mathbb{X}^L) \overset{\substack{\textit{uniform}\\\textit{bound}}}{\leq} \mathsf{P}\Big[\sup_{a \in A} \delta(a, X) \geq \varepsilon\Big] \overset{\substack{\textit{union}\\\textit{bound}}}{\leq} \sum_{a \in A} Q_\varepsilon(a, \mathbb{X}^L)$$

$$\overset{\substack{\textit{approxi-}\\\textit{mation}}}{\leq} |A| \cdot \tfrac{3}{2} \exp\big(-\varepsilon^2 \ell\big), \quad \textit{for } \ell = k.$$

$|A|$ — Shattering Coefficient,

$|A| \leq C_L^0 + C_L^1 + \cdots + C_L^h,\ \ h = \mathrm{VCdim}(A)$

Usually this bound is overestimated by 10^5–10^{11} times. Why?

1) uniform bound is loose if A *is split* by $\nu(a, \mathbb{X}^L)$

2) union bound is loose if most classifiers are *similar* or *connected*

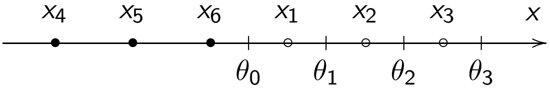
3) approximation bound is not so loose

Monotone chain of classifiers

One-dimensional threshold classifier (decision stump):

$$a_d(x) = [x \geq \theta_d], \qquad d = 0, \dots, D$$

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

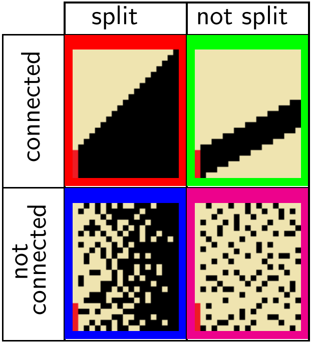


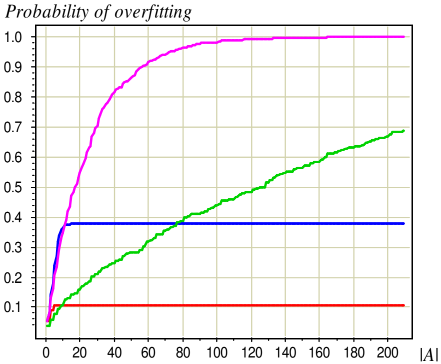
Loss matrix:

	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

Experiment with monotone chain of classifiers

$\ell = k = 100$, $\varepsilon = 0.05$, $N = 1000$ Monte-Carlo partitions.

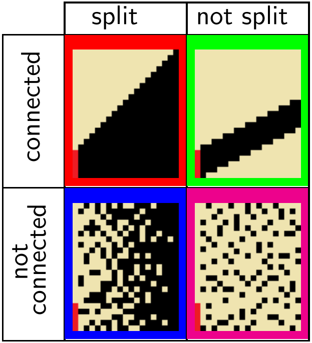


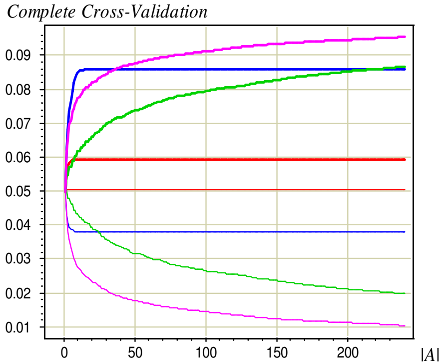


- With both splitting and connectivity a huge set does not overfit
- With no splitting and connectivity 30 classifiers may overfit

Experiment with monotone chain of classifiers

$\ell = k = 100$, $\varepsilon = 0.05$, $N = 1000$ Monte-Carlo partitions.





- The local complexity measure should depend on both splitting and connectivity properties of the set

Splitting-Connectivity graph (1-inclusion graph)

Define two binary relations on classifiers:

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

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

Definition (SC-graph)

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

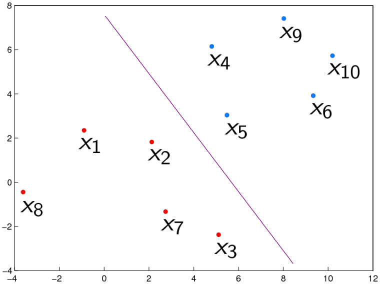
A — *a set of classifiers with distinct binary error vectors;*

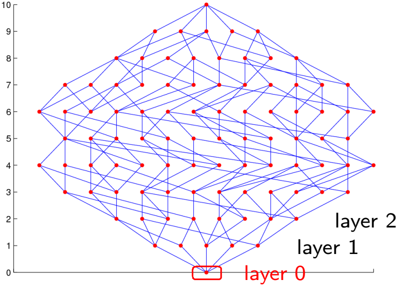
$E = \bigl\{(a,b)\colon a \prec b\bigr\}$.

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 $A_m = \bigl\{a \in A\colon \nu(a, \mathbb{X}^L) = \frac{m}{L}\bigr\}$, $m = 0, \dots, L+1$;

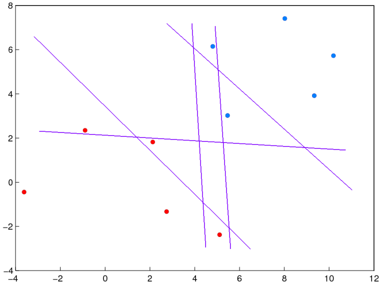
Example. Error matrix and SC-graph for a set of linear classifiers

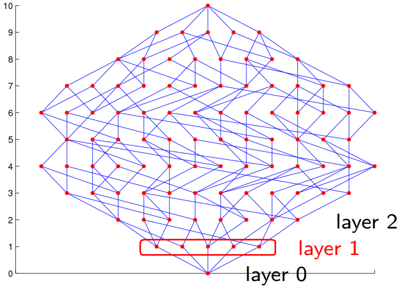




	layer 0
x_1	0
x_2	0
x_3	0
x_4	0
x_5	0
x_6	0
x_7	0
x_8	0
x_9	0
x_{10}	0

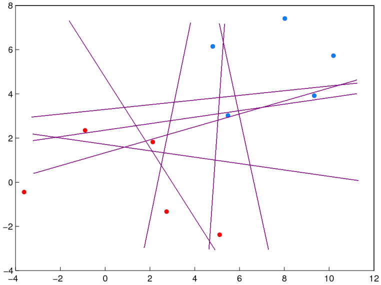
Example. Error matrix and SC-graph for a set of linear classifiers

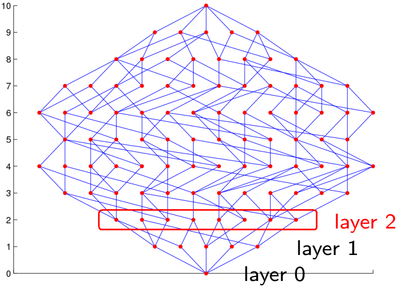




	layer 0	layer 1				
x_1	0	1	0	0	0	0
x_2	0	0	1	0	0	0
x_3	0	0	0	1	0	0
x_4	0	0	0	0	1	0
x_5	0	0	0	0	0	1
x_6	0	0	0	0	0	0
x_7	0	0	0	0	0	0
x_8	0	0	0	0	0	0
x_9	0	0	0	0	0	0
x_{10}	0	0	0	0	0	0

Example. Error 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 splitting coefficients of a classifier

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

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

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

Def. *Splitting coefficient* (*inferiority*) of a classifier $a \in A$

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

Splitting coefficient:

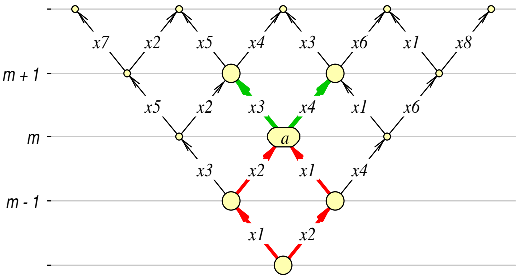
$d(a) \leq q(a) \leq L\nu(a, \mathbb{X}^L)$

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:

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

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

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

The properties of the SC-bound

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

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

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

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

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

The properties of the SC-bound

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

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

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

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.

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 CCV bound:

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

The Local Complexity Regularization

Main steps to use combinatorial Splitting-Connectivity bound:

1. Calculate SC-bound anyway (e.g. via random walks):

$$\mathsf{P}\big[(\mu X, \bar{X}) - \nu(\mu X, X) \geq \varepsilon\big] \;\leq\; \text{SCbound}(\varepsilon; A, \mathbb{X}^L) \equiv \eta$$

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

$$\nu(\mu X, \bar{X}) \leq \nu(\mu X, X) + \varepsilon(\eta; A, \mathbb{X}^L)$$

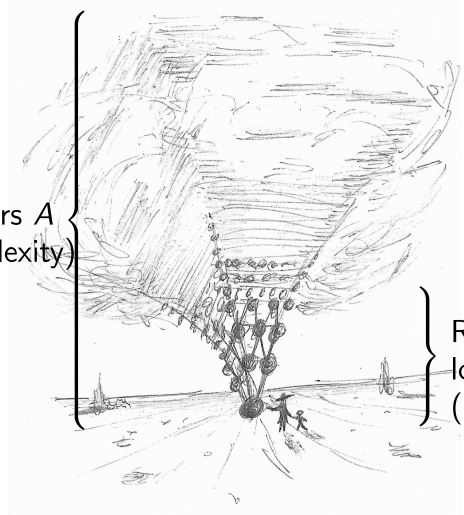
3. Use $\varepsilon(\eta; A, \mathbb{X}^L)$ as a penalty for features or model selection

---

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

Splitting gives an idea of effective SC-bound computation

All classifiers A
(global complexity)



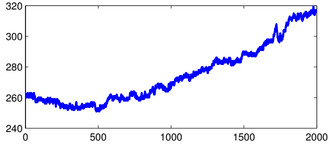
Really used classifiers,
lowest layers of A
(local complexity)

SC-bound computation via Random Walks

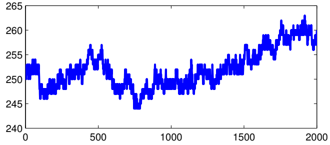
1. Learn a good classifier
2. Run a large number of short walks to get a subset $B \subset A$
3. Compute a partial sum $Q_\varepsilon \approx \sum\limits_{a \in B} \text{summand}(a)$

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:



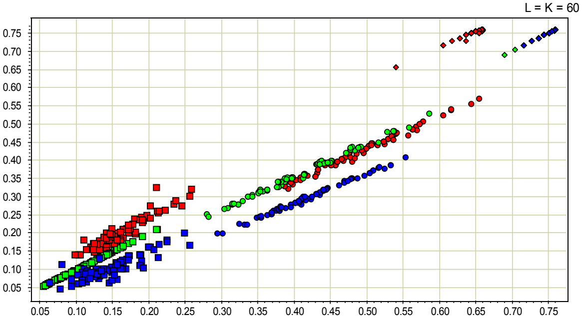
Making bounds observable

SCbound$(\mu, \mathbb{X}^L)$ depends on a hidden set $\bar{X}$, then we use SCbound(μ, X) instead.

Open problems: is it correct? why? may be not always?

Really EOF(μ, X) is well concentrated near to EOF$(\mu, \mathbb{X}^L)$:
Experiments on model data, $L = 60$, testing sample size $K = 60$

EOF$(\mu, \mathbb{X}^L)$



EOF(μ, X)

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 $b_t(x)$:

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

Main idea is to apply generalization bound
as features selection criterion in base classifiers

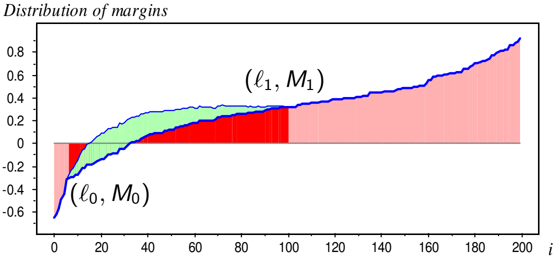
Our goals:
1) to reduce overfitting of base classifiers
2) to reduce the complexity of composition T

ComBoost: Committee boosting

Instead of objects reweighting ComBoost trains each base classifier on the training subset $X' \subset X$ in order to augment margins of the ensemble as much as possible:

$$X' = \{x_i \in X \colon M_0 \le \mathrm{Margin}(i) \le M_1\}$$

$$\mathrm{Margin}(i) = y_i \sum_{t=1}^{T} w_t b_t(x_i).$$



Learning ensembles of Conjunction Rules

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

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

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

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.

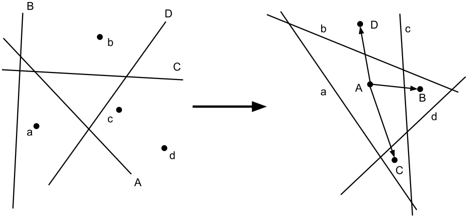
Vorontsov K. V., Ivahnenko A. A. Tight Combinatorial Generalization Bounds for Threshold Conjunction Rules // LNCS. PReMI'11, 2011. Pp. 66–73.

Liner classifiers and ensembles

Linear classifier: $a(x) = \operatorname{sign}\langle w, x\rangle$
Ensemble of low-dimensional linear classifiers

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

Random Walks for SC-bound computation
1) find all neighbor classifiers in the dual space:



2) lookup along random rays

Experiment 1: ComBoost ensemble of linear 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)
- SC — Splitting-Connectivity bound
- VC — Vapnik–Chervonenkis bound
- DD — Dimension-Dependent PAC-Bayes bound (Jin, 2012)

UCI Task	MC	SC	VC	PAC DD
glass	0.115	0.146	0.356	0.913
liver	0.095	0.533	0.595	1.159
ionosphere	0.083	0.149	0.238	1.259
wdbc	0.052	0.070	0.136	0.949
australian	0.043	0.244	0.277	0.798
pima	0.045	0.373	0.410	0.823

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
- and gives more accurate base classifiers in Ensemble Learning

Restrictions:

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

Further work:

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

Questions?

Konstantin Vorontsov
vokov@forecsys.ru

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

- User:Vokov