# Синергия алгоритмов классификации (SVM Multimodelling)

***С. Иванычев, А. Адуенко***
sergeyivanychev@gmail.com, aduenko1@gmail.com
Московский физико-технический институт

В данной статье рассматривается проблема агрегирования небольшого количества сильных классификаторов с целью улучшения решений задач классификации и регрессии. В качестве примера подобной системы рассматривается система SVM алгоритмов использующая kernel-trick с различными ядрами. Для комбинации решений и улучшения качества прогнозирования в задачах классификации и регрессии (SVR) авторы предлагают способ формирования новых признаков на основе сгенерированных отступов (*margins*) каждым классификатором, приводят алгоритм обучения на полученных объектах и анализируют отличия множеств опорных объектов для различных ядер. В качестве практической проверки были проведены эксперименты на различных реальных данных из репозитория UCI.

**Ключевые слова**: *двухклассовая классификация, композиция алгоритмов, SVM, SVR, бэггинг*

## Введение

Работа посвящена комбинированию небольшого количества сильных SVM, использующих kernel-trick с различными ядрами и получению агрегированного классификатора для улучшения решений задач классификации и регрессии.

SVM(Support Vector Machine) или *метод опорных векторов*[1, 2, 3] — это один из наиболее распространенных и эффективных методов в машинном обучении, которые используется для задач классификации и регрессии (SVR). Задача математического программирования сводится к двойственной задаче, функционалы в которой не зависят от векторов признаков как таковых, а лишь от их попарных скалярных произведений [4] . Использование особых функций, *ядер*, то есть скалярных произведений в сопряженном пространстве, позволяет получить разделяющие поверхности между классами более сложной формы [5]. Наша цель — скомбинировать SVM с различными примененными ядрами для улучшения решения, а также анализ множеств опорных объектов в случае разных использованных ядер.

Наиболее классическими методами агрегирования алгоритмов являются бэггинг (*bagging*)[6] и бустинг (*boosting*) [7], и их вариации, однако они работают только с большим количеством слабых классификаторов, что делает невозможным использование его использование для указанного множества базовых алгоритмов.

Среди способов агрегации для небольшого количества классификаторов можно выделить, например, выбор большинства классификаторов [8], комбинирование ранжирований (rankings) по классам, сделанных различными классификаторами [9]. В дальнейшем было показано, что все подобные методы есть особые случаи составного классификатора из [10], появляющиеся при особых условиях или способах аппроксимации.

Различные способы агрегации SVM используются во многих задачах анализа данных. [11] использовали совокупность SVM для уменьшения ошибочно негативных классификаций (FP) в задаче фильтрации спама среди электронных писем. Для этого на электронных письмах были введены различные метрики, для каждой из них был приспособлен SVM, а затем результат получался голосованием [10]. [12], решавшие задачу распознавания написанных рукой символов, делили множество признаков на четыре непересекающихся подмножества, и на каждом из них обучали SVM, увеличив этим самым коэффициент распознавания по сравнению с одним SVM.

В последнее время стал набирать популярность *метод многоядерного обучения* (MKL, multiple kernel learning) [13, 14, 15], который основывается на том, что линейная комбинация ядер также является ядром. Данный метод хорош при объединении данных из нескольких источников и полной автоматизации, так как суперпозиция функций может быть оптимизирована любым методом валидации (например кросс-валидацией).

Мы также предлагаем использовать накопившийся банк ядер, однако не на этапе обучения SVM, а на этапе агрегирования обученных алгоритмов. Известно, что алгоритм $b_i$ для объекта $x_j$ обучающей выборки генерирует *отступ* (margin). По отступу в общем случае можно определить не только предсказанный класс, но и насколько «уверен» в своем решении алгоритм. В случае банка с $n$ ядрами и обучающей выборки с $m$ сэмплами мы получим матрицу отступов $M \in \mathbb{R}^{m\times n}$. Отнормировав ее, мы получим новую матрицу «объект-признак», где вектором признаков каждого объекта будет вектор отнормированных отступов.

В этой работе предложен алгоритм обучения на матрице отступов, проведен анализ опорных объектов, генерируемые различными ядрами, а также проведено тестирование полученного алгоритма на реальных данных репозитория UCI.

## Постановка задачи

Пусть $X^l = (\boldsymbol{x}_i, y_i)_{i=1}^l$ — обучающая выборка, $\boldsymbol{x} \in \mathbb{R}^n, y \in Y$ где $|Y| < \infty$ — задача классификации, $|Y| = \infty$ — задача регрессии. В данной статье под $s$-й *моделью* будем понимать SVM с ядром $K_s$ где выбрано множество ядер:

$$\mathcal{K} = \{K_i\}_{j=1}^m$$

При обучении каждая модель дает классификатор или регрессор (в зависимости от типа $Y$). Например, для случая $Y \in \{-1, +1\}$ классификации алгоритм выглядит следующим образом:

$$b_s(\mathbf{x}) = \mathbf{sign} \sum_{i=1}^l \lambda_i y_i K_s(\boldsymbol{x}_i, \boldsymbol{x}) - w_0$$

Где $\{\lambda_i\}$ и $w_0$ находятся из решения задачи математического программирования[5]

$$\begin{cases} \sum_{i=1}^l \lambda_i + \frac{1}{2}\sum_{i=1}^l \sum_{j=1}^l \lambda_i\lambda_j y_i y_j K_s(\boldsymbol{x}_i, \boldsymbol{x}_j) \to \min\limits_{\lambda} \\ 0 \leqslant \lambda_s \leqslant c, \quad i = 1 \dots l \\ \sum_{i=1}^l \lambda_i y_i = 0 \end{cases}$$

Побочным результатом обучения является то, что модель для обучающей выборки генерирует вектор *отступов* (margins) для каждого объекта.

Если мы рассмотрим множество моделей, обученных на $X^l$, то мы получим матрицу отступов размерности $M \in \mathbb{R}^{l\times m}$, в котором $(i, j)$-й элемент — это отступ $i$-го объекта в SVM с $j$-м ядром.

Пусть $M$ — матрица «объект-признак», $\mathcal{A}$ — множество алгоритмов вида

$$\mathcal{A} = \{a(\boldsymbol{x}) = g(\boldsymbol{x}, \theta) | \theta \in \Theta\} \quad g : \mathbb{R}^m \to Y \tag{1}$$

Пару $(g, \mathcal{K})$ назовем *мультимоделью.* $\mathcal{L}(y, y^*)$ — функционал качества, тогда перед нами стоит задача минимизации

$$L(y, g(M, \theta)) \to \min_{\mathcal{A}, \Theta} \tag{2}$$

## Старая постановка задачи

Пусть $\mathcal{K} = \{K_i\}_{j=1}^m$ — множество ядер, выбранных для мультимоделирования SVM, $X^l = (\boldsymbol{x}_i, y_i)_{i=1}^l$ — обучающая выборка, $\boldsymbol{x} \in \mathbb{R}^n, y \in Y$. Тогда множество обученных SVM на данной выборке:

$$\mathcal{B} = \left\{ b_i | b_i = b_i(\boldsymbol{x}) = \mathbf{sign} \sum_{i=1}^{l} \lambda_i y_i K_i(\boldsymbol{x}_i, \boldsymbol{x}) - w_0 \right\}_{i=1}^{m} \tag{3}$$

Где $\lambda_s$ находятся из решения задачи математического программирования

$$\begin{cases} \sum_{i=1}^l \lambda_i + \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \lambda_i \lambda_j y_i y_j K_s(\boldsymbol{x}_i, \boldsymbol{x}_j) \to \min\limits_{\lambda} \\ 0 \leqslant \lambda_s \leqslant c, \quad i = 1 \ldots l \\ \sum_{i=1}^l \lambda_i y_i = 0 \end{cases}$$

И $w_0 = \sum_{i=1}^l \lambda_i y_i K_s(\boldsymbol{x}_i, \boldsymbol{x}_j) - y_j$ для такого $j$, что $\lambda_j > 0, M_j = 1$. Паре «выборка-обученные алгоритмы» соответствует матрица отступов размерности $M \in \mathbb{R}^{l \times m}$, в котором $(i, j)$-й элемент — это отступ $i$-го объекта в SVM с $j$-м ядром. Пусть $M$ — матрица «объект-признак», $\mathcal{A}$ — множество алгоритмов вида

$$\mathcal{A} = \{a(\boldsymbol{x}) = g(\boldsymbol{x}, \theta) | \theta \in \Theta\} \quad g : \mathbb{R}^m \to Y \tag{4}$$

$\mathcal{L}(y, y^*)$ — функционал качества, тогда перед нами стоит задача минимизации

$$L(y, g(M, \theta)) \to \min_{\mathcal{A}, \Theta} \tag{5}$$

## Литература

[1] Vladimir N Vapnik. Statistical Learning Theory. *Adaptive and learning Systems for Signal Processing, Communications and Control*, 2:1–740, 1998.

[2] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

[3] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.

[4] Konstantin Vyacheslavovich Vorontsov. Mathematical methods of learning from examples (machine lerning theory).

[5] Alex J Smola, Bernhard Sch, and B Schölkopf. A Tutorial on Support Vector Regression. *Statistics and Computing*, 14(3):199–222, 2004.

[6] Leo Breiman. Bagging Predictors. *Machine Learning*, 24(421):123–140, 1996.

[7] Y. Freund. Boosting a Weak Learning Algorithm by Majority, 1995.

[8] J. Franke and E. Mandler. A comparison of two approaches for combining the votes of\ncooperating classifiers. *Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems*, pages 1–4, 1992.

[9] Tin Kam T.K. Ho, JJ Hull, Sargur N SN Srihari, and Senior Member. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75, 1994.

[10] J. Kittler, M. Hater, and R. P W Duin. Combining classifiers. *Proceedings - International Conference on Pattern Recognition*, 2(3):897–901, 1996.

[11] Manuel Martin-merino and Manuel Mart. Combibing SVM Classifiers for Email Anti-spam Filtering. 4507(February), 2007.

[12] D. Gorgevik and D. Cakmakov. Handwritten Digit Recognition by Combining SVM Classifiers. *EUROCON 2005 - The International Conference on Computer as a Tool*, 2(February):1393–1396, 2005.

[13] Martin Dyrba, Michel Grothe, Thomas Kirste, and Stefan J. Teipel. Multimodal analysis of functional and structural disconnection in Alzheimer's disease using multiple kernel SVM. *Human Brain Mapping*, 36(6):2118–2131, 2015.

[14] S.S. Bucak, R. Jin, and Ak. Jain. Multiple Kernel Learning for Visual Object Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1354–1369, 2014.

[15] Salah Althloothi, Mohammad H. Mahoor, Xiao Zhang, and Richard M. Voyles. Human activity recognition using multi-features and multiple kernel learning. *Pattern Recognition*, 47(5):1800–1812, 2014.

[16] S. Amari and S. Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12(6):783–789, 1999.

[17] Catherine A. Shipp and Ludmila I. Kuncheva. Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion*, 3(2):135–148, 2002.

[18] Ben Fei and Jinbai Liu. Binary tree of SVM: A new fast multiclass training and classification algorithm. *IEEE Transactions on Neural Networks*, 17(3):696–704, 2006.

[19] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: learning using privileged information. *Neural networks : the official journal of the International Neural Network Society*, 22(5-6):544–57, jan 2009.

[20] European Signal and Processing Conference. BOOTSTRAP-BASED SVM AGGREGATION FOR CLASS IMBALANCE PROBLEMS S . Sukhanov , A . Merentitis , C . Debes AGT International J . Hahn , A . M . Zoubir Signal Processing Group Technische Universit at Darmstadt , Germany. pages 165–169, 2015.