

19-я Всероссийская конференция с международным участием «Математические методы распознавания образов»



26–29 ноября 2019, г. Москва, РАН



Российская Академия Наук

Метод средних решающих правил для быстрого двухклассового обучения в пространстве, порожденном потенциальной функцией



Макарова Александра, Сулимова Валентина

aleksarova@gmail.com, vsulimova@yandex.ru,

Тульский государственный университет
Лаборатория анализа данных



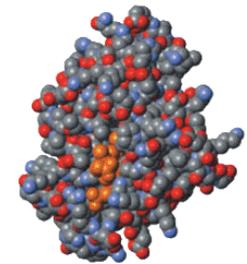
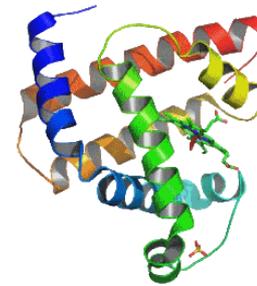
Задача двухклассового распознавания

Массовые источники прикладных задач:

- молекулярная биология;
- горнодобывающая и нефтяная промышленности;
- медицинские системы;
- системы видеонаблюдения;
- маркетинг;
- анализ текстов;
- биометрическая верификация личности;
- и многие другие.

Важная особенность современных задач:

- большие объемы данных, требующих обработки



Метод опорных векторов (SVM).

Обучение в линейном признаковом пространстве

Представление объектов в виде точек в m -мерном признаковом пространстве: $\mathbf{x}(\omega) \in R^m$

Обучающее множество: $\{\mathbf{x}_j, y_j\}$, $\mathbf{x}_j = \mathbf{x}_j(\omega)$, $j = 1, \dots, N$

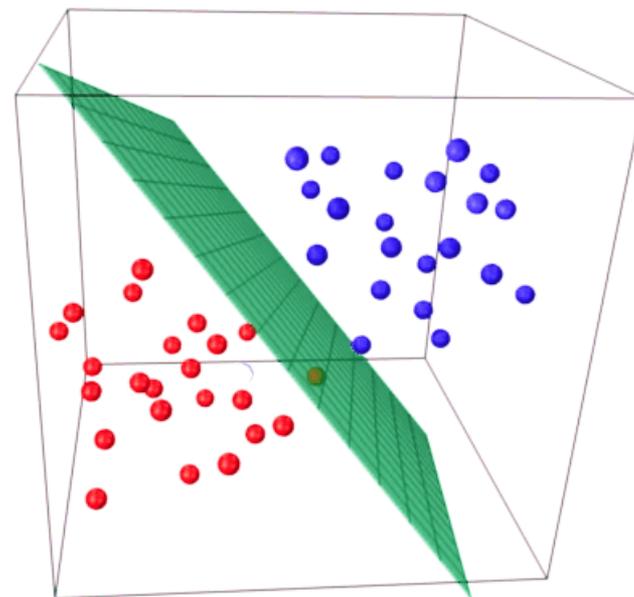
Решающее правило в виде линейной разделяющей гиперплоскости:

$$d(\mathbf{x}; \mathbf{a}, b) = \mathbf{a}^T \mathbf{x} + b \quad \begin{array}{l} \geq 0 \Rightarrow \hat{y}(\mathbf{x}) = +1, \\ < 0 \Rightarrow \hat{y}(\mathbf{x}) = -1, \end{array} \quad \begin{array}{l} \mathbf{a} \in R^m \text{ - направляющий вектор} \\ b \text{ - смещение вдоль направляющего вектора} \end{array}$$

Параметры оптимальной гиперплоскости
(в терминах множителей Лагранжа) $\lambda_j, j = 1, \dots, N$:

$$\mathbf{a} = \sum_{j=1}^N \lambda_j y_j \mathbf{x}_j$$

$$b = \frac{1}{2} \left[\min_{j: y_j=1} \sum_{k=1}^N \lambda_k y_k \mathbf{x}_j^T \mathbf{x}_k - \max_{j: y_j=-1} \sum_{k=1}^N \lambda_k y_k \mathbf{x}_j^T \mathbf{x}_k \right]$$



Метод опорных векторов (SVM).

Обучение в пространстве, порожденном потенциальной функцией

Потенциальная функция $K(\omega', \omega'')$ - функция сходства объектов $\omega', \omega'' \in \Omega$, матрица значений которой для любой конечной совокупности объектов неотрицательно определена

$K(\omega', \omega'')$ погружает множество объектов Ω в гипотетическое линейное пространство $\tilde{\Omega} \supset \Omega$ в котором играет роль скалярного произведения

Наиболее популярная потенциальная функция (радиальная):

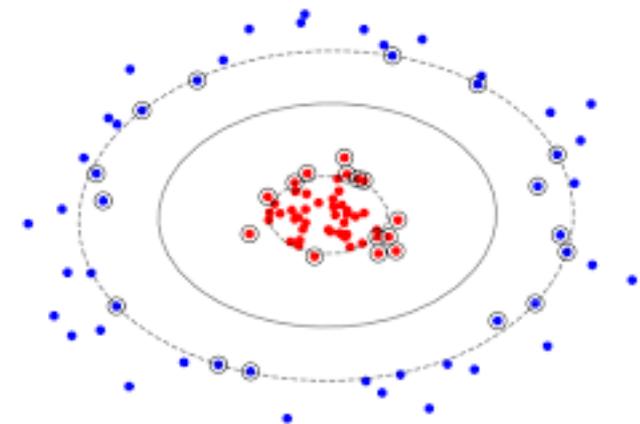
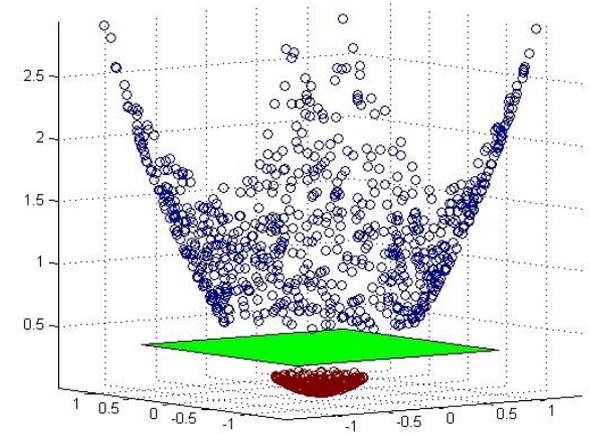
$$K(\omega', \omega'') = \exp[-\gamma \| \mathbf{x}(\omega') - \mathbf{x}(\omega'') \|^2]$$

где $\| \mathbf{x}(\omega') - \mathbf{x}(\omega'') \|$ - Евклидово расстояние

$$\text{Решающее правило : } d(\omega; \mathbf{a}, b) = \sum_{j=1}^N a_j K(\omega_j, \omega) + b$$

Параметры оптимальной гиперплоскости
(в терминах множителей Лагранжа) $\lambda_j, j = 1, \dots, N$:

$$b = \frac{1}{2} \left[\min_{j: y_j=1} \sum_{k=1}^N \lambda_k y_k K(\omega_j, \omega_k) - \max_{j: y_j=-1} \sum_{k=1}^N \lambda_k y_k K(\omega_j, \omega_k) \right]$$



Множество подходов к решению задачи SVM

1. Bottou L. (2004). Stochastic Learning. *Advanced Lectures on Machine Learning*, 146-168, Edited by Olivier Bousquet and Ulrike von Luxburg, Lecture Notes in Artificial Intelligence, LNAI 3176, Springer Verlag, Berlin, 2004.
2. Boser, B. E., Guyon, I. M., Vapnik, V. (1992). A Training Algorithm for Optimal Margin Classifiers, Fifth Annual Workshop on Computational Learning Theory, ACM, 1992
3. Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines (Technical Report MSR-TR-98-14). Microsoft Research.
4. Chang, C.-C., & Lin, C.-J. (2001). LIBSVM: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
5. Joachims T.(1999) "Making large-scale support vector machine learning practical" in Advances in kernel methods: support vector learning. Cambridge, MA, USA: MIT Press, 1999, pp. 169–184
6. Chu, C.-T., Kim, S. K., Lin, Y.-A., Yu, Y., Bradski, G., Ng, A. Y., & Olukotun, K. (2006). Map reduce for machine learning on multicore. NIPS
7. Zhao H. X., Magoules F. (2011). Parallel support vector machines on multi-core and multiprocessor systems. R. Fox. 11th International Conference on Artificial Intelligence and Applications (AIA 2011), Feb 2011, Innsbruck, Austria. IASTED, 2011, <10.2316/P.2011.717-056>
8. Graf, H. P., Cosatto, E., Bottou, L., Dourdanovic, I., & Vapnik, V. (2005). Parallel support vector machines: The cascade svm. In Advances in neural information processing systems 17, 521–528.
9. Hsieh C.-J., Dhillon S. Si, and I. S. (2013). A divide-and-conquer solver for kernel support vector machines. arXiv preprint arXiv:1311.0914, 2013.
10. Yang You, James Demmel, Kenneth Czechowski, Le Song and Richard Vuduc (2015). CA-SVM: Communication-Avoiding Parallel Support Vector Machines on Distributed Systems. EECS Department University of California, Berkeley Technical Report No. UCB/EECS-2015-9 February 27, 2015
11. A. Athanasopoulos, A. Dimou, V. Mezaris, I. Kompatsiaris, "GPU Acceleration for Support Vector Machines", Proc. 12th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2011), Delft, The Netherlands, April 2011
12. Carpenter, Austin. (2009). cuSVM: a CUDA implementation of support vector classification and regression.
13. и др.

Задача

В предыдущей работе*:

был предложен метод средних решающих правил (Mean Decision Rule, MDR), являющийся

- аппроксимацией метода решения задачи SVM в условиях большого числа объектов **в пространстве признаков,**

- **быстрым,**

- **экономичным по памяти** (обеспечивая возможность работы на одной вычислительной машине),

и обладающим **высокой степенью параллелизма** (для организации высокопроизводительных вычислений для систем с общей и распределенной памятью) .

Цель текущей работы:

разработка версии MDR для обучения в пространстве, порожденном потенциальной функцией.

*Макарова А. И., Сулимова В. В. Быстрое приближенное решение двухклассовой задачи SVM для больших обучающих совокупностей //СБОРНИК ТРУДОВ ИТНТ-2019. – 2019. – С. 25-34.

Метод средних решающих правил для обучения в признаковом пространстве

Исходная обучающая совокупность: $[X, Y]$, $X = [\mathbf{x}_j, j = 1, \dots, N]$, $\mathbf{x}_j \in R^m$
 $Y = [y_j, j = 1, \dots, N]$, $y_j \in \{-1; 1\}$

Набор случайных подвыборок из обучающей совокупности: $[X, Y]^{(i)} \in [X, Y]$, $i = 1, \dots, k$

Результат обучения по i -й подвыборке: $[X, Y]^{(i)}$
частное решающее правило с параметрами $[\mathbf{a}^{(i)}, b^{(i)}]$, $i = 1, \dots, k$

Усредненное решающее правило: $[\mathbf{a}, b]$, $\mathbf{a} = \frac{1}{k} \sum_{i=1}^k \mathbf{a}^{(i)}$, $b = \frac{1}{k} \sum_{i=1}^k b^{(i)}$

Обоснование:

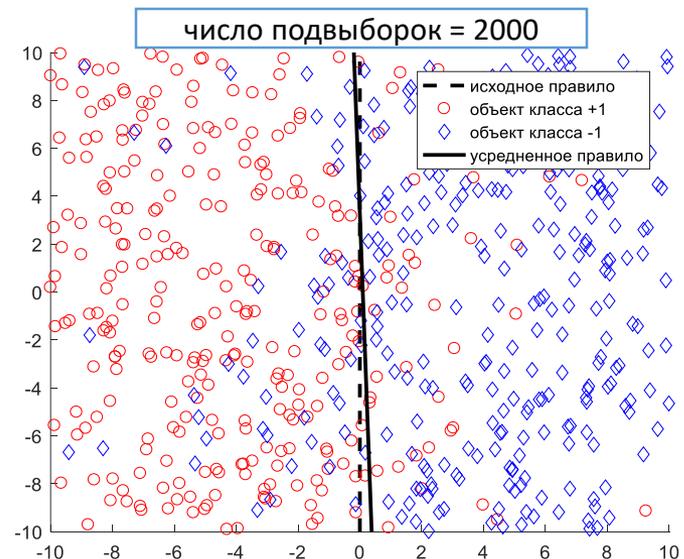
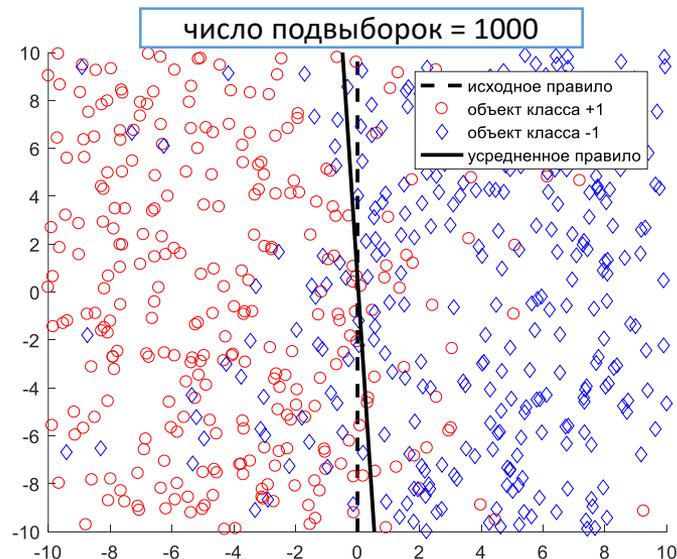
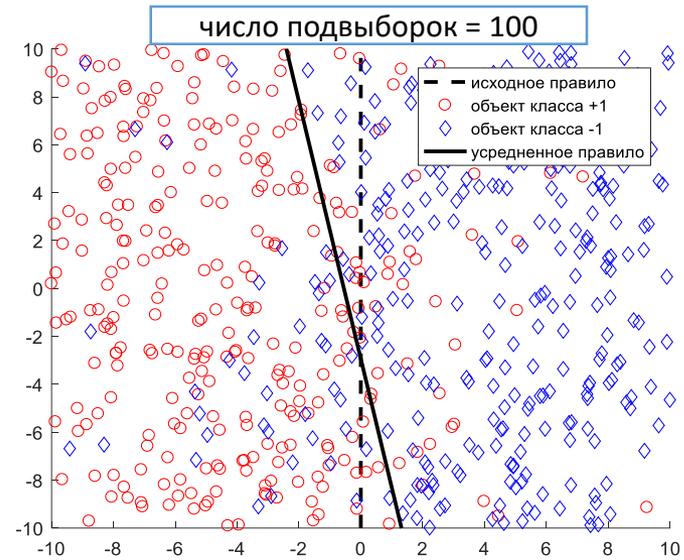
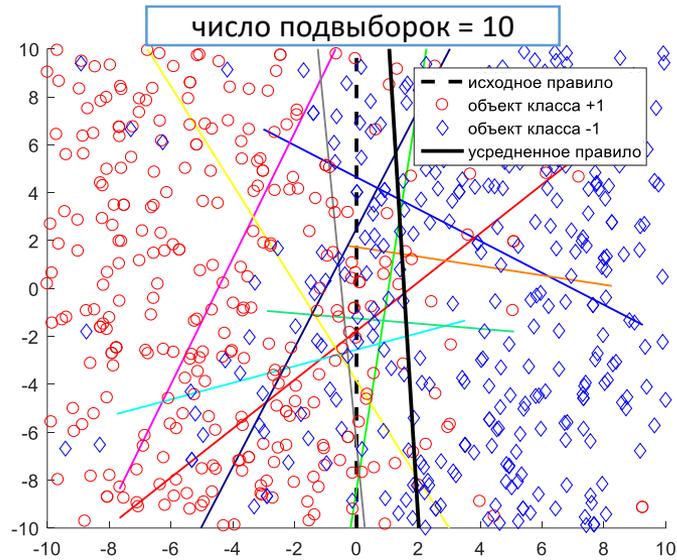
$[\mathbf{a}^{(i)}, b^{(i)}]$, $i = 1, \dots, k$ - случайные величины с характеристиками m и дисперсией d
Тогда $[\mathbf{a}, b]$ - случайная величина с характеристиками $M[\mathbf{a}, b] = m$ и $D[\mathbf{a}, b] = \frac{d}{k}$

Согласно закону больших чисел, усредненная оценка параметров решающего правила сходится по вероятности к математическому ожиданию соответствующей случайной величины:

$$\lim_{k \rightarrow \infty} P \left\{ \left| \frac{1}{k} \sum_{i=1}^k [\mathbf{a}^{(i)}, b^{(i)}] - m \right| < \varepsilon \right\} = 1$$

Т.о. при увеличении числа подвыборок **усредненное решающее правило стабилизируется и в пределе перестает быть случайной величиной**

Изменение положения усредненной гиперплоскости при изменении числа подвыборок



отображение сгенерированной для двумерного случая выборки (300 объектов каждого класса, степень перемешанности классов $s=0.8$) и решающих правил при разбиении на случайные пересекающиеся подвыборки

Метод средних решающих правил для потенциальных функций

Kernel-based Mean Decision Rule (KMDR) method

Исходная обучающая совокупность: $[\Omega, Y]$, $\Omega = [\omega_j, j = 1, \dots, N]$
 $Y = [y_j, j = 1, \dots, N]$, $y_j \in \{-1; 1\}$

Множество случайных подвыборок из обучающей совокупности: $[\Omega^{(i)} \subset \Omega, Y^{(i)} \subset Y] = [\Omega, Y]^{(i)}$, $i = 1, \dots, k$

Результат обучения на i ой подвыборке $[\Omega, Y]^{(i)}$:

Множители Лагранжа $\lambda_t^{(i)}, t = 1 \dots |\Omega^{(i)}|$ и смещение $b^{(i)}$

Множители по всей обучающей совокупности:

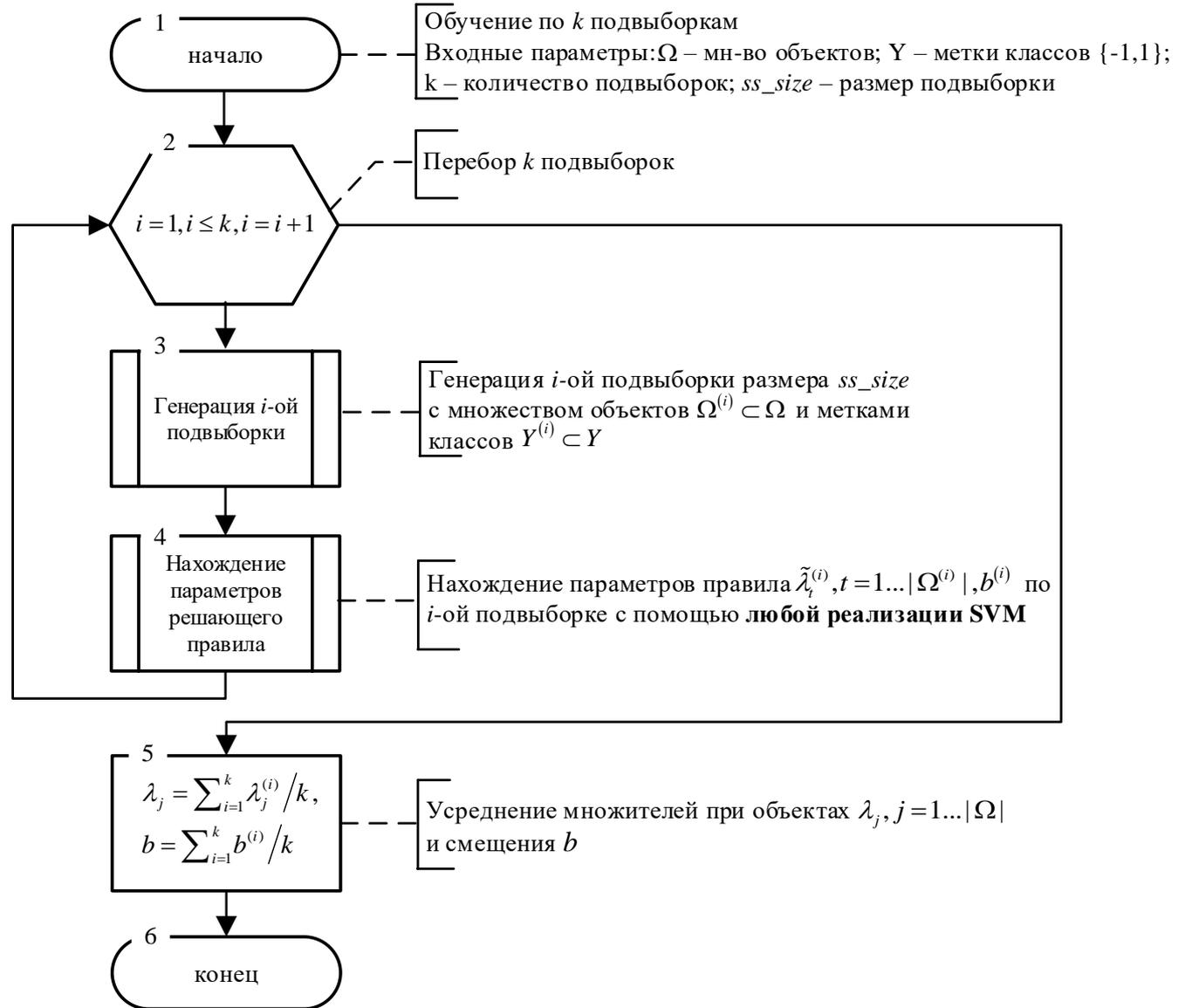
$$\lambda_j^{(i)} = \begin{cases} \lambda_{t(j)}^{(i)}, t(j) \in \{1 \dots |\Omega^{(i)}|\} & \text{если } \omega_j \in \Omega^{(i)}, \\ 0, & \text{иначе} \end{cases}$$

Усредненное решающее правило:

$$d(\omega; \lambda, b) = \frac{1}{k} \sum_{i=1}^k d(\omega; \lambda^{(i)}, b^{(i)}) = \sum_{j=1}^N \lambda_j y_j K(\omega_j, \omega) + b$$

$$\lambda_j = \frac{1}{k} \sum_{i=1}^k \lambda_j^{(i)}, \quad b = \frac{1}{k} \sum_{i=1}^k b^{(i)}$$

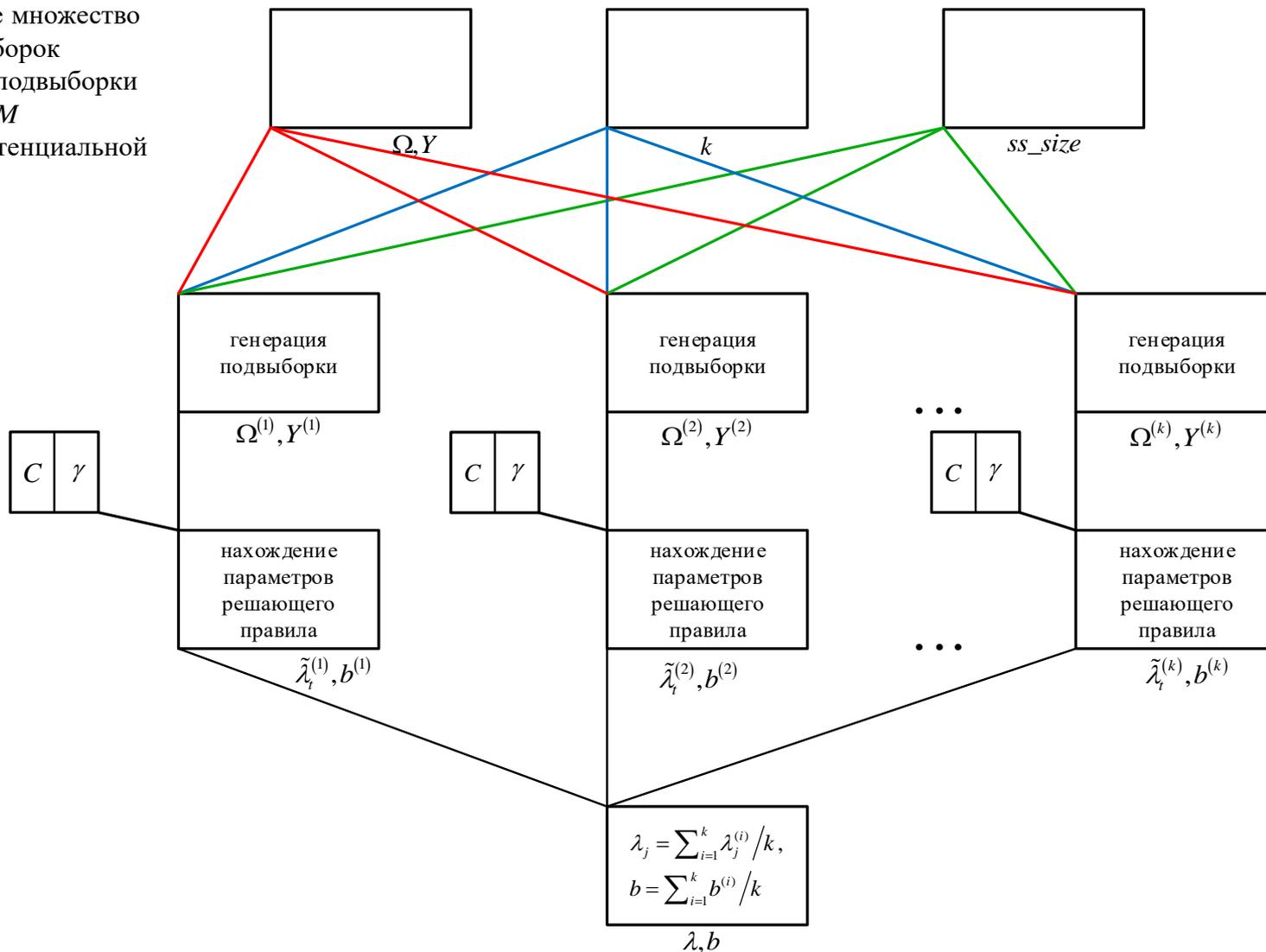
Последовательный алгоритм KMDR для быстрого приближенного решения задачи SVM



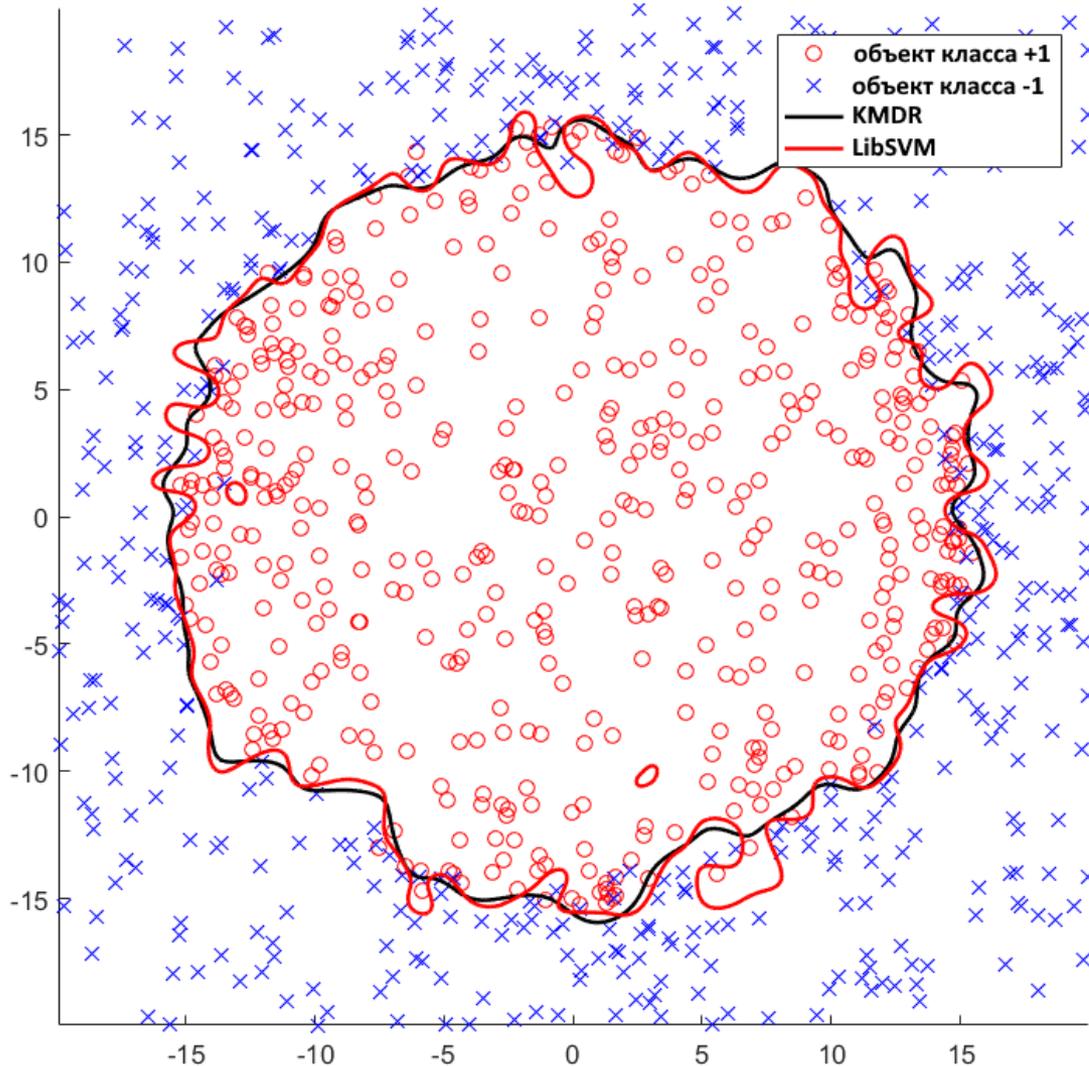
Модель параллельных вычислений KMDR

Граф операции-операнды для бесконечного числа процессов
(вершины – операции, ребра – зависимости по данным)

Ω, Y – обучающее множество
 k – число подвыборок
 ss_size – размер подвыборки
 C – параметр SVM
 γ – параметр потенциальной функции



LibSVM vs KMDR на модельных данных



Параметры набора

Число объектов: 1000

Число признаков: 2

Параметры SVM

$C = 10$

Kernel type: RBF

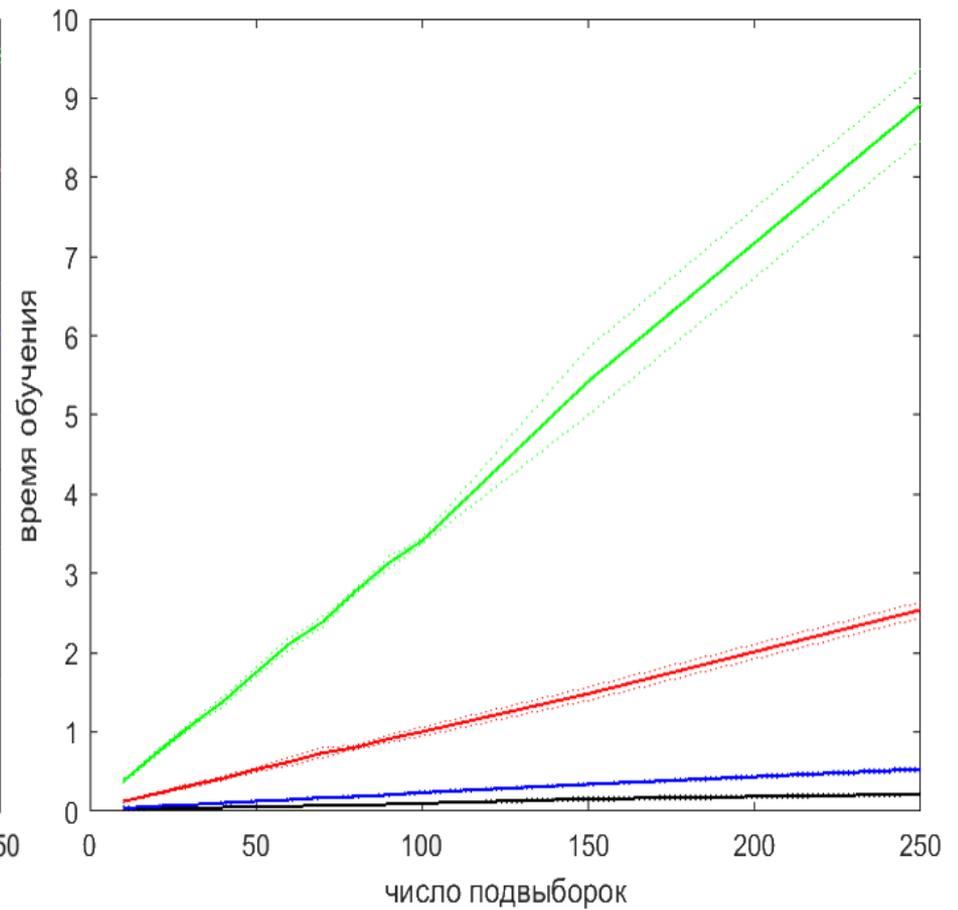
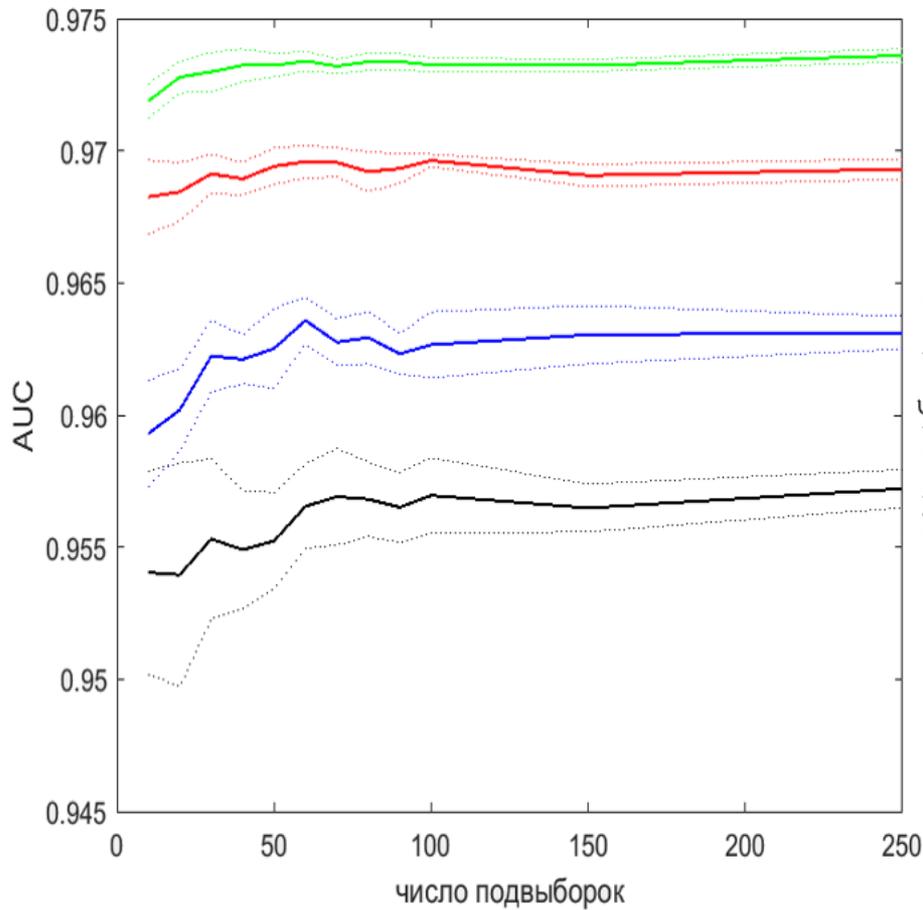
$\gamma = 0.75$

Параметры KMDR

Число подвыборок: 200

Число объектов в подвыборке: 50

Влияние параметров KMDR на качество и скорость решения задачи SVM



Число объектов в подвыборке:

—	50	—	250
—	100	—	500

Результаты тестирования на модельных данных

Число объектов	Число признаков	Метод	Размер подвыборки	AUC	Время обучения (с)	Время распознавания (с)
10 000	10	LibSVM	-	0.9873	6.977966	2.368463
		SVMlight	-	0.987299	6.91	2.32
		piSVM	-	0.987294	4.81	3.71
		KMDR	50	0.9702±0.0015	0.0199±0.009	1.099±0.04
			500	0.9849±0.0004	0.3493±0.029	2.431±0.31
1000	0.9867±0.0001		0.9841±0.015	2.68±0.021		
10 000	100	LibSVM	-	0.977504	92.589	32.242
		SVMlight	-	0.977092	41.93	14.92
		piSVM	-	0.976734	12.04	27.8
		KMDR	50	0.9552±0.0018	0.059±0.014	7.97±0.12
			500	0.9732±0.0004	1.749±0.054	28.75±0.19
1000	0.9760±0.0003		5.645±0.293	32.08±1.27		
100 000	10	LibSVM	-	0.990	3371.334	216.5087
		SVMlight	-	0.990	1871.75	203.07
		piSVM	-	0.990	1124.75	284.55
		KMDR	50	0.970±0.0015	0.0554±0.01	13.36±1.067
			500	0.9836±0.0003	0.4211±0.023	44.962±1.43
1000	0.9861±0.00008		1.0425±0.039	113.223±0.83		
100 000	100	LibSVM	-	0.985	4360.219	1053.589
		SVMlight	-	0.985	8558.51	1873.66
		piSVM	-	0.984	3708.81	2269.96
		KMDR	50	0.960±0.002	0.192±0.063	95.608±3.422
			500	0.977±0.0002	1.83±0.132	690.98±25.44
1000	0.979±0.0002		5.488±0.26	1097.26±44.61		

Спасибо за внимание!