

# Метод опорных векторов (Support Vector Machine, SVM)

ЦМФ

# Основная идея

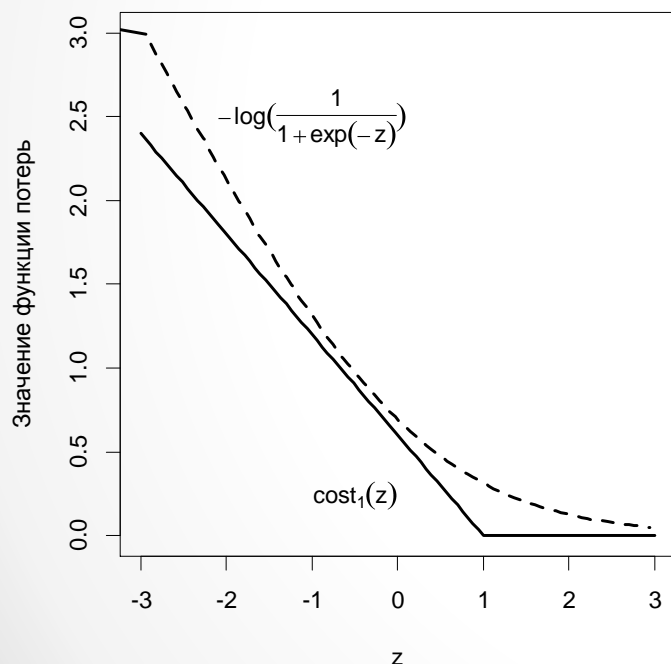
Основная идея состоит в модификации функции потерь, а также нелинейном преобразовании регрессоров и переводе их в пространство более высокой размерности, что позволяет строить сложные и эффективные разделяющие границы

# Модификация функции потерь

Ошибка на одном наблюдении в логистической регрессии:

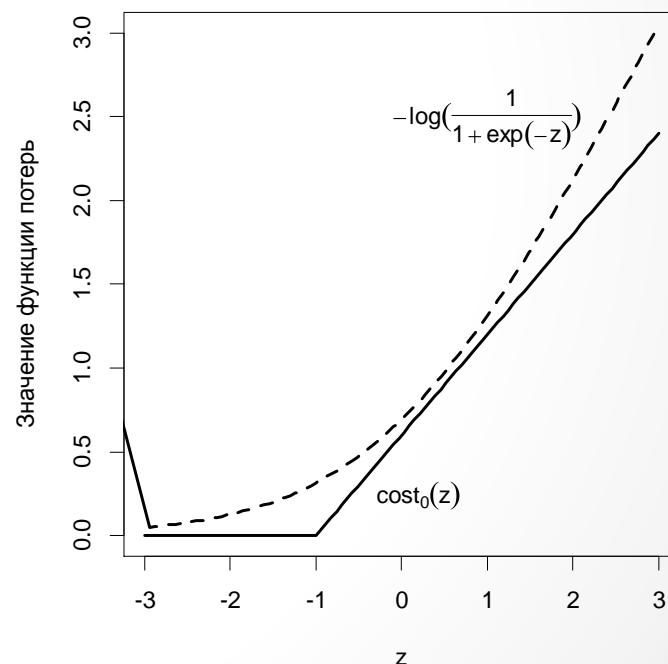
$$-y^{(i)} \log \frac{1}{1+e^{-\vec{\theta}^T \vec{x}^{(i)}}} - (1 - y^{(i)}) \log \left(1 - \frac{1}{1+e^{-\vec{\theta}^T \vec{x}^{(i)}}}\right)$$

Когда  $y^{(i)} = 1$ , мы хотим  
 $\vec{\theta}^T \vec{x}^{(i)} \geq 0$ ,



НО мы можем остановиться  
на  $\vec{\theta}^T \vec{x}^{(i)} \geq 1$

Когда  $y^{(i)} = 0$ , мы хотим  
 $\vec{\theta}^T \vec{x}^{(i)} < 0$ ,



НО мы можем остановиться  
на  $\vec{\theta}^T \vec{x}^{(i)} < -1$

# Функция потерь SVM

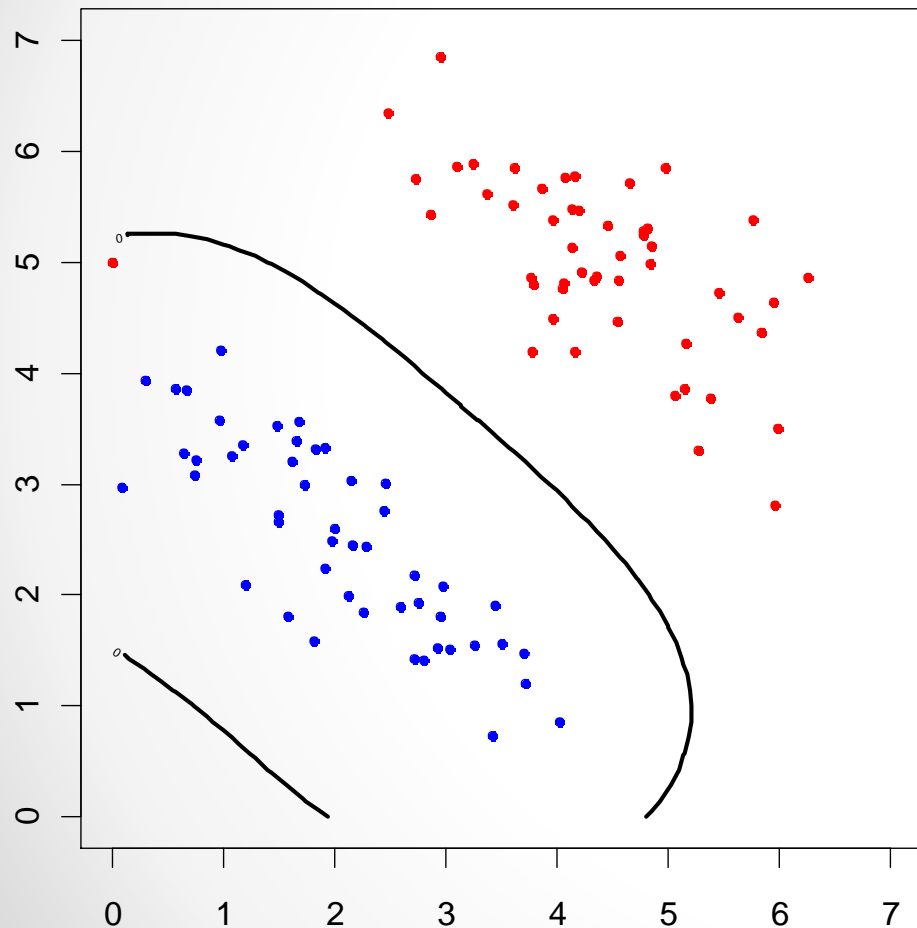
$$J(\vec{\theta}) = C \cdot \sum_{i=1}^m \left( y^{(i)} cost_1(\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) cost_0(\vec{\theta}^T \vec{x}^{(i)}) \right) + \\ + \frac{1}{2} \sum_{j=1}^n \theta_j^2 \rightarrow \min_{\vec{\theta}},$$

$C$  — параметр регуляризации,  $C \sim \frac{1}{\lambda}$

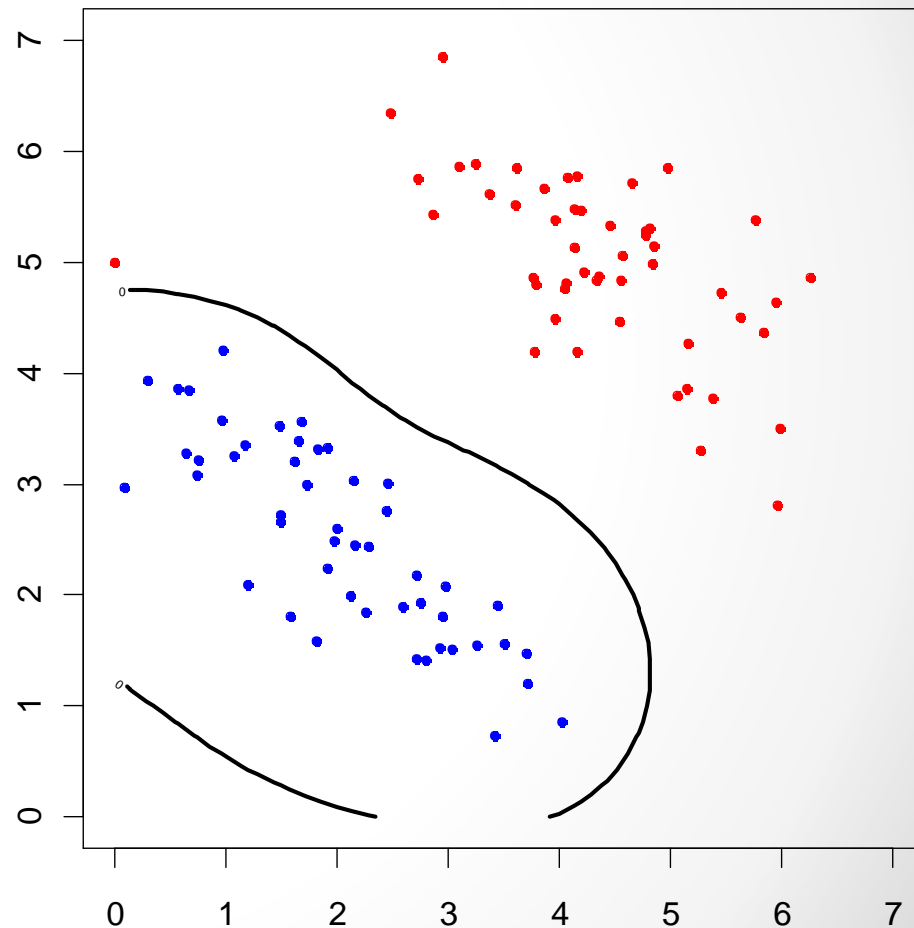
$$h_{\theta}(\vec{x}^{(i)}) = \begin{cases} 1, & \vec{\theta}^T \vec{x}^{(i)} \geq 0 \\ 0, & \vec{\theta}^T \vec{x}^{(i)} < 0 \end{cases}$$

# Влияние регуляционного параметра на разделительную границу

$C = 1$

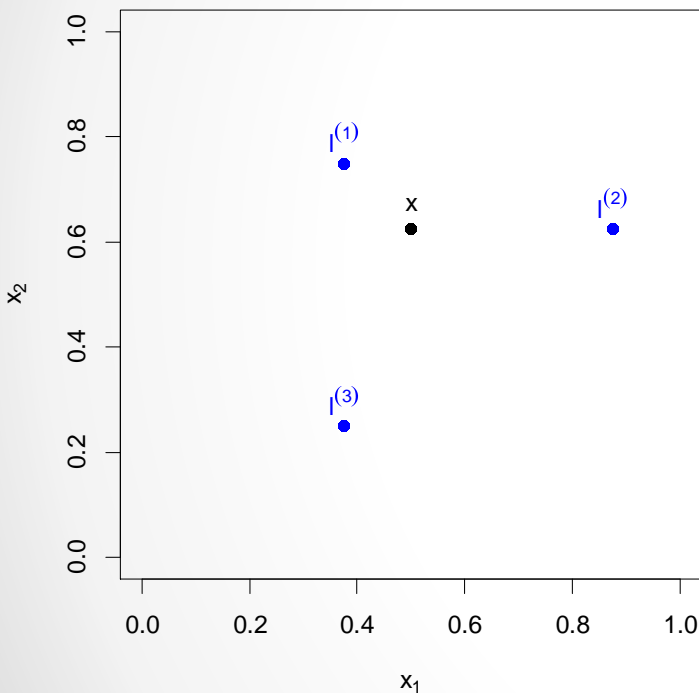


$C = 100$



# Модифицирование регрессоров, Kernel SVM

$$f_i = \exp\left(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{j=1}^n (x_j - l_j^{(i)})^2}{2\sigma^2}\right) \quad \leftarrow \text{ядро Гаусса}$$



$$x \approx l^{(i)} \Rightarrow f_i \approx 1$$

$x$  далеко от  $l^{(i)} \Rightarrow f_i \approx 0$ , т.е.

$$f_1 \approx 1, f_2 \approx f_3 \approx 0$$

$h_\theta = 1$ , если  $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$   
предсказание гипотезы зависит от того,  
насколько близко наблюдение находится  
к одним опорным точкам  $l^{(i)}$  и насколько  
далеко от других

В качестве опорных точек обычно берётся обучающая выборка

# Функция потерь SVM

$$\vec{x}^{(i)} \rightarrow \begin{pmatrix} f_0^{(i)}=1 \\ f_1^{(i)} \\ \vdots \\ f_i^{(i)}=1 \\ \vdots \\ f_m^{(i)} \end{pmatrix}, \quad \vec{f}^{(i)} \in R^{m+1}$$

$$J(\vec{\theta}) = C \cdot \sum_{i=1}^m \left( y^{(i)} cost_1(\vec{\theta}^T \vec{f}^{(i)}) + (1 - y^{(i)}) cost_0(\vec{\theta}^T \vec{f}^{(i)}) \right) + \\ + \frac{1}{2} \sum_{j=1}^m \theta_j^2 \rightarrow \min_{\vec{\theta}},$$

$$\vec{\theta} \in R^{m+1},$$

$$\vec{\theta}^T \vec{f} \geq 0 \Rightarrow h_{\theta}(\vec{f}) = 1$$

# Влияние параметров модели на ошибку и вариацию

Большое  $C \rightarrow$  малая ошибка, высокая вариация

Малое  $C \rightarrow$  большая ошибка, малая вариация

Большое  $\sigma \rightarrow$  большая ошибка, малая вариация

Малое  $\sigma \rightarrow$  малая ошибка, высокая вариация

Метод Kernel SMV следует применять, когда количество наблюдений  $m$  значительно превосходит их исходную размерность  $n$



# Метод опорных векторов в R

Пусть  $X$  — матрица регрессоров,  $y$  — вектор классов

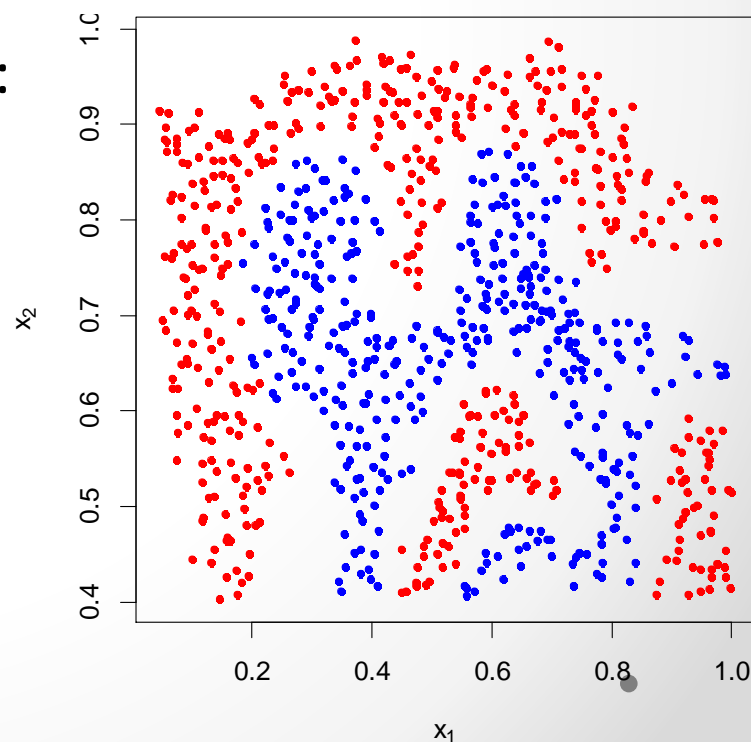
# разделение выборки на обучающую и экзаменующую

```
m <- nrow(X)
m.train <- round(0.8*m); m.cv <- m - m.train
train.obs <- sample(1:m,size=m.train,replace=FALSE)
cv.obs <- (1:m)[-train.obs]
X.train <- X[train.obs,]; X.cv <- X[cv.obs,]
y.train <- y[train.obs]; y.cv <- y[cv.obs]
```

# сетка экзогенных параметров модели:

#  $C$  и  $\sigma$

```
par <- c(0.01,0.05,0.1,0.5,1,5,10,50,
        100,500,1000)
par <- expand.grid(par,par)
# заголовки столбцов
dimnames(par)[[2]] <- c("C","sigma")
```



# Моделирование

```
res <- NULL # в неё будут записаны результаты моделирования

library(kernlab)
source("SVM_func.r") # файл с пользовательскими функциями

for (i in 1:nrow(par)) { # для каждой комбинации экз. параметров
  # подбор параметров  $\vec{\theta}$  на обучающей выборке
  model <- ksvm(X.train, y.train, type="C-svc",
               C = par$C[i], kern = "rbfdot",
               kpar = list(sigma=par$sigma[i]))
  # прогнозная классификация на экзаменующей выборке
  y.pred <- predict(model, newdata = X.cv, type = "response")
  # запись комбинации экзогенных параметров и статистик
  # прогноза, возвращаемых пользовательской функцией fitStats
  res <- rbind(res, c(par$C[i], par$sigma[i], fitStats(y.cv, y.pred)) )
}

dimnames(res)[[2]][1:2] <- c("C", "sigma") # заголовки столбцов
```

# Статистики прогноза

Для оценки качества бинарного классификационного алгоритма используют следующие показатели:

$$Accuracy = \frac{\#true\ pos. + \#true\ neg.}{m} \quad \text{— точность алгоритма}$$

$$Precision = \frac{\#true\ pos.}{\#true\ pos. + \#false\ pos.} \quad \text{— безошибочность выявления моделируемого признака}$$

$$Recall = \frac{\#true\ pos.}{\#true\ pos. + \#false\ neg.} \quad \text{— способность выявлять моделируемый признак}$$

$$F1.Score = 2 \frac{Precision \cdot Recall}{Precision + Recall} \quad \text{— общее качество алгоритма}$$

		$\vec{y}_{cv}^{(i)}$	
		1	0
$h_{\theta}(\vec{x}_{cv}^{(i)})$	1	true positive	false positive
	0	false negative	true negative

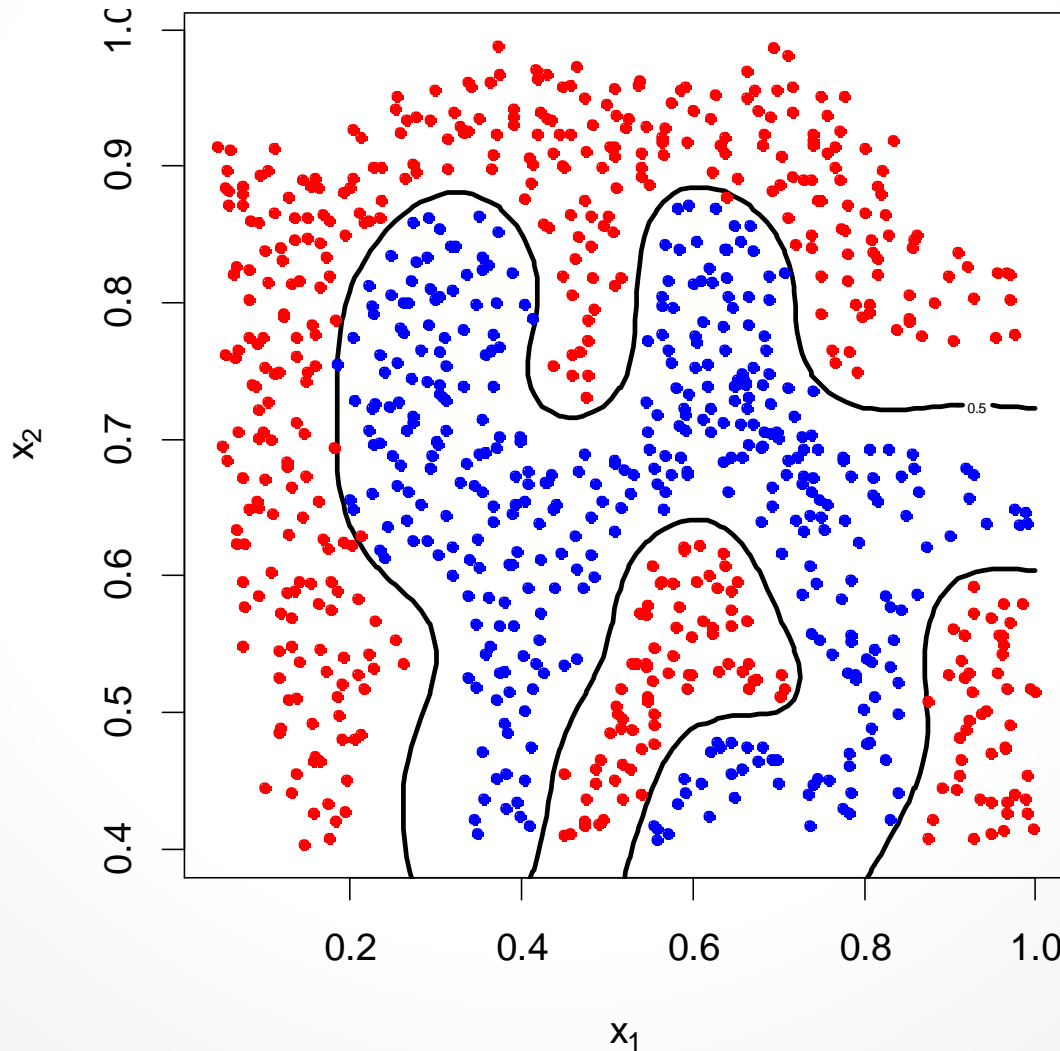
# Функция fitSats

```
# расчёт статистик качества прогноза
```

```
fitStats <- function(y,y.pred) {  
  ...  
  if (precision + recall == 0) f1.score <- 0  
  stat <- c(accuracy,precision,recall,f1.score)  
  names(stat) <- c("accuracy","precision","recall","f1.score")  
  stat  
}
```

# Выбор оптимальной комбинации экзогенных параметров

```
# номер комбинации параметров, максимизирующей f1.score  
j <- which.max(res[, "f1.score"])  
res[j,] # вывод на экран
```



# Домашнее задание

В файле «[elections\\_usa96\\_train.csv](#)» находятся данные экзит-пулов, проведённых во время президентских выборов в США в 1996-м году, в которых соревновались Билл Клинтон и Боб Доул

Вашей задачей является предсказание выбора респондентов (значения «Clinton» или «Dole») из тестовой выборки «[elections\\_usa96\\_test.csv](#)»

# Описание переменных

popul	TVnews	ClinLR	DoleLR	age	educ	income	vote
31	2	Lib	Con	36	BAdeg	\$75K-\$90K	Dole
22	7	sliLib	Mod	47	HS	\$25K-\$30K	Clinton
87	4	Mod	Con	41	Coll	\$30K-\$35K	Clinton
50	4	Lib	Con	44	HS	\$50K-\$60K	Clinton
9	7	extLib	Mod	79	MAdeg	\$30K-\$35K	Dole
75	4	Lib	extCon	62	BAdeg	\$105Kplus	Dole

popul — население города, в котором проживает респондент, тыс. чел.

TVnews — количество вечеров, проведённых за просмотром новостей за последнюю неделю

ClinLR — оценка респондентом политических взглядов Билла Клинтона, упорядоченные значения: extLib (крайне либеральные) → Lib (либеральные) → sliLib (умеренно либеральные) → Mod (в целом умеренные) → sliCon (умеренно консервативные) → Con (консервативные) → extCon (крайне консервативные)

DoleLR — оценка респондентом политических взглядов Боба Дойла, та же шкала  
age — возраст респондента

продолжение на следующем слайде

# Описание переменных

начало на предыдущем слайде

- educ — уровень образования респондента, упорядоченные значения: MS (начальная школа) → HSdrop (неоконченное среднее) → HS (среднее) → Coll (неоконченное высшее) → CCdeg (оконченный двухлетний колледж) → BAdeg (степень бакалавра) → MAdeg (степень магистра)
- income — уровень дохода респондента
- vote — кандидат, за которого отдан голос

Ответы принимаются на  
[https://kaggle.com/join/cmf\\_elections](https://kaggle.com/join/cmf_elections)