

# Soft margin SVM в pyomo

Городницкий Олег, 374 гр.

21 декабря 2016 г.

## Постановка задачи

Часто в алгоритмах машинного обучения возникает необходимость классифицировать данные. Каждый объект данных представлен как вектор (точка) в  $p$ -мерном пространстве (последовательность  $p$  чисел). Каждая из этих точек принадлежит только одному из двух классов. Нас интересует, можем ли мы разделить точки гиперплоскостью размерности  $(p-1)$ . Это типичный случай линейной разделимости. Таких гиперплоскостей может быть много. Поэтому вполне естественно полагать, что максимизация зазора между классами способствует более уверенной классификации. То есть можем ли мы найти такую гиперплоскость, чтобы расстояние от неё до ближайшей точки было максимальным. Это бы означало, что расстояние между двумя ближайшими точками, лежащими по разные стороны гиперплоскости, максимально. Если такая гиперплоскость существует, то она нас будет интересовать больше всего; она называется оптимальной разделяющей гиперплоскостью, а соответствующий ей линейный классификатор называется оптимально разделяющим классификатором.

Оптимизационная постановка Soft margin SVM:

$$\begin{cases} \sum_{i=1}^N \frac{1}{n} \xi_i + \lambda \|w\|^2 \rightarrow \min \\ \forall i \in \overline{1, N} \quad \xi_i \geq 0 \\ \forall i \in \overline{1, N} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i \end{cases}$$

## Модель pyomo

```
from __future__ import division
from pyomo.environ import *

model = AbstractModel()

model.n = Param(within=NonNegativeIntegers)
model.m = Param(within=NonNegativeIntegers)

model.I = RangeSet(1, model.n)
model.J = RangeSet(1, model.m)

model.X = Param(model.I, model.J)
model.y = Param(model.I)
model.ksis = Var(model.I, domain=NonNegativeReals)
model.b = Var()
model.w = Var(model.J)

def obj_expression(model):
    return sum(model.ksis[i] for i in model.I) + model.l*model.n*sum(model.w[j]**2 for j in model.J)

model.OBJ = Objective(rule=obj_expression, sense=minimize)

def constI(model, i):
    # return the expression for the constraint for i
    return model.y[i] * (sum(model.w[j] * model.X[i,j] for j in model.J) + model.b) >= 1 - model.ksis[i]

model.c1 = Constraint(model.I, rule=constI)
```

## Данные

В качестве входных данных использовались 6 точек на плоскости.

```
param n := 6;
param m := 2;
```

```

param X:
    1 2 :=
1 4 6
2 5 5
3 6 3
4 0 0
5 1 2
6 4 0
;
param y := [1] 1 [2] 1 [3] 1 [4] 0 [5] 0 [6] 0;
param l := 0.01;

```

## Решатель bonmin

Командная строка:

```

pyomo solve PYOMO_SVM.py PYOMO_SVM.dat --solver=bonmin
[ 0.00] Setting up Pyomo environment
[ 0.00] Applying Pyomo preprocessing actions
[ 0.00] Creating model
[ 0.11] Applying solver
[ 0.50] Processing results
Number of solutions: 1
Solution Information
    Gap: None
    Status: optimal
    Function Value: 0.0184615403012
Solver results file: results.json
[ 0.50] Applying Pyomo postprocessing actions
[ 0.51] Pyomo Finished

```

Результат:

```

{
  "Problem": [
    {
      "Lower bound": -Infinity,
      "Number of constraints": 0,
      "Number of objectives": 1,
      "Number of variables": 9,
      "Sense": "unknown",
      "Upper bound": Infinity
    }
  ],
  "Solution": [
    {
      "number of solutions": 1,
      "number of solutions displayed": 1
    },
    {
      "Constraint": "No values",
      "Gap": null,
      "Message": "bonmin\\x3a Optimal",
      "Objective": {
        "OBJ": {
          "Value": 0.018461540301163166
        }
      },
      "Problem": {},
      "Status": "optimal",
      "Variable": {
        "b": {
          "Value": -2.231346692117955
        },
        "w[1]": {

```

```

        "Value": 0.30783667280350335
    },
    "w[2]": {
        "Value": 0.46144221873381747
    }
}
],
"Solver": [
    {
        "Error rc": 0,
        "Id": 3,
        "Message": "bonmin\\x3a Optimal",
        "Status": "ok",
        "Termination condition": "optimal",
        "Time": 0.3874170780181885
    }
]
}

```

## Решатель couenne

Командная строка:

```

pyomo solve PYOMO_SVM.py PYOMO_SVM.dat --solver=couenne
[ 0.00] Setting up Pyomo environment
[ 0.00] Applying Pyomo preprocessing actions
[ 0.00] Creating model
[ 0.11] Applying solver
[ 0.15] Processing results
Number of solutions: 1
Solution Information
  Gap: None
  Status: optimal
  Function Value: 0.0184615402475
Solver results file: results.json
[ 0.15] Applying Pyomo postprocessing actions
[ 0.15] Pyomo Finished

```

Результат:

```

{
  "Problem": [
    {
      "Lower bound": -Infinity,
      "Number of constraints": 0,
      "Number of objectives": 1,
      "Number of variables": 9,
      "Sense": "unknown",
      "Upper bound": Infinity
    }
  ],
  "Solution": [
    {
      "number of solutions": 1,
      "number of solutions displayed": 1
    },
    {
      "Constraint": "No values",
      "Gap": null,
      "Message": "couenne\\x3a Optimal",
      "Objective": {
        "OBJ": {
          "Value": 0.018461540247523396
        }
      },
      "Problem": {},
      "Status": "optimal",

```

```

    "Variable": {
      "b": {
        "Value": -2.2313402820227397
      },
      "w[1]": {
        "Value": 0.3078350703728699
      },
      "w[2]": {
        "Value": 0.46144328677233293
      }
    }
  },
  "Solver": [
    {
      "Error rc": 0,
      "Id": 3,
      "Message": "couenne\\x3a Optimal",
      "Status": "ok",
      "Termination condition": "optimal",
      "Time": 0.028950929641723633
    }
  ]
}

```

## Решатель glpk

Командная строка:

```
pyomo solve PYOMO_SVM.py PYOMO_SVM.dat --solver=glpk
```

```

[ 0.00] Setting up Pyomo environment
[ 0.00] Applying Pyomo preprocessing actions
[ 0.00] Creating model
[ 0.11] Applying solver
[ 0.11] Pyomo Finished

```

ERROR: Unexpected exception while running model:

Selected solver is unable to handle objective functions with quadratic terms. Objective at issue: OBJ.

Результат:

```

{
  "Problem": [
    {
      "Lower bound": -Infinity,
      "Number of constraints": 6,
      "Number of objectives": 1,
      "Number of variables": 9,
      "Sense": "unknown",
      "Upper bound": Infinity
    }
  ],
  "Solution": [
    {
      "number of solutions": 1,
      "number of solutions displayed": 1
    },
    {
      "Constraint": "No values",
      "Gap": null,
      "Message": "Ipopt 3.12.4\\x3a Optimal Solution Found",
      "Objective": {
        "OBJ": {
          "Value": 0.01846154179248524
        }
      },
      "Problem": {},

```

```

    "Status": "optimal",
    "Variable": {
      "b": {
        "Value": -2.2313037694724223
      },
      "w[1]": {
        "Value": 0.3078259302907522
      },
      "w[2]": {
        "Value": 0.4614494120147086
      }
    }
  },
  ],
  "Solver": [
    {
      "Error rc": 0,
      "Id": 0,
      "Message": "Ipopt 3.12.4\\x3a Optimal Solution Found",
      "Status": "ok",
      "Termination condition": "optimal",
      "Time": 0.37284207344055176
    }
  ]
}

```

## Решатель ipopt

Командная строка:

```

pyomo solve PYOMO_SVM.py PYOMO_SVM.dat --solver=ipopt
[ 0.00] Setting up Pyomo environment
[ 0.00] Applying Pyomo preprocessing actions
[ 0.00] Creating model
[ 0.11] Applying solver
[ 0.49] Processing results
Number of solutions: 1
Solution Information
  Gap: None
  Status: optimal
  Function Value: 0.0184615417925
Solver results file: results.json
[ 0.49] Applying Pyomo postprocessing actions
[ 0.49] Pyomo Finished

```

Результат:

```

{
  "Problem": [
    {
      "Lower bound": -Infinity,
      "Number of constraints": 6,
      "Number of objectives": 1,
      "Number of variables": 9,
      "Sense": "unknown",
      "Upper bound": Infinity
    }
  ],
  "Solution": [
    {
      "number of solutions": 1,
      "number of solutions displayed": 1
    },
    {
      "Constraint": "No values",

```

```

    "Gap": null,
    "Message": "Ipopt 3.12.4\\x3a Optimal Solution Found",
    "Objective": {
      "OBJ": {
        "Value": 0.01846154179248524
      }
    },
    "Problem": {},
    "Status": "optimal",
    "Variable": {
      "b": {
        "Value": -2.2313037694724223
      },
      "w[1]": {
        "Value": 0.3078259302907522
      },
      "w[2]": {
        "Value": 0.4614494120147086
      }
    }
  }
],
"Solver": [
  {
    "Error rc": 0,
    "Id": 0,
    "Message": "Ipopt 3.12.4\\x3a Optimal Solution Found",
    "Status": "ok",
    "Termination condition": "optimal",
    "Time": 0.37341785430908203
  }
]
}

```

## Решатель scip

Командная строка:

```

pyomo solve PYOMO_SVM.py PYOMO_SVM.dat --solver=scip
[ 0.00] Setting up Pyomo environment
[ 0.00] Applying Pyomo preprocessing actions
[ 0.00] Creating model
[ 0.11] Applying solver
[ 0.53] Processing results
Number of solutions: 1
Solution Information
  Gap: None
  Status: optimal
  Function Value: 0.0184615505138
Solver results file: results.json
[ 0.53] Applying Pyomo postprocessing actions
[ 0.53] Pyomo Finished

```

Результат:

```

{
  "Problem": [
    {
      "Lower bound": -Infinity,
      "Number of constraints": 0,
      "Number of objectives": 1,
      "Number of variables": 9,
      "Sense": "unknown",
      "Upper bound": Infinity
    }
  ],
  "Solution": [

```

```

{
  "number of solutions": 1,
  "number of solutions displayed": 1
},
{
  "Constraint": "No values",
  "Gap": null,
  "Message": "optimal solution found",
  "Objective": {
    "OBJ": {
      "Value": 0.018461550513756426
    }
  },
  "Problem": {},
  "Status": "optimal",
  "Variable": {
    "b": {
      "Value": -2.232259668261285
    },
    "w[1]": {
      "Value": 0.30806491693233307
    },
    "w[2]": {
      "Value": 0.4612900557329215
    }
  }
}
],
"Solver": [
  {
    "Error rc": 0,
    "Id": 0,
    "Message": "optimal solution found",
    "Status": "ok",
    "Termination condition": "optimal",
    "Time": 0.41566991806030273
  }
]
}

```