機械学習特論

~ 理論とアルゴリズム ~

第 10 回

(Feature Selection)

講師：西郷浩人

Excercise from previous lecture

Algorithm: L2-Logistic Regression by IRLS

- Input:
  - X: n x p data matrix
  - y: n x 1 binary response vector
- Output
  - β: p x 1 coefficient vector
- Initialize
  - J=0; prev_J = LARGE_NUMBER;
- Repeat
  - Compute J $J(\boldsymbol{\beta}) = -\left(\boldsymbol{y}'\log h(\boldsymbol{X}\boldsymbol{\beta}) + (\mathbf{1}-\boldsymbol{y})'\log(\mathbf{1}-h(\boldsymbol{X}\boldsymbol{\beta}))\right) \underline{+\frac{\lambda}{2}\|\boldsymbol{\beta}\|_2}$
  - If prev_J – J < SMALL_NUMBER
    - Break
  - Compute w $\boldsymbol{W} = diag\left(h(\boldsymbol{X}\boldsymbol{\beta})(\mathbf{1}-h(\boldsymbol{X}\boldsymbol{\beta}))\right)$
  - Compute z $\boldsymbol{z} = \left(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{W}^{-1}(\boldsymbol{y}-h(\boldsymbol{X}\boldsymbol{\beta}))\right)$
  - Solve $(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X} \underline{+\lambda \boldsymbol{I}})\boldsymbol{\beta} = \boldsymbol{X}'\boldsymbol{W}\boldsymbol{z}$ for β

logisticIRLSL2.m

```
function [beta, J, w] = logisticIRLSL2(X,y,lambda)
[n p] = size(X);
beta = zeros(p,1);
itr = 0;
J = 0;
while 1
  itr = itr + 1;
  prev_J = J;
J = -(y'*log(h(X*beta))+(ones(n,1)-y)'*log(ones(n,1)-h(X*beta)))
   + lambda*norm(beta,2)/2;

  if abs(prev_J-J) < 1/n break; end

  W = diag(h(X*beta).*(ones(n,1)-h(X*beta)));
  z = X*beta + W\(y-h(X*beta));
  beta = (X'*W*X+lambda*eye(p))\(X'*W*z);

end

disp(['converged in ',num2str(itr),' iterations']);
disp(['sparsity: ',num2str(sum(abs(beta)<0.001)/length(beta))]);
disp(['likelihood: ',num2str(-J)]);
endfunction

function [y] = h(x)
y = 1.0 ./ (1.0 + exp(-x));
endfunction
```

$$J(\boldsymbol{\beta}) = -\left(\boldsymbol{y}'\log h(\boldsymbol{X}\boldsymbol{\beta}) + (\boldsymbol{1}-\boldsymbol{y})'\log(\boldsymbol{1}-h(\boldsymbol{X}\boldsymbol{\beta}))\right) + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2$$

$$\boldsymbol{\beta} = \left(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}'\boldsymbol{W}\boldsymbol{z}$$

Algorithm: L1-Logistic Regression by IRLS

- Input:
  - X: n x p data matrix
  - y: n x 1 binary response vector
- Output
  - β: p x 1 coefficient vector
- Initialize
  - J=0; prev_J = LARGE_NUMBER;
- Repeat
  - Compute J $J(\boldsymbol{\beta}) = -\left(\boldsymbol{y}'\log h(\boldsymbol{X}\boldsymbol{\beta}) + (\mathbf{1}-\boldsymbol{y})'\log(\mathbf{1}-h(\boldsymbol{X}\boldsymbol{\beta}))\right) + \lambda\|\boldsymbol{\beta}\|_1$
  - If prev_J – J < SMALL_NUMBER
    - Break
  - Compute w $\boldsymbol{W} = diag\left(h(\boldsymbol{X}\boldsymbol{\beta})(\mathbf{1}-h(\boldsymbol{X}\boldsymbol{\beta}))\right)$
  - Compute z $\boldsymbol{z} = \left(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{W}^{-1}(\boldsymbol{y}-h(\boldsymbol{X}\boldsymbol{\beta}))\right)$
  - Solve $(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X} + \lambda\mathbf{B})\boldsymbol{\beta} = \boldsymbol{X}'\boldsymbol{W}\boldsymbol{z}$ for β

logisticIRLSL1.m

```
function [beta, J, w] = logisticIRLSL1(X,y,lambda)
[n p] = size(X);
beta = zeros(p,1);
itr = 0;
J = 0;
while 1
  itr = itr + 1;
  prev_J = J;
J = -(y'*log(h(X*beta))+(ones(n,1)-y)'*log(ones(n,1)-h(X*beta)))
   + lambda*norm(beta,1);

  if abs(prev_J-J) < 1/n break; end

  W = diag(h(X*beta).*(ones(n,1)-h(X*beta)));
  z = X*beta + W\(y-h(X*beta));
  beta = (X'*W*X+lambda*pinv(diag(abs(beta))))\(X'*W*z);

end

disp(['converged in ',num2str(itr),' iterations']);
disp(['sparsity: ',num2str(sum(abs(beta)<0.001)/length(beta))]);
disp(['likelihood: ',num2str(-J)]);
endfunction

function [y] = h(x)
y = 1.0 ./ (1.0 + exp(-x));
endfunction
```

$$J(\boldsymbol{\beta}) = -\left(\boldsymbol{y}'\log h(\boldsymbol{X}\boldsymbol{\beta}) + (\mathbf{1}-\boldsymbol{y})'\log(\mathbf{1}-h(\boldsymbol{X}\boldsymbol{\beta}))\right) + \lambda\|\boldsymbol{\beta}\|_1$$

$$\boldsymbol{\beta} = \left(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X} + \lambda\mathbf{B}\right)^{-1}\boldsymbol{X}'\boldsymbol{W}\boldsymbol{z}$$

$$\mathbf{B} = diag\left(1/|\beta_1|, 1/|\beta_2|, \ldots, 1/|\beta_p|\right)$$

About feature selection

- *Optimal* feature selection is an *NP-hard* problem.
- Consider choosing 10 variables while paying attention to the *order*.
  - 10! = 3.63e6 approx. 3 million possible orders.
  - 20! = 2.43e18 possible orders. (2.43 京)
  - 50! = 3.04e64 possible orders. (3 無量大数)

Feature selection (a.k.a. Variable selection)

- Pros
  - We might get better prediction performance
  - Small number of variables is easier to interpret
  - Testing is faster.
- Cons
  - Performance often degrades
- Methods for Feature selection
  - Forward stepwise selection
  - Backward stepwise elimination
  - Hypothesis testing
  - Sparsity inducing methods (by L1-norm)

In case of Regression

Forward selection

$$\boldsymbol{y} \sim f(\boldsymbol{x}) = \beta_0 + \sum_{j=i}^{p} \beta_j x_j$$

- Think about the above regression case, where we have at most p variables.
- Forward selection is a simple approach that starts with a null model

$$f(\boldsymbol{x}) = \beta_0$$

  and adds a variable one by one.
- A variable that decreases RSS most is selected.

Backward elimination

- On the otherhand, backward elimination starts with a full model

$$\boldsymbol{y} \sim f(\boldsymbol{x}) = \beta_0 + \sum_{j=i}^{p} \beta_j x_j$$

and eliminates variables one by one.

- A variable that decreases RSS most is selected.

Hybrid

$$\boldsymbol{y} \sim f(\boldsymbol{x}) = \beta_0 + \sum_{j=i}^{p} \beta_j x_j$$

- Combination of forward selection and backward elimination. It starts with a null model

$$f(\boldsymbol{x}) = \beta_0$$

  and adds a variable one by one. Not only that, removal of a variable can occur as well.
- A variable that decreases RSS most is selected.

Excercise 1

- Download today's data. Typing "init.m" loads Longley data.
- Try
  - Forward stepwise selection

    ```
    >forward(X,y)
    ```
  - Backward stepwise elimination

    ```
    >backward(X,y)
    ```
  - Hybrid stepwise selection

    ```
    >hybrid(X,y)
    ```
- Choose a model with the smallest AIC.
- Do the results by the three methods agree ?

About Longley data

- The response variable (y) is the Total Derived Employment
- The predictor variables are
  - offset (x1),
  - GNP Implicit Price Deflator with Year 1954 = 100 (x2),
  - Gross National Product (x3),
  - Unemployment (x4),
  - Size of Armed Forces (x5),
  - Non-Institutional Population Age 14 & Over (x6),
  - Year (x7).

Variable selection by Correlation

Correlation

- Most simple idea
- Correlation betwen each feature and target response variable is calculated by

$$cor(\boldsymbol{x}, \boldsymbol{y}) = \frac{cov(\boldsymbol{x}, \boldsymbol{y})}{var(\boldsymbol{x})\, var(\boldsymbol{y})}$$

- In Octave, you can type "cor(x,y)" to calculate correlation.

Variable selection by Hypothesis Testing

Flowchart of hypothesis testing

Alternative Hypothesis H_A

Null Hypothesis H_0

Hypothesis test

Do not reject

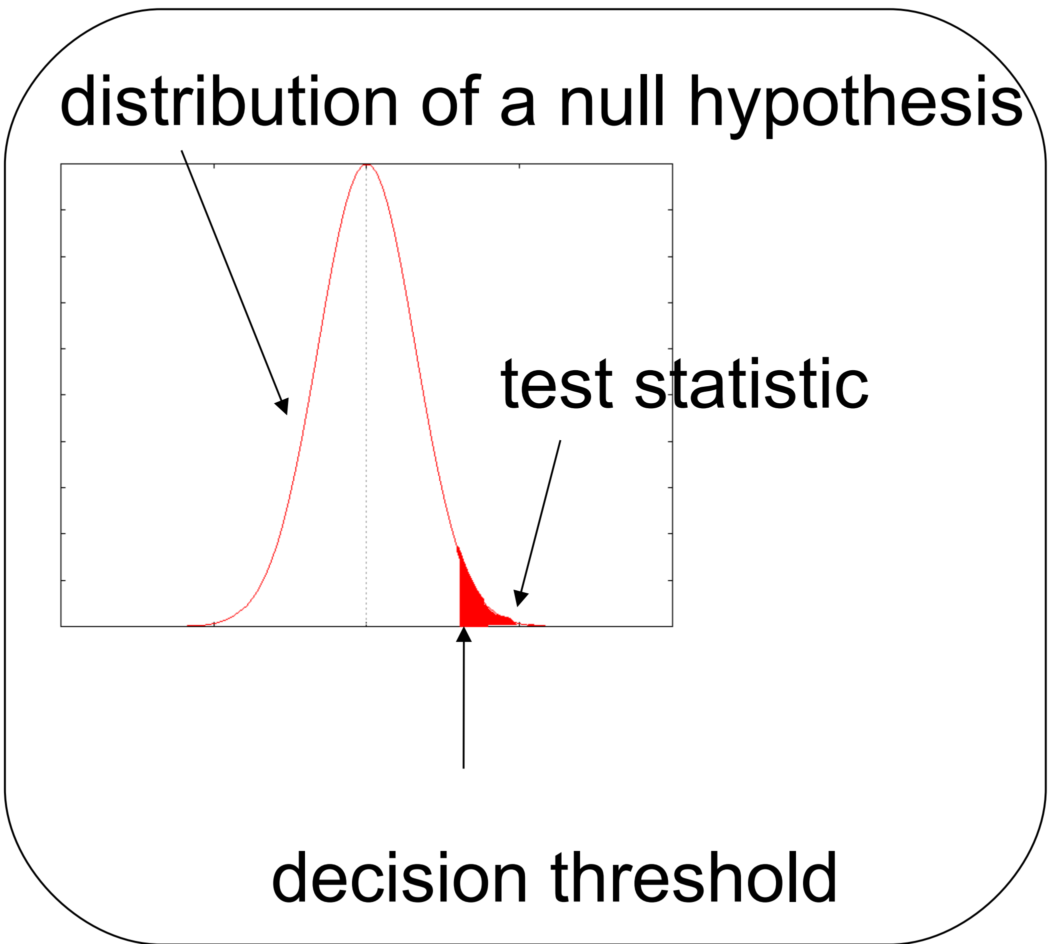
Reject

H_0 is likely

H_A is unlikely

H_0 is unlikely

H_A is likely



Testing if coefficient is 0 in regression

$$\boldsymbol{y} \sim f(\boldsymbol{x}) = \beta_0 + \sum_{j=i}^{p} \beta_j x_j$$

- Think about the above regression case.
- For each coefficient, we build the following null/alternative hypotheses, and perform a test.

$$H_A : \beta_j \neq 0 \qquad H_0 : \beta_j = 0$$

- Test statistic is $t = \dfrac{\beta_j}{SE(\beta_j)}$
  - where $SE(\beta_0) = \sqrt{C_{1,1}}$ $SE(\beta_j) = \sqrt{C_{j+1,j+1}}$ $\boldsymbol{C} = \dfrac{RSS}{df}(\boldsymbol{X}'\boldsymbol{X})^{-1}$

$$df = n - (p+1)$$

How to look at hypothesis testing results

- Coefficients with more *** are called as statistically significant.

Regression with 16 examples and 7 predictors
RSS: 0.067815 R-squared: 0.99548 logL: -5.4636 AIC: -26.011
residual standard error (sigma): 0.086804
F-statistic: 3.30e+02
F-test p-value: 4.98e-10(***)

Estimate	Std.Err	t-statistic	p-value
5.55e-17	2.17e-02	2.56e-15	1.00e+00( )
4.63e-02	2.61e-01	1.77e-01	8.63e-01( )
-1.01e+00	9.48e-01	-1.07e+00	3.13e-01( )
-5.38e-01	1.30e-01	-4.14e+00	2.54e-03(**)
-2.05e-01	4.25e-02	-4.82e+00	9.44e-04(***)
-1.01e-01	4.48e-01	-2.26e-01	8.26e-01( )
2.48e+00	6.17e-01	4.02e+00	3.04e-03(**)

Signif. codes: <0.000 (***) 0.001 (**) * 0.05 (.) 0.1 ( ) 1

Variable selection by LASSO

LASSO regression

$$min_{\boldsymbol{\beta}} \lambda \|\boldsymbol{\beta}\|_1 + \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|^2$$

- Minimizes RSS while placing **β** on a hyper polygon.

L1-norm regularization on a coefficient vector enforces sparsity; a few coefficients in **β** remains nonzero.

Excercise 2

- Try
  - Correlation

```
>b=cor(X,y)
>[S I]=sort(abs(b),'descend')
```

  - Hypothesis Testing

```
>HT_reg(X,y)
```

  - LASSO

```
>b=lasso(X,y,0.000001)
>find(abs(b)>0.001)
>[S I]=sort(abs(b),'descend')
```

- Do the results agree with Excercise 1 ?

Result of all subset regression 'leaps'

size	model	Cp
1	2	52.9
2	3,6	25.2
3	3,4,6	6.24
4	2,3,4,6	3.24
5	2,3,4,5,6	5.03
6	1,2,3,4,5,6	7.00

Methods and selected subsets

Forward	3,5,6,7
Backward	{}
Hybrid	3,5,6,7
Correlation	$3 \rightarrow 7 \rightarrow 2 \rightarrow 6 \rightarrow 4 \rightarrow 5$
Test	4,5,7
LASSO	$7 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 2 \rightarrow 1$
All subset	3,4,5,7

In case of classification

Testing if coefficient is 0 in classification

$$\boldsymbol{y} \sim f(\boldsymbol{x}) = \exp\left(\beta_0 + \sum_{j=i}^{p} \beta_j x_j\right)$$

- For each coefficient, we build the following null/alternative hypotheses, and perform a test.

$$H_A: \beta_j \neq 0 \qquad H_0: \beta_j = 0$$

- Test statistic is $z = \dfrac{\beta_j}{SE(\beta_j)}$
  - where $SE(\beta_0) = \sqrt{C_{1,1}}$ $\quad SE(\beta_1) = \sqrt{C_{2,2}}$ $\quad \boldsymbol{C} = (\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1}$
  - **W** is a diagonal matrix storing likelihood of each data point in the diagonal. Likelihood is obtained from the final iteration of the IRLS algorithm.

Variable selection by L1-norm Logistic Regression

L1-norm logistic regression

$$min_{\boldsymbol{\beta}} \lambda \|\boldsymbol{\beta}\|_1 - \left(\boldsymbol{y}' \log h(\boldsymbol{X}\boldsymbol{\beta}) + (\mathbf{1} - \boldsymbol{y})' \log(\mathbf{1} - h(\boldsymbol{X}\boldsymbol{\beta})) \right)$$

- Maximizes likelihood while placing **β** on a hyper polygon.
- L1-norm regularization on a coefficient vector enforces sparsity; a few coefficients in **β**remains nonzero.

(Excercise 3)

- Prepare a response vector for classification

```
>z=sign(y-mean(y));
>z(z==-1)=0
```

- Try
  - Correlation

    ```
    >b=cor(X,z)
    >[S I]=sort(abs(b),'descend')
    ```

  - Hypothesis Testing

    ```
    >HT_cls(X,z)
    ```

  - L1 Logistic Regression

    ```
    >b=logisticIRLSL1(X,z,0.1)
    >find(abs(b)>0.001)
    >[S I]=sort(abs(b),'descend')
    ```

(Excercise 4)

- Let's implement
  - Forward stepwise selection
  - Backward stepwise elimination
  - Hybrid stepwise selection

  for classification

Summary

- Because of infeasibility of optimal selection for moderately large *p*, exisiting all methods for variable selection are heuristics.
- In practice many variables are highly correlated to each other (e.g. Gene expression), in which the order of selecting variables makes difference in the final selected set.
- L1-norm variable selection is recently studied very well, since it can handle large number of variables, and perform variable selection by taking into account *correlation* in design matrix.

Overview on supervised learning

min (Regularizer + loss function)

- Most of modern supervised methods are designed to incorporate regularization, and written in the following form.

$$min_{\beta} \lambda \|\boldsymbol{\beta}\|_p + \sum_{i=1}^{n} loss\left(y_i, f\left(\boldsymbol{x_i}\right)\right)$$

$$min_{\beta} \lambda \|\boldsymbol{\beta}\|_p + \sum_{i=1}^{n} loss\left(y_i, f\left(\boldsymbol{x_i}\right)\right)$$

	Method	Regularizer	Loss
Regression	OLS	None	$\left(\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\right)^2$
	Ridge	$\|\boldsymbol{\beta}\|_2$	$\left(\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\right)^2$
	LASSO	$\|\boldsymbol{\beta}\|_1$	$\left(\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\right)^2$
	SVR	$\|\boldsymbol{\beta}\|_2$	$\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}-\epsilon$
Classification	LDA	None	$\left(\boldsymbol{1}-\boldsymbol{y}'\boldsymbol{X}\boldsymbol{\beta}\right)^2$
	SVM	$\|\boldsymbol{\beta}\|_2$	$max\left(\boldsymbol{0},\boldsymbol{1}-\boldsymbol{y}'\boldsymbol{X}\boldsymbol{\beta}\right)$
	LR	None	$-\log\left(1+\exp\left(-\boldsymbol{y}'\boldsymbol{X}\boldsymbol{\beta}\right)\right)$

Original LDA and LR does not have regularizers, but it is not difficult to add regularization as we have already seen.

Regularizers

- definition $$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

  - p =1: 1-norm (manhattan distance)

    $$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

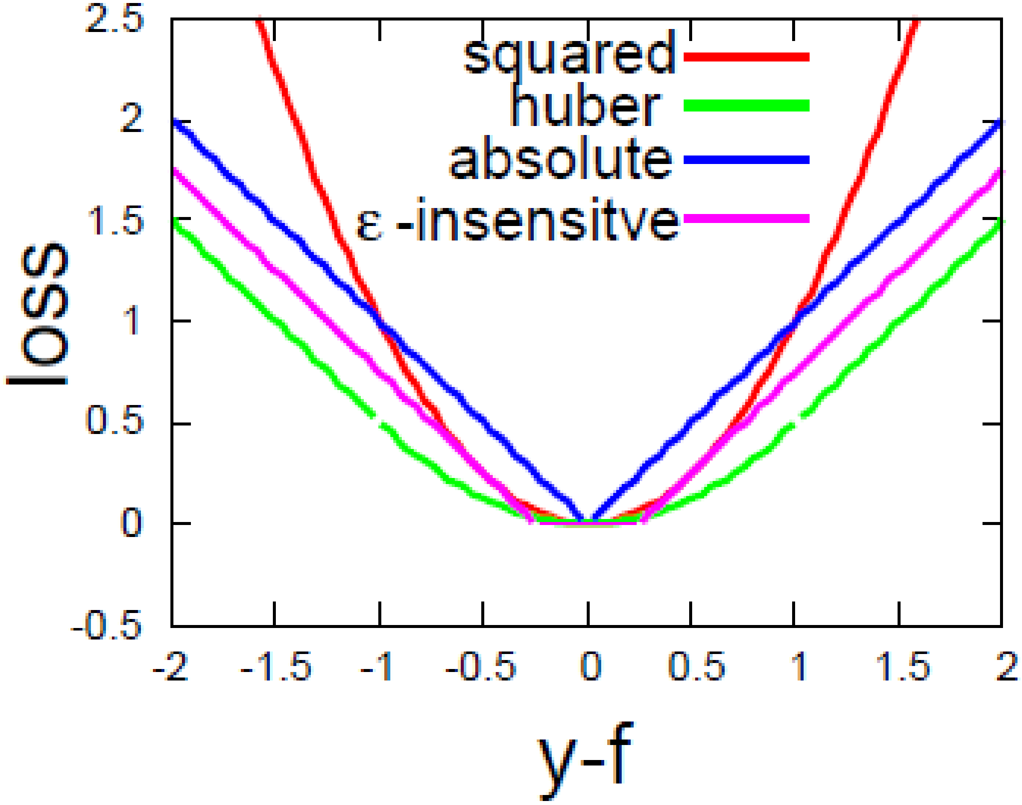
  - p = 2: 2-norm (euclidian distance)

    $$\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

  - p = ∞ : infinity-norm (chebyshev distance)

    $$\|\boldsymbol{x}\|_\infty = max_i |x_i|$$

Loss functions for regression

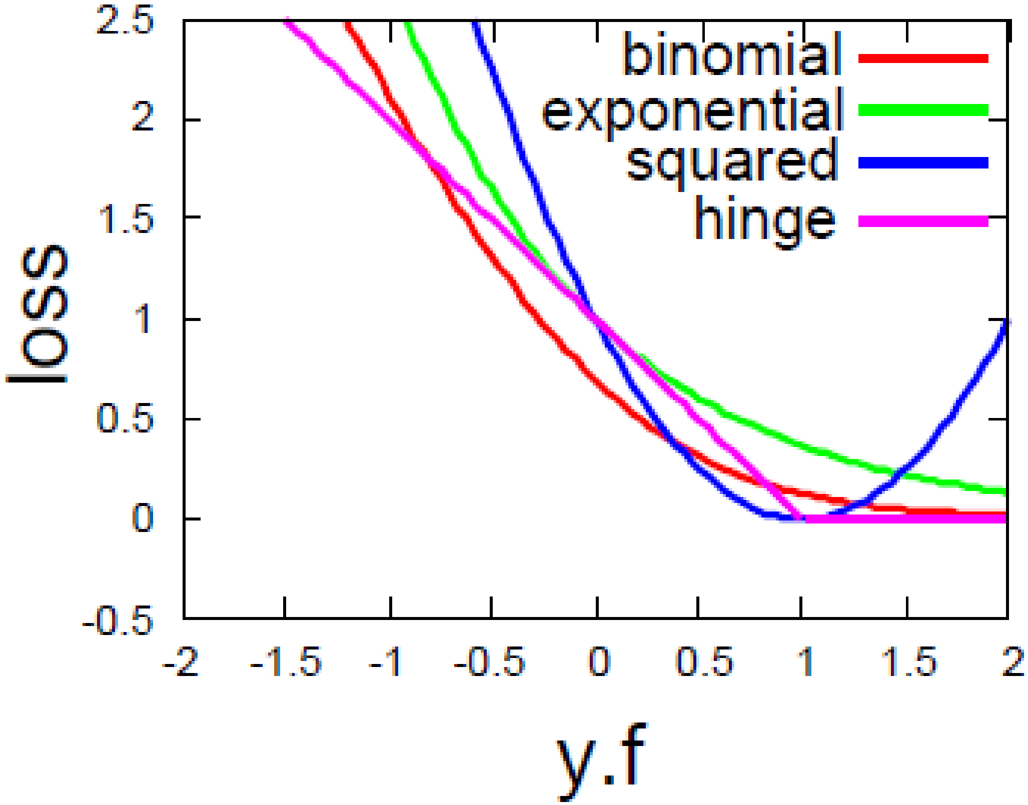


Method	Loss
OLS	$(X\beta - y)^2$
Ridge	$(X\beta - y)^2$
LASSO	$(X\beta - y)^2$
SVR	$X\beta - y - \epsilon$

OLS, Ridge, LASSO: squared loss

SVR: epsilon-insensitive loss

Loss functions for classification



Method	Loss
LDA	$(\mathbf{1} - \boldsymbol{y}'\boldsymbol{X}\boldsymbol{\beta})^2$
SVM	$max(\mathbf{0}, \mathbf{1} - \boldsymbol{y}'\boldsymbol{X}\boldsymbol{\beta})$
LR	$-\log(1 + \exp(-\boldsymbol{y}'\boldsymbol{X}\boldsymbol{\beta}))$

SVM: hinge loss

LR: binomial loss

LDA: squared loss