

Ordinal Regression Based on Learning Vector Quantization (LVQ)

Fengzhen Tang and Peter Tino

School of Computer Science, University of Birmingham
fxt126@cs.bham.ac.uk

December 14, 2014

Outline

- ① Introduction
- ② Ordinal Generalized Matrix LVQ (OGMLVQ)
- ③ Proposed Accumulative OGMLVQ
- ④ Experimental Results
- ⑤ Summary

Outline

① Introduction

② Ordinal Generalized Matrix LVQ (OGMLVQ)

③ Proposed Accumulative OGMLVQ

④ Experimental Results

⑤ Summary

What is Ordinal Regression?

Ordinal Regression: predicts categories of ordinal scale

- Labels are discrete
- More than two classes
- A natural order exists in class labels

E.g.

- Grade the exchange rate of currency: extremely down, down, up, extremely up
- Evaluate a product : very bad, bad, average, good, very good

What is Learning Vector Quantization (LVQ) ?

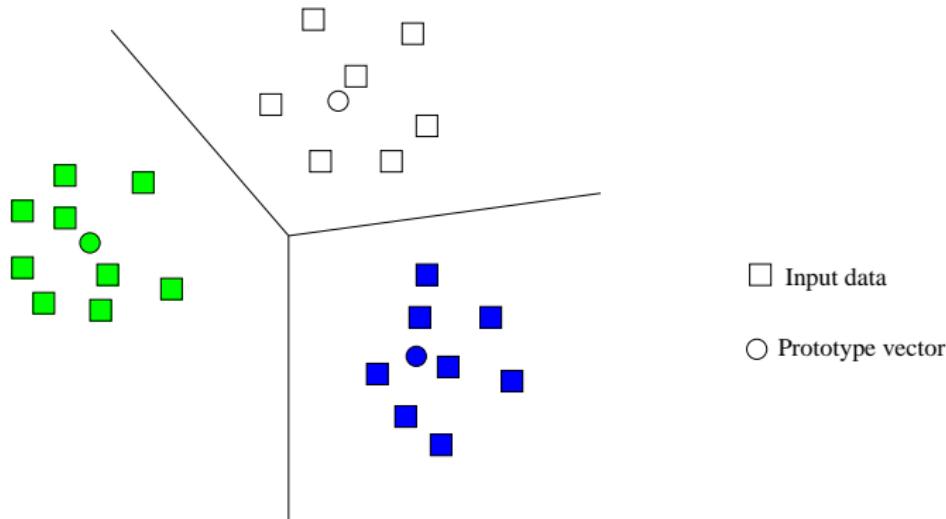


Figure: LVQ

- Prototype-based learning algorithms
- Winner takes all scheme

Advantages and Drawbacks of Original LVQ

Advantages:

- Naturally dealing with multi-class classification problem
- Decision rules: intuitive and interpretable
- Implementation: simple
- Scales well

Drawbacks:

- Euclidean distance scales every dimension equally – metric learning
- Original LVQ does not have principle formulation, preventing a proper mathematical investigation of the learning behaviour and generalization dynamics – generalized LVQ

Generalized Matrix LVQ (GMLVQ)

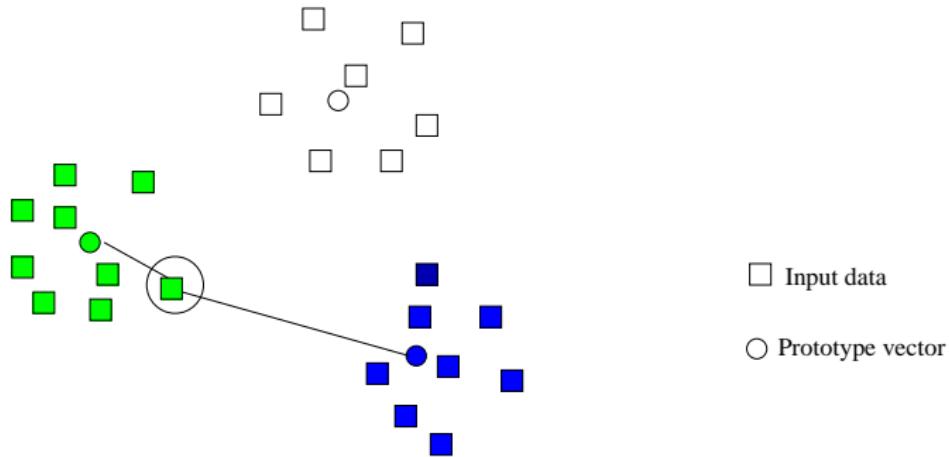


Figure: GMLVQ

Cost function: $E = \sum_i \Phi(\mu_i)$ where $\mu_i = \frac{d^\Lambda(x_i, w^+) - d^\Lambda(x_i, w^-)}{d^\Lambda(x_i, w^-) + d^\Lambda(x_i, w^+)}$

- w^+ and w^- : the closest correct and incorrect prototypes, respectively
- Squared distance $d^\Lambda(x_i, w) = (x_i - w)^T \cdot \Lambda \cdot (x_i - w)$, where $\Lambda = \Omega^T \Omega$



P. Schneider, P. Biehl, B. Hammer, Adaptive relevance matrices in learning vector quantization, Neural Computation 21 (12) (2009) 3532–3561.

Outline

1 Introduction

2 Ordinal Generalized Matrix LVQ (OGMLVQ)

3 Proposed Accumulative OGMLVQ

4 Experimental Results

5 Summary

Ordinal GMLVQ (OGMLVQ)

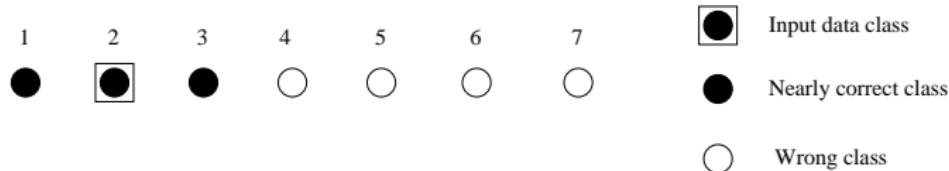


Figure: OGMLVQ

Cost function:

$$f = \sum_{i=1}^n \sum_{j=1}^r \Phi(\mu_j(\mathbf{x}_i)), \mu_j(\mathbf{x}_i) = \frac{\alpha_j^+ \cdot d^\Delta(\mathbf{x}_i, \mathbf{w}_j^+) - \alpha_j^- \cdot d^\Delta(\mathbf{x}_i, \mathbf{w}_j^-)}{\alpha_j^+ \cdot d^\Delta(\mathbf{x}_i, \mathbf{w}_j^+) + \alpha_j^- \cdot d^\Delta(\mathbf{x}_i, \mathbf{w}_j^-)}$$

- $\alpha_j^+ = \exp\left\{-\frac{h(c(\mathbf{x}_i) - c(\mathbf{w}_j^+))^2}{2\sigma^2}\right\}$
- $\alpha_j^- = \exp\left\{-\frac{(T - h(c(\mathbf{x}_i) - c(\mathbf{w}_j^-)))^2}{2\sigma^2}\right\} \cdot \exp\left\{-\frac{d^\Delta(\mathbf{x}_i, \mathbf{w})}{2\sigma'^2}\right\}$



S. Fouad and P. Tino. Adaptive metric learning vector quantization for ordinal classification. *Neural Computation*, 24(11):2825–2851, 2012.

OGMLVQ Updating Rule:

For each input \mathbf{x}_i :

$$\Delta \mathbf{w}_j^+ = \eta \cdot \Phi'(\mu_i) \cdot \gamma_j^+ \cdot \Lambda \cdot (\mathbf{x}_i - \mathbf{w}_j^+)$$

$$\Delta \mathbf{w}_j^- = -\eta \cdot \Phi'(\mu_i) \cdot \gamma_j^- \cdot \Lambda \cdot (\mathbf{x}_i - \mathbf{w}_j^-)$$

$$\Delta \Omega = -\epsilon \cdot \Phi'(\mu_i) \cdot$$

$$\left\{ \gamma_j^+ \cdot \Omega \cdot (\mathbf{x}_i - \mathbf{w}_j^+) (\mathbf{x}_i - \mathbf{w}_j^+)^T - \gamma_j^- \cdot \Omega \cdot (\mathbf{x}_i - \mathbf{w}^-) (\mathbf{x}_i - \mathbf{w}^-)^T \right\}$$

where $\gamma_j^+ = \frac{4\alpha_j^+ \alpha_j^- d^\Lambda(\mathbf{x}_i, \mathbf{w}^-)}{(\alpha_j^+ d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^+) + \alpha_j^- d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^-))^2}$, $\gamma_j^- = \frac{4\alpha_j^+ \alpha_j^- d^\Lambda(\mathbf{x}_i, \mathbf{w}^+)}{(\alpha_j^+ d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^+) + \alpha_j^- d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^-))^2}$,
and $j = 1, \dots, r$.

Problem

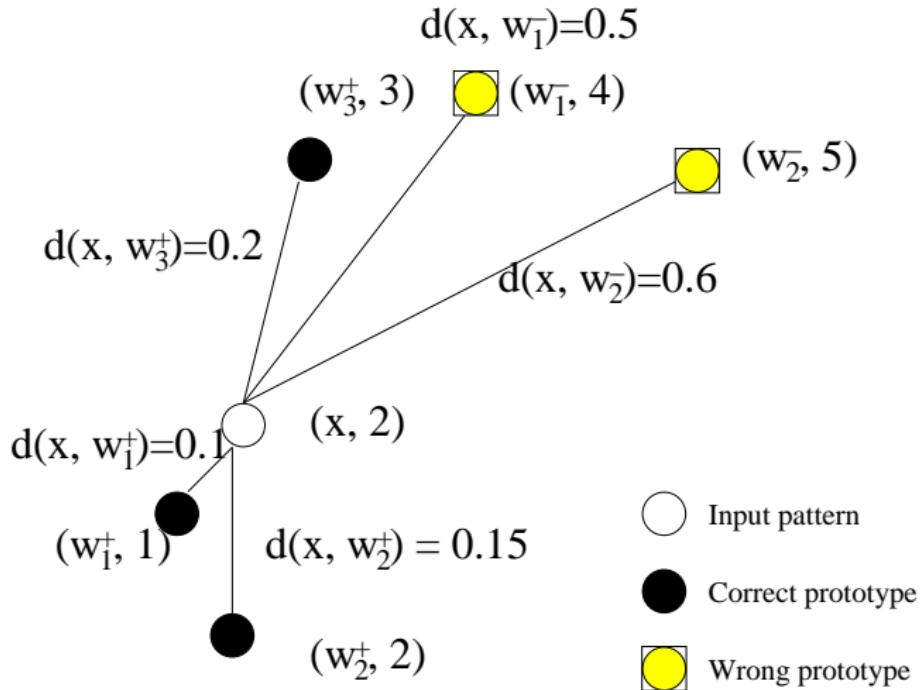


Figure: Problem in the existing Ordinal GMLVQ. Let σ and σ' be 1 and 10, respectively. $\gamma_1^+(2.7824)$ is greater than γ_2^+ (2.1372).

Outline

- ① Introduction
- ② Ordinal Generalized Matrix LVQ (OGMLVQ)
- ③ Proposed Accumulative OGMLVQ
- ④ Experimental Results
- ⑤ Summary

Proposed Cost Function

$$f = \sum_i \Phi(\mu(\mathbf{x}_i))$$

where

$$\mu(\mathbf{x}_i) = \frac{\sum_{j=1}^{r_1} \alpha_j^+ d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^+) - \sum_{j=1}^{r_2} \alpha_j^- d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^-)}{\sum_{j=1}^{r_1} \alpha_j^+ d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^+) + \sum_{j=1}^{r_2} \alpha_j^- d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^-)}$$

Differences in Terms of Cost Function

Ours:

$$f = \sum_i \Phi(\mu(\mathbf{x}_i)) \quad \mu(\mathbf{x}_i) = \frac{\sum_{j=1}^{r_1} \alpha_j^+ d^\wedge(\mathbf{x}_i, \mathbf{w}_j^+) - \sum_{j=1}^{r_2} \alpha_j^- d^\wedge(\mathbf{x}_i, \mathbf{w}_j^-)}{\sum_{j=1}^{r_1} \alpha_j^+ d^\wedge(\mathbf{x}_i, \mathbf{w}_j^+) + \sum_{j=1}^{r_2} \alpha_j^- d^\wedge(\mathbf{x}_i, \mathbf{w}_j^-)}$$

Original:

$$f = \sum_{i=1}^n \sum_{j=1}^r \Phi(\mu_j(\mathbf{x}_i)) \quad \mu_j(\mathbf{x}_i) = \frac{\alpha_j^+ \cdot d^\wedge(\mathbf{x}_i, \mathbf{w}_j^+) - \alpha_j^- \cdot d^\wedge(\mathbf{x}_i, \mathbf{w}_j^-)}{\alpha_j^+ \cdot d^\wedge(\mathbf{x}_i, \mathbf{w}_j^+) + \alpha_j^- \cdot d^\wedge(\mathbf{x}_i, \mathbf{w}_j^-)}$$

Updating Rule

$$\Delta \mathbf{w}_j^+ = \eta \cdot \Phi'(\mu_i) \cdot \mu_j^+ \cdot \Lambda \cdot (\mathbf{x}_i - \mathbf{w}_j^+), j = 1, \dots, r_1$$

$$\Delta \mathbf{w}_j^- = -\eta \cdot \Phi'(\mu_i) \cdot \mu_j^- \cdot \Lambda \cdot (\mathbf{x}_i - \mathbf{w}_j^-), j = 1, \dots, r_2$$

$$\begin{aligned} \Delta \Omega &= -\epsilon \cdot \Phi'(\mu_i) \cdot \left(\sum_{j=1}^{r_1} \mu_j^+ \cdot \Omega \cdot (\mathbf{x}_i - \mathbf{w}_j^+) (\mathbf{x}_i - \mathbf{w}_j^+)^T \right. \\ &\quad \left. \sum_{j=1}^{r_2} \mu_j^- \cdot \Omega \cdot (\mathbf{x}_i - \mathbf{w}_j^-) (\mathbf{x}_i - \mathbf{w}_j^-)^T \right) \end{aligned}$$

where $\mu_j^+ = \frac{4\alpha_j^+ \sum_{k=1}^{r_2} \alpha_k^- d^\Lambda(\mathbf{x}_i, \mathbf{w}_k^-)}{\left(\sum_{k=1}^{r_1} \alpha_k^+ d^\Lambda(\mathbf{x}_i, \mathbf{w}_k^+) + \sum_{k=1}^{r_2} \alpha_k^- d^\Lambda(\mathbf{x}_i, \mathbf{w}_k^-) \right)^2}$ and

$$\mu_j^- = \frac{4(1 - d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^-)/2\sigma_3^2) \alpha_j^- \sum_{k=1}^{r_1} \alpha_k^+ d^\Lambda(\mathbf{x}_i, \mathbf{w}_k^+)}{\left(\sum_{k=1}^{r_1} \alpha_k^+ d^\Lambda(\mathbf{x}_i, \mathbf{w}_k^+) + \sum_{k=1}^{r_2} \alpha_k^- d^\Lambda(\mathbf{x}_i, \mathbf{w}_k^-) \right)^2}$$

Differences in Terms of Updating Rule

Updating coefficients for correct prototypes:

$$\gamma_j^+ = \frac{4\alpha_j^+ \alpha_j^- d^\Lambda(\mathbf{x}_i, \mathbf{w}^-)}{\left(\alpha_j^+ d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^+) + \alpha_j^- d^\Lambda(\mathbf{x}_i, \mathbf{w}^-) \right)^2}$$
$$\mu_j^+ = \frac{4\alpha_j^+ \sum_{k=1}^{r_2} \alpha_k^- d^\Lambda(\mathbf{x}_i, \mathbf{w}_k^-)}{\left(\sum_{k=1}^{r_1} \alpha_k^+ d^\Lambda(\mathbf{x}_i, \mathbf{w}_k^+) + \sum_{k=1}^{r_2} \alpha_k^- d^\Lambda(\mathbf{x}_i, \mathbf{w}_k^-) \right)^2}$$

Updating coefficients for the incorrect prototypes:

$$\gamma_j^- = \frac{4\alpha_j^+ \alpha_j^- d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^+)}{\left(\alpha_j^+ d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^+) + \alpha_j^- d^\Lambda(\mathbf{x}_i, \mathbf{w}^-) \right)^2}$$
$$\mu_j^- = \frac{4 \left(1 - d^\Lambda(\mathbf{x}_i, \mathbf{w}_j^-)/2\sigma_3^2 \right) \alpha_j^- \sum_{k=1}^{r_1} \alpha_k^+ d^\Lambda(\mathbf{x}_i, \mathbf{w}_k^+)}{\left(\sum_{k=1}^{r_1} \alpha_k^+ d^\Lambda(\mathbf{x}_i, \mathbf{w}_k^+) + \sum_{k=1}^{r_2} \alpha_k^- d^\Lambda(\mathbf{x}_i, \mathbf{w}_k^-) \right)^2}$$

Learn the Hyperparameters

By minimizing the cost function, we automatically adapt the hyperparameters in the weighting functions:

$$\Delta\sigma_1 = -\epsilon \sum_{j=1}^{r_1} \mu_j^+ d^\wedge(\mathbf{x}_i, \mathbf{w}_j^+) \left(h(c(\mathbf{x}_i), c(\mathbf{w}_j^+)) \right)^2 / 2\sigma_1^4$$

$$\Delta\sigma_2 = \epsilon \sum_{j=1}^{r_2} \mu_j^+ \frac{d^\wedge(\mathbf{x}_i, \mathbf{w}_j^-)}{1 - d^\wedge(\mathbf{x}_i, \mathbf{w}_j^-)/2\sigma_3^2} \left(T - h(c(\mathbf{x}_i), c(\mathbf{w}_j^-)) \right)^2 / 2\sigma_2^4$$

$$\Delta\sigma_3 = \epsilon \sum_{j=1}^{r_2} \mu_j^+ \frac{d^\wedge(\mathbf{x}_i, \mathbf{w}_j^-)}{1 - d^\wedge(\mathbf{x}_i, \mathbf{w}_j^-)/2\sigma_3^2} d^\wedge(\mathbf{x}_i, \mathbf{w}_j^-) / 2\sigma_3^4$$

Outline

1 Introduction

2 Ordinal Generalized Matrix LVQ (OGMLVQ)

3 Proposed Accumulative OGMLVQ

4 Experimental Results

5 Summary

Experiments on Benchmark Datasets

Table: Brief description of the Ordinal Regression datasets

Dataset	Dimension	#Training	#Test
Pyrimidines	27	50	24
MachineCPU	6	150	59
Boston	13	300	206
Abalone	8	1000	3177
Bank	32	3000	5182
Computer	21	4000	4182
California	8	5000	15640
Census	8	6000	16784

Zero-one Error (MZE) of Benchmark Datasets

Table: Experimental results of mean zero-one error (MZE). Average MZE (\pm std) over 20 trials is given. The best performance is boldfaced. The smaller standard deviations of our cumulative OGMLVQ compared with OGMLVQ are marked by red.

Dataset	GMLVQ	OGMLVQ	A-OGMLVQ
Pyrimidines	$0.679 \pm (0.059)$	$0.645 \pm (0.106)$	$0.635 \pm (0.083)$
MachineCPU	$0.344 \pm (0.038)$	$0.415 \pm (0.096)$	$0.339 \pm (0.043)$
Boston	$0.468 \pm (0.029)$	$0.534 \pm (0.024)$	$0.443 \pm (0.031)$
Abalone	$0.538 \pm (0.005)$	$0.532 \pm (0.049)$	$0.499 \pm (0.010)$
Bank	$0.807 \pm (0.018)$	$0.750 \pm (0.008)$	$0.756 \pm (0.004)$
Computer	$0.593 \pm (0.031)$	$0.510 \pm (0.010)$	$0.488 \pm (0.004)$
California	$0.717 \pm (0.012)$	$0.680 \pm (0.007)$	$0.692 \pm (0.005)$
Census	$0.799 \pm (0.021)$	$0.735 \pm (0.014)$	$0.724 \pm (0.004)$

6/8 better performance, 7/8 smaller std, 5/8 better performance and smaller std.

Table: Experimental results of mean absolute error (MAE). Average MAE (\pm std) over 20 trials is given. The best performance is boldfaced. The smaller standard deviations of our cumulative OGMLVQ compared with OGMLVQ are marked by red.

Dataset	GMLVQ	OGMLVQ	A-OGMLVQ
Pyrimidines	$1.312 \pm (0.341)$	$0.985 \pm (0.169)$	$1.060 \pm (0.167)$
MachineCPU	$0.505 \pm (0.087)$	$0.630 \pm (0.176)$	$0.448 \pm (0.067)$
Boston	$0.623 \pm (0.039)$	$0.731 \pm (0.050)$	$0.579 \pm (0.053)$
Abalone	$0.775 \pm (0.011)$	$0.731 \pm (0.068)$	$0.599 \pm (0.013)$
Bank	$2.183(0.244)$	$1.462 \pm (0.009)$	$1.572 \pm (0.031)$
Computer	$0.994 \pm (0.154)$	$0.698 \pm (0.023)$	$0.648 \pm (0.007)$
California	$1.526 \pm (0.125)$	$1.208 \pm (0.018)$	$1.206 \pm (0.008)$
Census	$2.145 \pm (0.193)$	$1.582 \pm (0.018)$	$1.560 \pm (0.019)$

6/8 better performance, 5/8 smaller std, 4/8 better performance and smaller std.

Experiments on Real Datasets

Table: Description of the real datasets. d is number of previous values which function as inputs information

Dataset	training/test	d	# trials
FTSE100	1 year/1 month	5	36
Fish	9 folds/1 fold	5	10
wine	9 folds/1 fold	5	10
MonthlySOI	300/200	5	7
Birth	9 folds/1 fold	5	10

Experiments on Real Datasets

Table: Experimental results of MZE. Average MZE (\pm std) over folds listed in Table 4 is given. The best performance is boldfaced. The smaller standard deviations of our cumulative OGMLVQ compared with OGMLVQ are marked by red.

Dataset	GMLVQ	OGMLVQ	A-OGMLVQ
FTSE100	$0.624 \pm (0.092)$	$0.604 \pm (0.141)$	$0.604 \pm (0.113)$
Fish	$0.648 \pm (0.028)$	$0.554 \pm (0.046)$	$0.565 \pm (0.027)$
Wine	$0.689 \pm (0.108)$	$0.610 \pm (0.098)$	$0.541 \pm (0.100)$
MonthlySOI	$0.599 \pm (0.059)$	$0.534 \pm (0.047)$	$0.530 \pm (0.033)$
Birth	$0.625 \pm (0.088)$	$0.550 \pm (0.143)$	$0.600 \pm (0.056)$

3/5 better performance, 4/5 smaller std, 2/5 better performance and smaller std.

Experiments on Real Datasets

Table: Experimental results of MAE. Average MAE (\pm std) over folds listed in Table 4 is given. The best performance is boldfaced. The smaller standard deviations of our cumulative OGMLVQ compared with OGMLVQ are marked by red.

Dataset	GMLVQ	OGMLVQ	A-OGMLVQ
FTSE100	$0.814 \pm (0.227)$	$0.755 \pm (0.281)$	$0.740 \pm (0.228)$
Fish	$0.773 \pm (0.058)$	$0.607 \pm (0.061)$	$0.638 \pm (0.058)$
Wine	$1.015 \pm (0.218)$	$0.715 \pm (0.147)$	$0.703 \pm (0.204)$
MonthlySOI	$0.725 \pm (0.113)$	$0.586 \pm (0.054)$	$0.573 \pm (0.039)$
Birth	$0.850 \pm (0.105)$	$0.750 \pm (0.265)$	$0.825 \pm (0.112)$

3/5 better performance, 4/5 smaller std, 2/5 better performance and smaller std.

Outline

1 Introduction

2 Ordinal Generalized Matrix LVQ (OGMLVQ)

3 Proposed Accumulative OGMLVQ

4 Experimental Results

5 Summary

Summary

- Proposed new cost function for ordinal regression based on LVQ
- The cost function is more nature
- The updating rule is intuitive
- Automatically adapt the hyperparameters

Questions and Suggestions

Thank you very much for your attention!