Loss Functions for Preference Levels: Regression with Discrete Ordered Labels

Jason Rennie
Massachusetts Institute of Technology

Nati Srebro
University of Toronto

Supervised Learning Setting (Regression)

- Given a labeled training set:

objects, e.g. movies, options, etc, described as a feature vector

target labels

x_1 y_1

x_2 y_2

…

x_n y_n

- Learn a mapping

$$f(x) \mapsto y$$

in order to predict labels on future data:

x ?

Target Labels

- Common types of target labels:
  - Binary (positive/negative; ☹ ☺)
  - Multiclass (discrete, unordered categories)
  - Real valued

- Discrete ordinal labels

  ★ ★★ ★★★ ★★★★ ★★★★★

  ☹ 😐 ☺

  “undesirable”, “indifferent”, “preferred”

Background: Binary Regression

- Labeled training set $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn $z(x) = w'x+w_0$

such that $z(x)>0$ when $y=+1$,

and $z(x)<0$ when $y=-1$

minimizing $\sum_i loss(z(x_i);y_i)$

$$loss(z;y) = \begin{cases} 0 & yz>0 \\ 1 & \text{otherwise} \end{cases}$$

+1 / -1 labels

Focus on linear regression as an example. Same ideas apply to any other family of predictors

Background: Binary Regression

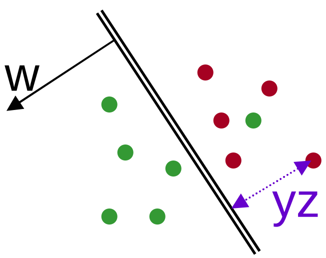
- Labeled training set $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

+1 / -1 labels

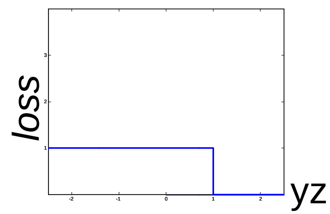
- Learn $z(x) = w'x + w_0$

  minimizing $\sum_i loss(z(x_i);y_i) + \lambda|w|^2$

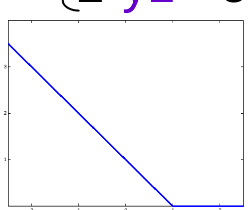
Focus on linear regression as an example. Same ideas apply to any other family of predictors



$$loss(z;y) = \begin{cases} 0 & yz>1 \\ 1 & \text{otherwise} \end{cases}$$

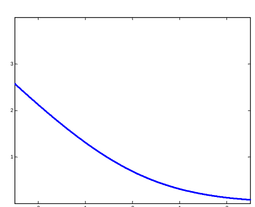


$$loss(z;y) = \begin{cases} 0 & yz>1 \\ 1-yz & \text{otherwise} \end{cases}$$



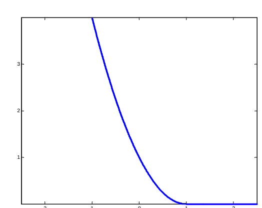
SVM

$$loss(z;y) = \log(1+e^{-yz})$$



logistic regression

$$loss(z;y) = (1-yz)^2_+$$



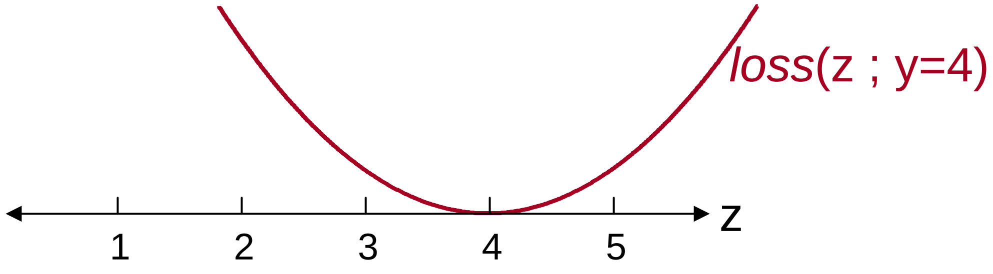
L_2 SVM

Discrete Ordinal Labels

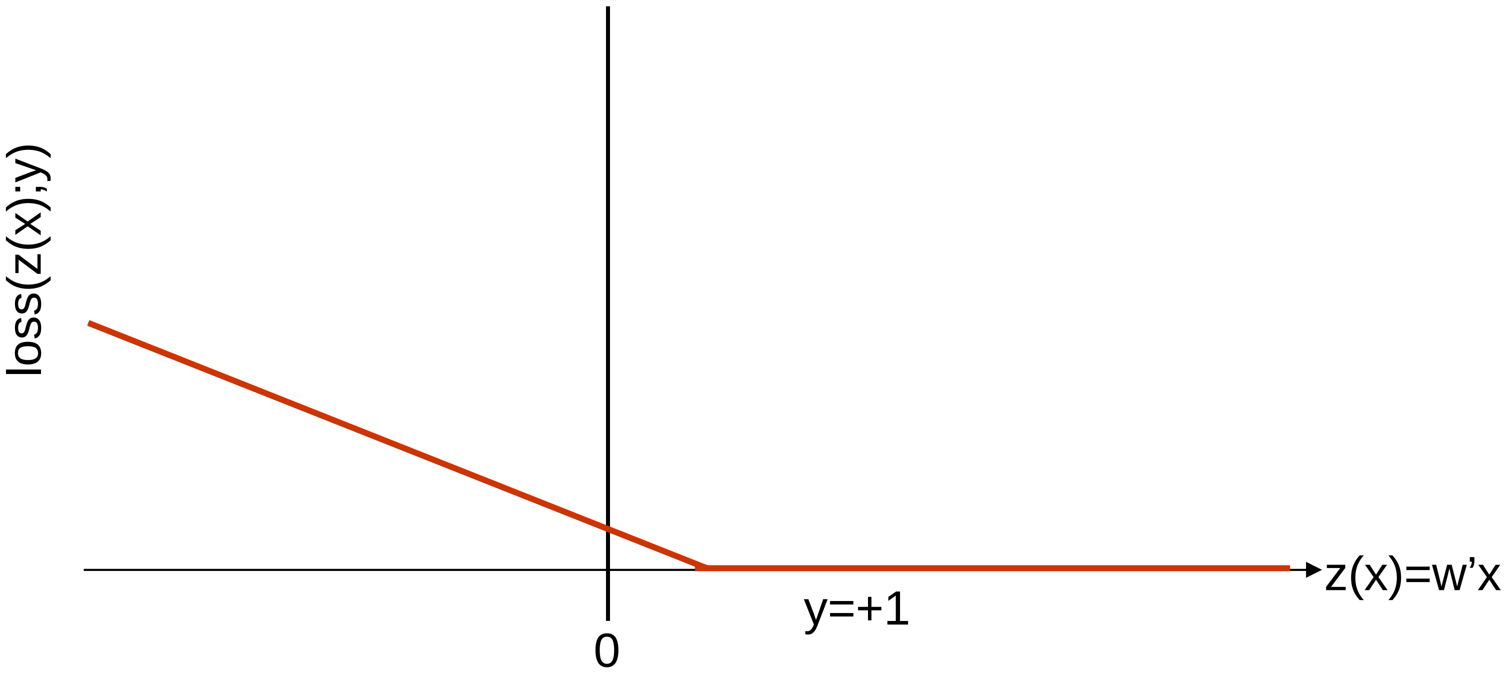
- Instead of y = -1 or +1,
  we have **y = 1, 2, 3, …, *k***

- Treat as *k* multiple unrelated classes, learn separate classifier for each value?

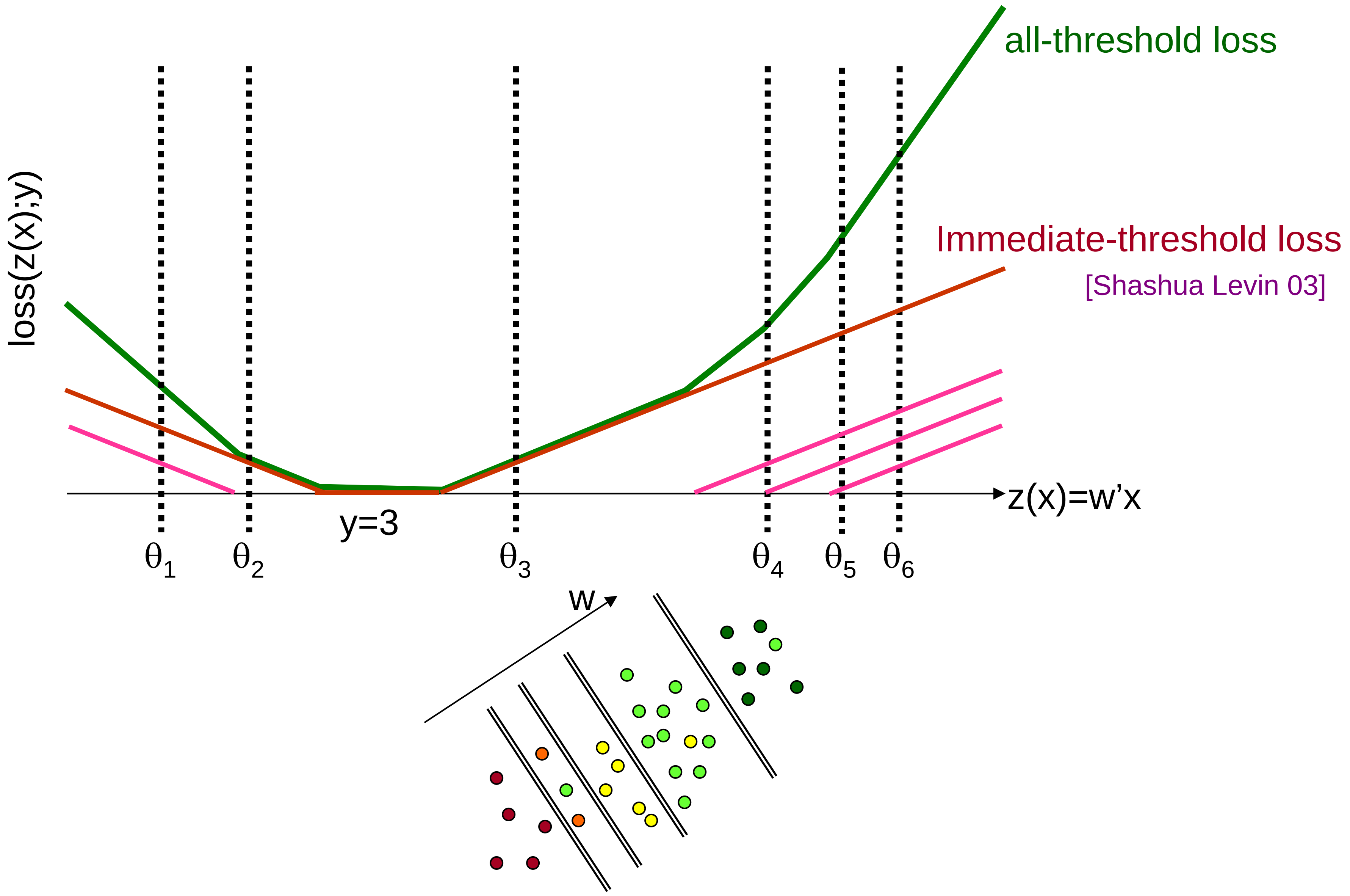
- Treat as a real valued objective, minimize, e.g. sum-squared error?



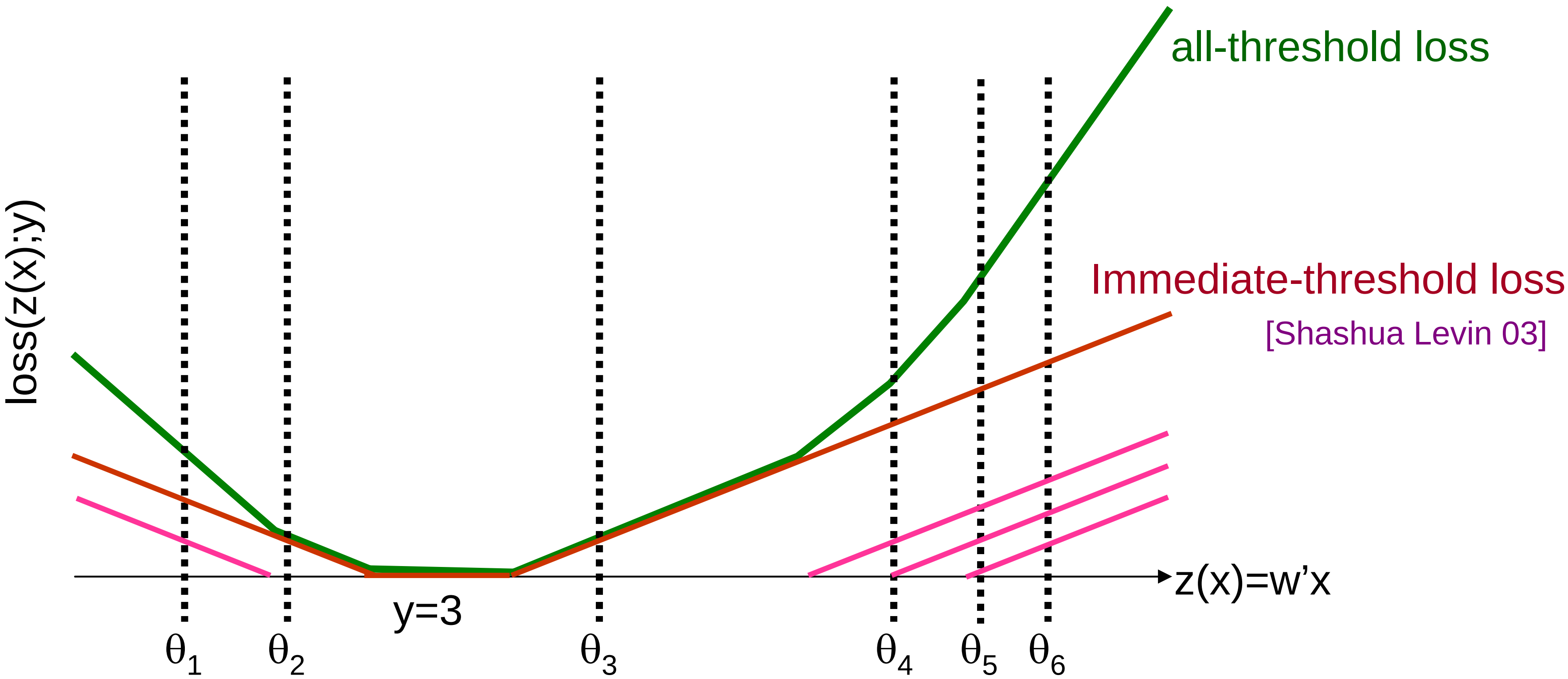
Threshold based approach



Threshold based approach



Threshold based approach

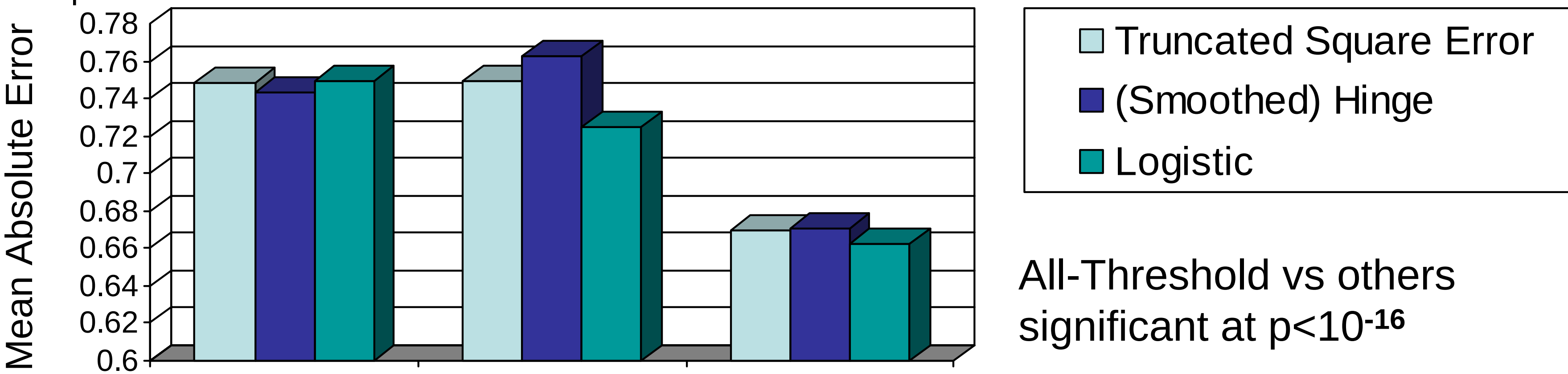


- All-threshold loss is a bound on the absolute rank-difference

- For both constructions:
  - can use any penalty function (e.g. logistic) instead of hinge
  - learn per-user θ's (different users use ratings differently)

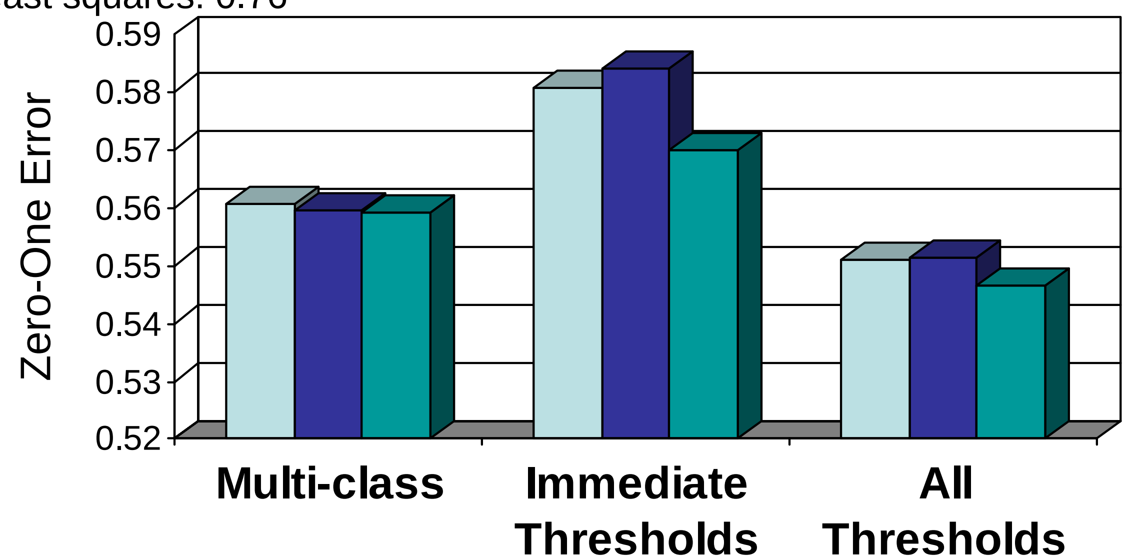
Results on MovieLens Data

Least squares: 1.33



All-Threshold vs others significant at $p<10^{-16}$

Least squares: 0.76



All-Threshold vs others barely significant at $p<0.14$

Beyond Linear Regression

- Same constructions can be used whenever a loss function is needed:
  - Kernel methods (SVMs)
  - Collaborative prediction (matrix completion)
    [Srebro Rennie Jaakkola NIPS'04]
    [Rennie Srebro ICML'05]

movies

users

	2		1			4				5	
	5		4				?		1		3
		3		5			2				
4			?			5		3		?	
		4		1	3				5		
			2				1	?			4
	1					5		5		4	
		2		?	5		?		4		
	3		3		1		5		2		1
	3				1			2		3	
	4			5	1			3			
		3				3	?			5	
2	?		1		1						
		5			2	?		4		4	
	1		3		1	5		4		5	
1		2			4				5	?	

Other Loss Functions

- Generalization to the logistic motivated by probabilistic generative model **(see paper)**
- Similar generative model with additive Gaussian “noise” **[Chu Ghahramani 2004]**

Alternative approach:

- Map ordinal labels to “<“ relationships
  [Herbrich *et al* 2000]
  – quadratic number of relationships

Summary

- Studied different constructions for loss-functions for discrete ordinal labels
- All-threshold construction best,
  much better then treating as multiclass or using squared error
- Can be used whenever a (scale sensative) loss function is needed