Pattern Recognition

Lecture 14. Linear Discriminant Functions: Gradient Descent and Perceptron Convergence

Dr. Shanshan ZHAO

shanshan.zhao@xjtlu.edu.cn

School of AI and Advanced Computing

Xi'an Jiaotong-Liverpool University

Academic Year 2021-2022

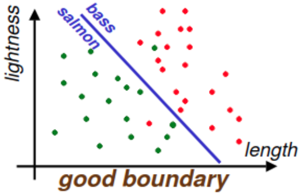
Table of Contents

Notations

- w : a scalar
- $\mathbf{w}$: a vector
- c : denotes the class

Recap

- Linear Discriminant function: $g(x) = w^T x + w_0$



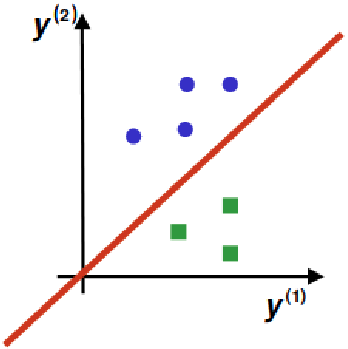
- need to estimate parameters w and w_0 from data
- Augment samples x get equivalent homogeneous problem in terms of samples y:

$$g(x) = \begin{bmatrix} w_0 & w^T \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix} = a^T y = g(y) \quad (1)$$

- Normalize by replacing all samples from class c_2 by their negative: $y_i \Rightarrow -y_i \quad \forall y_i \in c_2$

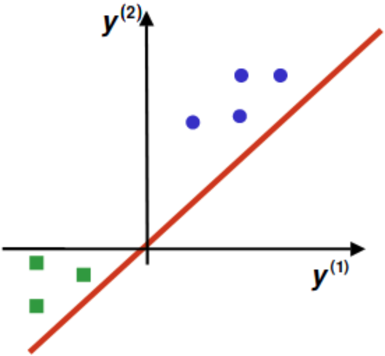
LDF: Problem "Normalization"

Before Normalization



Seek a hyperplane that separates patterns from different categories

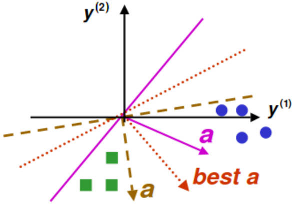
After Normalization

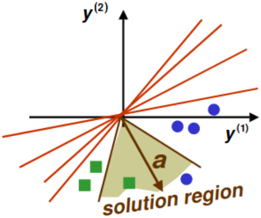


Seek a hyperplane that puts normalized patterns on the same side(should be positive)

LDF: Solution Region

- Find weight vector a, for all samples $y_1, ..., y_n$ (n is the number of samples):
  $a^T y_i = \sum_{j=0}^{d} a_j y_{ij} > 0$ (d is the dimension of vector y)





- In general, there are many such solutions a

Gradient Descent Procedures

Gradient Descent

For minimizing any function $J(x)$ set k = 1 and $x^{(1)}$ to some initial guess for the weight vector

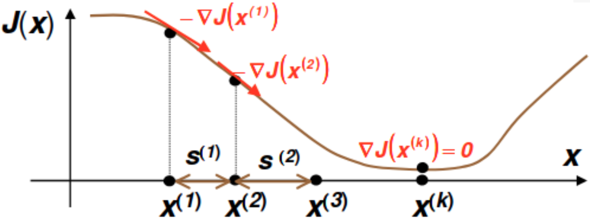
while $\eta^{(k)}|\nabla J(x^{(k)})| > \epsilon$ **do**

choose learning rate $\eta^{(k)}$

$x^{(k+1)} = x^{(k)} - \eta^{(k)}\nabla J(x^{(k)})$

$k = k + 1$

end



LDF: Criterion Function

- Find weight vector ***a*** , for all samples $y_1, ..., y_n$

$$a^T y_i = \sum_{j=0}^{d} a_j y_{ij} > 0 \tag{2}$$

- Need criterion function ***J(a)*** which is minimized when ***a*** is a solution vector
- Let Y_M be the set of samples misclassified by ***a***

$$Y_M(a) = \{\text{sample } y_i \quad s.t. \quad a^T y < 0\} \tag{3}$$

- First natural choice: number of misclassified samples

$$J(a) = |Y_M(a)| \tag{4}$$

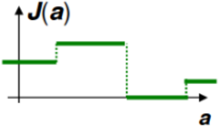


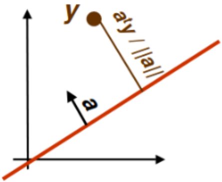
Figure: This is piecewise constant, gradient descent is useless

LDF: Perceptron Criterion Function

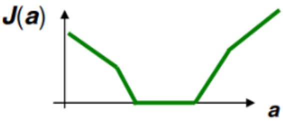
- Better choice: ***Perception*** criterion function

$$J_p(a) = \sum_{y \in Y_M} (-a^T y)$$

- If **y** is misclassified, $a^T y \leq 0$, so that $J_p(a) \geq 0$
- $J_p(a)$ is $||a||$ times the sum of distances of misclassified samples to decision boundary (figure a).
- $J_p(a)$ is piecewise linear and thus suitable for gradient descent (figure b).



(a)



(b)

LDF: Perceptron Batch Rule

- Perception criterion function

$$J_p(a) = \sum_{y \in Y_M} (-a^T y)$$

- Gradient of $J_p(a)$ is $\nabla J_p(a) = \sum_{y \in Y_M} (-y)$
  - Y_M are samples misclassified by $a^{(k)}$
  - It is not possible to solve $\nabla J_p(a) = 0$ analytically because of Y_M
- Update rule for gradient descent : $x^{(k+1)} = x^k - \eta^{(k)} \nabla J(x)$
- Thus gradient decent batch update rule for $J_p(a)$ is

$$a^{(k+1)} = a^{(k)} + \eta^{(k)} \sum_{y \in Y_M} y \tag{5}$$

- It is called batch rule here because it is based on all misclassified samples

LDF: Perceptron Single Sample Rule

- In comparison, gradient decent single sample rule for $J_p(a)$ is :

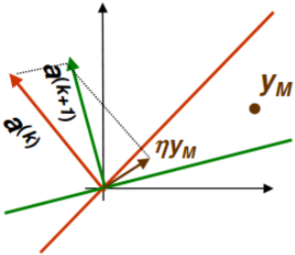
$$a^{(k+1)} = a^{(k)} + \eta^{(k)} y_M \tag{6}$$

  - note that y_M is one sample misclassified by $a^{(k)}$
  - must have a consistent way of visiting samples

- Geometric Interpretation
  - y_M misclassified by $a^{(k)}$

$$(a^{(k)})^T y_M \leq 0$$

  - y_M is on the wrong side of decision hyperplane
  - adding ηy_M to $\boldsymbol{a}$ moves new decision hyperplane in the right direction with respect to y_M



LDF: Perceptron Single Sample Rule

$$a^{(k+1)} = a^{(k)} + \eta^{(k)} y_M$$

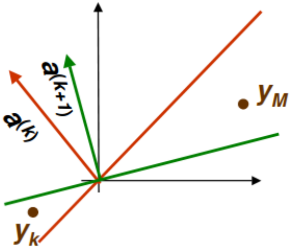


Figure: η is too large, previously correctly classified sample y_k is now misclassified

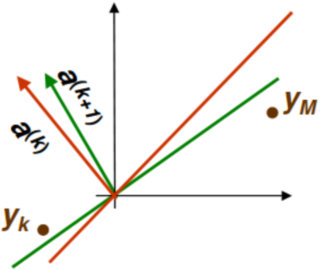


Figure: η is too small, y_M is still misclassified

LDF: Perceptron Example

	features				grade
name	**good attendance?**	**tall?**	**sleeps in class?**	**chews gum?**	
Jane	yes(1)	yes(1)	no(-1)	no(-1)	A
Steve	yes(1)	yes(1)	yes(1)	yes(1)	F
Mary	no(-1)	no(-1)	no(-1)	yes(1)	F
Peter	yes(1)	no(-1)	no(-1)	yes(1)	A

- **class 1**: students who get grade A
- **class 2**: students who get grade F

LDF: Perceptron Example

	features					grade
name	extra	**good attendance?**	**tall?**	**sleeps in class?**	**chews gum?**	
Jane	1	yes(1)	yes(1)	no(-1)	no(-1)	A
Steve	1	yes(1)	yes(1)	yes(1)	yes(1)	F
Mary	1	no(-1)	no(-1)	no(-1)	yes(1)	F
Peter	1	yes(1)	no(-1)	no(-1)	yes(1)	A

- Convert samples $x_1, .., x_n$ to augmented samples $y_1, ..., y_n$ by adding a new dimension of value 1.

LDF: Perceptron Example

	features					grade
name	extra	**good attendance?**	**tall?**	**sleeps in class?**	**chews gum?**	
Jane	1	yes(1)	yes(1)	no(-1)	no(-1)	A
Steve	-1	yes(-1)	yes(-1)	yes(-1)	yes(-1)	F
Mary	-1	no(1)	no(1)	no(1)	yes(-1)	F
Peter	1	yes(1)	no(-1)	no(-1)	yes(1)	A

- Replace all samples from class c_2 by their negative
  $y_i \Rightarrow -y_i \quad y_i \in c_2$
- Seek weight vector $\boldsymbol{a}$ s.t. $a^T y_i > 0 \quad \forall y_i$

LDF: Perceptron Example

	features					grade
name	extra	**good attendance?**	**tall?**	**sleeps in class?**	**chews gum?**	
Jane	1	yes(1)	yes(1)	no(-1)	no(-1)	A
Steve	-1	yes(-1)	yes(-1)	yes(-1)	yes(-1)	F
Mary	-1	no(1)	no(1)	no(1)	yes(-1)	F
Peter	1	yes(1)	no(-1)	no(-1)	yes(1)	A

- Sample is misclassified if $a^T y_i = \sum_{j=0}^{4} a_j y_{ij} < 0$

  (i is the numbering of the sample, j is the numbering of the dimension)
- Gradient descent single sample rule : $a^{(k+1)} = a^{(k)} + \eta^{(k)} \sum_{y \in Y_M} y$
- Here we set a fixed learning rate to $\eta^{(k)} = 1$

$$a^{(k+1)} = a^{(k)} + y_M$$

LDF: Perceptron Example

- set initial weights $a^{(1)} = [0.25, 0.25, 0.25, 0.25, 0.25]$
- visit all samples sequentially, modifying the weights for after finding a misclassified example

name	$a^T y$	misclassified?
Jane	0.25*1+0.25*1+0.25*1+0.25*(-1)+0.25*(-1) >0	no
Steve	0.25*(-1)+0.25*(-1)+0.25*(-1)+0.25*(-1)+0.25*(-1)<0	yes

- new weights

$$\begin{aligned} a^{(2)} = a^{(1)} + y_M &= [0.25 \quad 0.25 \quad 0.25 \quad 0.25 \quad 0.25] + \\ &+ [-1 \quad -1 \quad -1 \quad -1 \quad -1] \\ &= [-0.75 \quad -0.75 \quad -0.75 \quad -0.75 \quad -0.75] \end{aligned}$$

LDF: Perceptron Example

$$a^{(2)} = [-0.75 \quad -0.75 \quad -0.75 \quad -0.75 \quad -0.75]$$

name	$a^T y$	misclassified?
Mary	-0.75*(-1)-0.75*1-0.75*1-0.75*1-0.75*(-1) <0	yes

- new weights

$$\begin{aligned} a^{(3)} = a^{(2)} + y_M &= [-0.75 \quad -0.75 \quad -0.75 \quad -0.75 \quad -0.75] + \\ &+ [-1 \quad 1 \quad 1 \quad 1 \quad -1] \\ &= [-1.75 \quad 0.25 \quad 0.25 \quad 0.25 \quad -1.75] \end{aligned}$$

LDF: Perceptron Example

$$a^{(3)} = [-1.75 \quad 0.25 \quad 0.25 \quad 0.25 \quad -1.75]$$

name	$a^T y$	misclassified?
Peter	-1.75*1+0.25*1+0.25*(-1)+0.25*(-1)-0.75*1 <0	yes

- new weights

$$\begin{aligned} a^{(4)} = a^{(3)} + y_M &= [-1.75 \quad 0.25 \quad 0.25 \quad 0.25 \quad -1.75] + \\ &+ [1 \quad 1 \quad -1 \quad -1 \quad 1] \\ &= [-0.75 \quad 1.25 \quad -0.75 \quad -0.75 \quad -0.75] \end{aligned}$$

LDF: Perceptron Example

$$a^{(4)} = [-0.75 \quad 1.25 \quad -0.75 \quad -0.75 \quad -0.75]$$

name	$a^T y$	wrong?
Jane	-0.75*1+1.25*1-0.75*1-0.75*(-1)-0.75*(-1) >0	no
Steve	-0.75*(-1)+1.25*(-1)-0.75*(-1)-0.75*(-1)-0.75*(-1) >0	no
Mary	-0.75*(-1)+1.25*1-0.75*1-0.75*1-0.75*(-1) >0	no
Peter	-0.75*1+1.25*1-0.75*(-1)-0.75*(-1)-0.75*1 >0	no

- The discriminant function is

$$g(y) = -0.75 * y^{(0)} + 1.25 * y^{(1)} - 0.75 * y^{(2)} - 0.75 * y^{(3)} - 0.75 * y^{(4)}$$

- Converting back to the original features x:

$$g(x) = -0.75 + 1.25 * x^{(1)} - 0.75 * x^{(2)} - 0.75 * x^{(3)} - 0.75 * x^{(4)}$$

LDF: Perceptron Example

- Converting back to the original features x:

$$1.25 * x^{(1)} - 0.75 * x^{(2)} - 0.75 * x^{(3)} - 0.75 * x^{(4)} > 0.75 \Rightarrow gradeA$$
$$1.25 * x^{(1)} - 0.75 * x^{(2)} - 0.75 * x^{(3)} - 0.75 * x^{(4)} < 0.75 \Rightarrow gradeF$$

- This is just one possible solution vector
- If we started with weight $a^{(1)} = [0, 0.5, 0.5, 0, 0]$, the solution would be different: $[-1, 1.5, -0.5, -1, -1]$

week3 day4 exercise solution

3.3 Print the shape of the dataset

[12]:
```
# TODO
```

[13]:
```
print(hsi_image.shape)
```

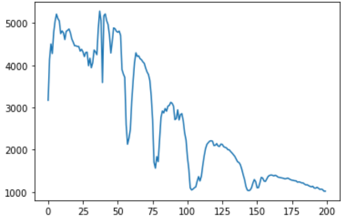
```
(145, 145, 200)
```

3.4 Plot the feature of one pixel

Is there any difference between pixels that belong to different class

[14]:
```
# get any position
feature = hsi_image[0,0,:]
plt.plot(feature)
```

[14]:
```
[<matplotlib.lines.Line2D at 0x7f72314f6910>]
```

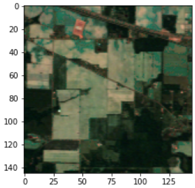


week3 day4 exercise solution

```
[17]: # get the three bands
r = hsi_image[:,:,31]
g = hsi_image[:,:, 170]
b = hsi_image[:,:,190]
# normalization
r = (r-np.min(r))/(np.max(r)-np.min(r))  if np.max(r)>0  else r
g = (g-np.min(g))/(np.max(g)-np.min(g))  if np.max(r)>0  else g
b = (b-np.min(b))/(np.max(b)-np.min(b))  if np.max(r)>0  else b
# merge three bands image
composite_color = np.stack([r,g,b], axis=2)
```

```
[18]: plt.imshow(composite_color)
```

```
[18]: <matplotlib.image.AxesImage at 0x7f640b68bf40>
```



week3 day4 exercise solution

3.6 Display the groundtruth

[18]:
```python
# print how many distinctive values in groundtruth
print(np.unique(y))
```

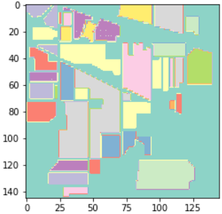
```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16]
```

[19]:
```python
y.shape
```

[19]:
```
(145, 145)
```

•[20]:
```python
plt.imshow(y, cmap='Set3') # you can change colormap : cmap
```

[20]:
```
<matplotlib.image.AxesImage at 0x7f72303ba3d0>
```



week3 day4 exercise solution

Task 4: Apply KDE codes to this dataset

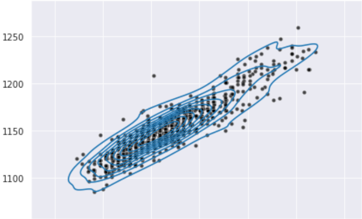
```
[21]: sns.set_style('darkgrid')
```

```
[104]: # since the label could be 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
       # you can choose any from them
       class_index = 6
```

```
[105]: # you can choose two features/channels/bands
       c1_feature1 = hsi_image[:,:,31][np.where(y==class_index)]
       c1_feature2 = hsi_image[:,:,170][np.where(y==class_index)]
```

```
[106]: sns.kdeplot(c1_feature1, c1_feature2)
       sns.scatterplot(c1_feature1, c1_feature2, color='black', alpha=0.7, s=20)
       plt.tight_layout();
```

```
/home/shanshan/anaconda3/envs/DTS201TC/lib/python3.9/site-packages/seaborn/_
he only valid positional argument will be `data`, and passing other argument
  warnings.warn(
/home/shanshan/anaconda3/envs/DTS201TC/lib/python3.9/site-packages/seaborn/_
2, the only valid positional argument will be `data`, and passing other argu
  warnings.warn(
```



Reference I

Lecture contents of week4 day1 and day2 borrow heavily from Prof. Olga Veksler's slide
https://www.csd.uwo.ca/~oveksler/Courses/CS434a_541a/Lecture9.pdf

Thank You !

$\mathcal{Q}$ & $\mathcal{A}$