LaTeXTemplate for English Report

Your name

April 17, 2024

Contents

1 LaTeX **1**
1.1 Text . . . 1
1.2 Figure . . . 2
1.2.1 One Figure . . . 2
1.2.2 Subfigure . . . 2
1.3 Table . . . 2
1.4 Pseudo-code . . . 3
1.5 Highlight . . . 4
1.6 Multiple Columns . . . 5
1.7 Math . . . 5
1.8 Ref . . . 5

Chapter 1

LaTeX

1.1 Text

1. normal
2. **bold**
3. *italic*
4. ~~delete line~~
5. *~~italic delete line~~*
6. wave
7. underline
8. underline
9. double underline
10. dash underline
11. dot underline
12. highlight

1.2 Figure

1.2.1 One Figure

Figure 1.1: Logo of SCUT

1.2.2 Subfigure

(a) subfig 1

(b) subfig 2

(c) subfig 3

(d) subfig 4

Figure 1.2: subfigures

1.3 Table

Table 1.1: Paramter Value

Parameter	Value
α	1
β	1

Table 1.2: Paramter Value

Module	Parameter	Value
contrastive model	number of RBF centers, $k_{\text{rbf\_c}}$	$\sqrt{n}$
	number of hidden neurons, k_{hidden}	$\frac{\sqrt{n}}{2}$
	dropout rate	0.3
regression model	repetition rate of offline data	10%
	number of centers of one RBFN, $k_{\text{rbf\_r}}$	$\sqrt{\frac{1.1n}{3}}$
topological sorting	threshold thr	$0.3 * nv_{\text{remain}}$
GA	distribution index η_c in SBX	15
	probability of crossover	100%
	distribution index η_m in PM	15
	probability of mutation	$\frac{1}{d}$

You cant take a screenshot, and throw the picture into the table environment, such as the table aboved.

1.4 Pseudo-code

Algorithm 1 KahnAlgorithm

Input: Graph $G(\mathbb{V}, \mathbb{E})$
Output: Sequence L

```
L ← an empty sequence
Q ← the vertices whose indegree is zero
while Q is not empty do
    u ← remove the top node of Q
    add u to L
    for each node v with an edge e from u to v do
        remove edge e from graph G
        if indegree of v is 0 then
            push v to Q
        end if
    end for
end while
return L
```

Algorithm 2 Framework

Input: Training data $\mathbb{D}$, Maximum generation $g_{\max}$, Population size n
Output: The best solution

Creating paired dataset $\mathbb{D}_{\text{cl}}$
Training contrastive model M_{con} from $\mathbb{D}_{\text{cl}}$
$i \leftarrow 0$
$P \leftarrow$ Latin hypercube sampling.
while $i < g_{\max}$ **do**
  $C \leftarrow$ apply SBX and PM on P
  $P \leftarrow P \cup C$
  $M_{\text{reg}} \leftarrow \text{BuildRegressionModel}(P, \mathbb{D})$
  $L \leftarrow \text{TopologicalSort}(P, M_{\text{con}}, M_{\text{reg}}, n)$
  $P \leftarrow P[L]$
  $i \leftarrow i + 1$
end while
return $P[0]$

You take a screenshot, and throw the picture into the algorithm environment, such as the algorithm aboved.

1.5 Highlight

```
#include <algorithm>
using namespace std;
void quickSort(int arr[],
               int begin,
               int end) {
  int i, j, t, pivot;
  if (begin > end)
    return;

  pivot = arr[begin];
  i = begin;
  j = end;
  while (i != j) {
    while (arr[j] >= pivot && i < j)
      j--;
    while (arr[i] <= pivot && i < j)
      i++;
    if (i < j)
      swap(arr[i], arr[j]);
  }

  arr[begin] = arr[i];
  arr[i] = pivot;
```

```
24    quickSort(arr, begin, i - 1);
25    quickSort(arr, i + 1, end);
26  }
```

1.6 Multiple Columns

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

1.7 Math

Interline Formula:

$$a_n = a_{n-1} + 1 \tag{1.1}$$

Inline Formula: This is a simple arithmetic progression formula $a_n = a_{n-1} + 1$.

1.8 Ref

- figure: Figure 1.1
- subfigure: Figure 1.2a
- table: Table 1.1

- pseudo-code: Algorithm 1
- equation: Equation 1.1
- chapter: chapter 1
- paper: [1]
- url 1: baidu
- url 2: `https://baidu.com`

Reference

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.