KIT 405 Tutorial 10

Learning Objective

Learn how to make an item-based collaborative learning using pre-defined sample list in PHP.

Tutorial Tasks

Tasks 1: Create sample list

1. Download sample_list in the MyLo

It is an utility matrix with book .

```
<?php

$books =  array(

    "Brigette" => array("the god delusion" => 3.5, "the shack" => 4,"the
birds in my life" => 2.5,"new moon" => 3.5),

    "Sonia" => array("the last lecture" => 2.5, "the god delusion" => 3.5,
"the shack" => 3.5, "the birds in my life" => 2.5, "new moon" => 1),

    "Yutao" => array("the shack" => 5, "new moon" => 3.5),
    "Navdeep" => array("the last lecture" => 2.5),

    "Jiuling" => array("the birds in my life" => 5, "new moon" => 3.5),

    "Liang" => array("the last lecture" => 3, "the god delusion" => 1.5,"the
shack" => 3.5),
  );
?>
```

2. Follow the above example to make another utility matrix regarding movie rating with same users and save it into above “sample_list.php” file.

Tasks 2: Item-based collaborative recommendation

1. Create main recommendation functions, download “recommend. php” from MyLo.

```
<?php

/**
 * PHP item based filtering
```

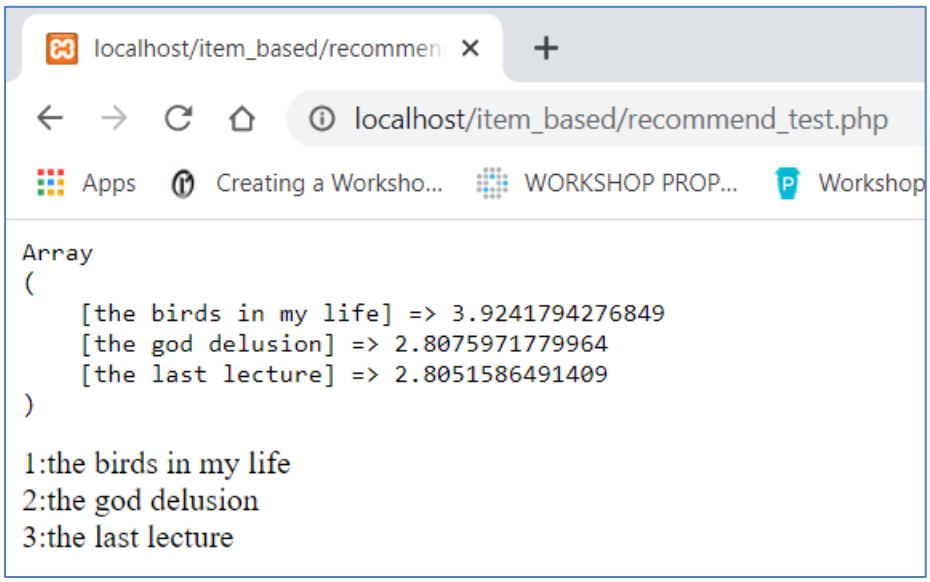
```
 * preferences including book and movie in this demo
 * @package   PHP item based filtering
 */

class Recommend {



    public function similarityDistance($preferences, $person1, $person2)
    {
        $similar = array();
        $sum = 0;

        foreach($preferences[$person1] as $key=>$value)
        {
            if(array_key_exists($key, $preferences[$person2]))
                $similar[$key] = 1;
        }

        if(count($similar) == 0)
            return 0;

        foreach($preferences[$person1] as $key=>$value)
        {
            if(array_key_exists($key, $preferences[$person2]))
                $sum = $sum + pow($value - $preferences[$person2][$key], 2);
        }

        return  1/(1 + sqrt($sum));
    }


    public function matchItems($preferences, $person)
    {
        $score = array();

            foreach($preferences as $otherPerson=>$values)
            {
                if($otherPerson !== $person)
                {
                    $sim = $this->similarityDistance($preferences, $person,
$otherPerson);

                    if($sim > 0)
                        $score[$otherPerson] = $sim;
                }
            }

        array_multisort($score, SORT_DESC);
        return $score;

    }


    public function transformPreferences($preferences)
    {
        $result = array();

        foreach($preferences as $otherPerson => $values)
        {
            foreach($values as $key => $value)
            {
                $result[$key][$otherPerson] = $value;
            }
        }

        return $result;
    }
```

```
    public function getRecommendations($preferences, $person)
    {
        $total = array();
        $simSums = array();
        $ranks = array();
        $sim = 0;

        foreach($preferences as $otherPerson=>$values)
        {
            if($otherPerson != $person)
            {
                $sim = $this->similarityDistance($preferences, $person,
$otherPerson);
            }

            if($sim > 0)
            {
                foreach($preferences[$otherPerson] as $key=>$value)
                {
                    if(!array_key_exists($key, $preferences[$person]))
                    {
                        if(!array_key_exists($key, $total)) {
                            $total[$key] = 0;
                        }
                        $total[$key] += $preferences[$otherPerson][$key] *
$sim;

                        if(!array_key_exists($key, $simSums)) {
                            $simSums[$key] = 0;
                        }
                        $simSums[$key] += $sim;
                    }
                }

            }
        }

        foreach($total as $key=>$value)
        {
            $ranks[$key] = $value / $simSums[$key];
        }

    array_multisort($ranks, SORT_DESC);
    return $ranks;

    }

}

?>
```

2. Test recommendation results with the preference (book), create a test php file as below, and show the top three ranking results of book for specific user “Yutao”.

```
<?php
require_once("recommend.php");
require_once("sample_list.php");


$re = new Recommend();

$ratinglist=$re->getRecommendations($books, "Yutao");
```

```
arsort($ratinglist);

echo "<pre>";
print_r($ratinglist);
echo "</pre>";

$i=1;
foreach($ratinglist as $key=>$value)
{
    if($i<4)
    {
       echo $i.":".$key."<br/>";
    }
       $i++;
}

?>
```



Submit the screenshot of book recommendation results!

3. Test the recommendation results with preference (movie) and show the top three ranking results.

Submit the screenshot of movie recommendation results!

Tutorial submission

Submit a zip file that contains all screenshots of results.
Due Date: 11:55PM Wednesday 12 May 2021
Submission: Use MyLo Dropbox (Tutorial_week 10)