

# Visualizing Matchup Data

## Problem

Matchup data set is, no doubt, an adjacency matrix. However, most of the matchup data visualization websites focus only on a single dimension (a row or a column) when presenting the matchup data. This makes comparison between players a relatively hard task, and does not provide a good insight on the entire dataset.

## Motivation

The goal is to automatically generate a clear visualization that makes the following tasks easy to perform.

1. Comparison between players
2. Player classification

The data set used here is the matchup data between 111 dota2 heroes.

## Approach

The visual encoding is red for comparative advantage and green for comparative disadvantage. Higher advantage/disadvantage values correspond to less transparent dots.

Finding an optimal visualization for a matrix is in general an NP-complete problem. Clearly the width of the matrix (111) does not allow any brute force approach.

Therefore, the approach to be introduced relies on the following observation: players/heroes of the same type tend to have similar matchup result. This means that similar players/heroes tend to produce similar row (column) vectors.

Therefore, when grouping similar players/heroes together, the visualization should produce a grid pattern, with noises in diagonal grids. Thus, a good visualization should emphasize this grid pattern.

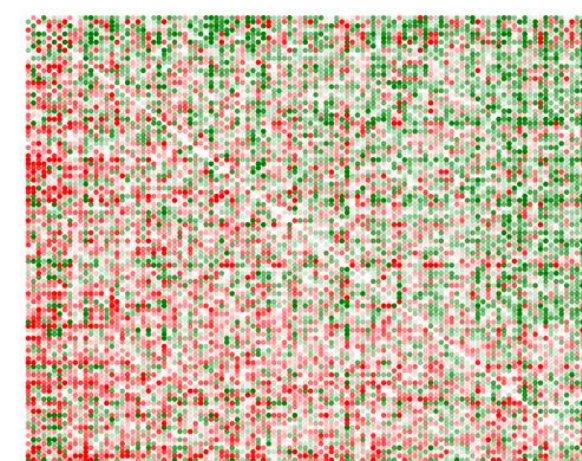
While grouping players/heroes into classes emphasizes similarity and differences, it also reduces the computational complexity because we only need to handle interclass and intraclass variances. In this approach, the heroes are separated into 10 groups.

From Cong Qiaoben, CS448B, Spring 2016

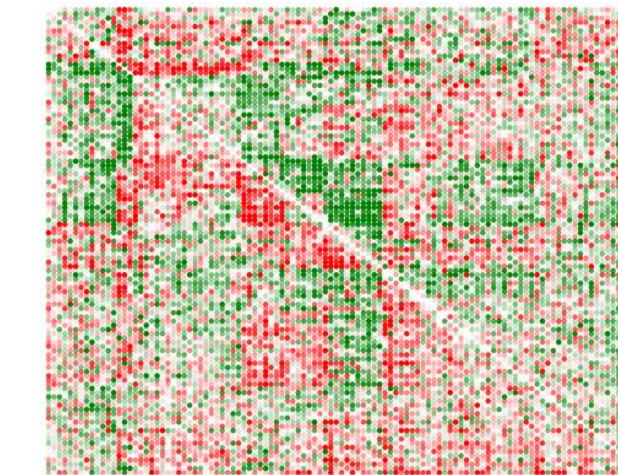
The group order is selected to visually emphasize the interclass difference. The heuristic used here is the sum of the distances between the averages of adjacent grids, each weighted by the inverse of the distance to the matrix diagonal. Under this heuristic, this matrix will produce a clear grid pattern with more visual weight on the diagonal.

The intergroup order is selected to have red on the bottom left and green on the top right. This order is better than a random order because it shows clear edges between adjacent grids. This order is achieved by sorting the rows and columns.

## Result



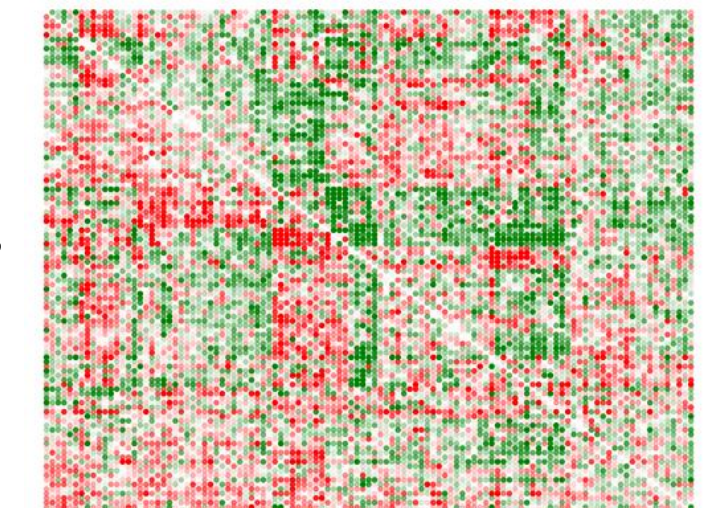
This image shows the ordered matrix. Although there is a clear pattern (more green dots on the upper right), it does not offer much insight into this dataset.



This image shows the matrix that maximizes the weights close to diagonal. Therefore there is a clear large rectangle near the middle. However,

it is relatively hard to see the grids clearly, because similarly colored grids are sometimes adjacent and tend to form a huge grid. Moreover, green dots and red dots are quite uniformly distributed in every single grid.

This image shows the final result. Grid pattern is clearer and green dots are quite separated from the red dots.



## Future Work

One interesting direction would be allowing interaction, because while this visualization shows patterns, it is hard to map the patterns back to specific players/heroes. This also means the algorithm above should be optimized to allow real-time interaction, because generating this matrix currently takes ~ 30 seconds. Most of the computation happens in arranging the order of the groups to form grid-like pattern with emphasis on the diagonal grids.