# The Basic Late Game Item Recommendation

## Baseline Hero-item Preference

Baseline hero-item preference defined as follows is the general preference of a certain hero to an item.

Where is the preference of hero to item . Suppose is the count of victories, as the count of defeats of matches, where hero is equipped with item by the end of match. With the introduction of two coefficients , an intuitive definition of the baseline preference is:

The value of and can be determined manually and some tuning process is expected in the following steps.

The baseline hero-item preference matrix could only reflect a general and high-level item effectiveness per hero. Obviously, it could be expected to recommend general items well simply based on hero personality, however, it does not discriminate between any matches and therefore none of the game dynamics are taken into consideration. This makes our recommendation no where better that the static recommendation, despite that the process can thus be automated. To distinct our method from static recommendations, we now take one of the most important game dynamics into consideration, which is the team draft.

## Team Draft based Item Preference Variation

The initial design and following updates of Dota2 have always maintained a subtle balance between heroes and items, so that no hero and item is perfect. For example, a nuker hero might be able to cast tremendous damages to enemy heroes in a short time, but they are mostly lack of health and armor, therefore they are vulnerable when targeted by carry heroes or disablers. Team draft can be seen as the first “fight” between the Radiant and Dire, in which each tries to utilize the game balance system to come up with a competitive 5-hero set that might most possibly lead to victory. As the draft is fixed, most players would have a basic idea on item purchases in this game. Inspired by this, we propose two voting matrices that captures this kind of relationship based on the balance system and item/hero neutralization effects. Mathematically, for each hero we define to capture the votes of hero to ally item where each element of is the probability of victory when item exists in ally heroes in all games that hero has played:

Similarly, we also define the Enemy Item Voting matrix where:

The point of having the probability minus 0.5 is to enable vote-against relationship between heroes and items. The final range of an element in both and is within . This is intuitive and can be analogized:

* **Friend’s friend is my friend:** if for some hero the existence of an item in ally is statistically associated with a higher victory rate, the hero would vote in favor of that item, vice versa.
* **Enemy’s enemy is my friend:** On the other hand, if for an enemy hero, the existence of an item in their enemy statistically results in a higher/lower winning rate for them (lower/higher winning rate for their enemy, which is us in this case), the hero would vote against/endorse it so that our chance of winning increases.

Apparently, the two voting matrices can be calculated based on historical match data (ideally in similar versions of the game) and stored persistently. As soon as the team draft is finished, we can apply the hero-item voting on the baseline preference to get a more accurate hero-item preference. We refer the hero-item voting as TD-based IPV (Team Draft-based Item Preference Variation), the new preference matrix as Matchwise Baseline Item Preference (MBIP). So far all the matrices are learnt purely through statistics.

Optimize the combination of baseline and TD-based IPV, for different heroes they have different roles in the team, therefore the weights of votes from different heroes should be different also. ML techniques might be applied here to learn the weight. (新坑)

## Evaluation

### Problem Formulation

Before evaluating, we first formulate our problem for better understanding. Without loss of generality, we suppose a certain version of Dota2 can be regarded as a function , the outcome of such a function is determinate, either 1(victory) or 0(defeat), the function will take a matrix as input, which is the game dynamics matrix:

Furthermore, the game dynamics matrix is composed of different factors (we also refer them as individual game dynamics) in a game as follows:yuuy7

Some examples of the individual game dynamics include item choice, team draft, etc.

Therefore, the eventual goal of an optimization approach is to achieve a determinate 1 as outcome. However, due to the high dimensionality and complex dynamics of Dota2, a determinate 1 cannot be achieved (need NP-hardness proof here) by optimizing only a portion of the game dynamics while it’s also impossible to control all game dynamics in real human-involved matches. Therefore, the goal can be tailored for a realistic system as follows:

Given game dynamics matrices , which can be rewritten in a new matrix:

Where columns are free while other columns are fixed.

Maximize:

Intuitively, the maximization target is mathematically equivalent to the winning rate of the team applied with the recommended dynamics in Dota2.

### Methodology

For baseline recommender, we don’t need to evaluate the performance as it’s based on pure statistical analysis based on all match data. Since it doesn’t consider game dynamics, it’s meaningless to consider the effectiveness of it with varying dynamics in different matches. However, we have to consider the accuracy of MBIP to show the effectiveness of our approach. The excessive factors that influence a Dota2 match make it challenging for us in the evaluation process. Besides the choice of items and hero draft, several other major factors that influences the outcome of a match include:

* **Player dynamics:** player dynamics include a series player-specific features like the level, ability (farm, push, gank in this case), special skills, etc. We argue that special skills and other factors are rare and can be percepted as noises in data, while the level and ability consistence between two teams can be maintained by applying only ladder or professional games.
* **Strategies:** the same item might have different affects when different strategies are applied. However, the good thing is that most of the time major strategies are applied among different games, while several corner cases can be also ignored if given a massive amount of data.
* **Other dynamics:** to reduce the problem complexity, we argue that strategy, player, team draft and item choice are the most influential factors in a game, other dynamics might also exist but are rare and of little significance in our recommendation system.

Assuming the influence of both player and strategy dynamics mentioned above is alleviated through applying appropriate match data. As the MBIP approach recommends items after team draft, the only variation of a game converges to items. Therefore, the effectiveness of our system can be evaluated by checking the winning rate

The ideal evaluation of an item recommendation system. Without loss of generality, we define the