Loss function

- Hemming loss:
  $\frac{1}{L}\sum_{i\in[1,2,\ldots L]}(\hat{y}_i \text{ xor } y_i)$
- exact loss
  0 if $\hat{y} = y$ else 1
- trade off
  $\sum_{i\in\hat{Y}}[1 - a - y_i]$

Algorithm

We can consider each "superarm" of the label powerset as a seperate class and then from there we can apply multiclass method.Let k be the cardinality of the labelset

for i = 1,2,....T:

1)input x

2)calculate $haty = argmax(W^T x)$

3)calculate $p_r = (1 - \gamma)|r = \hat{y}| + \frac{\gamma}{k}$

4)predict $\tilde{y}$ according to the distribution p and recieve $\tilde{y} \cap Y$

5)update w according to $w = w - \bigtriangledown L$ where L is the loss

Random k dataset for multilabel classification

Let L be the set of labels of size M , training set be D and labelset size be k. Define $m = [M/k]$.
Randomly Partition L into m disjoint labelsets(R_j) of size k. Next make training sets $D_j mi \in 1, 2, ...m$ such that for every example x_i in D , their annotation will be $Y_i \cap R_j$.
After this we train m different classifiers on $D_j s$. When we predict for the examples in test set we output the union of the predictions of m different classifiers

Idea to incorporate RAkEl for our bandit algorithm

Same as for the previous case we will be making m different classifiers. When an example x is input , we will take the union of the different classifiers and then predict that. When we get the feedback $\tilde{y} \cap Y$, we will consider $\tilde{y} \cap Y \cap R_j$ for updating the weights for each classifier.