Capturing confusion of user between labels

By: Aditya Kumar Akash

1 Situation in which user is confused

We first look at situations where user u_i could be said to be confused between two labels l_1 and l_2. Intuitively u_i should be confused, when majority of his labelling is inverse of what is established by consensus maximization. So consider following matrix with values being the number of videos :

$Consensus \downarrow, u_i \rightarrow$	l_1	l_2	l_1, l_2	Neither
l_1	a_1	a_2	a_3	a_4
l_2	b_1	b_2	b_3	b_4
l_1, l_2	c_1	c_2	c_3	c_4
Neither	d_1	d_2	d_3	d_4

The arrow represents from whom the label is obtained. Thus rows stand for labels obtained from consensus model, while column stand for labels given by user u_i.

Now we could say that a user u_i is confused between l_1 and l_2 when he uses them interchangeably. Thus relevant elements from matrix are :

Consensus	u_i	Video count
l_1	l_2	a_2
l_1	l_1, l_2	a_3
l_2	l_1	b_1
l_2	l_1, l_2	b_3

In these cases he uses the two labels in disagreement with the consensus prediction.

Let,
$s = \sum_i (a_i + b_i + c_i + d_i)$, i.e. total number of videos he watched.
$p_o(l_i|l_j)$ = Observed probability of putting label l_i given that consensus is on only l_j

Thus, $p_o(l_2|l_1) = (a_2 + a_3)/s$, $p_o(l_1|l_2) = (b_1 + b_3)/s$.
Now we must remove from these the probability of seeing label l_i, given label l_j is used, by the consensus model. An example which better explains this would be - Suppose we labelled something as *Computer*, then given that, what would be the probability of assigning label *Automata* or *Machine* or *System* or *Junk* to that thing. Thus it is akin to keeping a label bias, and then looking at label probability distribution.

2 Label Biased Probability distribution

Consider the MLCM-r model from the **Multilabel Consensus Classification, ICDM** paper. We get a label probability distribution with each group node, which stood for seeing a label l_i when being in a group node of label l_j. Let us call this $p(reach\ l_i \mid start\ g^k_{l_j})$, where $g^k_{l_j}$ stands for group node corresponding to label l_j of k^{th} labeller and $(reach\ l_i)$ stands for reaching any node group node with label l_i. It stands for probability of reaching any group node with label l_i from a given group node of label l_j, in the random walk amongst the graph of group nodes.

Thus probability of reaching any node with label l_i, given we are starting from any node with label l_j,

$$p(reach\ l_i \mid start\ l_j)\ =\ \sum_k\ p(reach\ l_i \mid start\ g^k_{l_j})\ *\ p(start\ g_{l_j} \mid start\ l_j)$$

Assuming that we uniformly select a group of given label to start with and there are m number of labellers, $p(start\ g_{l_j} \mid start\ l_j) = \frac{1}{m}$. Hence,

$$p(reach\ l_i \mid start\ l_j)\ =\ \frac{1}{m}\ *\ \sum_k\ p(reach\ l_i \mid start\ g^k_{l_j})$$

For us $p(reach\ l_i \mid start\ l_j)$ would mean probability that consensus model has of seeing/putting label l_i, given label l_j is used. This was what we wanted. Let us call this $p_e(l_i|l_j)$.

3 Confusion

Now based on the above sections, we would calculate confusion of a user between two labels.
Using similar concept as *Cohen Kappa* confusion could be given by :

$$Confusion\ =\ I_1 * \frac{p_o(l_1|l_2)\ -\ p_e(l_1|l_2)}{1\ -\ p_e(l_2|l_1)} + I_2 * \frac{p_o(l_2|l_1)\ -\ p_e(l_2|l_1)}{1\ -\ p_e(l_2|l_1)}$$

where $I_1,\ I_2$ are indicator variables to generate positive value of confusion, i.e. $I_1 = 1,\ if\ p_o(l_1|l_2)\ \geq\ p_e(l_1|l_2)$ else 0, similar for I_2.

This measure of confusion takes into account the probability of disagreement due to model itself, and hence is expected to be more robust.