Alternative Advice Framework Formulations and Preliminary Results

(based on Ideas of Ruozzi's class)

Projected functional gradient ascent

- Works just like projected gradient ascent in parametric space
- If the gradient step overshoots the constrained region, project the parameter back into the constrained region
- If Advice gradients treated as constrains, then it is $(I_A – P)$ where I_A is the advice label where advice applies or it is just the data gradient $(I – P)$
- For all examples collect all the data gradients $(I – P)$ and collect all the advice gradients.
- Then perform simple vector projection of data gradients onto advice gradients
- If the advice and data disagree, projection will lead to 0 gradients, this means they present orthogonal explanations for that example and a question can be asked to the expert.

Contrast with Phillip's work

Projected functional gradient ascent

- If the data and advice agree where it applies, the gradient remains the same as projection on to itself doesn't change the vector.
- Questions can be asked on examples with 0 or near 0 values after projection (data and advice orthogonal to each other)

Expert advice framework

- If the data and advice agree the gradient is inflated by alpha times $n_t - n_f$
- Questions are asked when upon prediction at a leaf, the entropy is high. The learners are weak and non parametric, I believe the variance will be high after the first gradient step almost at every leaf.

Franke Wolfe method

- If there is more than one piece of advice, how do we know who is an expert in which regions of the data set or if he even is an expert over any region of the data set.
- Maybe the data can teach us new things
- Franke Wolfe method can solve this problem

- Calculate Data and multiple advice gradients same way as before
- Solve for maximizing a convex combination of all the gradients using an LP solver
- The higher the coefficient, the more it contributes to the direction of maximum ascent
- Naturally provides a ranking over data and multiple advice clauses based on coefficient values.

Franke Wolfe method continued

- Using the same principle or intuition as before disagreement between data and multiple pieces of advice will lead to a minimal or near 0 shift in the gradient
- Questions can be asked to top k highest contributors to the gradient ranked by the LP solution coefficients on the gradients within allowable resource.
- Allows for data to over-ride all pieces of advice if it contributes to the most to the maximum direction of likelihood increase.

Contrast with Phillip's work

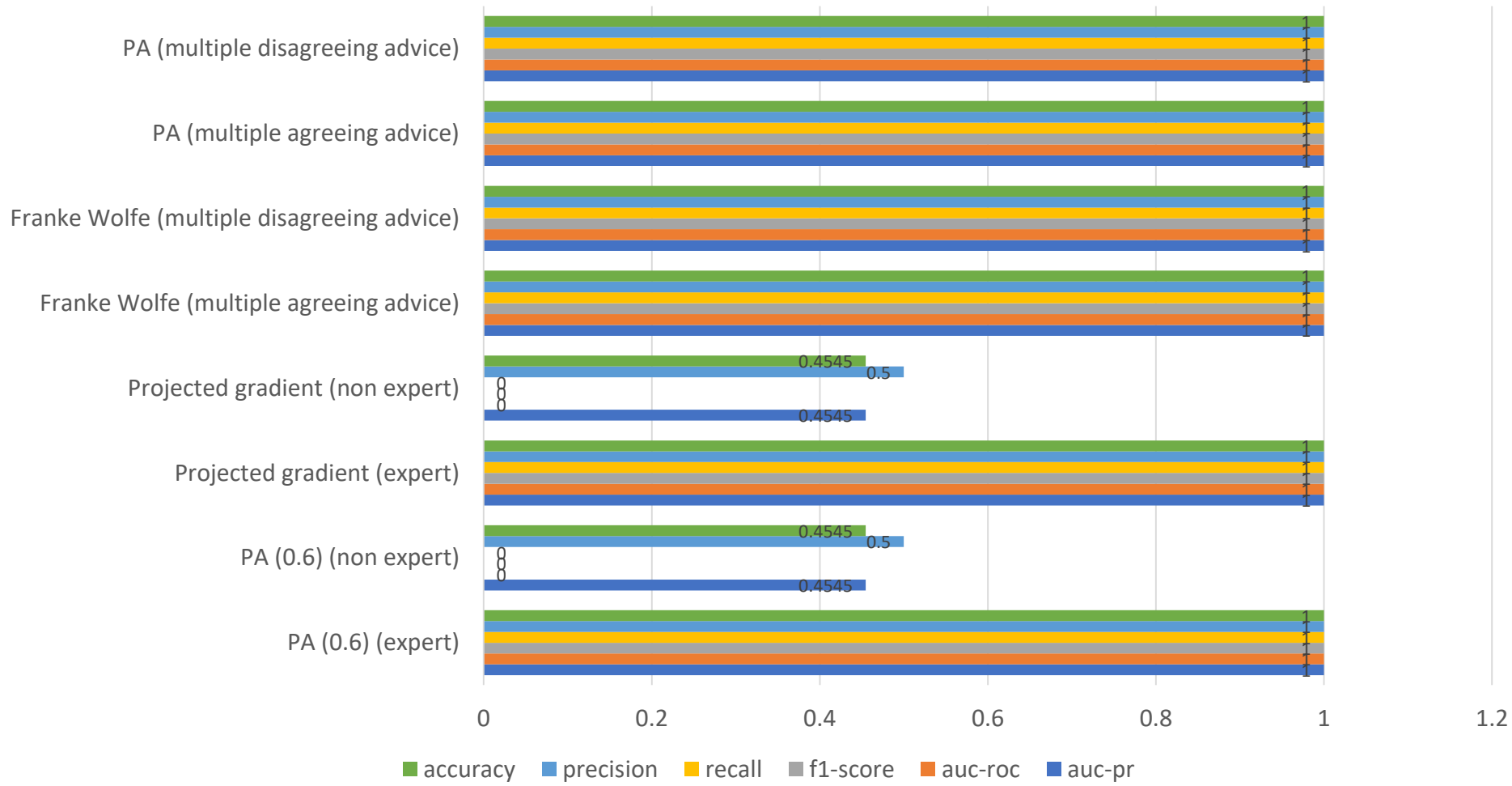
Franke Wolfe method

- LP solver polynomial time, returns weights over data and advice gradients upon maximizing convex combination.
- Advice seeking is more intuitive as the weights in the convex combination provide a natural order of expertise among the experts. When to ask? When gradient shift nears 0, who to ask? The one(s) with the top k weights.

Expert Advice framework

- If the data and advice agree the $n_t - n_f$ moves in the right direction (all though the data gradient is over-ridden unless alpha is chosen carefully), If multiple pieces of advice disagree then the data is chosen as the value of $n_t - n_f$ is small then.
- Advice seeking in the case of multiple advice is tricky as there is no clear reason to ask any expert over another for every example. Not to mention the problem of using entropy after each gradient step to decide when to seek advice still remains.

Comparison of different advice formulations with Phillip's Advice (PA) alpha value = 0.6 for heart attack data set.



Conclusion

- In evaluation metrics all methods in all settings evaluate the same as Phillip's advice framework
- However, under the hood the problem of inflated gradients still remains and how to choose alpha
- During advice seeking phase, the Projected Gradients orthogonality check (value = 0 + epsilon) and Franke Wolfe's ranking based on LP solution coefficients seems to address when and who to see advice from.
- All code and data available on github: https://github.com/kkroy36/advice-framework-experiments