Logistic Regression

Multiclass Decision Rule

- Weight vector w_y for each class
- Score (activation) of a class y $\triangleq w_y \cdot f(x)$

Classification

- The class vector most closely aligned with the result vector is the classification
- Prediction highest score wins: $y = \arg\max_y w_y \cdot f(x)$

Learning

- Iterative process; one update will not ensure the same f gets properly classified after update
- Start with $w_y = 0 \forall y$
- Pick up training examples $f(x), y*$ one by one
- Predict with current weights: $y = \arg\max_y w_y \cdot f(x)$
  - Correct $\implies$ no change
  - Wrong $\implies$ lower score of wrong answer; raise score of right answer:

$$w_y = w_y - f(x)$$
$$w_{y^*} = w_{y^*} + f(x)$$

Properties of Perceptrons

- Separability $\triangleq$ true if there exist some parameters that get the training set perfectly correct
- Convergence $\triangleq$ if the training is separable, perceptron will eventually converge (binary case)
- Mistake bound $\triangleq$ the maximum number of mistakes (binary case) related to the margin or degree of separability
  - $|\text{mistakes during training}| < \frac{|\text{features}|}{(\text{width of margin})^2}$
- If the data isn't separable, weights might thrash (won't converge)
  - Averaging weight vectors over time can help $\triangleq$ averaged perceptron
- Mediocre generalization is bad because it only finds a 'barely' separating solution
  - Want a boundary that is directly in between the two classes, but may not be
- Can result in overfitting

Probabilistic Decision (Logistic Regression)

- Have probabilities for each classification
- $z = w \cdot f(x)$ is positive $\triangleq$ want probability of + to approach 1
- $z = w \cdot f(x)$ is negative $\triangleq$ want probability of + to approach 0
- Sigmoid function: $\phi(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{e^z+1}$
  - Properties that matter: $\lim_{z\to\infty} \phi(z) = 1$, $\lim_{z\to-\infty} \phi(z) = 0$
- $\mathbb{P}(y = +1|x; w) = \frac{1}{1=e^{-w\cdot f(x)}}$
- $\mathbb{P}(y = -1|x; w) = 1 - \frac{1}{1+e^{-w\cdot f(x)}}$
- The center of the sigmoid defines the localization of the region of uncertainty
- The w in the probability sigmoid defines range of uncertainty
- Need to tune w during training / learning
  - Can use maximum likelihood estimation:

$$\begin{aligned}
&\text{likelihood} = \mathbb{P}(\text{training data}|w)\\
&= \textstyle\prod_i \mathbb{P}(\text{training datapoint } i|w)\\
&= \textstyle\prod_i \mathbb{P}(\text{point } x^{(i)} \text{ has label } y^{(i)}|w)\\
&= \textstyle\prod_i \mathbb{P}(y^{(i)}|x^{(i)}; w)\\
&\text{log likelihood} = \textstyle\sum_i \log \mathbb{P}(y^{(i)}|x^{(i)}; w)
\end{aligned}$$

- The probabilities can be drawn from the sigmoid definitions
- Allows us to maximize our confidence in our guesses

Multiclass Logistic Regression

- Softmax activation $\triangleq z_1, z_2, z_3 \rightarrow \frac{e^{z_1}}{e^{z_1}+e^{z_2}+e^{z_3}}, \frac{e^{z_2}}{e^{z_1}+e^{z_2}+e^{z_3}}, \frac{e^{z_3}}{e^{z_1}+e^{z_2}+e^{z_3}}$
  - Use instead of sigmoid function
  - Always positive
  - All values sum to 1
  - Its easy for one class to dominate the value
- $$\mathbb{P}(y|x;w) = \frac{e^{w}y \cdot f(x)}{\sum_{y'} e^{w}y' \cdot f(x)}$$
- Then maximum log likelihood estimation for learning:

$$\max_w ll(w) = \max_w \sum_i \log(\mathbb{P}(y^{(i)}|x^{(i)};w))$$

  - Note: cannot always set derivative to 0