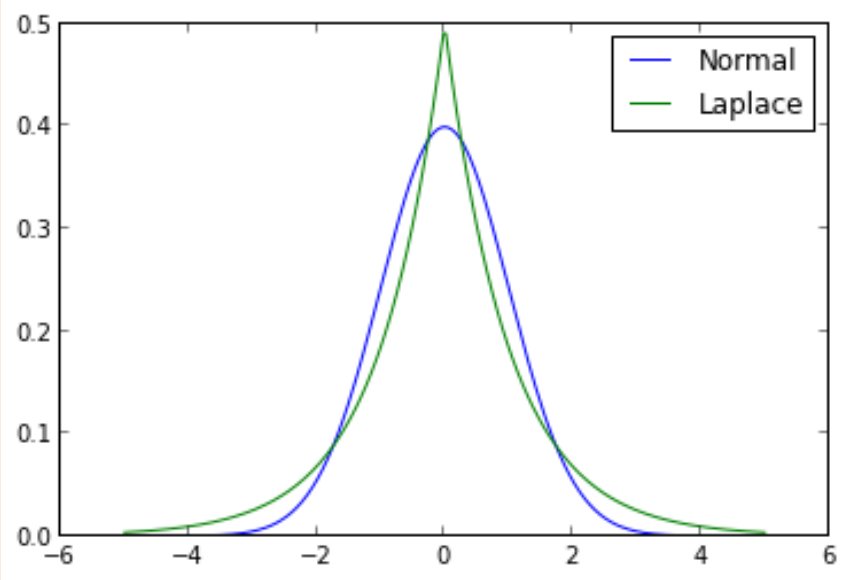
CPSC 340:
Machine Learning and Data Mining

MLE and MAP

Bonus slides

“Heavy” Tails vs. “Light” Tails

- We know that L1-norm is more robust than L2-norm.
  - What does this mean in terms of probabilities?



Here "fail" means "mass of the distribution away from the mean"

  - Gaussian has “light tails”: assumes everything is close to mean.
  - Laplace has “heavy tails”: assumes some data is far from mean.
  - Student ‘t’ is even more heavy-tailed/robust, but NLL is non-convex.

http://austinrochford.com/posts/2013-09-02-prior-distributions-for-bayesian-regression-using-pymc.html

Regularizing Other Models

- We can view priors in other models as regularizers.

- Remember the problem with MLE for naïve Bayes:
  - The MLE of p('lactase' = 1| 'spam') is: count(spam,lactase)/count(spam).
  - But this caused problems if count(spam,lactase) = 0.

- Our solution was Laplace smoothing:
  - Add "+1" to our estimates: (count(spam,lactase)+1)/(counts(spam)+2).
  - This corresponds to a "Beta" prior so Laplace smoothing is a regularizer.

Why do we care about MLE and MAP?

- Unified way of thinking about many of our tricks?
  - Probabilitic interpretation of logistic loss.
  - Laplace smoothing and L2-regularization are doing the same thing.

- Remember our two ways to reduce overfitting in complicated models:
  - Model averaging (ensemble methods).
  - Regularization (linear models).

- "Fully"-Bayesian methods (CPSC 540) combine both of these.
  - Average over all models, weighted by posterior (including regularizer).
  - Can use extremely-complicated models without overfitting.

Losses for Other Discrete Labels

- MLE/MAP gives loss for classification with basic labels:
  - Least squares and absolute loss for regression.
  - Logistic regression for binary labels {“spam”, “not spam”}.
  - Softmax regression for multi-class {“spam”, “not spam”, “important”}.

- But MLE/MAP lead to losses with other discrete labels (bonus):
  - Ordinal: {1 star, 2 stars, 3 stars, 4 stars, 5 stars}.
  - Counts: 602 ‘likes’.
  - Survival rate: 60% of patients were still alive after 3 years.
  - Unbalanced classes: 99.9% of examples are classified as +1.
- Define likelihood of labels, and use NLL as the loss function.

- We can also use ratios of probabilities to define more losses (bonus):
  - Binary SVMs, multi-class SVMs, and “pairwise preferences” (ranking) models.

Discussion: Least Squares and Gaussian Assumption

- Classic justifications for the Gaussian assumption underlying least squares:
  - Your noise might really be Gaussian. (It probably isn't, but maybe it's a good enough approximation.)
  - The central limit theorem (CLT) from probability theory. (If you add up enough IID random variables, the estimate of their mean converges to a Gaussian distribution.)

- I think the CLT justification is wrong as we've never assumed that the x_{ij} are IID across 'j' values. We only assumed that the examples x_i are IID across 'i' values, so the CLT implies that our estimate of 'w' would be a Gaussian distribution under different samplings of the data, but this says nothing about the distribution of y_i given $w^T x_i$.

- On the other hand, there are reasons *not* to use a Gaussian assumption, like it's sensitivity to outliers. This was (apparently) what lead Laplace to propose the Laplace distribution as a more robust model of the noise.
- The "student t" distribution (published anonymously by Gosset while working at the Guiness beer company) is even more robust, but doesn't lead to a convex objective.

Binary vs. Multi-Class Logistic

- How does multi-class logistic generalize the binary logistic model?
- We can re-parameterize softmax in terms of (k-1) values of z_c:

$$p(y|z_1, z_2, \cdots, z_{k-1}) = \frac{\exp(z_y)}{1 + \sum_{c=1}^{k-1} \exp(z_c)} \text{ if } y \neq k \quad \text{and} \quad p(y|z_1, z_2, \cdots, z_{k-1}) = \frac{1}{1 + \sum_{c=1}^{k-1} \exp(z_c)} \text{ if } y = k$$

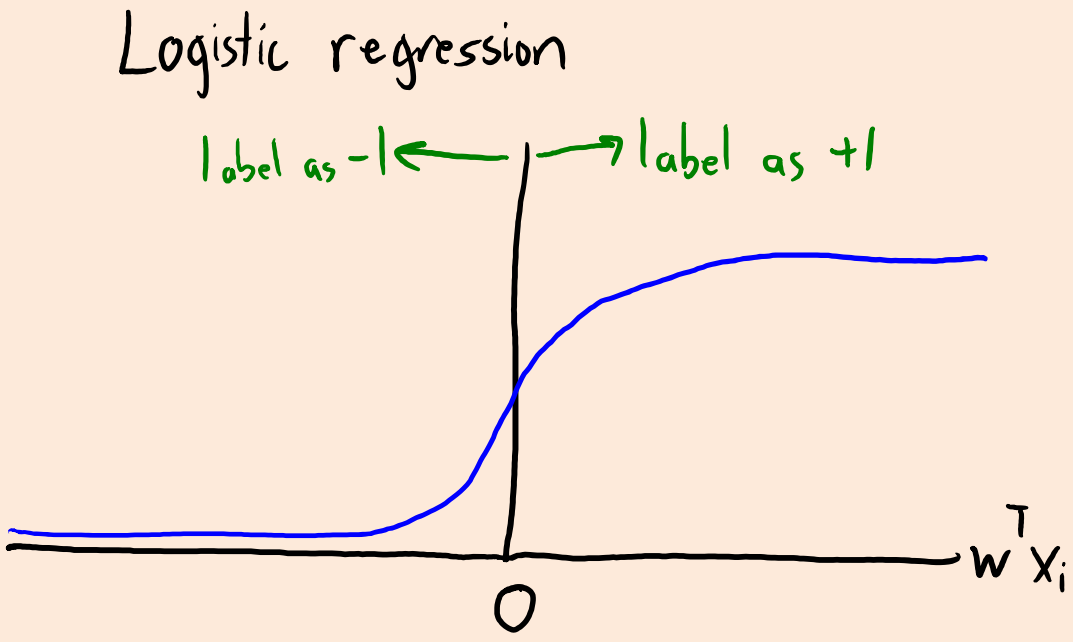
  - This is due to the "sum to 1" property (one of the z_c values is redundant).
  - So if k=2, we don't need a z_2 and only need a single 'z'.
  - Further, when k=2 the probabilities can be written as:

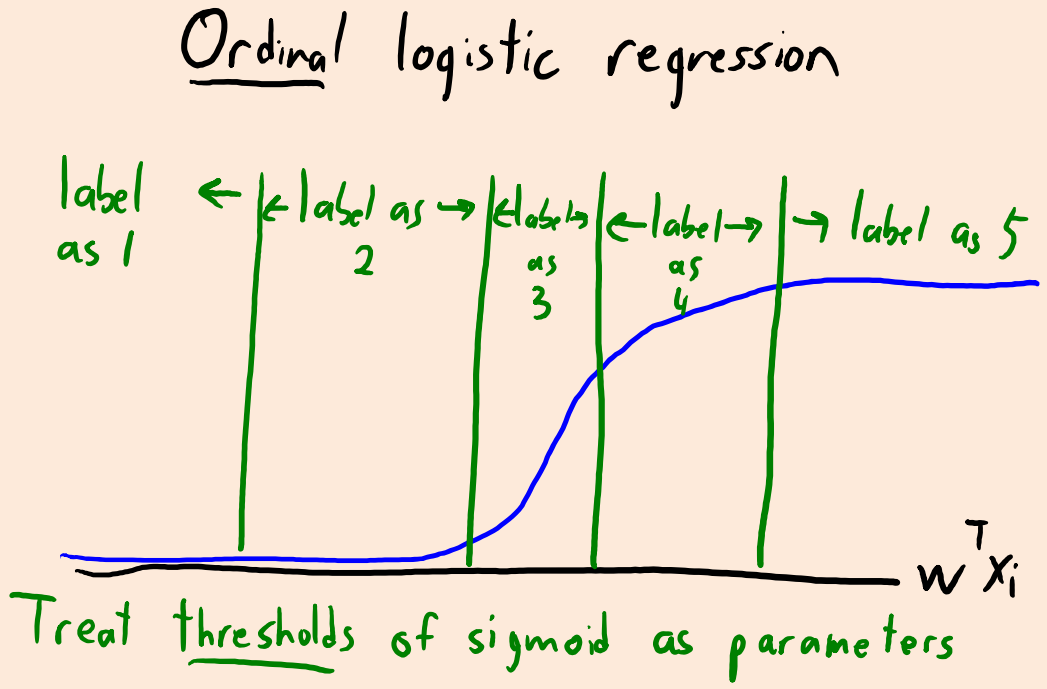
$$p(y=1|z) = \frac{\exp(z)}{1+\exp(z)} = \frac{1}{1+\exp(-z)} \qquad p(y=2|z) = \frac{1}{1+\exp(z)}$$

  - Renaming '2' as '-1', we get the binary logistic regression probabilities.

Ordinal Labels

- Ordinal data: categorical data where the order matters:
  - Rating hotels as {'1 star', '2 stars', '3 stars', '4 stars', '5 stars'}.
  - Softmax would ignore order.
- Can use 'ordinal logistic regression'.





Count Labels

- Count data: predict the number of times something happens.
  - For example, y_i = “602” Facebook likes.
- Softmax requires finite number of possible labels.
- We probably don’t want separate parameter for ‘654’ and ‘655’.
- Poisson regression: use probability from Poisson count distribution.
  - Many variations exist, a lot of people think this isn’t the best likelihood.

Censored Survival Analysis (Cox Partial Likelihood)

- Censored survival analysis:
  - Target y_i is last time at which we know person is alive.
    - But some people are still alive (so they have the same y_i values).
    - The y_i values (time at which they die) are "censored".
  - We use v_i=0 is they are still alive and otherwise we set $v_i = 1$.

- Cox partial likelihood assumes "instantaneous" rate of dying depends on x_i but not on total time they've been alive (not that realistic). Leads to likelihood of the "censored" data of the form:

$$p(y_i, v_i | x_i, w) = \exp(v_i w^T x_i) \exp(-y_i \exp(w^T x_i))$$

- There are many extensions and alternative likelihoods.

Other Parsimonious Parameterizations

- Sigmoid isn't the way to model a binary $p(y_i \mid x_i, w)$:
  - Probit (uses CDF of normal distribution, very similar to logistic).
  - Noisy-Or (simpler to specify probabilities by hand).
  - Extreme-value loss (good with class imbalance).
  - Cauchit, Gosset, and many others exist…

Unbalanced Training Sets

- Consider the case of binary classification where your training set has 99% class -1 and only 1% class +1.
  - This is called an "unbalanced" training set
- Question: is this a problem?
- Answer: it depends!
  - If these proportions are representative of the test set proportions, and you care about both types of errors equally, then "no" it's not a problem.
    - You can get 99% accuracy by just always predicting -1, so ML can only help with the 1%.
  - But it's a problem if the test set is not like the training set (e.g. your data collection process was biased because it was easier to get -1's)
  - It's also a problem if you care more about one type of error, e.g. if mislabeling a +1 as a -1 is much more of a problem than the opposite
    - For example if +1 represents "tumor" and -1 is "no tumor"

Unbalanced Training Sets

- This issue comes up a lot in practice!
- How to fix the problem of unbalanced training sets?
  - Common strategy is to build a "weighted" model:
    - Put higher weight on the training examples with y_i=+1.

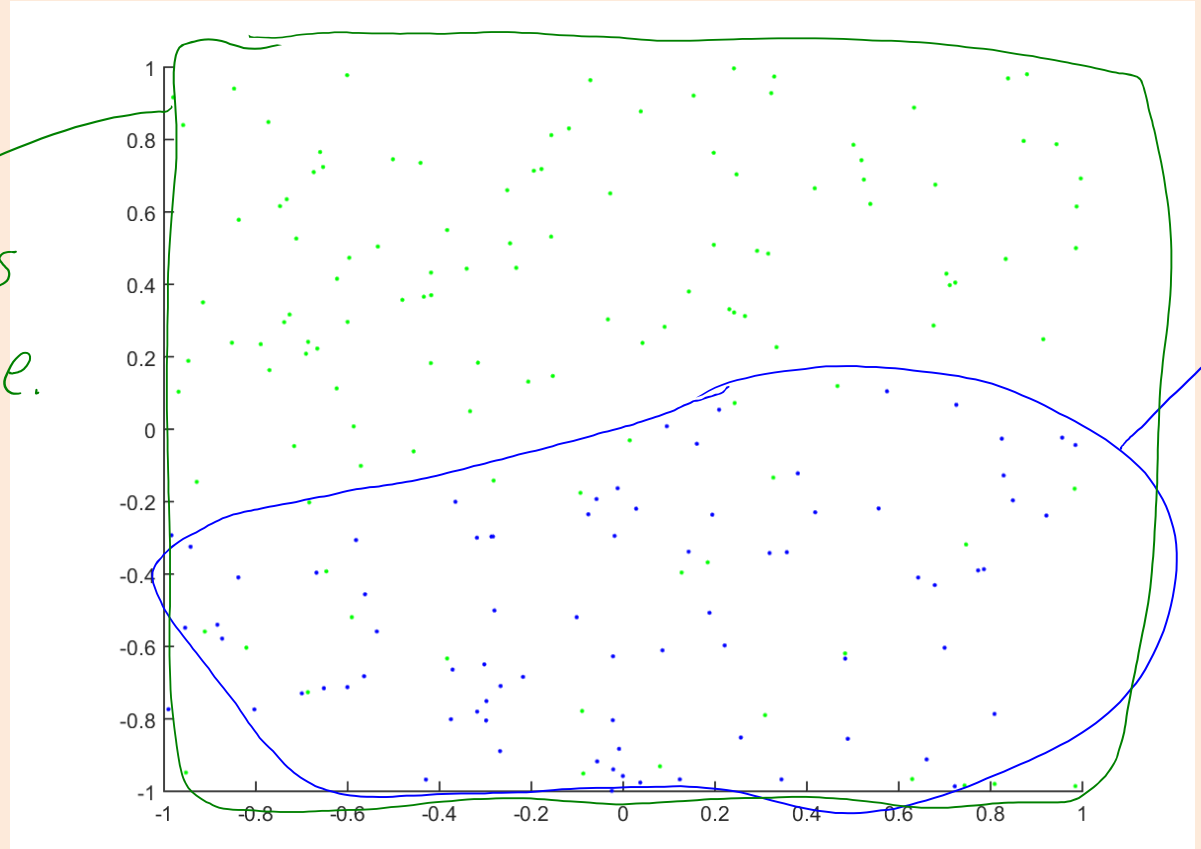
$$f(w) = \sum_{i=1}^{n} v_i \log(1+\exp(-y_i w^T x_i))$$

Make this weight bigger for under-represented class

    - This is equivalent to replicating those examples in the training set.
    - You could also subsample the majority class to make things more balanced.
    - Boostrap: create a dataset of size 'n' where n/2 are sampled from +1, n/2 from -1.
  - Another approach is to try to make "fake" data to fill in minority class.
  - Another option is to change to an asymmetric loss function (next slides) that penalizes one type of error more than the other.
  - Some discussion of different methods here.

Unbalanced Data and Extreme-Value Loss

- Consider binary case where:
  - One class overwhelms the other class ('unbalanced' data).
  - Really important to find the minority class (e.g., minority class is tumor).



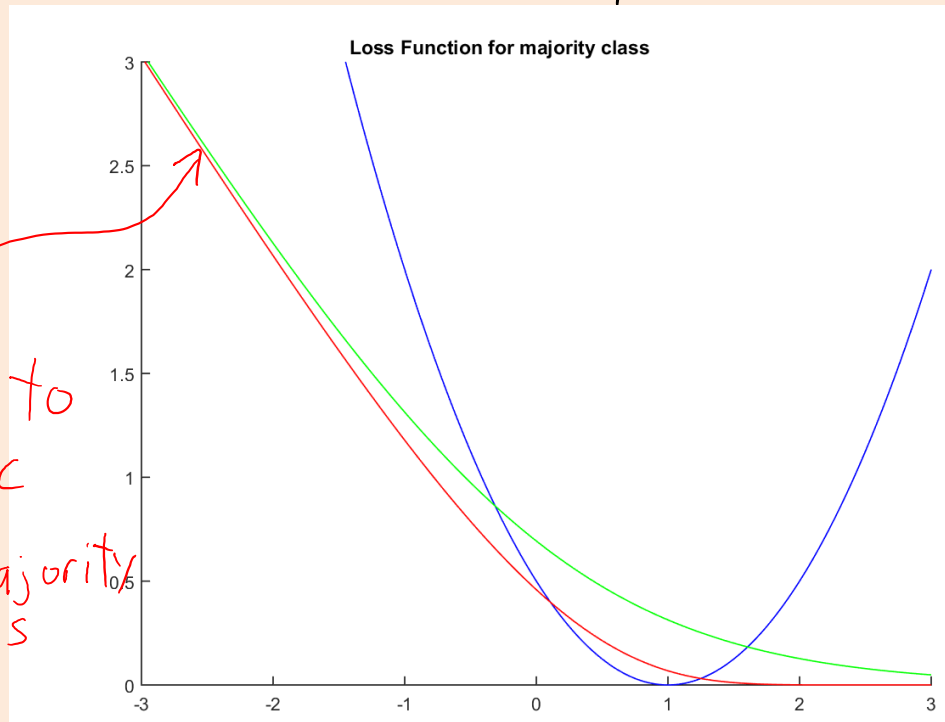
Unbalanced Data and Extreme-Value Loss

- Extreme-value distribution:

$$p(y_i = +1 \mid \hat{y}_i) = 1 - \exp(-\exp(\hat{y}_i)) \quad [+1 \text{ is majority class}]$$

asymmetric

To make it a probability, $p(y_i = -1 \mid \hat{y}_i) = \exp(-\exp(\hat{y}_i))$



Similar to logistic for majority class

big penalty for getting minority class wrong.

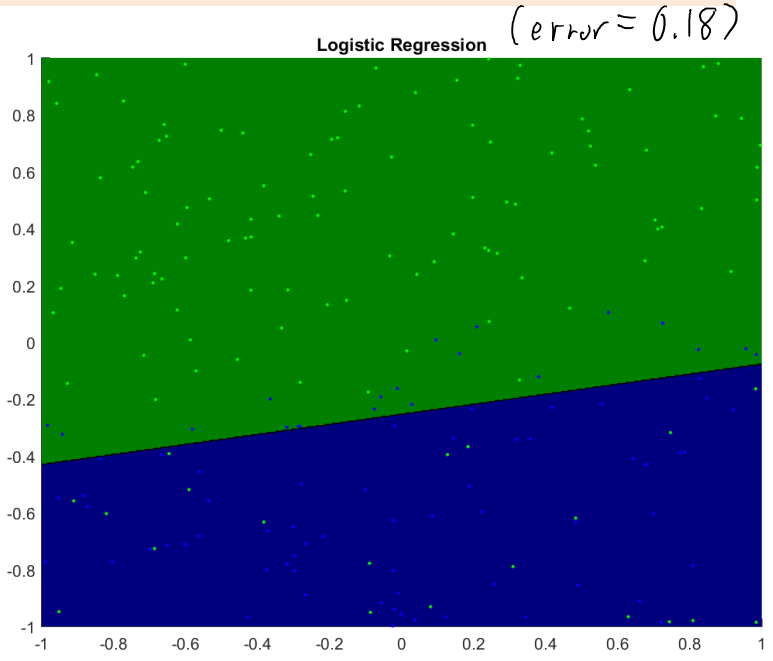
Unbalanced Data and Extreme-Value Loss

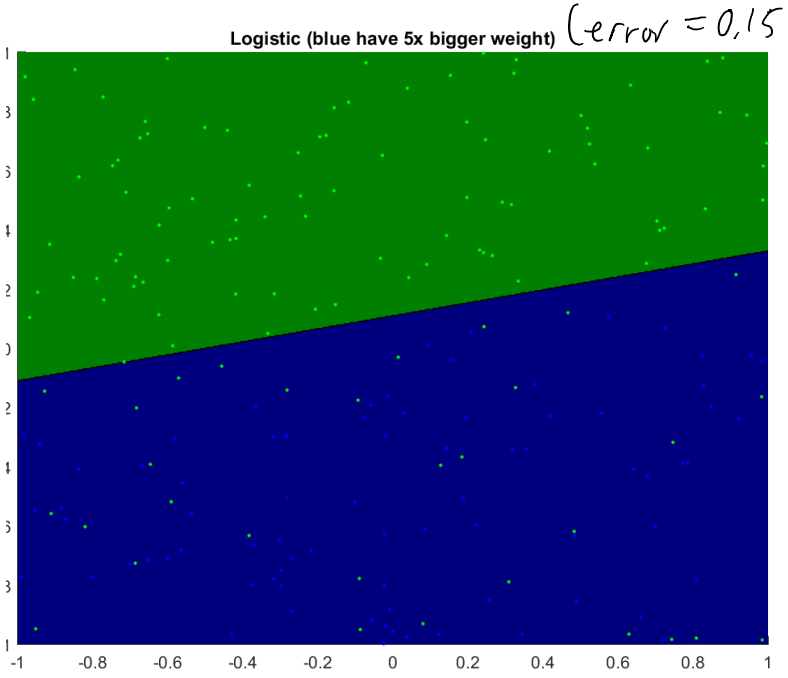
- Extreme-value distribution:

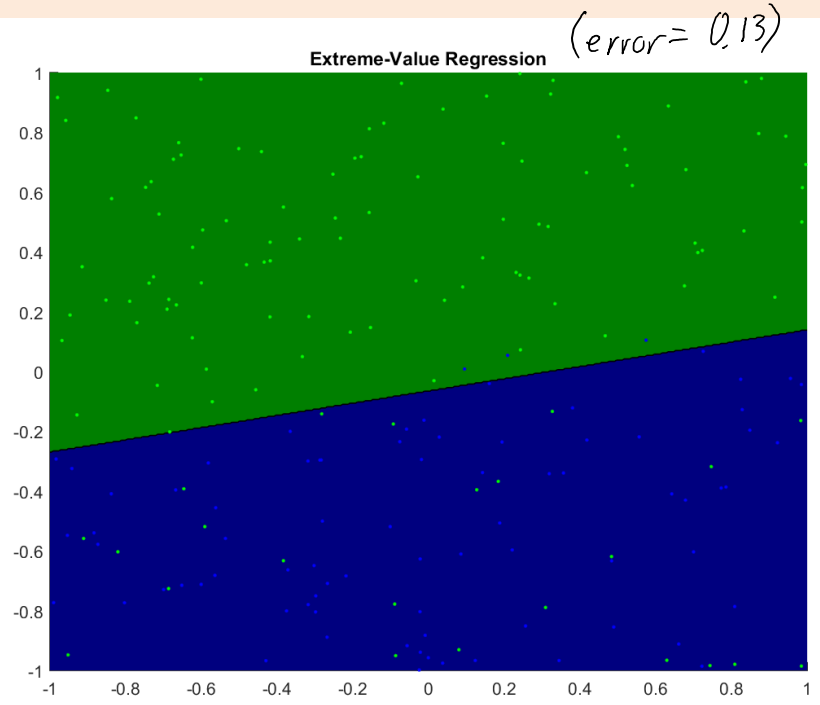
$$p(y_i = +1 \mid \hat{y}_i) = 1 - \exp(-\exp(\hat{y}_i)) \quad [+1 \text{ is majority class}]$$

asymmetric

To make it a probability, $p(y_i = -1 \mid \hat{y}_i) = \exp(-\exp(\hat{y}_i))$







Loss Functions from Probability Ratios

- We've seen that loss functions can come from probabilities:
  - Gaussian => squared loss, Laplace => absolute loss, sigmoid => logistic.
- Most other loss functions can be derived from probability ratios.
  - Example: sigmoid => hinge.

$$p(y_i \mid x_i, w) = \frac{1}{1 + \exp(-y_i w^T x_i)} = \frac{\exp(\frac{1}{2} y_i w^T x_i)}{\exp(\frac{1}{2} y_i w^T x_i) + \exp(-\frac{1}{2} y_i w^T x_i)} \propto \exp(\frac{1}{2} y_i w^T x_i)$$

Same normalizing constant for $y_i = +1$ and $y_i = -1$

Loss Functions from Probability Ratios

- We've seen that loss functions can come from probabilities:
  - Gaussian => squared loss, Laplace => absolute loss, sigmoid => logistic.
- Most other loss functions can be derived from probability ratios.
  - Example: sigmoid => hinge.

$$p(y_i \mid x_i, w) \propto \exp\left(\tfrac{1}{2} y_i w^T x_i\right)$$

To classify y_i correctly, it's sufficient to have $\frac{p(y_i \mid x_i, w)}{p(-y_i \mid x_i, w)} \geq \beta$ for some 'β' > 1

Notice that normalizing constant doesn't matter:

$$\frac{\exp\left(\frac{1}{2} y_i w^T x_i\right)}{\exp\left(-\frac{1}{2} y_i w^T x_i\right)} \geq \beta$$

Loss Functions from Probability Ratios

- We've seen that loss functions can come from probabilities:
  - Gaussian => squared loss, Laplace => absolute loss, sigmoid => logistic.
- Most other loss functions can be derived from probability ratios.
  - Example: sigmoid => hinge.

$$p(y_i \mid x_i, w) \propto \exp\left(\tfrac{1}{2} y_i w^T x_i\right)$$

We need: $\dfrac{\exp\left(\frac{1}{2} y_i w^T x_i\right)}{\exp\left(-\frac{1}{2} y_i w^T x_i\right)} \geq \beta$

$y_i w^T x_i \geq 1$ (if we choose $\log(\beta) = 1$)

Take log:

$$\log\left(\frac{\exp\left(\frac{1}{2} y_i w^T x_i\right)}{\exp\left(-\frac{1}{2} y_i w^T x_i\right)}\right) \geq \log(\beta) \iff \frac{1}{2} y_i w^T x_i + \frac{1}{2} y_i w^T x_i \geq \log(\beta)$$

Loss Functions from Probability Ratios

- We've seen that loss functions can come from probabilities:
  - Gaussian => squared loss, Laplace => absolute loss, sigmoid => logistic.
- Most other loss functions can be derived from probability ratios.
  - Example: sigmoid => hinge.

$$p(y_i \mid x_i, w) \propto \exp\left(\tfrac{1}{2} y_i w^T x_i\right)$$

We need: $\dfrac{\exp\left(\frac{1}{2} y_i w^T x_i\right)}{\exp\left(-\frac{1}{2} y_i w^T x_i\right)} \geq \beta$

Or equivalently:

$$y_i w^T x_i \geq 1 \quad (\text{for } \beta = \exp(1))$$

Define a loss function by amount of constraint violation:

$$\max\{0, 1 - y_i w^T x_i\}$$

when $1 - y_i w^T x_i \leq 0$ when $1 - y_i w^T x_i \geq 0$

We get SVMs by looking at regularized average loss:

$$f(w) = \sum_{i=1}^{n} \max\{0, 1 - y_i w^T x_i\} + \frac{\lambda}{2}\|w\|^2$$

Loss Functions from Probability Ratios

- General approach for defining losses using probability ratios:
  1. Define constraint based on probability ratios.
  2. Minimize violation of logarithm of constraint.
- Example: softmax => multi-class SVMs.

Assume: $p(y_i = c \mid x_i, w) \propto \exp(w_c^T x_i)$

Want: $\frac{p(y_i \mid x_i, w)}{p(y_i = c' \mid x_i, w)} \geq \beta$ for all c' and some $\beta > 1$

For $\beta = \exp(1)$ equivalent to

$$w_{y_i}^T x_i - w_{c'}^T x_i \geq 1 \quad \text{for } \underline{\text{all}}\ c' \neq y_i$$

Option 1: penalize all violations:

$$\sum_{c'=1}^{K} \max\{0, 1 - w_{y_i}^T x_i + w_{c'}^T x_i\}$$

Option 2: penalize only max violation:

$$\max_{c' \neq c} \{\max\{0, 1 - w_{y_i}^T x_i + w_{c'}^T x_i\}\}$$

Supervised Ranking with Pairwise Preferences

- Ranking with pairwise preferences:
  - We aren't given any explicit y_i values.
  - Instead we're given list of objects (i,j) where $y_i > y_j$.

Assume $p(y_i \mid X, w) \propto \exp(w^T x_i)$ is probability that object 'i' has highest rank.

Want: $\frac{p(y_i \mid X, w)}{p(y_j \mid X, w)} \geq \beta$ for all preferences (i,j)

For $\beta = \exp(1)$ equivalent to

$w^T x_i - w^T x_j \geq 1$

for preferences (i,j)

We can use $f(w) = \sum_{(i,j) \in R} \max\{0, 1 - w^T x_i + w^T x_j\}$

This approach can also be used to define losses for total/partial orderings. (but this information is hard to get)