Naive Bayes' classifier example w/Laplace smoothing

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

function TRAIN NAIVE BAYES(D, C) **returns** log $P(c)$ and log $P(w|c)$

for each class $c \in C$ # Calculate $P(c)$ terms
  N_{doc} = number of documents in D
  N_c = number of documents from D in class c
  $logprior[c] \leftarrow \log \frac{N_c}{N_{doc}}$
  $V \leftarrow$ vocabulary of D
  bigdoc[*c*] ← **append**(d) **for** d ∈ D **with** class *c*
  for each word *w* in V # Calculate $P(w|c)$ terms
    count(w,c) ← # of occurrences of *w* in *bigdoc*[*c*]
    $loglikelihood[w,c] \leftarrow \log \frac{count(w,c)\ +\ 1}{\sum_{w' \text{ in } V}\ (count\ (w',c)\ +\ 1)}$
return *logprior*, *loglikelihood*, *V*

function TEST NAIVE BAYES(*testdoc*, *logprior*, *loglikelihood*, C, V) **returns** best *c*

for each class $c \in C$
  sum[*c*] ← *logprior*[*c*]
  for each position *i* in *testdoc*
    word ← *testdoc[i]*
    if *word* ∈ *V*
      sum[*c*] ← *sum*[*c*]+ *loglikelihood*[*word,c*]
return $\text{argmax}_c\ sum[c]$

- Taken from Jurafsky & Martin, Chp 4

Text classification with discriminative models

- Discriminative modeling approach that learns P(y|x)

- Quick refresher about MLE & discriminative models:

$P(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}) \quad where \quad \mathbf{x}, \mathbf{y} \sim P_D\,, \quad \mathbf{y} \in \{0,1\}^K \longrightarrow$ one hot encoding

$$\begin{aligned}
\hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{M} -\log P(\mathbf{y}_i \mid \mathbf{x}_i; \boldsymbol{\theta}) \\
&= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \; -\mathbb{E}_{\mathbf{x},\mathbf{y}\sim P_D}\left[\log P(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta})\right] \\
&= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \; -\sum_{i=1}^{M} \mathbf{y} \log P(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}) \longrightarrow
\end{aligned}$$

Cross entropy between data distribution and the model output distribution, $H\left(P_D(\mathbf{y} \mid \mathbf{x}), P(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta})\right)$. Note, this holds for arbitrary PMFs (softmax, Bernoulli, etc...). Because the labels, $\mathbf{y}$, are one hot, they have zero entropy, and thus in this setting minimizing the cross entropy is equivalent to minimizing the KL divergence, $D_{KL}\left(P_D(\mathbf{y} \mid \mathbf{x}), P(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta})\right)$.