Feature Grammar Notes

Spring 2015

We want to design a model that intelligently expands its feature vocabulary using a "feature grammar". The feature grammar should consist of a set of rules that generate increasingly complex features, and the model should use some heuristics to navigate the infinite space of features generated by this grammar. Logistmar gramression is a variation of logistic regression that fulfills this goal by using heuristics defined over a feature grammar to expand an initial seed vocabulary of features. The heuristics given to logistmar gramression are a set of rules for navigating the grammar's infinite feature space in search of relevant features, and the logistmar gramression training procedure uses these rules to determine which features in the space should be selected to have non-zero weight. If the heuristics guide the model to all relevant parts of the space prior to irrelevant parts of the space, then logistmar gramression will produce the same classifier as l2 regularized logistic regression trained with a vocabulary consisting of mostly relevant features.

1 Some definitions and the basic idea for logistmar gramression

The following definitions are helpful in describing logistmar gramression:

- $D \subseteq X \times Y$ is a set of training examples. Each example $(x, y) \in X \times Y$ consists of a reference x to some aspect of the data (e.g. a noun phrase mention in text), and a label $y \in \{0, 1\}$. We write each $d \in D$ as $d = (x^d, y^d)$.
- $F^\infty = \{f : X \to [0, 1]\}$ is the set of all possible features generated by a grammar G. H is a set of heuristics for the exploration of G. The heuristics in H are functions $h : F^\infty \to F^\infty$. For some $f \in F^\infty$, $h(f)$ may produce no additional feature, in which case $h(f) = \varnothing$.[1]
- $F_0 \subset F^\infty$ is an initial feature vocabulary for the model, and $F_i = \{h(f_{i-1}) | h \in H, f_{i-1} \in F_{i-1}, h(f_{i-1}) \neq \varnothing\}$ is the set of features at the ith level of the heuristic search.
- For $f \in F^\infty$, $i(f)$ is defined such that $f \in F_{i(f)}$.
- $H^* : F^\infty \to 2^{F^\infty}$ gives the set of all descendants of a function in H's search graph.
- $H^{-*} : F^\infty \to 2^{F^\infty}$ gives the set of all ancestors of a function in H's search graph.
- $H^{-1} : F^\infty \to 2^{F^\infty}$ gives the set of immediate parents of a function in H's search graph. More precisely, $H^{-1}(f) = \{f' | \exists h \in H : h(f') = f\}$.
- $H_t^{-1} : F^\infty \to 2^{F^\infty}$ gives $H_t^{-1}(f) = H^{-1}(f) \cup \{f'(x) = t\}$.

[1] It would also be interesting to consider heuristics that take more than one feature as an argument. Modifying the model described below to incorporate such heuristics may result in something that may have a relationship to a neural network... I think...? This would be another interesting idea to explore later on.

Logistmar gramression learns a parameter vector $\mathbf{u}$ for a distribution of the form:

$$P(y = 1|x, \mathbf{u}) = \frac{e^{\sum_{i=0}^{\infty} \sum_{f \in F_i} f(x) c_{+,f}(\mathbf{u})}}{e^{\sum_{i=0}^{\infty} \sum_{f \in F_i} f(x) c_{+,f}(\mathbf{u})} + e^{\sum_{i=0}^{\infty} \sum_{f \in F_i} f(x) c_{-,f}(\mathbf{u})}} \tag{1}$$

Where c_{f+} and c_{f-} are functions of the parameters with outputs in $\mathbb{R}$ that are analogous to the weights from traditional logistic regression in their relation to the behavior of the model. As shorthand, we write $\mathbf{c}(\mathbf{u})$ for the vector resulting from applying all $c_{+,f}$ and $c_{-,f}$ to $\mathbf{u}$.

Letting $r_{d,\mathbf{u}} = P(y_d = 1|x_d, \mathbf{u})$, the negative conditional log-likelihood of the training data D under distribution given by Equation 1 is:

$$L_r(\mathbf{u}) = -\sum_d y_d \log r_{d,\mathbf{u}} + (1 - y_d) \log(1 - r_{d,\mathbf{u}}) \tag{2}$$

And with l2 regularization, this becomes:

$$L_{r2}(\mathbf{u}) = L_r(\mathbf{u}) + \beta \|\mathbf{c}(\mathbf{u})\|_2^2 \tag{3}$$

Logistmar gramression chooses the l2 regularized maximum likelihood estimate (MLE) of $\mathbf{u}$ by minimizing Equation 3. When designing logistmar gramression, we chose $\mathbf{c}$ functions such that:

- Each $c_{+,f}, c_{-,f} \in \mathbf{c}$ is a non-negative convex function of the parameters $\mathbf{u}$. This ensures that logistmar gramression loss function will also be convex, and have a global minimum.
- Each $c_{+,f}, c_{-,f} \in \mathbf{c}$ is subdifferentiable.
- The infinite sums in Equation 1 converge when $\mathbf{u}$ is chosen by MLE.[2]
- Computing the MLE using subgradient descent only requires computing a finite number of terms in the gradient, even though the sums in Equation 1 have an infinite number of terms.
- If successive applications of the heuristics in H to F_0 produce all and only relevant features prior to irrelevant features, then the MLE of $\mathbf{u}$ will produce a model that is the same as the model produced by l2 regularized logistic regression with a feature vocabulary containing mostly relevant features.

The first five requirements for $\mathbf{c}$ listed above are necessary to be able to use subgradient descent to compute the minimum of Equation 3. The final requirement in the list ensures that logistmar gramression produces a sensible classifier under an assumption about the grammar heuristics and initial feature vocabulary.[3]

In order to meet the above requirements, we let $t > 0$ be a constant threshold, constrain the parameter $\mathbf{u}$ to be a non-negative infinite dimensional vector, and define $\mathbf{c}$ by:

$$\forall f \in F^{\infty}, s \in \{+, -\} : c_{s,f}(\mathbf{u}) = \begin{cases} u_{s,f} & : f \in F_0 \\ u_{s,f} \max_{f' \in H_t^{-1}(f), s' \in \{+,-\}} \left(c_{s',f'}(\mathbf{u}) - t \right) & : f \notin F_0 \end{cases} \tag{4}$$

[2]Further, we only considered choices of $\mathbf{c}$ which force the sums in Equation 1 to have a finite number of non-zero terms, but it would also be interesting to consider models which rely on probability distributions over truly infinite dimensional spaces... without kernel stuff... but just convergent infinite series? I'm not sure whether that would turn out to make any sense, but there appears to be some existing math for such a thing (see `http://www.math.cornell.edu/~neldredge/7770/7770-lecture-notes.pdf`). It looks complicated.)

[3]Interesting alternative models might be constructed using various choices of $\mathbf{c}$ that reflect assumptions about the grammar heuristics other than the one given in the final requirement in the list.

The basic idea for this choice of c is that $c_{s,f}(\mathbf{u}) > 0$ only if $f \in F_0$ or f has some parent feature $f' \in H^{-1}(f)$ for which $c_{+,f'}(\mathbf{u}) > t$ or $c_{-,f'}(\mathbf{u}) > t$. In other words, logistmar gramression will give a feature non-zero weight only if it has a parent feature in the heuristic search that has weight greater than threshold t. If the MLE of $\mathbf{u}$ produces $c_{s,f}$ that correlate with the relevance of f, then logistmar gramression will tend to assign non-zero weight to features only if their ancestors in the heuristic search are relevant. This fact entails the fulfillment of the final requirement for $\mathbf{c}$ in the above list, which we formally prove in Section 3.

The remainder of this document will focus on establishing properties of the logistmar gramression model with the above choice of $\mathbf{c}$. The next section will first give some proofs of the relationships between several variations on standard logistic regression. Section 3 will give several formal assumptions about the feature grammar heuristics that are sufficient for logistmar gramression to produce the same classifier as l2 regularized logistic regression with a vocabulary of mostly relevant features. Lastly, Section 4 will give an example grammar that could be used with logistmar gramression.

2 Relationships between logistic regression variations

Consider the following conditional distributions. We want to find the relationships between the values of their parameters given by MLE with and without l2 regularization.

$$p_{d,\mathbf{w}_+,\mathbf{w}_-} = P(y = 1|x, \mathbf{w}_+, \mathbf{w}_-) = \frac{e^{\mathbf{w}_+^T \mathbf{f}(x^d)}}{e^{\mathbf{w}_+^T \mathbf{f}(x^d)} + e^{\mathbf{w}_-^T \mathbf{f}(x^d)}} \tag{5}$$

$$q_{d,v} = P(y = 1|x^d, \mathbf{v}) = \frac{e^{\mathbf{v}^T \mathbf{f}(x^d)}}{1 + e^{\mathbf{v}^T \mathbf{f}(x^d)}} \tag{6}$$

2.1 Without regularization

MLE without reglarization for the above distributions is equivalent to finding parameters that minimize the following convex loss functions:

$$L_p(\mathbf{w}_+, \mathbf{w}_-) = -\sum_d y_d \log p_{d,\mathbf{w}_+,\mathbf{w}_-} + (1 - y_d) \log(1 - p_{d,\mathbf{w}_+,\mathbf{w}_-}) \tag{7}$$

$$L_q(\mathbf{v}) = -\sum_d y_d \log q_{d,\mathbf{v}} + (1 - y_d) \log(1 - q_{d,\mathbf{v}}) \tag{8}$$

Note that 8 is strictly convex, and so has a unique minimum $\mathbf{v}^*$, whereas 7 is non-strictly convex, and so has several minima (see `http://qwone.com/~jason/writing/convexLR.pdf`). Let $\mathbf{w}_+^*, \mathbf{w}_-^*$ be an arbitrary minimum of 7.

Theorem 2.1. $\forall d, \mathbf{w}_+, \mathbf{w}_- : p_{d,\mathbf{w}_+,\mathbf{w}_-} = q_{d,\mathbf{w}_+ - \mathbf{w}_-}$, *and therefore* $L_p(\mathbf{w}_+, \mathbf{w}_-) = L_q(\mathbf{w}_+ - \mathbf{w}_-)$.

$$p_{d,\mathbf{w}_+,\mathbf{w}_-} = \frac{e^{\mathbf{w}_+^T\mathbf{f}(x^d)}}{e^{\mathbf{w}_+^T\mathbf{f}(x^d)} + e^{\mathbf{w}_-^T\mathbf{f}(x^d)}} \tag{9}$$

$$= \frac{\frac{e^{\mathbf{w}_+^T\mathbf{f}(x^d)}}{e^{\mathbf{w}_-^T f(x^d)}}}{\frac{e^{\mathbf{w}_+^T\mathbf{f}(x^d)}}{e^{\mathbf{w}_-^T\mathbf{f}(x^d)}} + 1} \tag{10}$$

$$= \frac{e^{(\mathbf{w}_+^T - \mathbf{w}_-^T)\mathbf{f}(x^d)}}{e^{(\mathbf{w}_+^T - \mathbf{w}_-^T)\mathbf{f}(x^d)} + 1} \tag{11}$$

$$= q_{d,\mathbf{w}_+ - \mathbf{w}_-} \tag{12}$$

□

Theorem 2.2. *For MLE parameters* $\mathbf{w}_+^*$, $\mathbf{w}_-^*$, *and* $\mathbf{v}^*$, $\mathbf{v}^* = \mathbf{w}_+^* - \mathbf{w}_-^*$.

$L_q(\mathbf{w}_+^* - \mathbf{w}_-^*)$ must be a minimum for L_q. Otherwise, there would be some $\mathbf{u}_+, \mathbf{u}_- \neq \mathbf{w}_+^*, \mathbf{w}_-^*$ for which $L_q(\mathbf{w}_+^* - \mathbf{w}_-^*) > L_q(\mathbf{u}_+ - \mathbf{u}_-) \rightarrow L_p(\mathbf{w}_+^*, \mathbf{w}_-^*) > L_p(\mathbf{u}_+, \mathbf{u}_-)$, which contradicts that $\mathbf{w}_+^*, \mathbf{w}_-^*$ minimizes L_p. The strict convexity of L_q implies that $\mathbf{w}_+^* - \mathbf{w}_-^*$ is the unique minimum of L_q, so $\mathbf{v}^* = \mathbf{w}_+^* - \mathbf{w}_-^*$. □

Theorem 2.3. $\mathbf{v}^{(i)} = \mathbf{w}_+^{(i)} - \mathbf{w}_-^{(i)}$ *at every every step* i *of gradient descent on* L_q *and* L_p *if the learning rate for the* L_q *descent is twice the learning rate of the* L_p *descent.*

Proof is by induction on i. □

Theorem 2.4. $\mathbf{w}_+^*, \mathbf{w}_-^* \geq 0$

Suppose the objective $L_p(\mathbf{w}_+, \mathbf{w}_-)$ were minimized with the constraint that $\mathbf{w}_+, \mathbf{w}_- \geq 0$. The objective $L_p'(\mathbf{w}_+, \mathbf{w}_-, \lambda_+, \lambda_-)$ augmented with Lagrange multipliers $\lambda_+, \lambda_- \geq 0$ would be minimized for $\mathbf{w}_+^*, \mathbf{w}_-^*$ such that:

$$0 = \sum_d (y^d - p_{d,\mathbf{w}_+^*,\mathbf{w}_-^*})\mathbf{f}(x^d)_i - \lambda_{+,i} \tag{13}$$

$$0 = \sum_d (1 - y^d - (1 - p_{d,\mathbf{w}_+^*,\mathbf{w}_-^*}))\mathbf{f}(x^d)_i - \lambda_{-,i} \tag{14}$$

Adding these two equations gives $\lambda_{+,i} = -\lambda_{-,i}$, and this implies that $\lambda_+ = \lambda_- = 0$ (since Lagrange multipliers are non-negative). This means that the $\mathbf{w}_+, \mathbf{w}_- \geq 0$ condition does not constrain the minimum.

(See `http://people.duke.edu/~hpgavin/cee201/LagrangeMultipliers.pdf` for helpful Lagrange multiplier notes.)

□

2.2 With l2 regularization

MLE with l2 regularization for the above distributions is equivalent to finding parameters that minimize the following strictly convex loss functions:

$$L_{p2}(\mathbf{w}_+, \mathbf{w}_-) = L_p(\mathbf{w}_+, \mathbf{w}_-) + \beta\|\mathbf{w}_+\|_2^2 + \beta\|\mathbf{w}_-\|_2^2 \tag{15}$$

$$L_{q2}(\mathbf{v}) = L_q(\mathbf{v}) + \beta\|\mathbf{v}\|^2 \tag{16}$$

Let $\mathbf{w}^*_{+2}, \mathbf{w}^*_{-2} \geq 0$ be parameters that minimize 15 under a non-negativity constraint, and let $\mathbf{v}^*_2$ be parameters that minimize 16.

Lemma 2.1. *For all i, either $\mathbf{w}^*_{+2,i} = 0$ or $\mathbf{w}^*_{-2,i} = 0$*

Suppose for contradiction that for some i, $\mathbf{w}^*_{+2,i} > 0$ and $\mathbf{w}^*_{-2,i} > 0$.

Let $\mathbf{u}_{+,j} = \mathbf{w}^*_{+2,j}$ and $\mathbf{u}_{-,j} = \mathbf{w}^*_{-2,j}$ for $j \neq i$. If $\mathbf{w}^*_{+2,i} \geq \mathbf{w}^*_{-2,i}$, then and let $\mathbf{u}_{+,i} = \mathbf{w}^*_{+2,i} - \mathbf{w}^*_{-2,i}$ and $\mathbf{u}_{-,i} = 0$; otherwise, let $\mathbf{u}_{+,i} = 0$ and $\mathbf{u}_{-,i} = \mathbf{w}^*_{-2,i} - \mathbf{w}^*_{+2,i}$ (so that $\mathbf{u}_+, \mathbf{u}_- \geq 0$ in either case).

$$
\begin{aligned}
L_{p2}(\mathbf{w}^*_{+2}, \mathbf{w}^*_{-2}) &= L_p(\mathbf{w}^*_{+2}, \mathbf{w}^*_{+2}) + \beta\|\mathbf{w}^*_{+2}\|_2^2 + \beta\|\mathbf{w}^*_{-2}\|_2^2 && (17)\\
&= L_q(\mathbf{w}^*_{+2} - \mathbf{w}^*_{+2}) + \beta\|\mathbf{w}^*_{+2}\|_2^2 + \beta\|\mathbf{w}^*_{-2}\|_2^2 && (18)\\
&= L_q(\mathbf{u}_+ - \mathbf{u}_-) + \beta\|\mathbf{w}^*_{+2}\|_2^2 + \beta\|\mathbf{w}^*_{-2}\|_2^2 && (19)\\
&= L_p(\mathbf{u}_+, \mathbf{u}_-) + \beta\|\mathbf{w}^*_{+2}\|_2^2 + \beta\|\mathbf{w}^*_{-2}\|_2^2 && (20)\\
&= L_p(\mathbf{u}_+, \mathbf{u}_-) + \beta\Big(\sum_{j\neq i}\big(\mathbf{w}^{*2}_{+2,j} + \mathbf{w}^{*2}_{-2,j}\big) + \mathbf{w}^{*2}_{+2,i} + \mathbf{w}^{*2}_{-2,i}\Big) && (21)\\
&= L_p(\mathbf{u}_+, \mathbf{u}_-) + \beta\Big(\sum_{j\neq i}\big(\mathbf{u}^2_{+,j} + \mathbf{u}^2_{-,j}\big) + \mathbf{w}^{*2}_{+2,i} + \mathbf{w}^{*2}_{-2,i}\Big) && (22)\\
&> L_p(\mathbf{u}_+, \mathbf{u}_-) + \beta\Big(\sum_{j\neq i}\big(\mathbf{u}^2_{+,j} + \mathbf{u}^2_{-,j}\big) + \mathbf{w}^{*2}_{+2,i} + \mathbf{w}^{*2}_{-2,i} - 2\mathbf{w}^{*2}_{+2i}\mathbf{w}^{*2}_{-2i}\Big) && (23)\\
&= L_p(\mathbf{u}_+, \mathbf{u}_-) + \beta\Big(\sum_{j\neq i}\big(\mathbf{u}^2_{+,j} + \mathbf{u}^2_{-,j}\big) + \mathbf{u}^2_{+,i} + \mathbf{u}^2_{-,i}\Big) && (24)\\
&= L_p(\mathbf{u}_+, \mathbf{u}_-) + \beta\|\mathbf{u}_{+2}\|_2^2 + \beta\|\mathbf{u}_{-2}\|_2^2 && (25)\\
&= L_{p2}(\mathbf{u}_+, \mathbf{u}_-) && (26)
\end{aligned}
$$

This contradicts that $\mathbf{w}^*_{+2}, \mathbf{w}^*_{-2}$ minimizes 7 subject to non-negativity constraints.

(The intuition here is just that if both $\mathbf{w}^*_{+2,i}$ and $\mathbf{w}^*_{-2,i}$ were non-negative, a smaller minimum could be constructed which sets one of them to 0, and shifts its weight into the other.)

□

Lemma 2.2. *If $\mathbf{u} = \mathbf{u}_+ - \mathbf{u}_-$ and either $\mathbf{u}_{+,i} = 0$ or $\mathbf{u}_{-,i} = 0$ for all i, then $L_{q2}(\mathbf{u}) = L_{p2}(\mathbf{u}_+, \mathbf{u}_-)$.*

$$
\begin{aligned}
L_{q2}(u) &= L_q(\mathbf{u}) + \beta\|\mathbf{u}\|_2^2 && (27)\\
&= L_q(\mathbf{u}_+ - \mathbf{u}_-) + \beta\|\mathbf{u}_+ - \mathbf{u}_-\|_2^2 && (28)\\
&= L_p(\mathbf{u}_+, \mathbf{u}_-) + \beta\sum_i(\mathbf{u}_{+,i} - \mathbf{u}_{-,i})^2 && (29)\\
&= L_p(\mathbf{u}_+, \mathbf{u}_-) + \beta\sum_i \mathbf{u}^2_{+,i} + \beta\sum_i \mathbf{u}^2_{-,i} && (30)\\
&= L_p(\mathbf{u}_+, \mathbf{u}_-) + \beta\|\mathbf{u}_+\|_2^2 + \beta\|\mathbf{u}_-\|_2^2 && (31)\\
&= L_{p2}(\mathbf{u}_+, \mathbf{u}_-) && (32)
\end{aligned}
$$

□

Theorem 2.5. *For all i, $\mathbf{w}^*_{+2,i} = \max(0, \mathbf{v}^*_{2,i})$ and $\mathbf{w}^*_{-2,i} = \max(0, -\mathbf{v}^*_{2,i})$. And as a result, $\mathbf{v}^*_2 = \mathbf{w}^*_{+2} - \mathbf{w}^*_{-2}$.*

Let $\mathbf{v}^*_{+2}$ and $\mathbf{v}^*_{-2}$ be defined so that $\mathbf{v}^*_{+2,i} = \max(0, \mathbf{v}^*_{2,i})$ and $\mathbf{v}^*_{-2,i} = \max(0, -\mathbf{v}^*_{2,i})$. Lemma 2.2 implies that $L_{q2}(\mathbf{v}^*_2) = L_{p2}(\mathbf{v}^*_{+2}, \mathbf{v}^*_{-2})$. Furthermore, Lemma 2.2 and Lemma 2.1 together imply that $L_{q2}(\mathbf{w}^*_{+2} - \mathbf{w}^*_{-2}) = L_{p2}(\mathbf{w}^*_{+2}, \mathbf{w}^*_{-2})$.

It must be the case that $\mathbf{v}^*_{+2}, \mathbf{v}^*_{-2} = \mathbf{w}^*_{+2}, \mathbf{w}^*_{-2}$. Otherwise since $\mathbf{w}^*_{+2}, \mathbf{w}^*_{-2}$ is the unique minimum of L_{p2},

$$\begin{aligned} L_{q2}(\mathbf{v}^*_2) &= L_{p2}(\mathbf{v}^*_{+2}, \mathbf{v}^*_{-2}) && (33)\\ &> L_{p2}(\mathbf{w}^*_{+2}, \mathbf{w}^*_{-2}) && (34)\\ &= L_{q2}(\mathbf{w}^*_{+2} - \mathbf{w}^*_{-2}) && (35)\\ & && (36) \end{aligned}$$

And this would contradict that $\mathbf{v}^*_2$ minimizes L_{q2}. $\square$

3 Some properties of logistmar gramression

In this section, we will prove several properties of the distributions defined by Equation 1 with parameters $\mathbf{u}$ obtained from logistmar gramression through the minimization of Equation 3. For all of these proofs, assume the l2 regularization parameter β and the threshold t used in the definition of $\mathbf{c}$ (see Equation 4) are fixed. Also, assume the following definitions:

- $\mathcal{M}^F$ is the l2 regularized logistic regression model defined by minimization of Equation 15 with non-negative constrained parameters and feature vocabulary F. $\mathcal{M}^F_+(f)$ and $\mathcal{M}^F_-(f)$ give the values of the $\mathbf{w}_{+,f}$ and $\mathbf{w}_{-,f}$ parameters in $\mathcal{M}^F$. $\mathcal{M}^F(f)$ gives the overall magnitude of the weight of the feature as $\max\left(\mathcal{M}^F_+(f), \mathcal{M}^F_-(f)\right)$ (see Theorem 2.5 for why this $\max$ gives the overall magnitude).
- $\mathcal{M}^{F_0,H}$ is logistmar gramression defined by minimization of Equation 3 with non-negative constrained parameters, an initial feature vocabulary F_0, and heuristics H. Let $\mathbf{u}^*$ be the parameters found by this model. Then $\mathcal{M}^{F_0,H}_+(f)$ and $\mathcal{M}^{F_0,H}_-(f)$ give the values of the $c_{+,f}(\mathbf{u}^*)$ and $c_{-,f}(\mathbf{u}^*)$ parameters in $\mathcal{M}^{F_0,H}$. $\mathcal{M}^{F_0,H}(f)$ gives the overall magnitude of the weight of the feature $\max\left(\mathcal{M}^{F_0,H}_+(f), \mathcal{M}^{F_0,H}_-(f)\right)$.
- $F^+(\mathcal{M})$ is the set of features f that model $\mathcal{M}$ gives non-zero weight (i.e. the set of features f for which $\mathcal{M}(f) > 0$).
- Let $F \subset F^\infty$. A feature f is F-relevant iff $\exists F' \subset F^\infty : \mathcal{M}^{F \cup F' \cup \{f\}}(f) > t$. A feature F is strongly-F-relevant iff $\mathcal{M}^{F \cup \{f\}}(f) > t$.
- $R_F = \{f | f \text{ is } F\text{-relevant}\}$, and $R_{H(F)} = \{h(f) | h \in H, f \in R_F\}$.
- The heuristics H are F-relevance complete iff $\forall f \in F : f$ is strongly-F-relevant $\Rightarrow \forall f' \in H^{-1}(f) : f'$ is strongly-F-relevant.[4]
- The heuristics H are F-relevance preserving iff $\forall f \in F^\infty : f$ is F-relevant $\Rightarrow \forall F \subseteq S \subseteq (F \cup R_{H(F)}) : f$ is strongly-S-relevant.

Theorem 3.1. *The size of $F^+(\mathcal{M}^{F_0,H})$ is finite.*

[4] I'm not sure whether it will be reasonable to expect heuristics to have this property, but it is necessary for several of the proofs given below. I think logistmar gramression will produce posteriors that are still related to logistic regression posteriors even if this property is weakened, but the proofs of those relationships will be more difficult.

Assume for contradiction that $F^+(\mathcal{M}^{F_0,H})$ is infinite, so that there are an infinite number of features f for which $\mathcal{M}^{F_0,H}(f) > 0$. By the definition of $\mathbf{c}$ in Equation 4, we know:

$$\forall f \notin F_0 : \mathcal{M}^{F_0,H}(f) > 0 \Rightarrow \forall f' \in H^{-1}(f) : \mathcal{M}^{F_0,H}(f') > t \tag{37}$$

From this, it follows that there are an infinite number of features f for which $\mathcal{M}^{F_0,H}(f) > t$. This means there are an infinite number of terms in $\mathbf{c}(\mathbf{u}^*)$ that are greater than t, and the remaining are non-negative. So, the regularization term in $L_{r2}(\mathbf{u}^*)$ is $+\infty$, which implies that $L_{r2}(\mathbf{u}^*)$ is either $+\infty$ or undefined. So for $\mathbf{u}^*$ to be the parameter found by logistmar gramression, there can be no $\mathbf{u}$ for which $L_{r2}(\mathbf{u})$ is finite. But $L_{r2}(\mathbf{0})$ is finite, which is a contradiction. $\square$

Lemma 3.1. *Assume $f : S \to T$ has a unique minimum and $g : R \to S$ is surjective. If $s^* = \operatorname{argmin}_s f(s)$ and $r^* = \operatorname{argmin}_r f(g(r))$, then $g(r^*) = s^*$.*

Proof is in the pudding. (I'll write this up later if necessary, but I'm pretty sure it's good.) $\square$

Theorem 3.2. *Let the size of S be finite with $F^+(\mathcal{M}^{F_0,H}) \subseteq S \subset F^\infty$, and assume H is S-relevance complete. Then, $\forall f \in S : \mathcal{M}_+^{F_0,H}(f) = \mathcal{M}_+^S(f)$ and $\mathcal{M}_-^{F_0,H}(f) = \mathcal{M}_-^S(f)$, and so the posterior distributions given by $\mathcal{M}^{F_0,H}$ and $\mathcal{M}^S$ are the same. Furthermore, by Theorem 2.5, logistmar gramression outputs the same posterior distributions as the standard logistic regression that minimizes Equation 16 without non-negativity constraints under feature vocabulary S.*

This follows from Lemma 3.1 and Theorem 3.1. To see this, let the function f from Lemma 3.1 be L_{q2} with finite feature vocabulary $S \supseteq F^+(\mathcal{M}^{F_0,H})$, and assume H is S-relevance complete. Note that Theorem 3.1 guarantees that S can be finite. Let the g function from Lemma 3.1 be defined to take an infinite length vector $\mathbf{u}$, and output the vector of all elements of $\mathbf{c}(\mathbf{u})$ that have corresponding features in S. By the definition of $\mathbf{c}$, g can produce a vector $(\mathbf{w}_+, \mathbf{w}_-)$ if and only if for all $f \in S$, $\mathbf{w}_{+,f} > t$ or $\mathbf{w}_{-,f} > t$ implies $\mathbf{w}_{+,f'} > t$ or $\mathbf{w}_{-,f'} > t$ for any $f' \in H^{-1}(f)$. Let the set T from Lemma 3.1 be the set of all such vectors, so that g is surjective with respect to T. Since H is S-relevance complete, the minimum of L_{q2} over the full range of possible vector inputs will be in T, and so we can restrict the domain of L_{q2} to T without a change in the parameters found by logistic regression in minimizing L_{q2}.

By the definitions of f and g, $\mathbf{u}^* = \operatorname{argmin}_{\mathbf{u}} L_{r2}(\mathbf{u}) = \operatorname{argmin}_{\mathbf{u}} f(g(\mathbf{u}))$ since the minimum point of L_{r2} will be preserved if we remove terms in the sums of L_{r2} that are zero at the minimum point. By Lemma 3.1, if $\mathbf{w}_+^*, \mathbf{w}_-^* = \operatorname{argmin}_{\mathbf{w}_+,\mathbf{w}_-} f(\mathbf{w}_+, \mathbf{w}_-)$, then $\mathbf{w}_+^*, \mathbf{w}_-^* = g(\mathbf{u}^*)$. $\square$

Theorem 3.3. *Assume H is $\left(F_0 \cup F^+(\mathcal{M}^{F_0,H})\right)$-relevance complete. Then, $F^+(\mathcal{M}^{F_0,H}) \subseteq F_0 \cup R_{H(F_0)}$*

Let $f \in F^+(\mathcal{M}^{F_0,H})$. Suppose $f \notin F_0$. Then by the definition of $\mathbf{c}$, there is some $f' \in H^{-1}(f)$ (meaning $h(f') = f$), for which $\mathcal{M}^{F_0,H}(f') > t$. By Theorem 3.2, since H is $\left(F_0 \cup F^+(\mathcal{M}^{F_0,H})\right)$-relevance complete, this implies $\mathcal{M}^{F_0 \cup F^+(\mathcal{M}^{F_0,H})}(f') > t$ which means that f' is F_0-relevant, and therefore $f \in R_{H(F_0)}$. So $f \in F_0$ or $f \in R_{H(F_0)}$. $\square$

Theorem 3.4. *Let H be $\left(F_0 \cup F^+(\mathcal{M}^{F_0,H})\right)$-relevance complete and F_0-relevance preserving. Then $R_{F_0} \subseteq F^+(\mathcal{M}^{F_0,H})$.*

Assume H is $\left(F_0 \cup F^+(\mathcal{M}^{F_0,H})\right)$-relevance complete and F_0-relevance preserving. Let $f \in R_{F_0}$. By Theorem 3.3, $F_0 \cup F^+(\mathcal{M}^{F_0,H}) \subseteq F_0 \cup R_{H(F_0)}$, and since H is F_0-relevance preserving, f is strongly-$\left(F_0 \cup F^+(\mathcal{M}^{F_0,H})\right)$-relevant. This means that $\mathcal{M}^{\left(F_0 \cup F^+(\mathcal{M}^{F_0,H})\right) \cup \{f\}}(f) > t$. By Theorem 3.2, it follows that $f \in F^+(\mathcal{M}^{F_0,H})$. $\square$

4 A possible grammar

In general, we can define a context free feature grammar $G = (V, \Sigma, R, S)$ by:

- V is a set of non-terminal functions.
- Σ is a set of all possible feature functions $f : X \to [0, 1]$.
- R is a set of production rules. For all functions $T : \mathcal{S} \to \mathcal{T}$ and $U : \mathcal{T} \to \mathcal{U}$ in V, there is a relation $T \to (U \circ T)$ in R.
- S is some function $I \in V$ that has domain X.

For the noun-phrase mention categorization task, let $\mathcal{X}$ be the set of all locations in documents, let $\mathcal{S}$ be the set of all strings over the documents' alphabets, let $\mathcal{P}$ be the set of all part-of-speech tag sequences, and let $\mathcal{F}_{S,T}$ be the set of all functions with domain S and target T. Assume that all sets have some well-defined ordering, and S_i gives the element at index i in set S. Then, the start symbol and non-terminal functions of the grammar are defined as:

- Start symbol S is a function $I : \mathcal{X} \to (2^{\mathcal{X}} \times \mathcal{F}_{\mathcal{X},2^{\mathcal{X}}})$ defined as $I(x) = (I'(x), I')$. The I' is an internal function defined in Table 1.
- Symbols other than the start symbol in V are either ending functions or non-ending functions. Ending functions output a terminal symbol, whereas non-ending functions do not. Most non-ending functions $N : (T \times \mathcal{F}_{S,T}) \to (U \times \mathcal{F}_{S,U})$, are defined in terms of an 'internal' non-ending function N' as $N(t, F) = (N'(t), N' \circ F)$ (see Table 1). The only exception is the X training data application function defined as $X(t, F) = (\bigcup_{x \in D} F(x), F)$. Notice that the 'internal' symbol functions are necessary in order to include higher order functions like X in the grammar. For this example grammar, the only ending functions will be the set containment indicators $E_i : (2^T \times \mathcal{F}_{X,2^T}) \to \mathcal{F}_{X,[1,0]}$ defined as $E_i(t, F) = f_i$ such that $f_i(x) = \mathbf{1}(t_i \in F(x))$.

Non-terminal internal functions	**Definition**
$I' : \mathcal{X} \to 2^{\mathcal{X}}$	$I'(x) = \{x\}$
$S' : 2^{\mathcal{X}} \to 2^{\mathcal{S}}$	Outputs strings at given locations
$P' : 2^{\mathcal{X}} \to 2^{\mathcal{P}}$	Outputs part-of-speech sequences at given locations
$Pr'_n : 2^{\mathcal{X}} \to 2^{\mathcal{S}}$	Outputs sequences of length n prefixes of tokens at given locations
$Su'_n : 2^{\mathcal{X}} \to 2^{\mathcal{S}}$	Outputs sequences of length n suffixes of tokens at given locations
$M'_n : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$	Outputs n-gram locations contained in given locations
$S'_n : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$	Outputs n-gram locations contained in sentences of given locations
$D'_n : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$	Outputs n-gram locations contained in document of given locations
$C'_{b,n} : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$	Outputs n-gram locations immediately before given locations
$C'_{a,n} : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$	Outputs n-gram locations immediately after given locations
$D'_{p,t,n} : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$	Outputs n-gram locations that are dependency parents of given locations
$D'_{c,t,n} : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$	Outputs n-gram locations that are dependency children of given locations
$F'_{\mathcal{S},p} : \mathcal{S} \to 2^{\mathcal{X}} \to 2^{\mathcal{X}}$	Filters locations to those that are prefixed by a given string
$F'_{\mathcal{S},s} : \mathcal{S} \to 2^{\mathcal{X}} \to 2^{\mathcal{X}}$	Filters locations to those that are suffixed by a given string
$F'_{\mathcal{P},p} : \mathcal{P} \to 2^{\mathcal{X}} \to 2^{\mathcal{X}}$	Filters locations to those that are prefixed by a given part-of-speech tag sequence
$F'_{\mathcal{P},s} : \mathcal{P} \to 2^{\mathcal{X}} \to 2^{\mathcal{X}}$	Filters locations to those that are suffixed by a given part-of-speech tag sequence

Table 1: Feature grammar non-terminal non-ending internal function symbols. These 'internal' symbols are necessary to allow for higher order functions in the grammar.

Given this grammar, it's possible to construct some some interesting features. For example, $(E_i \circ X \circ S \circ C_{b,2} \circ D_{p,t_1,1} \circ D_{c,t_2,1} \circ I)(x)$ indicates whether the a bigram immediately before one of x's dependents' governors is the ith element of the ordered vocabulary of such bigrams computed over the training data.

An initial seed feature vocabulary F_0 and some possible heuristic functions H that logistmar gramression can use are given in Table 2 and Table 3.

Feature type	**Description**
$E_i \circ X \circ S \circ M_1 \circ I$	Unigrams in mention
$E_i \circ X \circ S \circ M_2 \circ I$	Bigrams in mention
$E_i \circ X \circ S \circ S_1 \circ I$	Unigrams in sentence
$E_i \circ X \circ S \circ D_{c,t,1} \circ I$	Unigram dependents
$E_i \circ X \circ S \circ D_{p,t,1} \circ I$	Unigram governors
$E_i \circ X \circ S \circ C_{b,1} \circ I$	Unigram before
$E_i \circ X \circ S \circ C_{a,1} \circ I$	Unigram after
$E_i \circ X \circ P \circ C_{b,1} \circ I$	Part-of-speech before
$E_i \circ X \circ P \circ C_{a,1} \circ I$	Part-of-speech after
$E_i \circ X \circ Pr_3 \circ M_1 \circ I$	Length 3 prefixes in mention
$E_i \circ X \circ Su_3 \circ M_1 \circ I$	Length 3 suffixes in mention

Table 2: Initial seed vocabulary for logistmar gramression.

Heuristic
n-grams in sentence to $(n+1)$-grams in sentence $(E_i \circ X \circ S \circ F_{\mathbb{S},p}(s) \circ S_n \circ I) \rightarrow (E_j \circ X \circ S \circ F_{\mathbb{S},p}((X \circ S \circ F_{\mathbb{S},p}(s) \circ S_n \circ I)_{1,i}) \circ S_{n+1} \circ I)$ $(E_i \circ X \circ S \circ F_{\mathbb{S},s}(s) \circ S_n \circ I) \rightarrow (E_j \circ X \circ S \circ F_{\mathbb{S},s}((X \circ S \circ F_{\mathbb{S},s}(s) \circ S_n \circ I)_{1,i}) \circ S_{n+1} \circ I)$
n-grams in sentence to n-grams in document $(E_i \circ X \circ S \circ F_{\mathbb{S},p}(s) \circ S_n \circ I) \rightarrow (E_j \circ X \circ S \circ F_{\mathbb{S},p}((X \circ S \circ F_{\mathbb{S},p}(s) \circ S_n \circ I)_{1,i}) \circ D_n \circ I)$ $(E_i \circ X \circ S \circ F_{\mathbb{S},s}(s) \circ S_n \circ I) \rightarrow (E_j \circ X \circ S \circ F_{\mathbb{S},s}((X \circ S \circ F_{\mathbb{S},s}(s) \circ S_n \circ I)_{1,i}) \circ D_n \circ I)$
n-grams in document to $(n+1)$-grams in document $(E_i \circ X \circ S \circ F_{\mathbb{S},p}(s) \circ D_n \circ I) \rightarrow (E_j \circ X \circ S \circ F_{\mathbb{S},p}((X \circ S \circ F_{\mathbb{S},p}(s) \circ D_n \circ I)_{1,i}) \circ D_{n+1} \circ I)$ $(E_i \circ X \circ S \circ F_{\mathbb{S},s}(s) \circ D_n \circ I) \rightarrow (E_j \circ X \circ S \circ F_{\mathbb{S},s}((X \circ S \circ F_{\mathbb{S},s}(s) \circ D_n \circ I)_{1,i}) \circ D_{n+1} \circ I)$
n-grams before to $(n+1)$-grams before $(E_i \circ X \circ S \circ F_{\mathbb{S},s}(s) \circ C_{b,n} \circ I) \rightarrow (E_j \circ X \circ S \circ F_{\mathbb{S},s}((X \circ S \circ F_{\mathbb{S},s}(s) \circ C_{b,n} \circ I)_{1,i}) \circ C_{b,n+1} \circ I)$
n-grams after to $(n+1)$-grams after $(E_i \circ X \circ S \circ F_{\mathbb{S},p}(s) \circ C_{a,n} \circ I) \rightarrow (E_j \circ X \circ S \circ F_{\mathbb{S},p}((X \circ S \circ F_{\mathbb{S},p}(s) \circ C_{a,n} \circ I)_{1,i}) \circ C_{a,n+1} \circ I)$
n part-of-speech before to $(n+1)$ part-of-speech before $(E_i \circ X \circ S \circ F_{\mathcal{P},s}(s) \circ C_{b,n} \circ I) \rightarrow (E_j \circ X \circ S \circ F_{\mathcal{P},s}((X \circ S \circ F_{\mathcal{P},s}(s) \circ C_{b,n} \circ I)_{1,i}) \circ C_{b,n+1} \circ I)$
n part-of-speech after to $(n+1)$ part-of-speech after $(E_i \circ X \circ S \circ F_{\mathcal{P},p}(s) \circ C_{a,n} \circ I) \rightarrow (E_j \circ X \circ S \circ F_{\mathcal{P},p}((X \circ S \circ F_{\mathcal{P},p}(s) \circ C_{a,n} \circ I)_{1,i}) \circ C_{a,n+1} \circ I)$
Length n prefixes in mention to length $(n+1)$ prefixes in mention $(E_i \circ X \circ Pr_n \circ F_{\mathcal{P},p}(s) \circ M_1 \circ I) \rightarrow (E_j \circ X \circ Pr_{n+1} \circ F_{\mathcal{P},p}((X \circ Pr_n \circ F_{\mathcal{P},p}(s) \circ M_1 \circ I)_{1,i}) \circ M_1 \circ I)$
Length n suffixes in mention to length $(n+1)$ suffixes in mention $(E_i \circ X \circ Su_n \circ F_{\mathcal{P},s}(s) \circ M_1 \circ I) \rightarrow (E_j \circ X \circ Su_{n+1} \circ F_{\mathcal{P},s}((X \circ Su_n \circ F_{\mathcal{P},s}(s) \circ M_1 \circ I)_{1,i}) \circ M_1 \circ I)$

Table 3: Heuristic functions for logistmar gramression. Note that $F_{\mathbb{S},}$ applied to the empty string is the identity function, and so the left hand sides of the above rules will match the initial features shown in Table 2 for s as the empty string.