COMP 652: ASSIGNMENT 1

CARLOS G. OLIVER

260425853

1. Regression

1.1. Q1 (c):Objective for logistic regression with L_2 regulariztion.
The logistic regression objective is the cross entropy error function between true outputs y_i and predictions $h(x_i)$ with the additional regularization term proportional to some constant λ and the L_2 norm of the weight vector $\mathbf{w}$.

$$(1) \qquad J_{\mathbf{w}} = -\left(\sum_{i=1}^{m} y_i \log(h(\mathbf{x_i})) + (1 - y_i)\log(1 - h(\mathbf{x_i})) \right) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

1.2. Q1 (d): Regularization.
For all subsequent experiments, I present summary statistics of K-fold cross validation with shuffled input data. We compute a value of K so as to have approximately a training-test split ratio of $80 - 20\%$. This resulted in 4 splits. The bars one each point represent one standard deviation from the mean over the 4 splits.

1.3. Q1 (e): Gaussian Basis Functions.
Next, we apply a univariate Gaussian basis function $\phi_j(x_i)$ to each value in input vectors x to compute a new feature mapping. Where

$$(2) \qquad \phi(x)_j = exp\left[\frac{(x_i - \mu_k)^2}{2\sigma^2} \right]$$
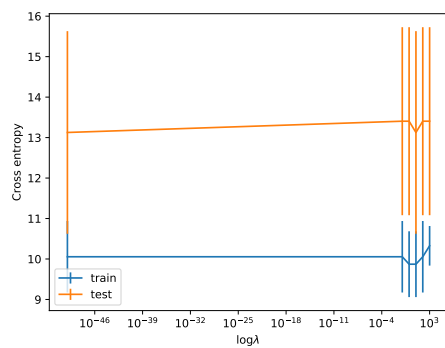
We apply this basis with a fixed σ chosen from the set $\{0.1, 0.5, 1, 5, 10\}$ at each iteration. For each variable we compute 5 basis functions $(\phi_1(x), ...\phi_5(x))$ with $\mu \in \{-10, -5, 0, 5, 10\}$ resulting in 5 feature vectors for each original feature vector in X. This results in a new input matrix Φ.

1.4. Q1 (f): Effect of σ on regression.

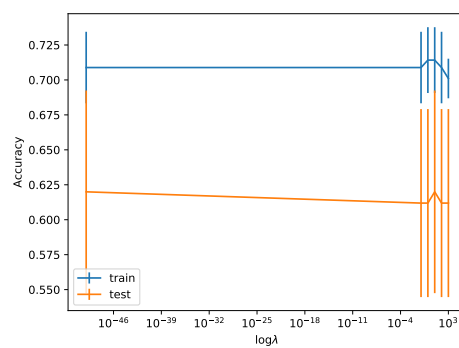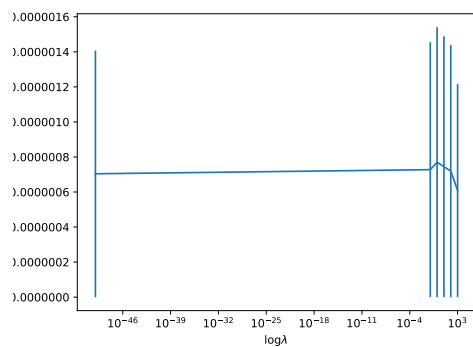### 1.5. Q1 (g): Basis function and regularization.
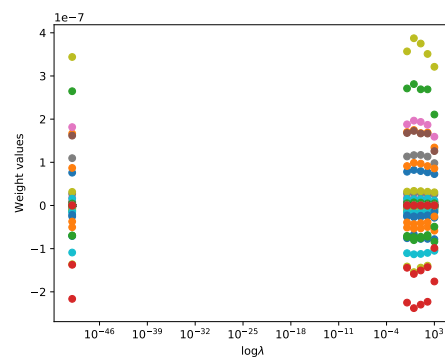

Date: February 5, 2017.
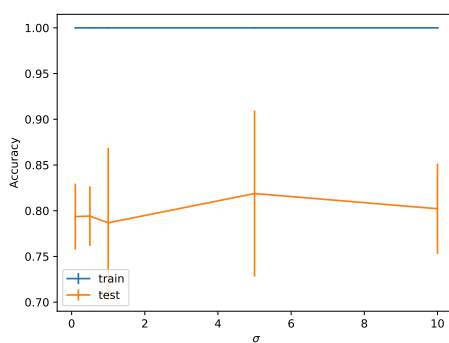
(A) MFE Sampling

(B) Suboptimal Sampling

(C) Suboptimal Sampling

(D) Suboptimal Sampling

FIGURE 1. Frequency of sequence-structure pair in Boltzmann ensemble for MFE and suboptimal sampling in RNAmutants.

1.6. Q1 (h): Gaussian basis function capturing relationship between inputs.
Instead of using a univariate Gaussian basis function to transform each input variable, we
could use multivariate Gaussians to capture the relationship between inputs as covariance.
We can use a similar basis as before, but introduce a covariance parameter matrix Σ that
can be estimated from the data. As we would be increasing the complexity of our model,
we are likely to reduce the bias but make the model more sensitive to variations in the
data.

1.7. Q1 (i): Adaptive gaussian center placement. Previously, we have been exter-
nally fixing the centers μ_i of the Gaussian basis functions. We can instead use the data to
adaptively compute the placement of these centers. This would effectively be a clustering
task, where each cluster would define the center of a Gaussian basis. With a fixed σ, the
only parameter we need to estimate is the center vector μ for each center k. Our basis
functions would then take the form:

$$(3) \qquad \phi(\mathbf{x}) = exp\left(\frac{\|\mathbf{x} - \mu\|_2^2}{2\sigma_k^2} \right)$$

> **input** : X, λ, $\mathbf{y}$, $[\sigma_0, ..., \sigma_j]$
> **output:** ω
> $currentTestErr \leftarrow \infty$;
> $prevTestErr \leftarrow 0$;
> $bestMode \leftarrow Null$;
> **while** $currentTestErr > prevTestErr$ **do**
> > $\mu_1, ..., \mu_j \leftarrow$ KMeans(X, k);
> > $X' \leftarrow$ GaussianBasis$(X, [\mu_1, ..., \mu_k])$;
> > $model, \mathbf{w} \leftarrow$ LogisticRegression(X', y, λ);
> > $currentTestErr \leftarrow$ accuracy$(model, X, y)$;
>
> **end**

1.8. Q1 (j): Convergence of algorithm.

1.9. Q2 (a): Dual view of logistic regression. The hypothesis $h(\mathbf{x})$ in logistic regres-
sion takes the form:

$$(4) \qquad h(\mathbf{x}) = P(Y = 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + e^{\mathbf{w^T x}}}$$

If we wish to obtain the probability distribution of y condition on $\mathbf{x}$ and $\mathbf{w}$ we get:

$$(5) \qquad p(y|\mathbf{x}, \mathbf{w}) = \sigma(y\mathbf{w^T x}) = \frac{1}{1 + e^{-y\mathbf{w^T x}}}$$

We can then use this form to find the parameters that maximize the log likelihood of the data given a choice of model. (The following derivation is based on work published by Thomas P. Minka in 2007 [1].)

$$(6) \qquad l(\mathbf{w}) = \sum_{i=1}^{m} \log \sigma(-y_i \mathbf{w}^\mathbf{T} \mathbf{x}) - \frac{\lambda}{2} \mathbf{w}^\mathbf{T} \mathbf{w}$$

In order to derive a dual expression of this objective, we aim to find a tight linear upper bound parametrized by a new set of parameters α_i. Such a function will let us reverse the order of the optimization while maintaining the same optima. In our case, if we wish to maximize $l(\mathbf{w})$ we can instead minimize some function $l(\mathbf{w}, \alpha)$ that is an upper bound to the first. We can then find the minimal value of $l(\mathbf{w}, \alpha)$ that maximizes $l(\mathbf{w})$.

We use the function:

$$(7) \qquad H(\alpha) = -\alpha \log \alpha - (1 - \alpha) \log(1 - \alpha), \quad \alpha \in [0, 1]$$

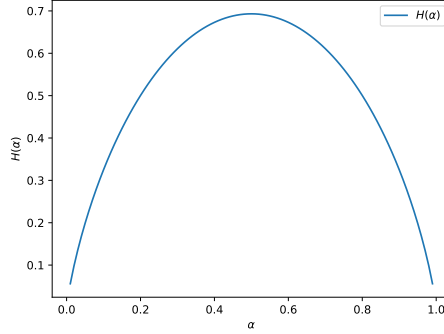Plotting $H(x)$ we see that the function is quadratic.

FIGURE 2. Parameter function

Using $H(x)$ we can bound $\log(\sigma(x))$ above as follows:

$$(8) \qquad \log \sigma(x) \leq \alpha x - H(x)$$

Plotting these functions with $\alpha = \{0, 0.5, 1\}$ we see that indeed, $\log \sigma(x)$ is tightly bounded above.

We can then apply these constraints to the original log likelihood function optimization problem:

$$(9) \qquad l(\mathbf{w}) = \sum_{i}^{m} \log \sigma(y_i \mathbf{w}^\mathbf{T} \mathbf{x}) - \frac{\lambda}{2} \mathbf{w}^\mathbf{T} \mathbf{w} \leq l(\mathbf{w}, \alpha)$$
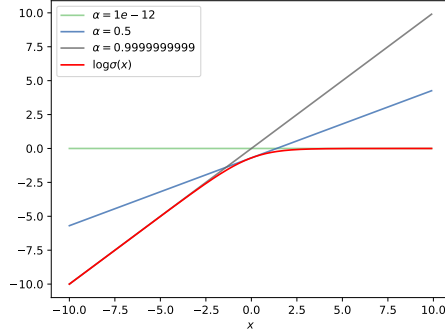
FIGURE 3. Upper bounds on log likelihood function

$$\text{(10)} \qquad \text{Where} \quad l(\mathbf{w}, \alpha) = \sum_i^m \alpha_i y_i \mathbf{w}^\mathbf{T} \mathbf{x_i} - H(\alpha_i) - \frac{\lambda}{2} \mathbf{w}^\mathbf{T} \mathbf{w}$$

We can now write the problem as a minimization problem over α

$$\text{(11)} \qquad \min_\alpha \max_\mathbf{w} l(\mathbf{w}, \alpha)$$

Taking the derivative of $l(\mathbf{w}, \alpha)$ with respect to $\mathbf{w}$ and setting to zero we get an expression for $\mathbf{w}$

$$\text{(12)} \qquad \mathbf{w}(\alpha) = -\lambda \sum_i^m \alpha_i y_i \mathbf{x_i}$$

Which we can plug into the original $l(\mathbf{w}, \alpha)$ and now optimize with respect to α to obtain

$$\text{(13)} \qquad J(\alpha) = \frac{1}{2\lambda} \sum_{i,j}^{m,n} \alpha_i \alpha_j y_i y_j \mathbf{x_j^T} \cdot \mathbf{x_i} - \sum_i^m H(\alpha_i)$$

This is the dual view of logistic regression, where the objective is expressed in terms of weighted dot products between input instances and not in terms of weight vector $\mathbf{w}$. In this form, we can apply any valid kernel to the dot product $\mathbf{x_i^T} \cdot \mathbf{x_j}$ as $K(\mathbf{x_i}, \mathbf{x_j})$. It is important to note that $\alpha \in \mathbb{R}^m$ where m is the number of input instances, and $\mathbf{w} \in \mathbb{R}^n$ where n is the number of features. When using kernel methods n can grow exponentially, making the dual form more computationally tractable when the number of instances is low compared to the number of features.

In order to solve for the optimal α Minka proposes to use coordinate wise Newton method instead of Newton's method to avoid inverting the Hessian.

$$(14) \qquad g_i = \frac{dJ(\alpha)}{d\alpha_i} = y_i \mathbf{w}(\alpha)^T x_i + \log\left(\frac{\alpha}{1 - \alpha_i}\right)$$

Using the first derivative we can iteratively step in the direction of the gradient over each α_i.

$$(15) \qquad \alpha_i^{new} = \alpha_i - \frac{g_i}{\lambda^{-1}\mathbf{x_i}^T\mathbf{x_i} + \frac{1}{\alpha_i(1 - \alpha_i)}}$$

Having obtained the vector α through this procedure, we can make predictions using the original logistic model with a kernel.

$$(16) \qquad P(Y = 1 | x, \alpha) = \sigma\left(\sum_i^m \alpha_i K(\mathbf{x}, \mathbf{x_i})\right)$$

1.10. **Q2 (b): Implementation of kernelized logistic regression.**

1.11. **Q2 (c): Kernelized logistic regression vs. non-kernel logistic regression results.**

1.12. **Q2 (d): Advantages and disadvantages of kernelized logistic regression.**

1.13. **Q3: (a) Kernels.** *Statement:* $K(\mathbf{x}, \mathbf{z}) = aK_1(\mathbf{x}, \mathbf{z}) + bK_2(\mathbf{x}, \mathbf{z})$ is a kernel given that K_1 and K_2 are kernels and $a, b \in \mathbb{R}$ and $a, b > 0$.

Proof. If K_1 and K_2 are kernels then we can write them as:

$$(17) \qquad K_1(\mathbf{x}, \mathbf{z}) = \phi^1(\mathbf{x})^T \phi^1(\mathbf{z})$$

$$(18) \qquad K_2(\mathbf{x}, \mathbf{z}) = \phi^2(\mathbf{x})^T \phi^2(\mathbf{z})$$

We define two new feature mappings $\phi^a(\mathbf{x})$ and $\phi^b(\mathbf{x})$ which are scalar products of the original $\phi(\mathbf{x})$.

$$(19) \qquad \phi^a(\mathbf{x}) = (\sqrt{a}x_1, \sqrt{a}x_2 ... \sqrt{a}x_n)$$

$$(20) \qquad \phi^b(\mathbf{x}) = (\sqrt{b}x_1, \sqrt{b}x_2 ... \sqrt{b}x_n)$$

Taking their dot products we can construct a new kernel $K^a(\mathbf{x}, \mathbf{z})$

$$(21) \quad K^a(\mathbf{x}, \mathbf{z}) = \phi^a(\mathbf{x})^T \cdot \phi^a(\mathbf{z}) = (\sqrt{a}\sqrt{a}x_1 z_1 + \sqrt{a}\sqrt{a}x_2 z_2 + ...\sqrt{a}\sqrt{a}x_n z_n) = aK^2(\mathbf{x}, \mathbf{z})$$

We can do the same for $\phi^b(\mathbf{x})$.

By conctatenating feature maps $\phi^a(\mathbf{x})$ and $\phi^b(\mathbf{x})$ into $\phi(\mathbf{x})$ we can express $aK_1 + bK_2$ as a dot product of this new feature map.

(22)
$$\phi(\mathbf{x})^T \cdot \phi(\mathbf{z}) = (\sqrt{a}\phi^a(x_1)\sqrt{a}\phi^a(x_1) + \sqrt{a}\phi^a(x_1)\sqrt{a}\phi^a(x_1) + ... + \sqrt{b}\phi^b(x_1)\sqrt{b}\phi^b(x_1) + \sqrt{b}\phi^b(x_1)\sqrt{b}\phi^b(x_1) + ...)$$

Which reduces to

(23)
$$K(\mathbf{x}, \mathbf{z}) = \phi^a(\mathbf{z})^T \cdot \phi^b(\mathbf{z})$$

$\square$

1.14. **Q3 (b) Kernels.** *Statement:* $K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z})K_2(\mathbf{x}, \mathbf{z})$

Proof. We can expand each kernel into dot products of their respective feature vectors ϕ^1 and ϕ^2

(24)
$$K(\mathbf{x}, \mathbf{z}) = \phi^1(\mathbf{x}) \cdot \phi^2(\mathbf{z})$$

$\square$

References

[1] Thomas P Minka. A comparison of numerical optimizers for logistic regression. *Unpublished draft*, 2003.