MLPR Exam Project: Gender Detection

Mattia Rosso [s294711]

July 26, 2022

This project is inteded to show a binary classification task on a datased made of 12 continuous observations coming from speking embeddings. A speaker embedding represents a smal-dimensional, fixed size representation of an utterance. Features can be seen as points in the m-dimensional embedding space (and the embeddings have already been computed). This is a task where classes are balanced both in training and evaluation set

1 Dataset analysis

1.1 Training and evaluation sets

The training set contains:

- Training Set: 3000 samples belonging to Male class (Label = 0) and 3000 samples belonging to Female class (Label = 1).
- Evaluation Set: 2000 samples belonging to Male class (Label = 0) and 2000 samples belonging to Female class (Label = 1).

1.2 Training set features analysis

1.3 Features Statistics

All the features are contiguous and their main statics can be showed through a boxplot in figure 1.

1.4 Z-normalization

A useful operation that can be applied in order to avoid to deal with numerical issues and to make data more uniform is to apply Z-normalization as a preprocessing step:

$$z_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j} \forall x_i \in D$$

Where $z_{i,j}$ is the Z-normalized value correspongin to the feature j of sample i while μ_j and σ_j are, respectively, the mean and the variance computed over all values for feature j.

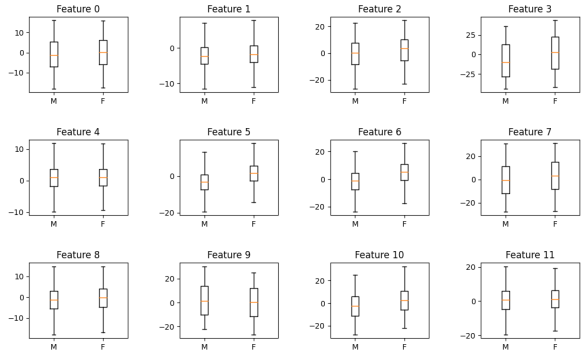

Figure 1: *Features boxplot*

1.5 Features distribution

By plotting historams for each feature, separated for male and female, it is possible to show how much each feature follows a gaussian distribution in order to understend whether a pre processing like gaussianization can be useful to go on with our classification task

We can notice from figure 2 that almost all the features are already well-distributed except for features 3, 7, 9. We can apply an additional pre-processing step in order to make all the features following a gaussian distribution.

1.5.1 Gaussianization

Gaussianization is a pre processing step that maps each fature to values whose empirical cumulative

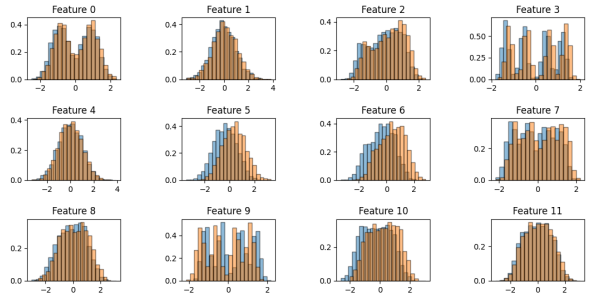

Figure 2: *Z-normalized features distribution*

distribution is well approximated by a Gaussian cumulative distribution function. For each feature x that we want to gaussianize we firstly compute the rank over the dataset:

$$r(x) = \frac{\sum_{i=1}^{N} \mathbb{I}[x_i < x] + 1}{N+2}$$

where $\mathbb{I}$ is the indicator function (1 when the condition inside [] is true, 0 otherwise). Actually, we are counting how many samples in the dataset D have a greater value with respect to the feature we are computing the rank on.
The next step is to compute the transformed feature as $y = \Phi^{-1}(r(x))$ where Φ is the inverse of the cumulative distribution function.

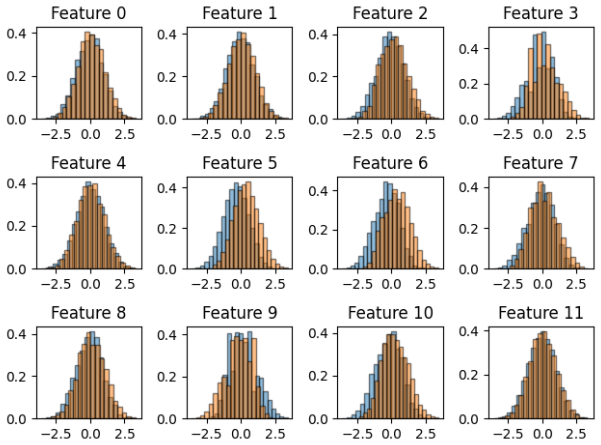

Figure 3: *Gaussianized features distribution*

1.6 Features correlation

We can show how much features are correlated by using a heatmap plot showing a darker color inside cells $[i, j]$ for which it exists an high correlation among feaure i and feature j. We are going to use the Pearson correlation coefficient to compute the correlation among feature X and feature Y:

$$|\frac{Cov(X,Y)}{\sqrt{Var(X)}\sqrt{Var(Y)}}|$$

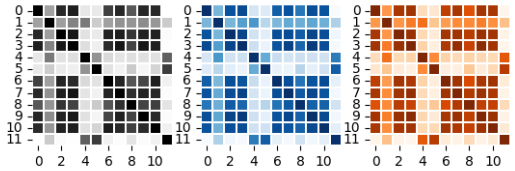

Figure 4: *Z-normalized+Gaussianized features correlation: grey the whole dataset, orange F class, blue M class*

Correlation is quite high among most of the features, we can understand that applying PCA would be meaningful for values of m not below 10 or 9 at most. For smaller values of m we would lose important information coming from high-correlated features.

2 Dimensionality reduction

Before proceeding with the classification task i would spend some words on the possible dimensioanlity reduction rechniques that can be applied: PCA, LDA.

2.1 PCA

As already anticipated, given the heatmap in figure 4 we can observe that PCA with reasonable values of m can be applied. PCA is a dimensionality reduction technique that, given a centered dataset $X = x_1, ..., x_k$, it aims to find the subspace of $\mathbb{R}^n$ that allows to preserve most of the information (the directions with the highest variance).
Starting from the sample covariance matrix

$$C = \frac{1}{K} \sum_i (x_i - \bar{x})(x_i - \bar{x})^T$$

we compute the eigen-decomposition of $C = U \Sigma U^T$ and project the data in the subspace spanned by the m columns of U corresponding to the m highest eigenvalues:

$$y_i = P^T (x_i - \bar{x})$$

In order to select the optimal m we can use a cross-validation approach by inspecting how much of the total variance of data we are able to retain by using that value of m. We exploit the fact that each eigenvalue correspond to the variance along the corresponding axis and the eigenvalues are the elements of the diagonal of the matrix Σ. We selecte m as:

$$\min_m \; s.t \frac{\sum_i^m \sigma_i}{\sum_i^n \sigma_i} >= t, \; t \geq 95\%$$

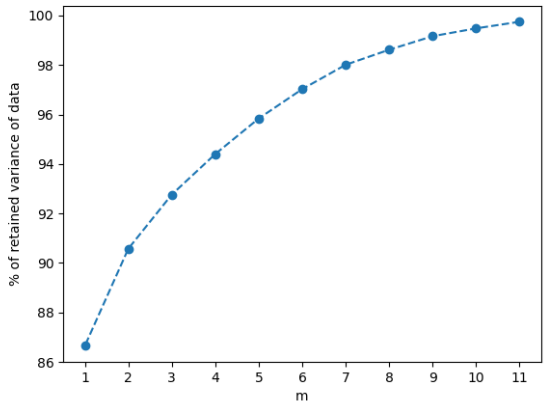

Figure 5: *3-fold cross validation for PCA impact evaluation*

We can clearly understand from figure 5 that values of $m < 9$ would rapidly decrease the amout of retained variance. We will see better later on how this would badly impact the performance of the classifiers

2.2 LDA

PCA technique is unsupervised so we have no guarantee of obtainig discriminant directions. Despite the fact that LDA allows to find at most $C-1$, where C is the number of classes, directions, so it makes no sense to apply it as a dimensionality reduction technique, it can be used as a linear classifier and we can understand it by its definition; LDA mazimizes the between-class variability over the within-class variability ratio for the transformed samples:

$$\mathcal{L}(w) = \frac{s_B}{s_W} = \max_w \frac{w^T S_B w}{w^T S_W w}$$

It can be proved that optimal solution correspond to the eigenvector of $S_W^- 1 S_B$ corresponding to the largest eigenvalue. Once that we have estimated w we can project the test samples over w and assign the class label looking at the score obtained:

$$f(x) = \begin{cases} 1, & \text{if } x < 0. \\ 0, & \text{otherwise.} \end{cases}$$

It will be showed later the result of the classifier

3 Classification models analysis

3.1 Premises

In the next paragraphs we are going to compare different classification models. We will employ a k-fold cross validation technique (with $k = 3$) for model evaluation and the best models will be chosen to train the entire training set for performing a final comparison. We will consider three types of applications:

$$(\tilde{\pi}, C_{fp}, C_{fn}) = (0.1, 1, 1)$$
$$(\tilde{\pi}, C_{fp}, C_{fn}) = (0.5, 1, 1)$$
$$(\tilde{\pi}, C_{fp}, C_{fn}) = (0.9, 1, 1)$$

and the target application will be

$$(\tilde{\pi}, C_{fp}, C_{fn}) = (0.5, 1, 1)$$

We are interested in selecting the most promising approach and we will infact perform measures in term of minimum detection cost:

$$DCF = \frac{DCF_u(\pi_T, C_{fn}, C_{fp})}{min(\pi_T, C_{fn}, (1-\pi_T C_{fp}))} = \frac{\pi_T C_{fn} P_{fn} + (1-\pi_T) C_{fp} P_{fp}}{min(\pi_T, C_{fn}, (1-\pi_T C_{fp}))}$$

and for $minDCF$ computation we will look for:

$$t' = -log(\frac{\tilde{\pi}}{1-\tilde{\pi}})$$

that allows us to obtain the lowest possible DCF (as if knew in advance this optimal threshold)

3.2 Gaussian models

The first class of models we are going to analyze are the generative gaussian models. Model assumption are that, given the dataset X we assume that the sample x_t is a realization of the R.V. X_t. A simple model consists in assuming that our data, given the class, can be described by a Gaussian distribution:

$$(X_t | C_t = c) \sim (X | C = c) \sim \mathcal{N}(x_t | \mu_c, \Sigma_c)$$

We will assign a probabilistic score to each sample in term of the class-posterior log-likelihood ratio:

$$log\ r(x_t) = log\ \frac{P(C=h_1|x_t)}{P(C=h_0|x_t)}$$

We can expand this expression by writing:

$$log\ r(x_t) = log\ \frac{f_{X|C}(x_t|h_1)}{f_{X|C}(x_t|h_0)} + log\ \frac{\pi}{1-\pi}$$

While the training phase consists in estimating the model parameters the scoring phase consists in computing the log-likelihood ratio (first term of the equation) for each sample. It will be then compared with a threshold specific for each application for computing the $minDCF$. What differentiate the different Gaussian models is the way how we estimate the model parameters.

3.2.1 MVG Gaussian Classifier

The ML solution to the previous desribed problem is given by the empirical mean and covariance matrix for each class:

$$\mu_c^* = \frac{1}{N_c} \sum_{i=1}^{N} x_{c,i}$$
$$\Sigma_c^* = \frac{1}{N_c} \sum_{i=1}^{N} (x_{c,i} - \mu_c^*)(x_{c,i} - \mu_c^*)$$

We will then compute the log densities for each sample by using the estimated model parameters

3.2.2 Naive Bayes Classifier

The Naive Bayes assumption simplifies the MVG full covariance model stating that if we knew that for each class the componenets are approximately independent we can assume that the distribution $X|C$ can be factorized over its components. The ML solution to this problem is:

$$\mu^*_{c,[j]} = \frac{1}{N_c} \sum_{i|c_i=c} x_{i,[j]}$$
$$\sigma^2_{c,[j]} = \frac{1}{N_c} \sum_{i|c_i=c} (x_{i,[j]} - \mu_{c,[j]})^2$$

The density of a sample x can be expressed as $\mathcal{N}(x|\mu_c, \Sigma_c)$ where μ_c is an array where each element $\mu_{c,[j]}$ is the the mean for each class for each component while Σ_c is a diagonal covariance matrix. The Naive Bayes classifier corresponds to the MVG full covariance classifier with a diagonal covariance matrix

3.2.3 Tied Gaussian Classifier

This model assumes that the covariance matrices of the different class are tied (we consider only one covariance matrix common to all classes). We are assuming that:

$$f_{X|C}(x|c) = \mathcal{N}(x|\mu_c, \Sigma)$$

so each class has its own mean but the covariance matrix is the same for all the classes. The ML solution to this problem is:

$$\mu^*_c = \frac{1}{N_c} \sum_{i=1}^{N} x_{c,i}$$
$$\Sigma^* = \frac{1}{N} \sum_c \sum_{i|c_i=c} (x_i - \mu^*_c)(x_i - \mu^*_c)$$

This model is strongly related to LDA (used as a linear classfication model). By considering the binary log-likelihood ratio of the tied model we obtain a linear decision function:

$$llr(x) = log \frac{f_{X|C}(x|h_1)}{f_{X|C}(x|h_0)} = x^T b + c$$

where b and c are functions of class means and (tied) covariance matrix. On the other hand, projecting over the LDA subspace is, up to a scaling factor k, given by:

$$w^T x = k \cdot x^T \Lambda(\mu_1 - \mu_0)$$

where $\Lambda(\mu_1 - \mu_0) = b$. The LDA assumption that all the classes have the same within class covariance is related to the assumption done for the tied model.

3.2.4 Gaussian Models Comparison

Table 1: *MVG*

	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	
Z-normalized features - no PCA				
Full Cov	0.128	0.048	0.125	
Tied Cov	0.122	0.046	0.127	
Naive Bayes	0.822	0.567	0.856	
Z-normalized features - PCA(m=10)				
Full Cov	0.303	0.115	0.267	
Tied Cov	0.293	0.112	0.264	
Naive Bayes	0.306	0.121	0.283	
Z-normalized features - PCA(m=9)				
Full Cov	0.398	0.153	0.369	
Tied Cov	0.392	0.151	0.367	
Naive Bayes	0.416	0.159	0.362	
Gaussianized features - no PCA				
Full Cov	0.218	0.078	0.191	
Tied Cov	0.208	0.078	0.189	
Naive Bayes	0.813	0.586	0.847	
Gaussianized features - PCA(m=10)				
Full Cov	0.223	0.084	0.211	
Tied Cov	0.212	0.082	0.207	
Naive Bayes	0.279	0.103	0.254	
Gaussianized features - PCA(m=9)				
Full Cov	0.270	0.105	0.247	
Tied Cov	0.265	0.103	0.244	
Naive Bayes	0.305	0.119	0.282	

A graphical version of the table can be helpful in analyzing results:

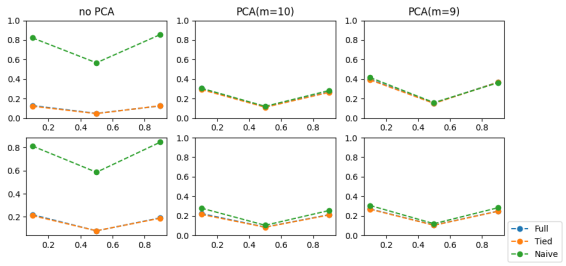

Figure 6: *Top: Z-Normalized feautres, Bottom: Gaussianized features*

Some observations:

- We can notice that Full-Cov model and Tied-Cov model achieve very good and similar results with slighlty better performances for the tied model.
- Gaussianization pre processing does't really help in achieving better results maybe because data are already well distributed according to the Gaussian assumptions
- Naive Bayes assumptio doesn't hold well, in particular if PCA is not applied. When PCA is applied it behaves slightly worse than the other two models. In particular Naive Bayes classfier achieves better results when gaussianization is applied

3.3 Logistic Regression Classifier

Logistic Regression is a discriminative classification model. Starting from the results obtained from the Gaussian classifiers we consider the linear decision function obtained from the expression of the posterior log-likelihood ratio:

$$l(x) = log\ \frac{P(C=h_1|x)}{P(C=h_0|x)} = log\ \frac{f_{X|C}(x|h_1)}{f_{X|C}(x|h_0)} + log\ \frac{\pi}{1-\pi} = w^T x + b$$

where b takes into account all the prior information. Given w and b we can compute the expression for the posterior class probability:

$$P(c = h_1|x, w, b) = \frac{e^{(w^T x+b)}}{1+e^{(w^T x+b)}} = \sigma(w^T x + b)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. Decision rules will be hyperplanes orthogonal to w.

3.3.1 Linear Logistic Regression

We are going to look for the minimizer of the function:

$$J(w, b) = \frac{\lambda}{2}||w||^2 + \frac{1}{n}\sum_{i=1}^{n} log(1 + e^{-z_i(w^T x_i+b)})$$

where λ is an hyperparameter that represents the regularization term (needed to make the problem solvable in case of linearly separable classes).

Table 2: *Linear Logistic Regression - 3-fold cross validation*

	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
Z-normalized features - no PCA			
Linear LR ($\lambda = 10^{-3}$)	0.170	0.170	0.155
Linear LR ($\lambda = 10^{-5}$)	0.132	0.047	0.127
Linear LR ($\lambda = 10^{-6}$)	0.132	0.047	0.126
Z-normalized features - PCA(m=10)			
Linear LR ($\lambda = 10^{-3}$)	0.299	0.113	0.263
Linear LR ($\lambda = 10^{-5}$)	0.297	0.114	0.263
Linear LR ($\lambda = 10^{-6}$)	0.297	0.114	0.262
Z-normalized features - PCA(m=9)			
Linear LR ($\lambda = 10^{-3}$)	0.390	0.153	0.369
Linear LR ($\lambda = 10^{-5}$)	0.388	0.152	0.363
Linear LR ($\lambda = 10^{-6}$)	0.388	0.152	0.362

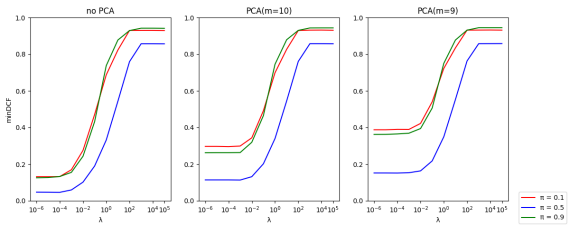

Figure 7: *minDCF for different values of λ and different priors*

Best performances are obtained for small values of λ.

3.3.2 Quadratic Logistic Regression

Now we are going to train a Quadratic LR model by performing features expansion. For binary linear LR the separation surfaces are linear decision function as already discussed (and we obtain the same form as for the Tied Gaussian classifier). By looking instead at the separation surface obtained through the MVG gaussian classifier we have:

$$log\ \frac{P(C=h_1|x)}{P(C=h_0|x)} = x^T Ax + b^T x + c = s(x, A, b, c)$$

This expression is quadratic in x but linear in A and b. We could rewrite it to obtain a decision function that is linear for the expanded features space but quadratic in the original features space. Features expansion is defined as:

$$\Phi(x) = \begin{bmatrix} vec(xx^T) \\ x \end{bmatrix}, w = \begin{bmatrix} vec(A) \\ b \end{bmatrix}$$

where $vec(X)$ is the operator that stacks the columns of X. In this way the posterior log-likelihood is expressed as:

$$s(x, w, c) = s^T \phi(x) + c$$

We are now going to train the Linear Logistic Regression model using features vectors $\phi(x)$

Table 3: *Linear Logistic Regression - 3-fold cross validation*

	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
Z-normalized features - no PCA			
Quadratic LR ($\lambda = 10^{-3}$)	0.187	0.063	0.149
Quadratic LR ($\lambda = 10^{-5}$)	0.153	0.052	0.142
Quadratic LR ($\lambda = 10^{-6}$)	0.150	0.053	0.141
Z-normalized features - PCA(m=10)			
Quadratic LR ($\lambda = 10^{-3}$)	0.299	0.111	0.241
Quadratic LR ($\lambda = 10^{-5}$)	0.307	0.109	0.249
Quadratic LR ($\lambda = 10^{-6}$)	0.305	0.109	0.248
Z-normalized features - PCA(m=9)			
Quadratic LR ($\lambda = 10^{-3}$)	0.373	0.151	0.337
Quadratic LR ($\lambda = 10^{-5}$)	0.378	0.149	0.349
Quadratic LR ($\lambda = 10^{-6}$)	0.378	0.149	0.350

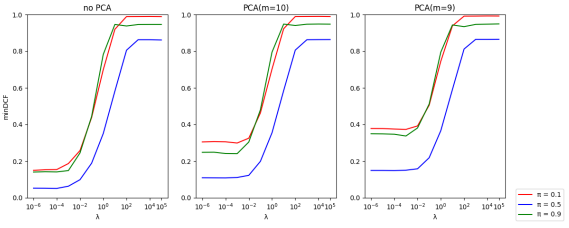

Figure 8: *minDCF for different values of λ and different priors*

3.4 Subsection

Nam ante risus, tempor nec lacus ac, congue pretium dui. Donec a nisl est. Integer accumsan mauris eu

ex venenatis mollis. Aliquam sit amet ipsum laoreet, mollis sem sit amet, pellentesque quam. Aenean auctor diam eget erat venenatis laoreet. In ipsum felis, tristique eu efficitur at, maximus ac urna. Aenean pulvinar eu lorem eget suscipit. Aliquam et lorem erat. Nam fringilla ante risus, eget convallis nunc pellentesque non. Donec ipsum nisl, consectetur in magna eu, hendrerit pulvinar orci. Mauris porta convallis neque, non viverra urna pulvinar ac. Cras non condimentum lectus. Aliquam odio leo, aliquet vitae tellus nec, imperdiet lacinia turpis. Nam ac lectus imperdiet, luctus nibh a, feugiat urna.

- First item in a list
- Second item in a list
- Third item in a list

Nunc egestas quis leo sed efficitur. Donec placerat, dui vel bibendum bibendum, tortor ligula auctor elit, aliquet pulvinar leo ante nec tellus. Praesent at vulputate libero, sit amet elementum magna. Pellentesque sodales odio eu ex interdum molestie. Suspendisse lacinia, augue quis interdum posuere, dolor ipsum euismod turpis, sed viverra nibh velit eget dolor. Curabitur consectetur tempus lacus, sit amet luctus mauris interdum vel. Curabitur vehicula convallis felis, eget mattis justo rhoncus eget. Pellentesque et semper lectus.

First This is the first item
Last This is the last item

Donec nec nibh sagittis, finibus mauris quis, laoreet augue. Maecenas aliquam sem nunc, vel semper urna hendrerit nec. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Maecenas pellentesque dolor lacus, sit amet pretium felis vestibulum finibus. Duis tincidunt sapien faucibus nisi vehicula tincidunt. Donec euismod suscipit ligula a tempor. Aenean a nulla sit amet magna ullamcorper condimentum. Fusce eu velit vitae libero varius condimentum at sed dui.

3.5 Subsection

In hac habitasse platea dictumst. Etiam ac tortor fermentum, ultrices libero gravida, blandit metus. Vivamus sed convallis felis. Cras vel tortor sollicitudin, vestibulum nisi at, pretium justo. Curabitur placerat elit nunc, sed luctus ipsum auctor a. Nulla feugiat quam venenatis nulla imperdiet vulputate non faucibus lorem. Curabitur mollis diam non leo ullamcorper lacinia.

Morbi iaculis posuere arcu, ut scelerisque sem. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Mauris placerat urna id enim aliquet, non consequat leo imperdiet. Phasellus at nibh ut tortor hendrerit accumsan. Phasellus sollicitudin luctus sapien, feugiat facilisis risus consectetur eleifend. In quis luctus turpis. Nulla sed tellus libero. Pellentesque metus tortor, convallis at tellus quis, accumsan faucibus nulla. Fusce auctor eleifend volutpat. Maecenas vel faucibus enim. Donec venenatis congue congue. Integer sit amet quam ac est aliquam aliquet. Ut commodo justo sit amet convallis scelerisque.

1. First numbered item in a list
2. Second numbered item in a list
3. Third numbered item in a list

Aliquam elementum nulla at arcu finibus aliquet. Praesent congue ultrices nisl pretium posuere. Nunc vel nulla hendrerit, ultrices justo ut, ultrices sapien. Duis ut arcu at nunc pellentesque consectetur. Vestibulum eget nisl porta, ultricies orci eget, efficitur tellus. Maecenas rhoncus purus vel mauris tincidunt, et euismod nibh viverra. Mauris ultrices tellus quis ante lobortis gravida. Duis vulputate viverra erat, eu sollicitudin dui. Proin a iaculis massa. Nam at turpis in sem malesuada rhoncus. Aenean tempor risus dui, et ultrices nulla rutrum ut. Nam commodo fermentum purus, eget mattis odio fringilla at. Etiam congue et ipsum sed feugiat. Morbi euismod ut purus et tempus. Etiam est ligula, aliquam eget porttitor ut, auctor in risus. Curabitur at urna id dui lobortis pellentesque.

4 Section

Figure 9: *A majestic grizzly bear*