Probabilistic Methods for Color Classification and Blue Recycling Bin Detection

Srinidhi Kalgundi Srinivas
Department of Electrical and Computer Engineering
University of California San Diego
skalgundisrinivas@ucsd.edu

***Abstract*—This report presents different probabilistic methods used to classify pixels of image into different colors and use colors as features to detect blue recycle bins based on the classified pixel values. A comparison of accuracy of different classification models such as logistic regression and Gaussian Discriminant Analysis is presented.**

***Index Terms*—Logistic Regression, Maximum Likelihood Estimation (MLE), Gaussian Naive Bayes**

I. Introduction

Object detection is an important problem in the field of Computer Vision. Accurate detection of objects in an input image is essential for downstream tasks such as auto focus on the face of a person in an image, detection of pedestrians in the field of view and detection of traffic signs in an autonomous driving situation and many more. Different probabilistic methodologies exist in detecting these objects and these approaches rely on the probabilistic distribution of features in the training data. Pixel intensity values serve as features for detection of colors and an object such as recycling bins of a particular color in the image. Since the models are probabilistic in nature, it is of utmost importance to collect quality data as the models are only as good as the data they are trained on. There are two types of models, *Generative Models* and *Discriminative Models*. While the latter models the decision boundary between the classes in the data, the former models the distribution the data is sampled from. A Generative Model learns the joint probability distribution of input and output classes parameterized by a parameter $P(Y, X; \theta)$ and the discriminative model learns the conditional probability of $P(Y = y \mid X; \theta)$. Examples of Generative Models are Gaussian Naive Bayes [2], Bayesian Networks, Markov Random Fields whereas examples of Discriminative Models are Logistic Regression [1], Neural Networks, Conditional Random Fields etc.,

For classifying pixels as red, green or blue, logistic regression model is trained and the weights of the trained model is used to predict the color of a new pixel. This classifier model is later used to classify each pixels in an input image and generate a mask. Shape statistics is used to detect the location of blue recycling bins in the given mask and obtain bounding box coordinates of the bin.

II. Problem Formulation

A. Supervised Learning [4]

Classification of colors and recycling bin detection can be viewed of as a supervised learning problem where we have a dataset D composed of n Independent and Identically Distributed(iid) samples.

$$D := \{X, \mathbf{y}\}$$

where,

$$X \in \mathbb{R}^{nxd}, y \in \mathbb{R}^{n}$$

where, d is the number of features in each input samples of X, I number of features in our problem in each input pixel is 3, i.e., d = 3, hence,

$$X \in \mathbb{R}^{nx3}$$

For the logistic regression model, the aim to choose a model $P(y \mid \mathbf{x}; \theta)$ with parameters θ. With this conditional probability model, the label for unknown sample x is obtained.

The goal is to obtain θ^* that maximizes the likelihood of the training data.

$$\omega^* = \arg\max_{\theta} P(\mathbf{y} \mid X; \omega)$$

Obtained ω^* is then used to obtain the label for an unknown sample x_*

$$y^* = \arg\max_{y} P(y \mid \boldsymbol{x_*}; \omega^*)$$

B. Logistic Regression

At the very heart of logistic regression is the Sigmoid function which is mathematically written as,

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \tag{1}$$

The sigmoid function has a special property of mapping the continuous input value **z** to Bernoulli Probability Mass function.

For the problem of pixel color classification, three output classes are considered, $\{$*Red, Green and Blue*$\}$. Input vector **X** is a vector in $\mathbb{R}^{NX3}$ and y is a vector containing in $\mathbb{R}^{N}$ containing class labels = $\{$1: *Red*, 2: *Green, 3: Blue*$\}$.

Problem can be formulated as maximizing the parameter ω of the distribution $P(y \mid x;\omega)$.

$$\omega^* = \arg\max P(y \mid x;\omega)$$

$$\omega^* = \arg\max \sigma(\omega^T x)$$

The optimal value of ω is determined by the gradient descent algorithm,

$$\omega^{(t+1)} = \omega^t - \alpha * \nabla_\omega(P(y \mid x;\omega))$$

These equations are implemented as is to obtain the results.

III. Technical Approach

A. Logistic Regression

For optimal implementation of the logistic regression and to save on the run time, numpy vectorization methods have been used.

One versus all multilogistic regression is used where each class is one-shot encoded. Example of one shot encoding for 3 classes is as shown below;

$$y_i = \begin{cases} 1, & \text{i=j} \\ 0, & \text{otherwise} \end{cases}$$

Solution for the color classification models for classifying colors and classifying recycling bin blue color comprises of the following stages;

B. Data Collection

Training data for problem 1 is provided as images and the pixels of the provided images are grouped into a matrix X of size Nx3, where N is the number of pixels in the training dataset. These pixels are hand classified into one of the 3 classes {Red, Green and Blue} for problem 1 and later used to train the logistic regression model.

For blue recycling bin detection, 5 classes were used with the logistic regression model to enable better separation between colors of different classes. Training data samples were collected using *roipoly* method. The 5 classes that are utilized to train the classifier for problem 2 are {*Bin Blue, Other Blue, Green, Red/Brown, Gray/Black*}.

C. Training the model

The only difference between the classifiers used for color classification and blue recycling bin detection is the number of classes used to train the model. For the former, 3 classes were used and for the latter, 5.

As discussed in the previous section, logistic regression model uses the sigmoid function to map continuous values to Bernoulli probability mass function. Softmax function achieves the same result where it maps continuous values to categorical probability mass function that can be utilized to classify different classes in y.

$$y \in \{1, 2, ..K\}$$

The parameters are W^{Kxd}, where K is the number of classes.

For classification of colors in 3 classes, the weight matrix is of size W^{3x3} and the weight matrix for classifying blue recycling bins is of size W^{5x3}. Before the training process was started, a column of 1's was added to account for the bias term.

To obtain optimal weight of the matrix, gradient descent algorithm was used, where the update equation used is,

$$\nabla_\omega(P(y \mid x;\omega)) = (y - \sigma(W^T X))X$$

where,

$$X \in \mathbb{R}^{nx3}, y \in \mathbb{R}^n$$

This equation above is the MLE formulation for calculation of parameters ω. Gradient descent is run until convergence (where the error between original and predicted classes is less than a tolerance value=1e-5) Obtain weight matrices after training are presented in tables [XI and X]. Note that the number of columns is one more than the number of features to account for the bias term.

D. Prediction

The obtained optimal weight matrix is used to predict the classes of new pixels as;

$$y_* = \arg\max_y \sigma(W^T X)$$

To account for the bias term, a column of 1's was added to the image input before prediction was performed.

For recycling bin classification, a mask image is generated where the classified *bin blue* pixels are set to a value of 255 and rest of the pixels of other classes are set to 0.

Once the mask image is generated, morphological image processing techniques such as dilation and erosion is used to separate close objects and complete holes within the recycling bins and remove noise.

OpenCV's *findContours* method was used to obtain the regions of interest and bounding box coordinates were calculated based on the ratio of height to width of the rectangular regions obtained.

Intersection of Union (IoU) metric was used to calculate the accuracy of the calculate bounding boxes. Prediction was found to be accurate when the IoU was greater than 0.5.

IV. Results

A. RGB Classification

Logistic Regression model was trained for 150 epochs to obtain optimal results. On the validation set, the accuracy of classification is as shown in I.

TABLE I: Accuracy obtained using Logistic Regression for RGB classifier

Red	100%
Green	100%
Blue	100 %

B. Blue Recycling Bin Classifier

5 color classes were used to train the logistic regression classifier model. Model was tested on the validation set which is comprised of 10 images. Bounding boxes and accuracy obtained for the validation set is tabulated in IX.

The classification outputs of recycling bin classification are shown in Fig. 3 through Fig. 12. It can be observed in these images that there are mis-classification of some of the pixels that results in reduced accuracy, however, bounding box calculation is accurate because of image post processing. When the image contains no bins, we rely on the shape statistics to accurately determine whether there are bins in the image even where the pixels are false positives.

A good example of this is the output for 068.jpg shown in Fig. 10. Even though the cover is determined as blue, since it does not follow the shapre of the recycling bin, no bounding box is determined for the same. A false negative of the validation set is shown in Fig. 9. The resulting mask image accurately depicts the classification of blue pixels of the recycling bins. However, the accuracy of the bounding box coordinates is reduces as the 2 recycling bins are close to one another. These bins can be separated using opening operation of image processing, but better score is achieved without hard coding the operation for this particular image.

As an extension, Gaussian Naive Bayes model is implemented and the model parameters are calculated as shown in tables [II III IV V VI VII]. A comparison of accuracy is presented in VIII. Training losses for Color Classification and Recycling Bin Detection are shown in fig.1 and fig.2. It can be observed that the loss decreases as the training progresses.

TABLE II: Parameter of Gaussian Naive Bayes model for color classification: Mean

	Red	Green	Blue
Red	0.75	0.35	0.35
Green	0.35	0.74	0.33
Blue	0.35	0.33	0.74

TABLE III: Parameter of Gaussian Naive Bayes model for color classification: Variance

	Red	Green	Blue
Red	0.037	0.0619	0.0620
Green	0.0557	0.0347	0.0560
Blue	0.0545	0.0568	0.0357

TABLE IV: Parameter of Gaussian Naive Bayes model for color classification: Priors

Red	Green	Blue
0.365	0.325	0.309

TABLE V: Parameter of Gaussian Naive Bayes model for Recycling Bin Detection: Mean

	Hue	Saturation	Value
Bin Blue	0.44	0.73	0.60
Other Blue	0.36	0.15	0.77
Green	0.15	0.35	0.49
Red	0.21	0.36	0.42
Black	0.31	0.23	0.58

TABLE VI: Parameter of Gaussian Naive Bayes model for color classification: Variance

	Hue	Saturation	Value
Bin Blue	0.001	0.035	0.042
Other Blue	0.03	0.022	0.053
Green	0.04	0.029	0.069
Red	0.004	0.04	0.04
Black	0.03	0.0075	0.03

TABLE VII: Parameter of Gaussian Naive Bayes model for color classification: Priors

Bin Blue	Other Blue	Green	Red	Black
0.0604	0.011	0.471	0.23	0.23

TABLE VIII: Comparison of accuracy of Logistic Regression and Gaussian Naive Bayes

	Logistic Regression	Gaussian Naive Bayes
Red	100%	100%
Green	100%	97.06%
Blue	100%	97.56%

TABLE IX: Table Type Styles

Image	**Bounding Box Coordinates**	**IoU $\geq$ 0.5**
061.jpg	[182, 101, 313, 295]	True
062.jpg	[25, 347, 133, 497]	True
063.jpg	[168, 64, 300, 239]	True
064.jpg	[349, 104, 467, 264]	True
065.jpg	[762, 416, 924, 622]	True
066.jpg	[]*	True
067.jpg	[]^F	True
068.jpg	[]*	True
069.jpg	[]*	True
070.jpg	[]*	True

*: No bins in the image
^{F}: False Negative

TABLE X: Color Classification Weight matrix with additional column for bias

-0.06	4.00	-1.92	-1.88
0.04	-2.00	3.85	-1.93
0.023	-2.00	-1.92	3.81

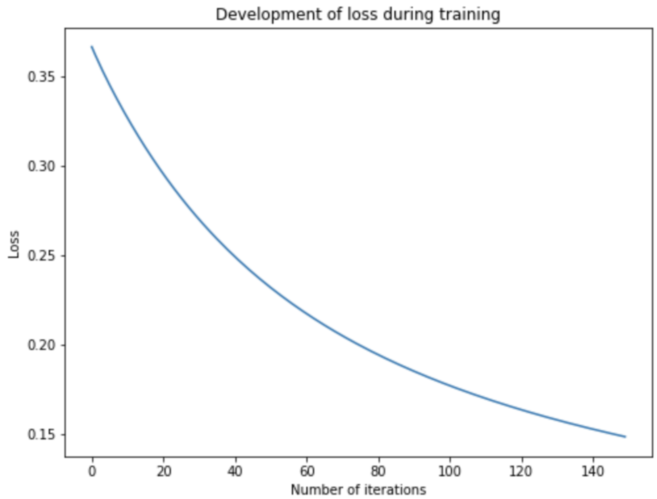

Fig. 1: Logistic Regression training loss for Color Classification

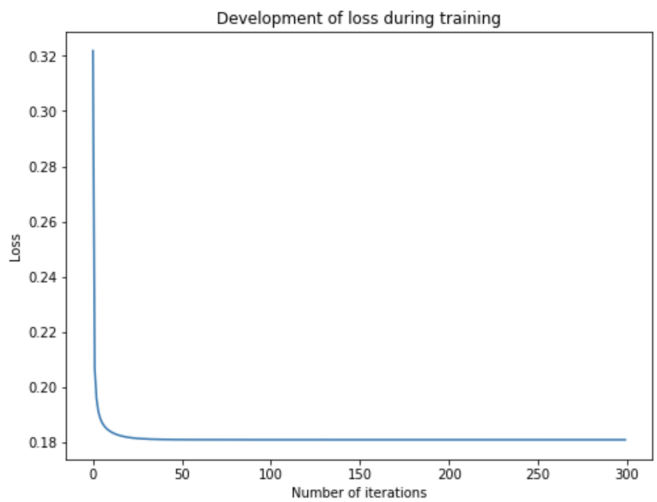

Fig. 2: Logistic Regression training loss for Recycling Bin detection

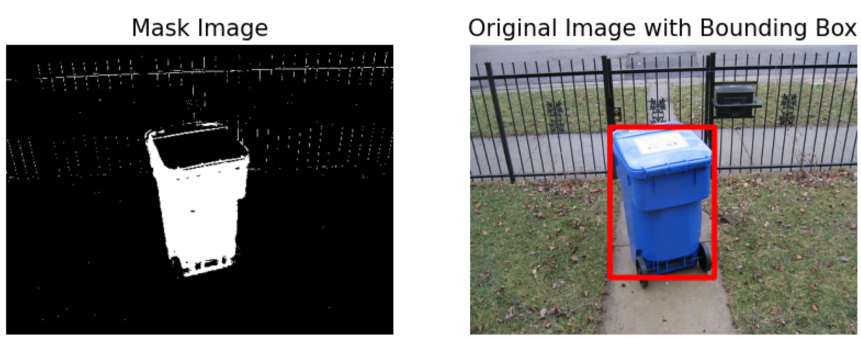

Fig. 3: figure
Recycling Bin Detector Output - Image 0061.png

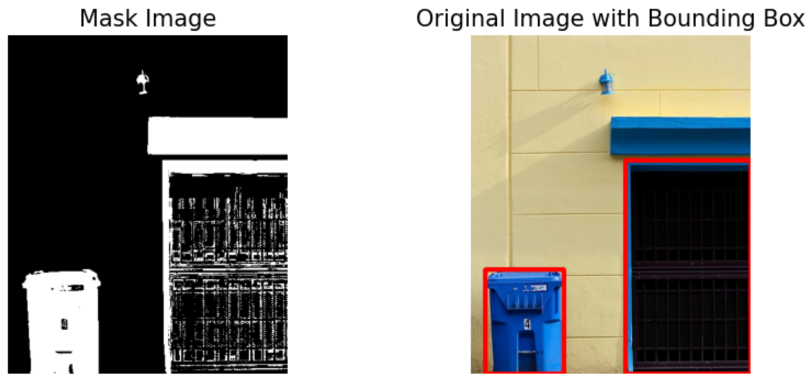

Fig. 4: Recycling Bin Detector Output - Image 0062.png

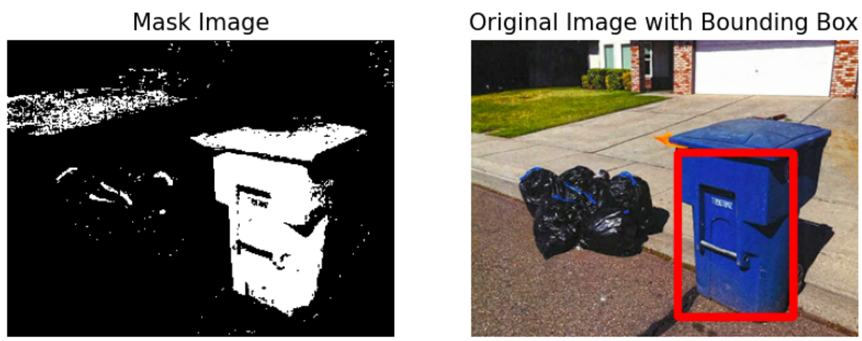

Fig. 5: Recycling Bin Detector Output - Image 0063.png

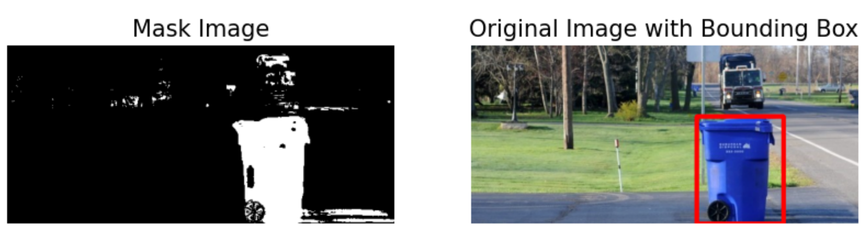

Fig. 6: Recycling Bin Detector Output - Image 0064.png

Fig. 7: Recycling Bin Detector Output - Image 0065.png

Fig. 8: Recycling Bin Detector Output - Image 0066.png

Fig. 9: Recycling Bin Detector Output - Image 0067.png

Fig. 10: Recycling Bin Detector Output - Image 0068.png

Fig. 11: Recycling Bin Detector Output - Image 0069.png

Fig. 12: Recycling Bin Detector Output - Image 0070.png

TABLE XI: Recycling bin Classification Weight matrix with additional column for bias

-8.65	8.19	10.46	1.81
-4.28	3.16	-7.54	5.68
6.09	-7.15	0.08	-3.99
4.89	-4.83	1.09	-4.65
1.94	0.64	-4.08	1.17

V. Acknowledgements

To all the instructors and students who asked and answered questions on Piazza.

References

[1] Cox, D.R., 1958. The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), 20(2), pp.215–232.

[2] Hand, D. J., Yu, K. (2001). Idiot's Bayes—not so stupid after all?. International statistical review, 69(3), 385-398.

[3] Ruthotto, L. and Haber, E., 2021. An introduction to deep generative modeling. GAMM-Mitteilungen, p.e202100008.

[4] Cunningham P., Cord M., Delany S.J. (2008) Supervised Learning. In: Cord M., Cunningham P. (eds) Machine Learning Techniques for Multimedia. Cognitive Technologies. Springer, Berlin, Heidelberg.

ECE276A - Assignment-1

Srinidhi Bharadwaj Kalgundi Srinivas

February 2022

1 Problem 4

Let X be a Gaussian random variable with known variance σ^2 but unknown mean. Let $x_i{}_{i=1}^n$ n independent samples obtained from X.

(a) Formulate a maximum likelihood estimation (MLE) problem to determine the unknown mean of X.
Solution: Given,

$$P(X \mid \mu) = N(\mu, \sigma^2)$$

Since $x_i{}_{i=1}^n$ are n-independent samples, the probability distribution can be written as;

$$P(x_1, x_2, ..., x_n \mid \mu) = \prod_{i=1}^{n} P(X_i \mid \mu)$$

Formulation as MLE, i.e., maximizing the function on the RHS with μ as the parameter,

$$P(x_1, x_2, ..., x_n \mid \mu) = \arg\max_{\mu} \prod_{i=1}^{n} P(X_i \mid \mu)$$

$$P(x_1, x_2, ..., x_n \mid \mu) = \arg\max_{\mu} \prod_{i=1}^{n} P(X_i \mid \mu)$$

Using log as it is monotonic and increasing a function implies increasing log of the function,

$$log(P(x_1, x_2, ..., x_n \mid \mu)) = \arg\max_{\mu} log(\prod_{i=1}^{n} P(X_i \mid \mu))$$

$$= \arg\max_{\mu} \sum_{i=1}^{n} log(P(X_i \mid \mu))$$

$$= \arg\max_{\mu} \sum_{i=1}^{n} log(\frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{(x_i-\mu)^2}{2\sigma^2}))$$

$$= \arg\max_{\mu} \sum_{i=1}^{n} (-\frac{1}{2} log(2\pi\sigma^2) - \frac{(x_i-\mu)^2}{2\sigma^2})$$

As the variance (σ^2) is not dependent on μ, they do not contribute to the equation and can be removed.

$$= \arg\max_{\mu} \sum_{i=1}^{n} -\frac{(x_i-\mu)^2}{2\sigma^2}$$

The equation above is the MLE formulation for determining the unknown mean μ of X.

(b) Solve the problem in part (a) to obtain the maximum likelihood estimate $\hat{\mu}_{MLE}$.
Solution: Using the equation obtained above, i.e.,

$$log(P(x_1, x_2, ..., x_n \mid \mu)) = \arg\max_{\mu} \sum_{i=1}^{n} -\frac{(x_i-\mu)^2}{2\sigma^2}$$

To find the max value, taking the derivative of the above equation with respect to μ,

$$\frac{\partial(log(P(x_1, x_2, ..., x_n \mid \mu))}{\partial\mu} = \sum_{i=1}^{n} \frac{(x_i-\mu)}{\sigma^2}$$

Setting the above equation to 0 to obtain the critical point,

$$0 = \sum_{i=1}^{n} \frac{(x_i - \hat{\mu}_{MLE})}{\sigma^2}$$

$$0 = \sum_{i=1}^{n} (x_i - \hat{\mu}_{MLE})$$

$$0 = \sum_{i=1}^{n} (x_i) - n\hat{\mu}_{MLE})$$

$$n\hat{\mu}_{MLE} = \sum_{i=1}^{n} (x_i)$$

$$\hat{\mu}_{MLE} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

The Maximum Likelihood Estimate of the mean is the same mean of the distribution.

(c) Formulate a maximum a posteriori (MAP) problem to determine the unknown mean of X. Suppose that a prior Gaussian distribution N(μ_0; σ_0^2)
Solution: The task is to find μ that maximizes the equation,

$$P(\mu|X)$$

From Bayes' rule,

$$P(\mu|X) = \frac{P(X|\mu)P(\mu)}{P(X)}$$

$$P(\mu|x_1, x_2, .., x_n) = \frac{P(x_1, x_2, .., x_n|\mu)P(\mu)}{P(x_1, x_2, .., x_n)}$$

$$P(\mu|x_1, x_2, .., x_n) = \arg\max_{\mu} \frac{P(x_1, x_2, .., x_n|\mu)P(\mu)}{P(x_1, x_2, .., x_n)}$$

$$P(\mu|x_1, x_2, .., x_n) = \arg\max_{\mu} P(x_1, x_2, .., x_n|\mu)P(\mu)$$

The above equation is the Maximum Aposteriori Probability. We know that,

$$P(x_1, x_2, .., x_n|\mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{(x_i - \mu)^2}{2\sigma^2})$$

$$P(\mu) = \frac{1}{\sqrt{2\pi\sigma_0^2}} exp(-\frac{(\mu - \mu_0)^2}{2\sigma_0^2})$$

Using log,

Using these equation in MAP equation and log,

$$log(P(\mu|x_1, x_2, .., x_n)) = \arg\max_{\mu}[log(P(x_1, x_2, .., x_n|\mu)) + log(P(\mu))]$$

$$log(P(\mu|x_1, x_2, .., x_n)) = \arg\max_{\mu}[\sum_{i=1}^{n} -\frac{1}{2}log(2\pi\sigma^2) - \frac{(x_i - \mu)^2}{2\sigma^2} - \frac{1}{2}log(2\pi\sigma_0^2) - \frac{(\mu - \mu_0)^2}{2\sigma_0^2}]$$

Since, variance of both the distributions are known,

$$log(P(\mu|x_1, x_2, .., x_n)) = \arg\max_{\mu}[\sum_{i=1}^{n} -\frac{(x_i - \mu)^2}{2\sigma^2} - \frac{(\mu - \mu_0)^2}{2\sigma_0^2}]$$

This is the MAP optimization problem that needs to be solved to obtain μ_{MAP}

(d) Solve the problem in part (c) to obtain the maximum a posteriori estimate $\hat{\mu}_{MAP}$

Solution: Using the equation obtained in the previous problem,

$$log(P(\mu|x_1, x_2, .., x_n)) = \arg\max_{\mu}[\sum_{i=1}^{n} -\frac{(x_i - \mu)^2}{2\sigma^2} - \frac{(\mu - \mu_0)^2}{2\sigma_0^2}]$$

To maximize this, taking the derivative with respect to μ and setting it to 0,

$$\frac{\partial(log(P(x_1, x_2, ..., x_n \mid \mu))}{\partial\mu} = 0$$

$$0 = \sum_{i=1}^{n} \frac{(x_i - \mu)}{\sigma^2} - \frac{\mu - \mu_0}{\sigma_0^2}$$

$$\frac{\hat{\mu}_{MAP} - \mu_0}{\sigma_0^2} = \sum_{i=1}^{n} \frac{(x_i - \hat{\mu}_{MAP})}{\sigma^2}$$

$$\hat{\mu}_{MAP} - \mu_0 = \frac{\sigma_0^2}{\sigma^2}(\sum_{i=1}^{n}(x_i - \hat{\mu}_{MAP}))$$

$$\hat{\mu}_{MAP} - \mu_0 = \frac{\sigma_0^2}{\sigma^2}\sum_{i=1}^{n} x_i - \frac{n\sigma_0^2}{\sigma^2}\hat{\mu}_{MAP}$$

$$\hat{\mu}_{MAP} + \frac{n\sigma_0^2}{\sigma^2}\hat{\mu}_{MAP} = \frac{\sigma_0^2}{\sigma^2}\sum_{i=1}^{n} x_i + \mu_0$$

$$\frac{\sigma^2\hat{\mu}_{MAP} + n\sigma_0^2\hat{\mu}_{MAP}}{\sigma^2} = \frac{\sigma_0^2\sum_{i=1}^{n} x_i + \sigma^2\mu_0}{\sigma^2}$$

$$\hat{\mu}_{MAP} = \frac{\sigma_0^2\sum_{i=1}^{n} x_i + \sigma^2\mu_0}{\sigma^2 + n\sigma_0^2}$$

As the number of samples, n `->` ∞, $\hat{\mu}_{MAP}$ `->` $\hat{\mu}_{MLE}$.