An overview of probabilistic SVMs

presented by

Mark Jayson Cortez and Manoj Subedi

Outline

1. Bayesian Inference and MCMC Methods
2. Bayesian Regression
3. Bayesian SVMs
4. Comparisons with deterministic counterparts

Bayesian Inference

Bayesian inference provides a key framework for quantifying the knowledge of the data and prior belief about some parameters of a model. This expressed mathematically through Baye's theorem.

Baye's Theorem gives,

$$\boxed{p\left(\theta \middle| Y_{obs}\right)} = \frac{\boxed{p\left(Y_{obs} \middle| \theta\right)}\boxed{p\left(\theta\right)}}{p\left(Y_{obs}\right)}$$

Posterior
Likelihood
Prior

$p(\theta|Y_{obs})$ is the posterior distribution, the target output of the algorithm

$p(Y_{obs}|\theta)$ is the likelihood, i.e. the probability of observing the data Y_{obs} given θ

$p(Y_{obs})$ is the prior distribution, which represents the knowledge about the parameters before taking into account the observed data

MCMC Method

To determine the posterior distribution, we generate a Markov Chain in which the stationary distribution is the target posterior distribution. This is called Markov Chain Monte Carlo (MCMC), and a common method is the Metropolis-Hastings (MH) method.

1. Initialize $\theta^{(0)}$.
2. Generate a proposal sample, $\theta^* \sim q\left(\theta \middle| \theta^{(n)}\right)$.
3. Calculate acceptance probability

$$\alpha = \min\left(1, \frac{p\left(Y \middle| \theta^*\right) p\left(\theta^*\right) q\left(\theta^* \middle| \theta^{(n)}\right)}{p\left(Y \middle| \theta^{(n)}\right) p\left(\theta^{(n)}\right) q\left(\theta^{(n)} \middle| \theta^*\right)}\right).$$

4. Accept θ^* with probability α.

Bayesian Regression

Before we discuss the regression process itself, we contextualize this as a problem of the identification of a trend between weight and height, with height as the predicting factor for weight. We therefore try to predict weight as a scalar multiplied to height plus a baseline value. We denote by $\hat{y}$ the predicted weight, and x the height, so that

$$\hat{y} = \beta_1 x + \beta_0.$$

We also have to describe the random variation of the actual weights y around the predicted weight $\hat{y}$, that is, for an observation i

$$y_i = \hat{y}_i + \xi_i$$

for some value ξ. For the sake of simplicity we can assume that all these variations are normally distributed with mean zero and standard deviation σ. It then follows that $y_i \sim N(\hat{y}_i, \sigma^2)$. This analysis then provides us with a way to define the likelihood of the prediction $\hat{y}_i$. Hence, given independent observations $\{(y_i, x_i)\}_{i=1}^{N}$, the total likelihood is given by

$$\prod_{i=1}^{N} \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \beta_1 x_i + \beta_0)^2}{2} \right) \right].$$

Bayesian Regression: Example

To better understand the process, we proceed to an example. First we generate a data set where the true regression line takes the form $y = x + 1$. We take M points (x_i, y_i) with $x_i \in [0, 1]$ along the true regression line. For each of these points, we add to y_i, a Gaussian-distributed noise from $N(0, \sigma^2)$. For our purpose, we specify $M = 50$ and $\sigma = 0.5$. These new set of points then will serve as our data set, seen in Figure 1.

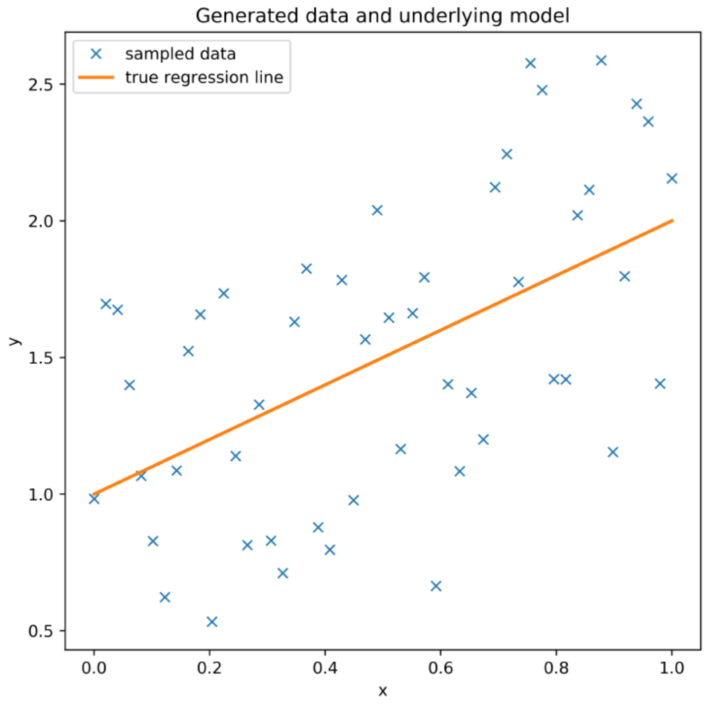


Figure 1: Data points together with the true regression line.

Bayesian Regression: Example

First, we define the priors and assume that the parameters are independent. Recall from the previous discussion that we have to specify priors for β_0, β_1, and σ. For both β_0 and β_1, we set the informative Gaussian prior $N(0, 20)$. In the case of σ, we take the suggestion given in [2], to take a half-Cauchy distribution.

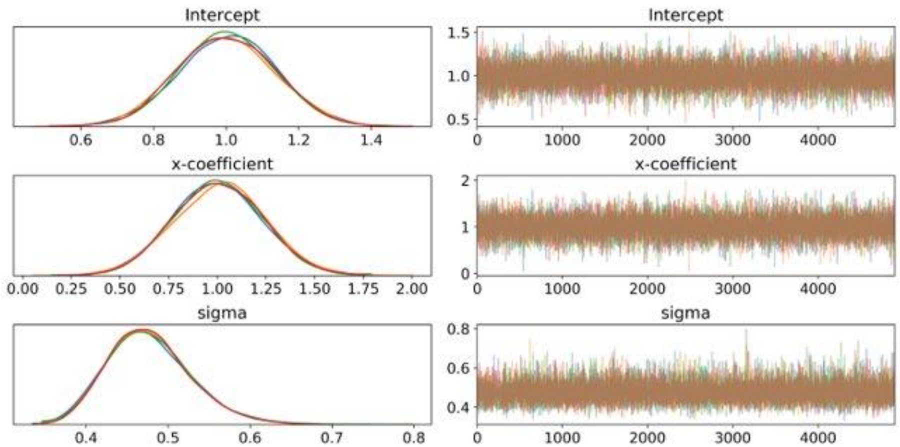
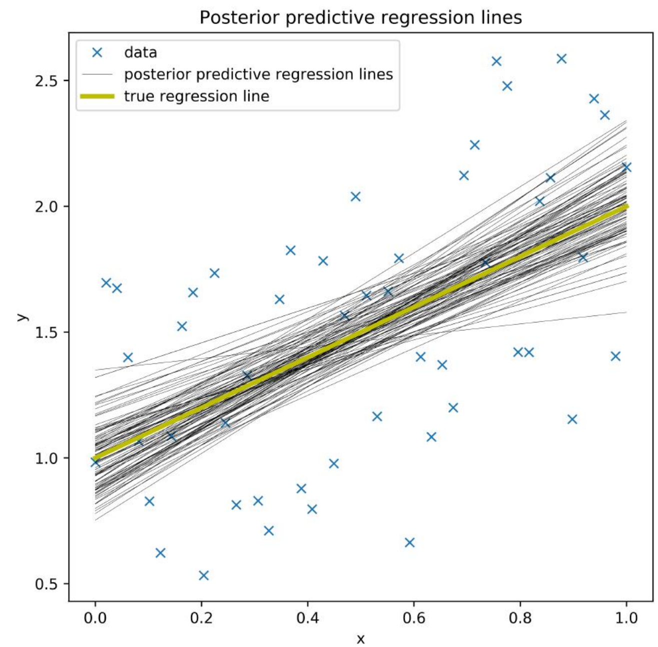


Figure 2: Posterior distributions obtained using pymc3 together with the traceplot of samples.

	true regression line	mean	std. dev.	HDI 95%
Intercept β_0	1	1.002404	0.133168	$[0.746450, 1.266259]$
x-coefficient β_1	1	1.004820	0.228079	$[0.558677, 1.446214]$
error st. dev. σ	0.5	0.477677	0.050640	$[0.382568, 0.577510]$

Table 1: Comparison of regression line parameters and the inferred parameter values obtained through sampling from the posterior distribution.



Support Vector Machine

Given data points $\left\{\left(x_i, y_i\right)\right\}_{i=1}^{N}$ where $x_i \in \mathbb{R}^d$ and $y_i \in \left\{\pm 1\right\}$, the task is to find **w** and **b** such that $w \bullet \phi\left(x\right) + b$ separates the data with the largest margin.

If the points are not linearly separable, we can reformulate the problem into

$$\min_{\mathbf{w} \in \mathbb{Y}, b \in \mathbb{R}} L(\mathbf{w}, b, C)$$

where

$$L(\mathbf{w}, b, C) := \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{n} l\left(y_i \left(\mathbf{w} \cdot \phi(x_i) + b\right)\right) \qquad (1)$$

Bayesian SVM: Prior Distribution

$$L(\mathbf{w}, b, C) := \boxed{\frac{1}{2}\|\mathbf{w}\|^2} + C\sum_{i=1}^{n} l\left(y_i\left(\mathbf{w}\cdot\phi(x_i) + b\right)\right)$$

The deterministic SVM classifier given by equation (1) is described as the maximum a posteriori (MAP) solution of the corresponding probabilistic problem. The first summand induces Gaussian priors on **w** and **b** which can be described as

$$Q(\mathbf{w}, b) \propto \exp\left(-\frac{1}{2}\|\mathbf{w}\|^2 - \frac{1}{2}b^2 B^{-2}\right)$$

Since only the latent variable $\theta(x) = \mathbf{w}\cdot\phi(x) + b$ appear in the 2nd term, we can express the prior directly as a distribution over θ with covariance

$$\langle \theta(x)\cdot\theta(x')\rangle = \langle\left(\phi(x)\cdot\mathbf{w}\right)\left(\mathbf{w}\cdot\phi(x')\right)\rangle + B^2 = \phi(x)\cdot\phi(x') + B^2$$

Bayesian SVM: Likelihood Function

$$\boxed{L(\mathbf{w},b,C) := \frac{1}{2}\|\mathbf{w}\|^2 + \boxed{C\sum_{i=1}^{n} l\left(y_i\left(\mathbf{w}\cdot\phi(x_i)+b\right)\right)}}$$

The second part of equation (1) can be taken as a negative log-likelihood when we assume the probability of getting output y for a given x is

$$Q(y=\pm 1 \mid x,\theta) = \kappa\, exp\left(-C\left(y\theta\left(x\right)\right)\right)$$

where κ is a normalizing constant.

The total likelihood therefore is $Q\left(D\middle|\theta\right) = \prod_{i=1}^{N} Q\left(y_i\middle|x_i,\theta\right)Q\left(x_i\right)$

where $Q\left(x_i\right)$ is the distribution of x_i.

Bayesian SVM: Example

We are now ready for an example. Consider data points (x_1, x_2) circling the origin, whose coordinates were independently sampled from a standard normal distribution with variance 0.1 for the x_1-coordinate and variance 5 for the x_2-coordinate. Suppose we assign to class 0, all points whose distance from the origin is less than or equal to 2, and all others to class 1 (see Figure 4). We assign the label $y = 1$ to those belonging in class 0, and the label $y = -1$ to those belonging to class 1.

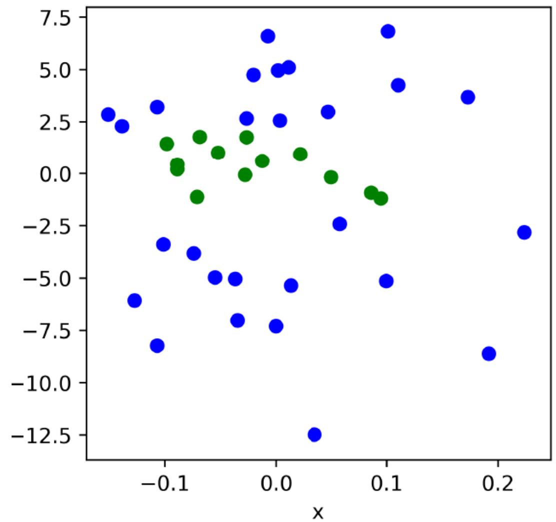


Figure 4: Generated data for binary classification. Green points belong to class 0, while blue ones belong to class 1.

Bayesian SVM: Example

Since the feature space is a subset of $\mathbb{R}^3$, the separating hyperplane takes the form

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \phi(x_1, x_2) + b = 0.$$

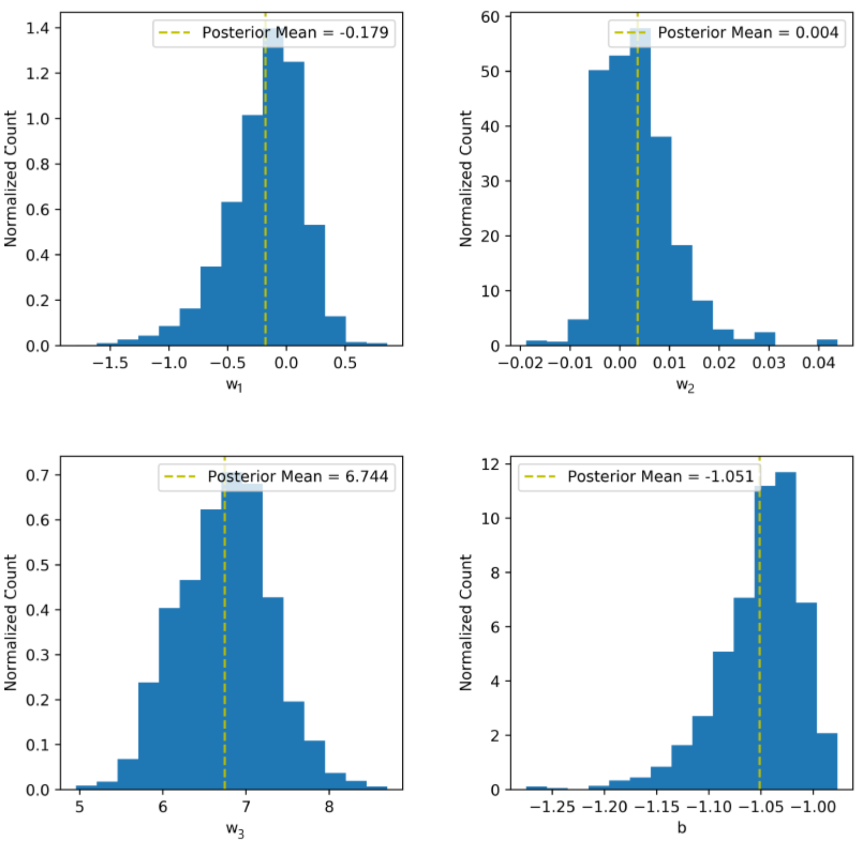
We assume that the distribution of the data is known, so that the usual SVM optimization problem translates to the total posterior

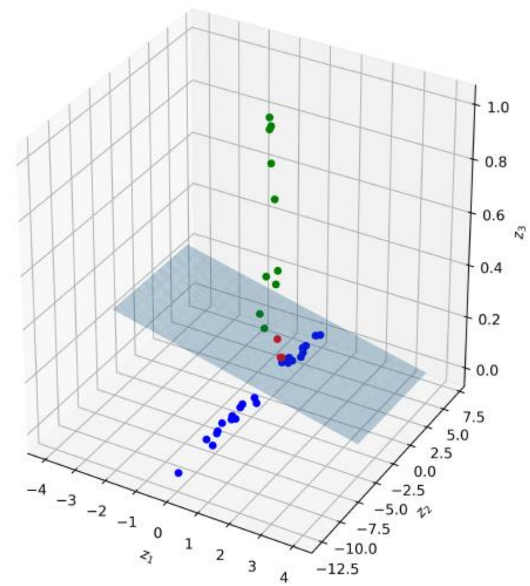
$$w, b | x, y \propto \prod_{i=1}^{N} \left\{ \exp\left[-Cl\left(y_i\left(\phi\left(x_i\right) + b\right)\right)\right] \cdot \frac{1}{\sqrt{0.1\,(2\pi)}} \exp\left(-\frac{{x_{1i}}^2}{2\,(0.1)}\right) \cdot \frac{1}{\sqrt{5\,(2\pi)}} \exp\left(-\frac{{x_{2i}}^2}{2\,(5)}\right) \right\}$$
$$\cdot \left\{ \prod_{j=1}^{3} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{{w_j}^2}{2}\right) \right\} \frac{1}{B\sqrt{2\pi}} \exp\left(-\frac{b^2}{2B^2}\right)$$

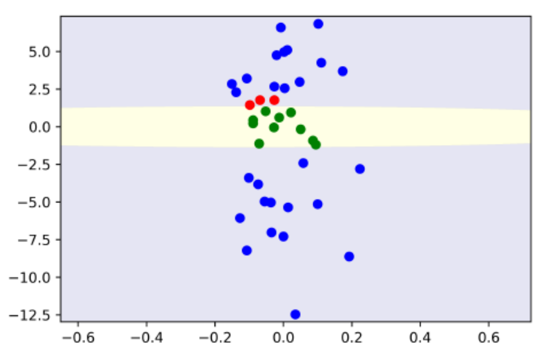
We apply the Metropolis-Hastings algorithm to sample the marginal distributions of each component of w and b, with specified parameters $C = 10$ and $B = 20$.

Bayesian SVM: Example

$$\phi_1(x) = \left(x_1, x_2, e^{-(x_1^2 + x_2^2)}\right)$$

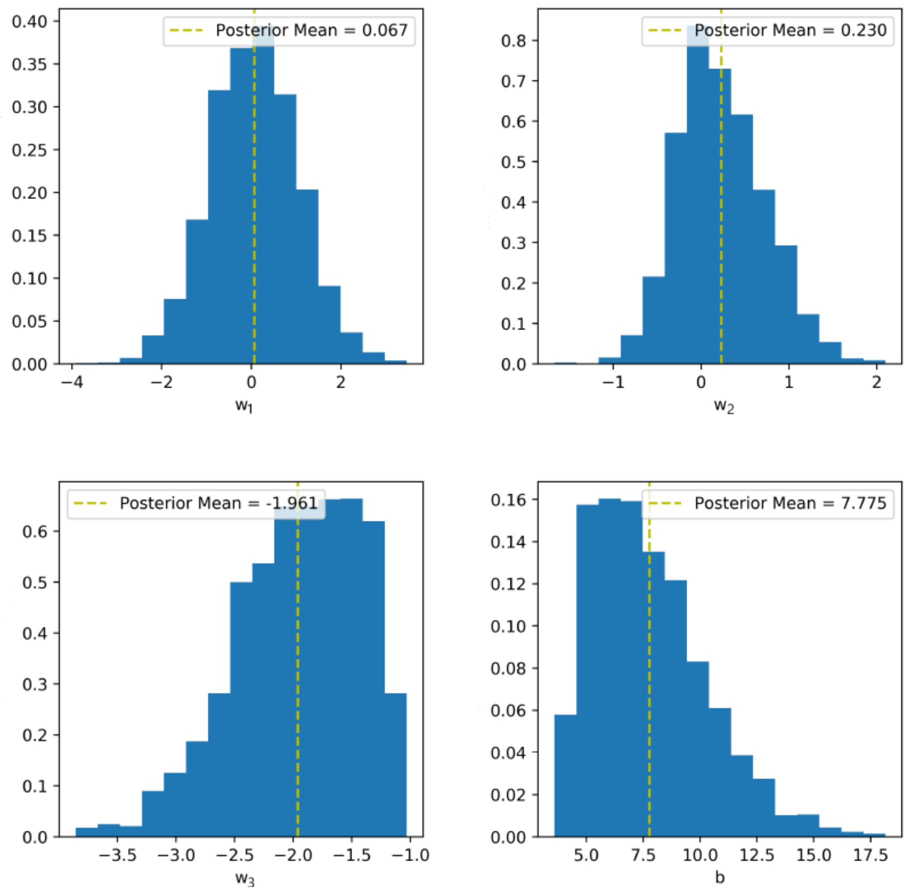


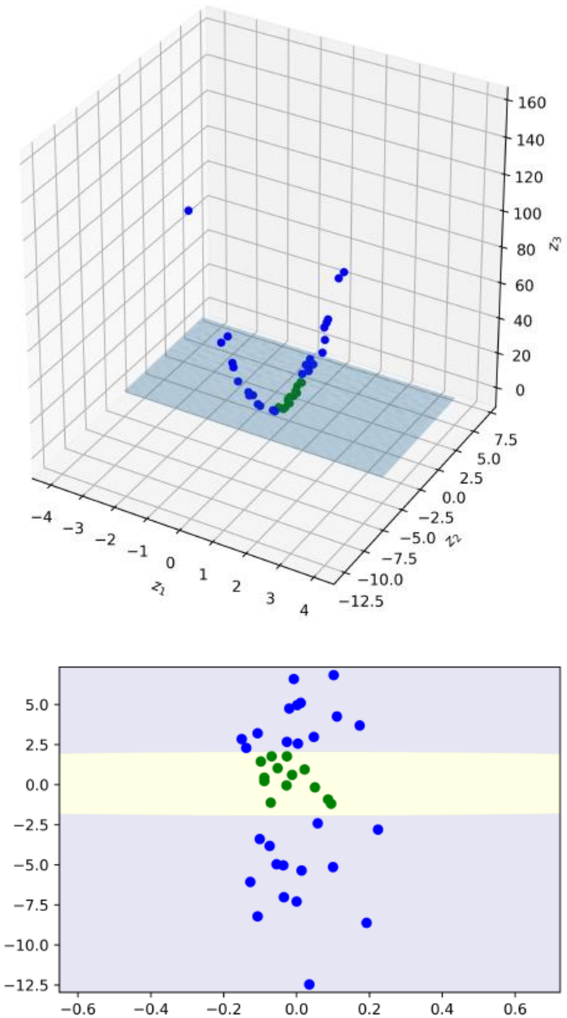




Bayesian SVM: Example

$$\phi_2(x) = (x_1, x_2, x_1^2 + x_2^2)$$





Comparison with deterministic counterpart: **Linear Regression**

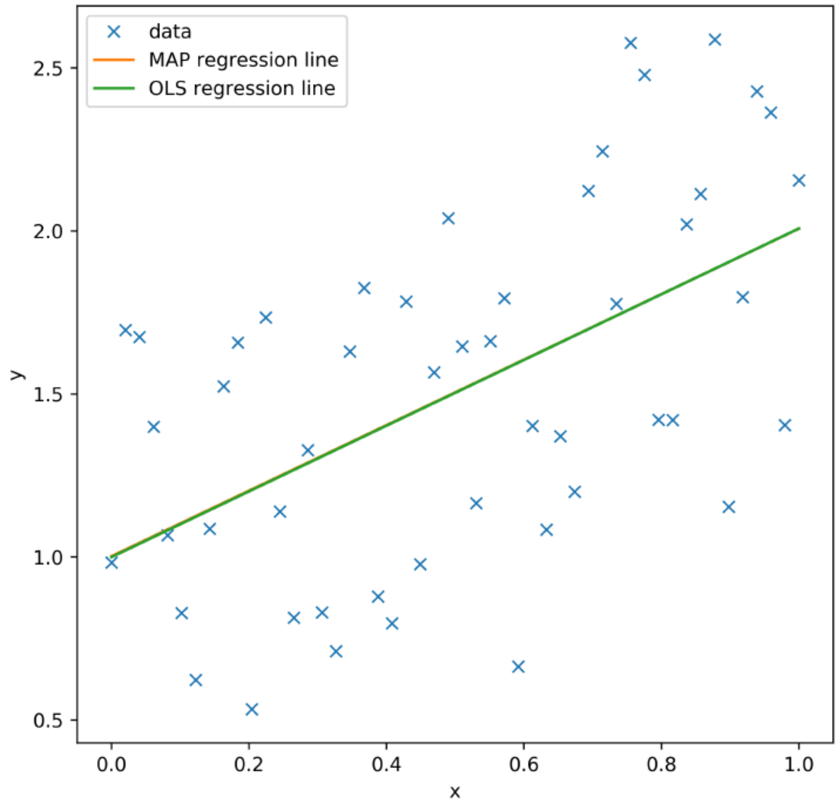


Figure 9: Comparison between OLS and MAP regression lines. Separating hyperplanes are identical with intercept-coefficient pairs (1.00767, 0.99986) for ordinary least squares and (1.00482,1.00240) for Bayesian.

Comparison with deterministic counterpart: **Support Vector Machine**

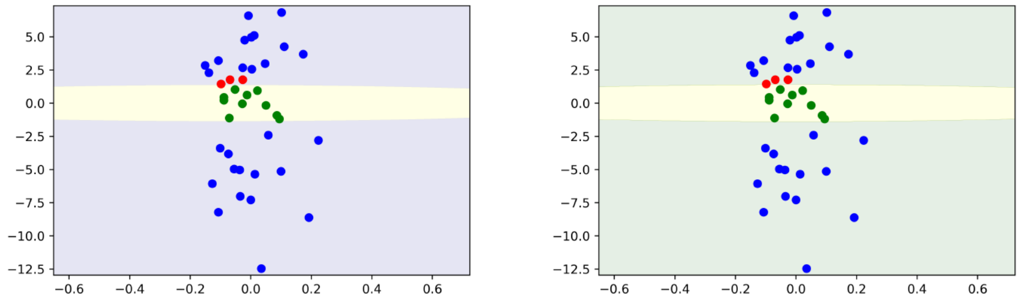


Figure 10: Resulting decision boundaries using Bayesian (left) and deterministic (right) SVMs which uses a feature map $\phi_1 : x \mapsto x \mapsto (x_1, x_2, e^{-(x_1^2+x_2^2)})$. Decision boundaries in both cases are identical both with three misclassifications of the same data points.

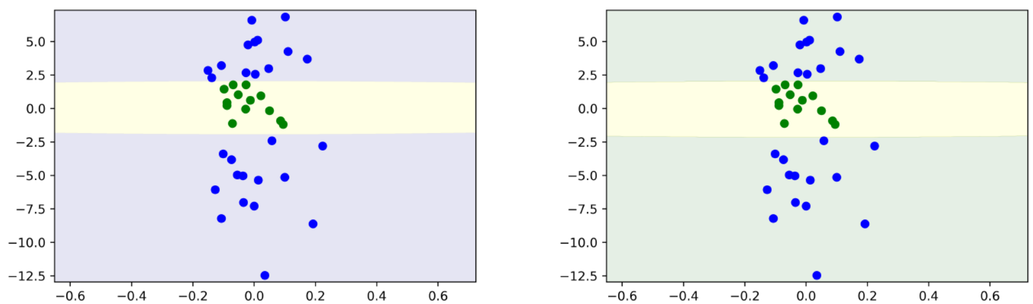


Figure 11: Resulting decision boundaries using Bayesian (left) and deterministic (right) SVMs which uses a feature map $\phi_2 : (x_1, x_2, x_1^2 + x_2^2)$. Decision boundaries in both cases are identical both performing perfect classification of the data points.

References

[1] M. Betancourt, ”A conceptual introduction to Hamiltonian Monte Carlo”, *arXiv:1701.02434v2*, 2018.

[2] A. Gelman, ”Prior distributions for variance parameters in hierarchical models”, Bayesian Analysis, 1(3), 515-533, 2006.

[3] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. Biometrika, 57:97–109, 1970.

[4] J. Kruschke, ”Doing Bayesian data analysis : a tutorial with R, JAGS, and Stan 2nd Edition”, Elsevier Inc., 2015.

[5] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, ’Equation of state calculations by fast computing machines”, *The Journal of Chemical Physics* 1953, *21*, 1087-1092.

[6] J. Salvatier, T.V. Wiecki, C. Fonnesbeck C., “Probabilistic programming in Python using PyMC3”, PeerJ Computer Science 2:e55 DOI: 10.7717/peerj-cs.55, 2016.

[7] M. Seeger, “Bayesian model selection for support vector machines, Gaussian process and other kernel classifiers.” *Advances in neural information processing systems,* 12, 603-609, Cambridge, MA. 2000.

[8] P. Solich, Bayesian Methods for Support Vector Machines: Evidence and Predictive Class Probabilities. Machine Learning, 46, 21-52, 2002.

[9] V. Vapnik, “An Overview of Statistical Learning Theory” *IEEE Transactions on Neural Network,* Vol 10, No. 5, 1999.

[10] D.J. Warne, R.E. Baker, and M.J. Simpson, ”Simulation and inference algorithms for stochastic biochemical reaction networks: from basic concepts to state-of-the-art”, *arXiv:1812.05759v1*, 2018.