ELEN E4903 Homework 1

Kliment Mamykin UNI 2770

Problem 1

Given a sequence of N observations $X = (x_1, \dots, x_N)$ where $x_i \overset{iid}{\sim} Bern(\pi)$, with p.d.f

$$p(x_i|\pi) = \pi^{x_i}(1-\pi)^{1-x_i} \tag{1}$$

(a) What is the likelihood of the data $(x_1, \dots, x_N)$?

The likelihood of data given parameters of the model (in the case of $Bern(\pi)$ the parameter is a single value π) is:

$$p(X \mid \pi) = p(x_1, \dots, x_N|\pi) = \prod_{i=1}^{N} \pi^{x_i}(1-\pi)^{1-x_i} \tag{2}$$

Where the assumption that the observations are independent allow factorization of the joint distribution into a product of individual observation probabilities.

(b) Maximum likelihood estimate $\hat{\pi}_{ml}$ for π

Maximum likelihood estimate $\hat{\pi}_{ml}$ is defined as

$$\hat{\pi}_{ml} = \arg\max_{\pi} \log p(X \mid \pi) \tag{3}$$

Here $\hat{\pi}_{ml}$ is a point estimate (a scalar), and X is treated as a r.v.. We are selecting a value π that maximizes the log probability of data given model parameters. For the given model ($x_i \sim Bern(\pi)$)

$$\begin{aligned}
\hat{\pi}_{ml} &= \arg\max_{\pi} \log p(x_1, \ldots, x_N \mid \pi) \\
&= \arg\max_{\pi} \log \prod_{i=1}^{N} \pi^{x_i} (1-\pi)^{1-x_i} \\
&= \arg\max_{\pi} \log\Big(\pi^{\sum_{i=1}^{N} x_i} (1-\pi)^{\sum_{i=1}^{N}(1-x_i)}\Big) \\
&= \arg\max_{\pi} \sum_{i=1}^{N} x_i \log \pi + \sum_{i=1}^{N} (1-x_i) \log(1-\pi) \qquad (4)
\end{aligned}$$

To find a maximunm of a function, we take a gradient with respect to the parameters of maximization, and find a point where the gradient is 0.

$$\begin{aligned}
0 &= \frac{d}{d\pi}\left(\sum_{i=1}^{N} x_i \log \pi + \sum_{i=1}^{N} (1-x_i) \log(1-\pi)\right)\Bigg|_{\hat{\pi}_{ml}} \\
&= \frac{\sum_{i=1}^{N} x_i}{\hat{\pi}_{ml}} - \frac{\sum_{i=1}^{N} (1-x_i)}{1-\hat{\pi}_{ml}} \Rightarrow \qquad (5) \\
\hat{\pi}_{ml} &= \frac{\sum_{i=1}^{N} x_i}{\sum_{i=1}^{N} x_i + \sum_{i=1}^{N} (1-x_i)} = \frac{\sum_{i=1}^{N} x_i}{N} \qquad (6)
\end{aligned}$$

(c) Maximum a posteriori (MAP) estimate $\hat{\pi}_{MAP}$ for π

MAP estimate $\hat{\pi}_{MAP}$ is defined as

$$\hat{\pi}_{MAP} = \arg\max_{\pi} \log p(\pi \mid X) \tag{7}$$

Here we consider a conditional distribution of parameters of the model, treated as a r.v. conditioned on the data X. X generally is also treated as a r.v. but for conditioning we pick one random instantiation of the dataset. Using Bayes formula we express the posterior through the likelihood, the prior and the data evidence, plug in likelihood of $Bern(\pi)$ and the prior pdf of $Beta(\alpha, \beta)$.

$$\begin{aligned}\hat{\pi}_{MAP} &= \arg\max_{\pi} \log \frac{p(X \mid \pi)p(\pi)}{p(X)} \\ &= \arg\max_{\pi} \log p(X \mid \pi) + \log p(\pi) - \log p(X) \\ &= \arg\max_{\pi} \log \prod_{i=1}^{N} \pi^{x_i}(1-\pi)^{1-x_i} + \log C\pi^{\alpha-1}(1-\pi)^{\beta-1} - \log p(X) \\ &= \arg\max_{\pi} \left((\sum_{i=1}^{N} x_i + \alpha - 1)\log\pi + (\sum_{i=1}^{N}(1-x_i) + \beta - 1)\log(1-\pi) + \log C - \log p(X) \right) \quad (8\end{aligned}$$

$$\begin{aligned}0 &= \frac{d}{d\pi}\left((\sum_{i=1}^{N} x_i + \alpha - 1)\log\pi + (\sum_{i=1}^{N}(1-x_i) + \beta - 1)\log(1-\pi) \right)\Big|_{\hat{\pi}_{MAP}} \\ &= \frac{\sum_{i=1}^{N} x_i + \alpha - 1}{\hat{\pi}_{MAP}} - \frac{\sum_{i=1}^{N}(1-x_i) + \beta - 1}{1-\hat{\pi}_{MAP}} \end{aligned} \tag{9}$$

$$\hat{\pi}_{MAP} = \frac{\sum_{i=1}^{N} x_i + \alpha - 1}{\sum_{i=1}^{N} x_i + \alpha - 1 + \sum_{i=1}^{N}(1-x_i) + \beta - 1} = \frac{\sum_{i=1}^{N} x_i + \alpha - 1}{N + \alpha + \beta - 2} \tag{10}$$

(d) Posterior distribution of π

$$\begin{aligned}
x_i \sim Bernoulli(\pi) &\Rightarrow p(X \mid \pi) = \prod_{i=1}^{N} \pi^{x_i}(1-\pi)^{1-x_i} \\
\pi \sim Beta(\alpha, \beta) &\Rightarrow p(\pi) = C\pi^{\alpha-1}(1-\pi)^{\beta-1} \\
p(\pi \mid X) &= \frac{p(X \mid \pi)p(\pi)}{p(X)} \propto \pi^{\sum_{i=1}^{N} x_i+\alpha-1}(1-\pi)^{\sum_{i=1}^{N}(1-x_i)+\beta-1} \Rightarrow \\
p(\pi \mid X) &= Beta(\alpha', \beta') \qquad \alpha' = \sum_{i=1}^{N} x_i + \alpha \quad \beta' = \sum_{i=1}^{N}(1-x_i) + \beta \qquad (11)
\end{aligned}$$

Distribution families sich that when multiplied by a likelihood function produce a re-parameterized distribution from the same family are called conjugate priors. As shown in (11) $Beta(\alpha, \beta)$ is a conjugate prior to $Bern(\pi)$ likelihood. Some other examples: Gaussian distribution is a conjugate prior for the mean in a Gaussian likelihood, Gamma is a prior for Gaussian precision parameter [3].

(e) Mean and variance of π under posterior distribution, relationship to $\hat{\pi}_{MAP}$ and $\hat{\pi}_{ml}$

From (11) we have that posterior $p(\pi) = Beta(\alpha', \beta')$ from which follows (using Beta distribution mean and variance from [2])

$$E[\pi] = \frac{\alpha'}{\alpha' + \beta'} = \frac{\sum_{i=1}^{N} x_i + \alpha}{N + \alpha + \beta} \tag{12}$$

$$Var[\pi] = \frac{\alpha' \beta'}{(\alpha' + \beta')^2(\alpha' + \beta' + 1)} \tag{13}$$

An interpretation of (12) is that the parameters of the prior encode prior belief on the number of successes (α) and failures (β) seen (or believed to be seen) so far, also called *pseudo observations*. And the parameters of the posterior represent an updated belief of the number of success (α') and failues (β') after seeing more data.

Comparing (6) and (10) implies that $\hat{\pi}_{MAP}$ becomes $\hat{\pi}_{ml}$ when prior's parameters $\alpha = \beta = 1$, so the Baysian MAP estimator is a generalization of the maximum likelihood approach with the prior selected as $Beta(1, 1)$

MAP estimator $\hat{\pi}_{MAP}$ picks the mode in the posterior distribution, a point with the highest density. However the follwing derivations show that MAP estimator is biased.

Calculate $E(\hat{\pi}_{ml})$ and $Var(\hat{\pi}_{ml})$ from (6) considering linearity of expectation of a sum of r.v.'s $E(\sum_{i=1}^{N} a_i X_i) = \sum_{i=1}^{N} a_i E[X_i]$ and for independent r.v.s $Var(\sum_{i=1}^{N} a_i X_i + b) = \sum_{i=1}^{N} a_i^2 Var[X_i]$ and $Var(X \sim Bern(\pi)) = \pi(1 - \pi)$

$$E[\hat{\pi}_{ml}] = E\left[\frac{\sum_{i=1}^{N} x_i}{N}\right] = \frac{\sum_{i=1}^{N} E[x_i]}{N} = \frac{N\pi}{N} = \pi \tag{14}$$

from which we can conclude that $E[\hat{\pi}_{ml}]$ is an unbiased estimate of π with variance

$$Var[\hat{\pi}_{ml}] = Var\left[\frac{\sum_{i=1}^{N} x_i}{N}\right] = \frac{1}{N^2}\sum_{i=1}^{N} Var[x_i] = \frac{1}{N^2} N\pi(1 - \pi) = \frac{1}{N}\pi(1 - \pi) \tag{15}$$

And the same calculations for $\hat{\pi}_{MAP}$ yield:

$$E[\hat{\pi}_{MAP}] = E\left[\frac{\sum_{i=1}^{N} x_i + \alpha - 1}{N + \alpha + \beta - 2}\right] = \frac{\sum_{i=1}^{N} E[x_i] + \alpha - 1}{N + \alpha + \beta - 2} = \frac{N\pi + \alpha - 1}{N + \alpha + \beta - 2} = \frac{\pi + \frac{\alpha - 1}{N}}{1 + \frac{\alpha + \beta - 2}{N}} \tag{16}$$

indicating that $E[\hat{\pi}_{MAP}]$ is a biased estimate of π with variance

$$Var[\hat{\pi}_{MAP}] = Var\left[\frac{\sum_{i=1}^{N} x_i + \alpha - 1}{N + \alpha + \beta - 2}\right] = \frac{1}{(N + \alpha + \beta - 2)^2}\sum_{i=1}^{N} Var[x_i] = \frac{N}{(N + \alpha + \beta - 2)^2}\pi(1 -$$

$Var[\hat{\pi}_{MAP}] \leq Var[\hat{\pi}_{ml}]$ implies that on average MAP estimate will be closer to the true but unknown value of π, assuming of course that the prior was selected correctly.

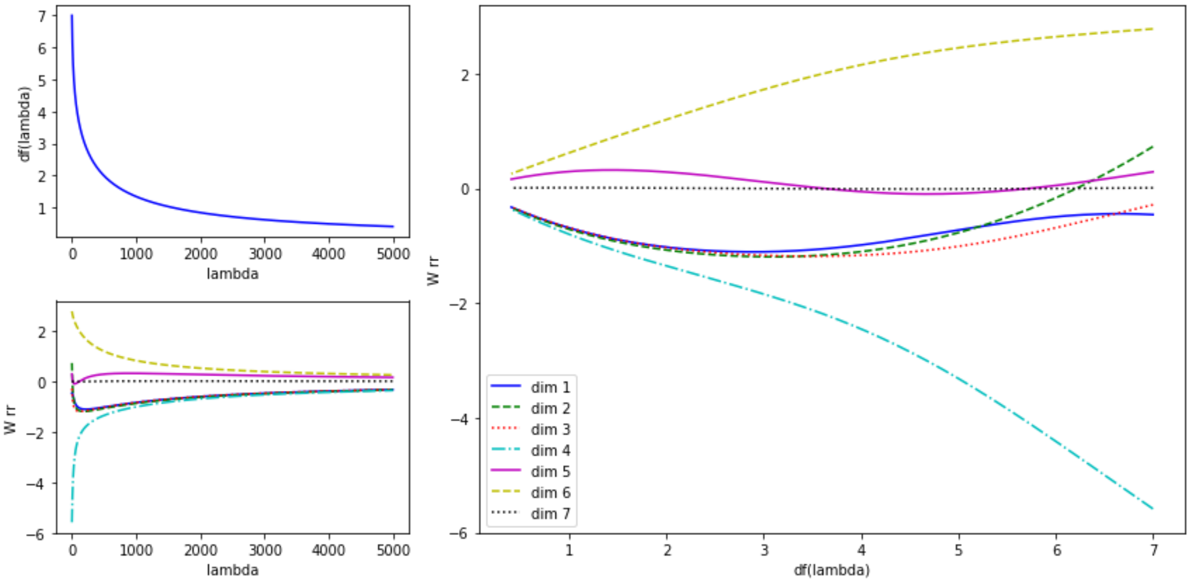
Problem 2

```
In [17]: %load_ext autoreload
         %autoreload 2
         %matplotlib inline
         from hw1 import regression
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

Part 1 (a): Weights vs $df(\lambda)$ plot

```
In [18]: regression.part1a()
```



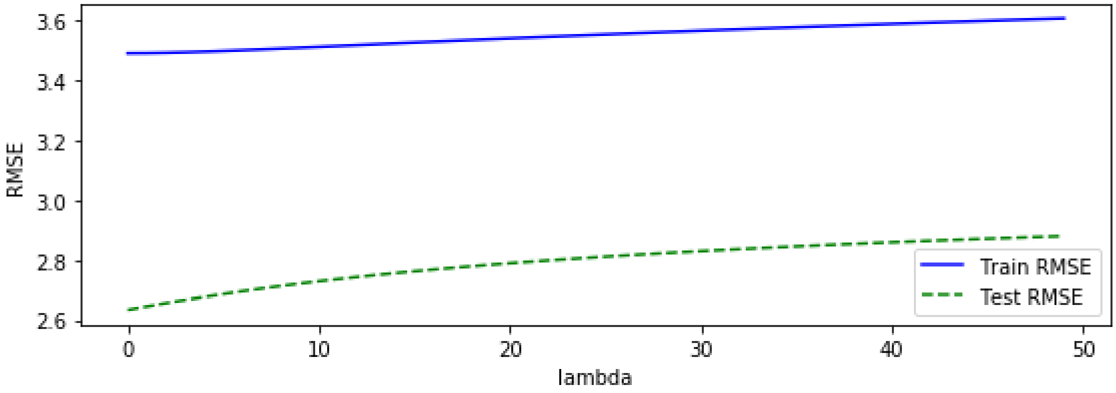
Two figures on the left are added for my own understanding of the relationship between lambda and df(lambda), and between lambda and Wrr.

Part 1 (b): Why dim 4 and 6 clearly stand out on the plot?

When degrees of freedom = 7, which corresponds to $\lambda = 0$ or no regularization, the magnitude of weights, especially for dim 2 and dim 6 are larger then the other dimentions. This means that dim 2 and dim 6 are more correlated to the outcome of the model when the magnitude of the weights is not penalized. Without regularization, the wights are allowed to grow unconstrained, as long as the model fits training data well. With ridge regularization, as λ increases and $df(\lambda)$ decreases, the magnitudes of all weights are shrunk to small values (left side of the plot).

Part 1 (c): RMSE vs λ plot

```
In [19]: regression.part1c()
```

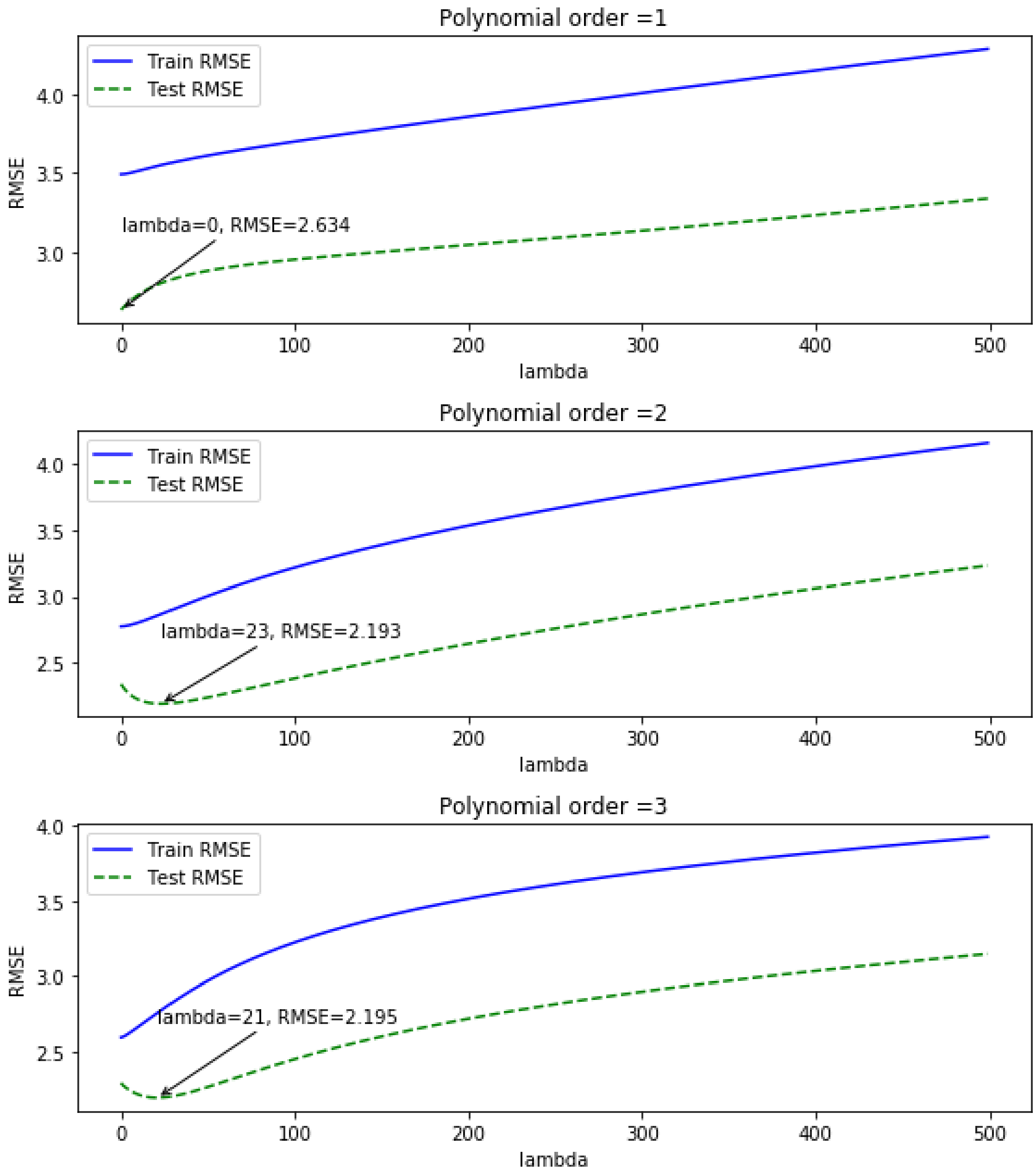


On the plot we can see that Root Mean Squared Error is monotonicaly increasing for both train and test set when lambda increases. That indicates that the optimal lambda in this case is 0, which is the same solution as no regularization and equal to the least squares solution.

My explanation is that for a simple linear model without expanding features (using polynomials or other functions) does not have the capacity to overfit the training set (it is a hyperplane), and by imposing a regularization on the weights we can only make the testing error larger, and as such is unnecessary in this case.

Part 2 (d): Polynomial features model

```
In [20]: regression.part2d()
```



With polynomial expansions (second and third plot) the capacity of the model to fit and overfit the training set increases. Here regularization becomes important, and we see that there is a dip in the testing RMSE plot for polynomial order 2 and 3. The arrow annotations indicate the optimal lambda with the minimum RMSE on the testing set.

Based on the minimal values of RMSE we have that polynomial order 2 features get a slightly smaller RMSE = 2.193, then polynomial order 3 with RMSE = 2.195. Given this, the polynomial order 2 is prefered, because RMSE is slightly less on the training set and more importantly, the model is less complex and will probably generalize better on the unseed data.

References

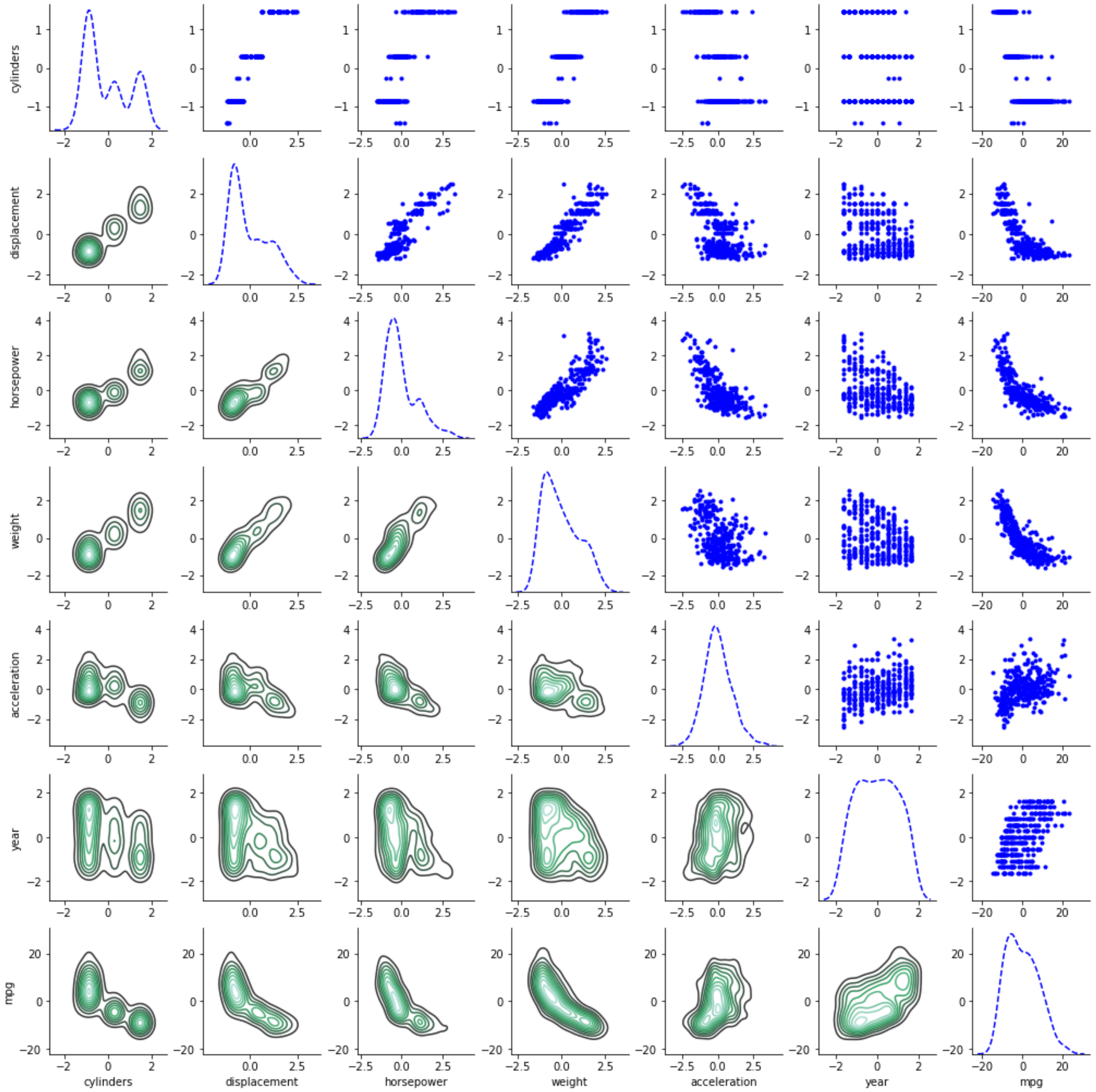
1. Lars Buitinck et all, "API design for machine learning software: experiences from the scikit-learn project", https://arxiv.org/pdf/1309.0238.pdf (https://arxiv.org/pdf/1309.0238.pdf)
2. Beta Distribution, Wikipedia, https://en.wikipedia.org/wiki/Beta_distribution (https://en.wikipedia.org/wiki/Beta_distribution)
3. Conjugate Prior, Wikipedia, https://en.wikipedia.org/wiki/Conjugate_prior (https://en.wikipedia.org/wiki/Conjugate_prior)
4. Source Code for this report, https://github.com/kmamykin/ELENE4903 (https://github.com/kmamykin/ELENE4903)

Appendix

Data visualization of the training dataset using pairplot with scatter plot in the upper dioganal and kernel density on the lower diagonal.

```
In [21]: regression.pairplot(regression.make_dataframe(X_train, y_train))
```

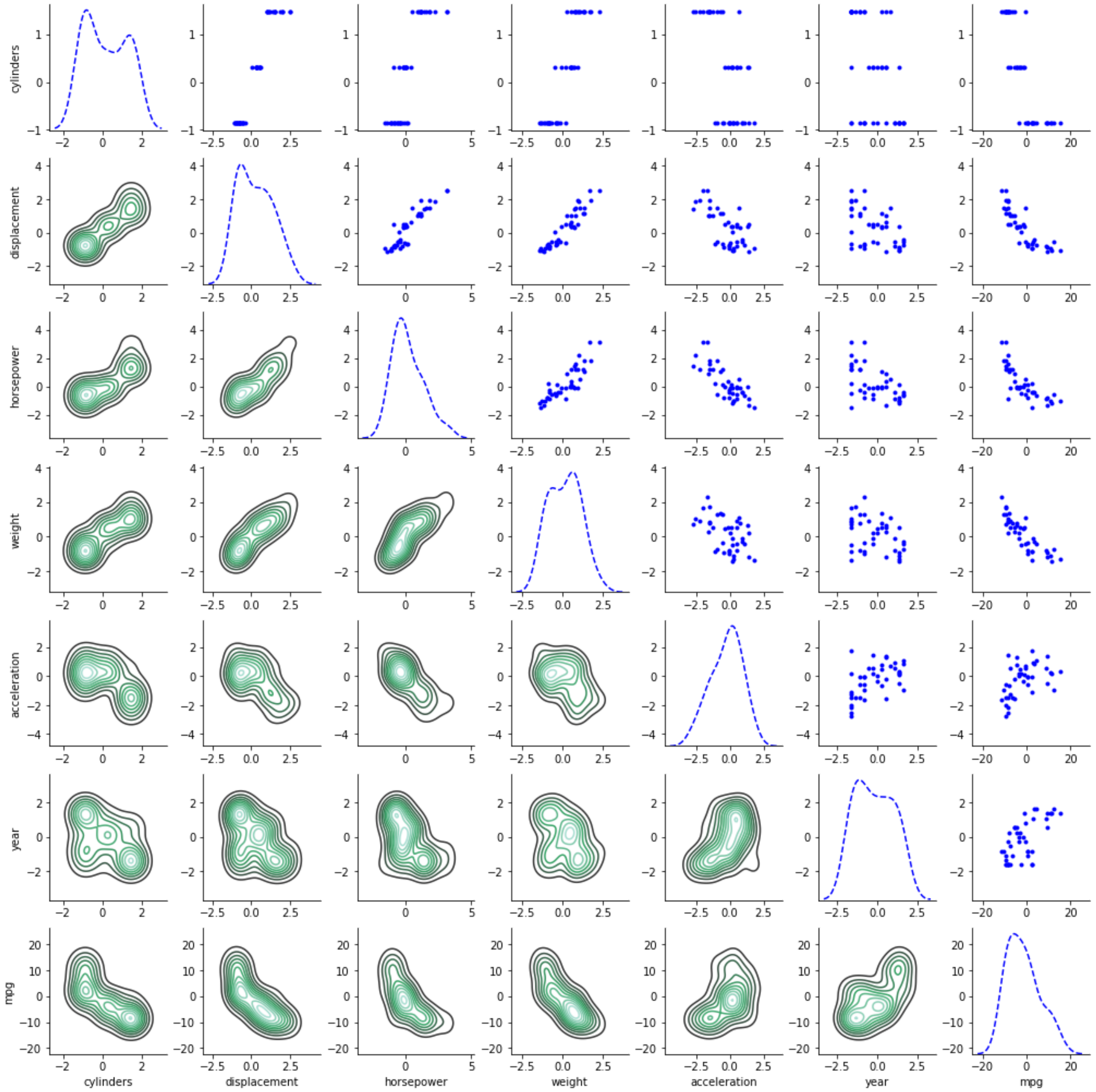
```
/Users/kmamykin/anaconda3/envs/hearts/lib/python3.6/site-packages/matpl
otlib/contour.py:967: UserWarning: The following kwargs were not used b
y contour: 'label', 'color'
  s)
```



Similar visualization for the test dataset

```
In [22]: regression.pairplot(regression.make_dataframe(X_test, y_test))
```

```
/Users/kmamykin/anaconda3/envs/hearts/lib/python3.6/site-packages/matpl
otlib/contour.py:967: UserWarning: The following kwargs were not used b
y contour: 'label', 'color'
  s)
```


cylinders
displacement
horsepower
weight
acceleration
year
mpg