

CPSC 440: Advanced Machine Learning

Bayesian Regression

Mark Schmidt

University of British Columbia

Winter 2022

Last Time: L2-Regularized Least Squares and Gaussians

- We started discussing **regression**:
 - Supervised learning with a **continuous output** y^i .
- **Linear regression** models make predictions using $\hat{y}^i = w^T x^i$.
 - For regression weights w .
- A common training objective is L2-regularized least squares,

$$\operatorname{argmin}_w \frac{1}{2\sigma^2} \|Xw - y\|^2 + \frac{\lambda}{2} \|w\|^2.$$

- This corresponds to MAP estimation with a **Gaussian likelihood and prior**,

$$y^i \sim \mathcal{N}(w^T x^i, \sigma^2), \quad w_j \sim \mathcal{N}(0, \lambda^{-1}).$$

- The unique MAP estimate is given by:

$$w_{\text{MAP}} = \frac{1}{\sigma^2} \left(\frac{1}{\sigma^2} X^T X + \lambda I \right)^{-1} X^T y.$$

Bayesian Linear Regression

- Consider linear regression with **Gaussian likelihood and prior**,

$$y^i \sim \mathcal{N}(w^T x^i, \sigma^2), \quad w_j \sim \mathcal{N}(0, \lambda^{-1}).$$

- By some tedious Gaussian identities, the **posterior** has the form

$$w \mid X, y \sim \mathcal{N} \left(w_{\text{MAP}}, \left(\frac{1}{\sigma^2} X^T X + \lambda I \right)^{-1} \right),$$

which is a **Gaussian centered at the MAP** estimate.

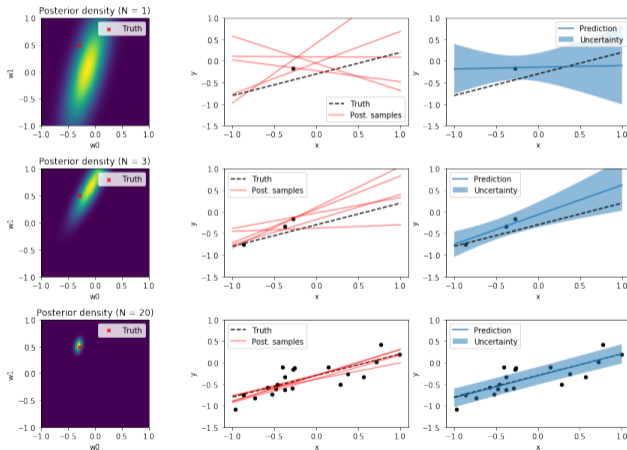
- The variance tells us **how much variation** we have around the MAP estimate.
 - Note that **in other models the MAP is usually not the mean** of the posterior.
- By more tedious Gaussian identities the **posterior predictive** has the form

$$\tilde{y} \mid X, y, \tilde{x} \sim \mathcal{N}(w_{\text{MAP}}^T \tilde{x}, \sigma^2 + \tilde{x}^T \left(\frac{1}{\sigma^2} X^T X + \lambda I \right)^{-1} \tilde{x}).$$

- Decoding in posterior predictive gives MAP predictions (special for Gaussians).
 - But working with the full posterior predictive gives us **variance of predictions**.

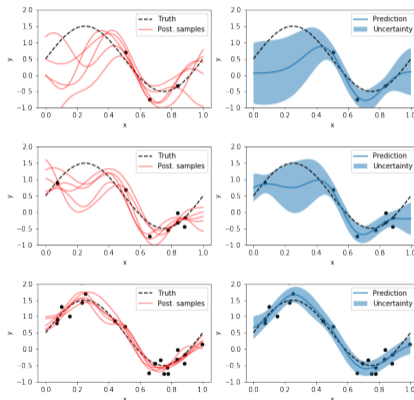
Bayesian Linear Regression

- Bayesian perspective gives us **variability in w and predictions**:



Bayesian Linear Regression

- Bayesian linear regression with Gaussian RBFs as features:



<http://krasserm.github.io/2019/02/23/bayesian-linear-regression>

- We have not only a prediction, but Bayesian inference gives “error bars”.
 - Gives an idea of “where model is confident” and where it is not.

Digression: Gaussian Processes

- In CPSC 340 you may have seen the **kernel trick**:
 - Re-writes L2-regularized least squares linear/prediction in terms of inner products.
 - Allows us to efficiently use some exponential-sized or infinite-sized feature sets.
- We can use **kernel trick on posterior** in Gaussian likelihood/prior model.
 - Allows us to efficiently use some exponential-sized or infinite-sized feature sets.
 - Posterior in this case can be written as a **Gaussian process (GP)**.
- Notation: a **stochastic process** is an **infinite collection of random variables**.
- Gaussian process is a stochastic process where **any finite sample is Gaussian**.
 - Defined in terms of a **mean function** and a **covariance function**.
 - The set of **possible covariance functions is the set of possible kernel functions**.
 - A popular book on this topic if you want to read more:
 - <http://www.gaussianprocess.org/gpml/chapters/RW.pdf>
- We will **assume we have explicit features**, but you could kernels/GPs instead.

Setting Hyper-Parameters with Empirical Bayes

- To set hyper-parameters like σ^2 and λ , we could use a validation set.
- But could also use **empirical Bayes** and optimize the **marginal likelihood**,

$$\hat{\sigma}^2, \hat{\lambda} \in \underset{\sigma^2, \lambda}{\operatorname{argmax}} p(y | X, \sigma^2, \lambda).$$

- The **marginal likelihood integrates** over the parameters w ,

$$p(y | X, \sigma^2, \lambda) = \int_w p(y, w | X, \sigma^2, \lambda) dw = \int_w p(y | X, w, \sigma^2) p(w | \lambda) dw \quad (w \perp X).$$

- This is the marginal in a product of Gaussians, which is (with some work):

$$p(y | X, \sigma^2, \lambda) = \frac{(\lambda)^{d/2}}{(\sigma\sqrt{2\pi})^n | \frac{1}{\sigma^2} X^T X + \lambda I |^{1/2}} \exp \left(-\frac{1}{2\sigma^2} \|X w_{\text{MAP}} - y\|^2 - \frac{\lambda}{2} \|w_{\text{MAP}}\|^2 \right)$$

- You could **run gradient descent** on the negative log of this to set hyper-parameters.
 - You could do “projected” gradient to handle parameters with constraints.

Setting Hyper-Parameters with Empirical Bayes

- Consider having a hyper-parameter λ_j for each w_j ,

$$y^i \sim \mathcal{N}(w^T x^i, \sigma^2), \quad w_j \sim \mathcal{N}(0, \lambda_j^{-1}).$$

- Too expensive for cross-validation, but can still do empirical Bayes.
 - You can do projected **gradient descent to optimize the λ_j** .
- Weird fact: this yields **sparse** solutions.
 - It can send some $\lambda_j \rightarrow \infty$, concentrating posterior for w_j at exactly 0.
 - This is L2-regularization, but **empirical Bayes naturally encourages sparsity**.
 - “Automatic relevance determination” (ARD)
- Non-convex and theory not well understood:
 - Tends to yield much sparser solutions than L1-regularization.

Setting Hyper-Parameters with Empirical Bayes

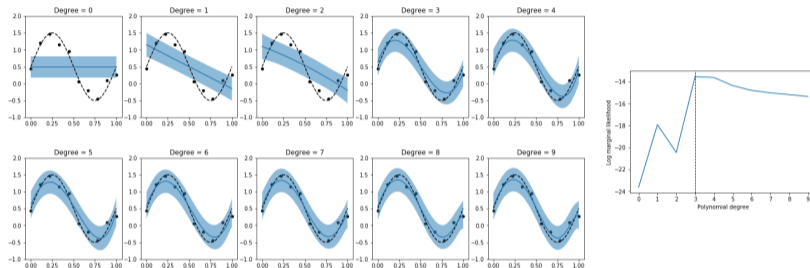
- Consider also having a hyper-parameter σ_i for each i ,

$$y^i \sim \mathcal{N}(w^T x^i, \sigma_i^2), \quad w_j \sim \mathcal{N}(0, \lambda_j^{-1}).$$

- You can also use empirical Bayes to optimize these hyper-parameters.
- The “automatic relevance determination” selects training examples ($\sigma_i \rightarrow \infty$).
 - This is like the support vectors in SVMs, but tends to be much more sparse.
- Type II MLE can also be used to learn kernel parameters like RBF variance.
 - Do gradient descent on the σ values in the Gaussian kernel.
- Bonus slides: Bayesian feature selection gives probability that w_j is non-zero.
 - Posterior can be more informative than standard sparse MAP methods.

Choosing Polynomial Degree with Empirical Bayes

- Using empirical Bayes to choose degree hyper-parameter with polynomial basis:



<http://krasserm.github.io/2019/02/23/bayesian-linear-regression>

- Marginal likelihood (“evidence”) is highest for degree 3.
 - “Bayesian Occam’s Razor”: prefers simpler models that fit data well.
 - $p(y | X, \sigma^2, \lambda, k)$ is smaller for degree 4 polynomials since they can fit more datasets.
 - It’s actually **non-monotonic** it prefers degree 1 and 3 over degree 2.
 - Model selection criteria like BIC are approximations to marginal likelihood as $n \rightarrow \infty$.

Choosing Polynomial Degree with Empirical Bayes

- Why is the marginal likelihood **higher for degree 3 than 7?**
 - Marginal likelihood for degree 3 (ignoring conditioning on hyper-parameters):

$$p(y | X) = \int_{w_0} \int_{w_1} \int_{w_2} \int_{w_3} p(y | X, w) p(w | \lambda) dw$$

- Marginal likelihood for degree 7:

$$p(y | X) = \int_{w_0} \int_{w_1} \int_{w_2} \int_{w_3} \int_{w_4} \int_{w_5} \int_{w_6} \int_{w_7} p(y | X, w) p(w | \lambda) dw.$$

- Higher-degree integrates over high-dimensional volume:
 - A non-trivial **proportion** of degree 3 functions fit the data really well.
 - There are many degree 7 functions that fit the data even better, but they are a **much smaller proportion** of all degree 7 functions.

Choosing Between Bases with Empirical Bayes

- We could compare **marginal likelihood between different non-linear transforms**:

$$p(y | X, \text{polynomial basis}) > p(y | X, \text{Gaussian RBF as basis})?$$

- This is the idea behind **Bayes factors** for hypothesis testing (see bonus slides).
 - Alternative to classic hypothesis tests like t-tests.
- Usual warning: empirical Bayes can sometimes become degenerate.
 - May **need a non-vague prior on the hyper-parameters**.
- But we could have a **hyper-prior over possible non-linear transformations**.
 - And use empirical Bayes in this hierarchical model to learn basis and parameters.

Application: Automatic Statistician

- Can be viewed as an **automatic statistician**:
<http://www.automaticstatistician.com/examples>

An automatic report for the dataset : 01-airline

The Automatic Statistician

Abstract

This report was produced by the Automatic Bayesian Covariance Discovery (ABCD) algorithm.

1 Executive summary

The raw data and full model posterior with extrapolations are shown in figure 1.

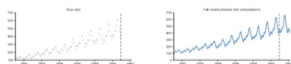


Figure 1: Raw data (left) and model posterior with extrapolation (right)

The structure search algorithm has identified four additive components in the data. The first 2 additive components explain 98.5% of the variation in the data as shown by the coefficient of determination (R^2) values in table 1. The first 3 additive components explain 99.8% of the variation in the data. After the first 3 components the cross validated mean absolute error (MAE) does not

#	R^2 (%)	ΔR^2 (%)	Residual R^2 (%)	Cross validated MAE	Reduction in MAE (%)
-	-	-	-	280.30	-
1	85.4	85.4	85.4	34.03	87.9
2	98.5	13.2	89.9	12.44	63.4
3	99.8	1.3	85.1	9.10	26.8
4	100.0	0.2	100.0	9.10	0.0

Table 1: Summary statistics for cumulative additive fits to the data. The residual coefficient of determination (R^2) values are computed using the residuals from the previous fit as the target values; this measures how much of the residual variance is explained by each new component. The mean absolute error (MAE) is calculated using 10 fold cross validation with a contiguous block design; this measures the ability of the model to interpolate and extrapolate over moderate distances. The model is fit using the full data and the MAE values are calculated using this model; this double use of data means that the MAE values cannot be used reliably as an estimate of out-of-sample predictive performance.

2 Detailed discussion of additive components

2.1 Component 1 : A linearly increasing function

This component is linearly increasing.
 This component explains 85.4% of the total variance. The addition of this component reduces the cross validated MAE by 87.9% from 280.3 to 34.0.

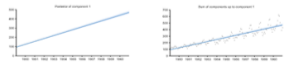


Figure 2: Pointwise posterior of component 1 (left) and the posterior of the cumulative sum of components with data (right)

from 34.03 to 12.44.

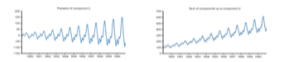


Figure 4: Pointwise posterior of component 2 (left) and the posterior of the cumulative sum of components with data (right)


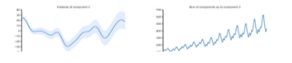


Figure 5: Pointwise posterior of residuals after adding component 2

2.3 Component 3 : A smooth function

This component is a smooth function with a typical lengthscale of 8.1 months.
 This component explains 85.1% of the residual variance; this increases the total variance explained from 98.5% to 99.8%. The addition of this component reduces the cross validated MAE by 26.81% from 12.44 to 9.10.



Outline

- 1 Bayesian Linear Regression
- 2 Rejection and Importance Sampling

Motivation: Bayesian Logistic Regression

- A classic way to fit a binary classifier is **L2-regularized logistic loss**,

$$\hat{w} \in \operatorname{argmax}_w \sum_{i=1}^n \log(1 + \exp(-y^i w^T x^i)) + \frac{\lambda}{2} \|w\|^2.$$

- This corresponds to using a sigmoid likelihood and Gaussian prior,

$$p(y^i | x^i, w) = \frac{1}{1 + \exp(-y^i w^T x^i)}, \quad w_j \sim \mathcal{N}\left(0, \frac{1}{\lambda}\right).$$

- In **Bayesian logistic regression**, we would work with the posterior.
 - But the posterior is not a Gaussian, so this is **not a conjugate prior**.
 - We do not have a nice expression for the posterior predictive or marginal likelihood.

Motivation: Monte Carlo for Bayesian Logistic Regression

- Posterior predictive in Bayesian logistic regression has the form

$$\begin{aligned} p(\tilde{y}^i | \tilde{x}^i, X, y, \lambda) &= \int_w p(\tilde{y}^i | \tilde{x}^i, w) p(w | y, X, \lambda) dw \\ &= \mathbb{E}[p(\tilde{y}^i | \tilde{x}^i, w)]. \end{aligned}$$

- If we could sample from the **posterior**, we could compute this with **Monte Carlo!**
 - But we **do not know how to generate IID samples from this posterior.**
- We will later cover **MCMC**, which is a standard method in scenarios like this.
- Today we will cover simpler **rejection sampling** and **importance sampling**.
 - These assume you can **generate from a simple distribution q** (like a Gaussian).
 - But you really want to solve an integral for a **complicated distribution p** .
 - Like the posterior for Bayesian logistic regression.

Previously: Rejection Sampling to Compute Conditionals

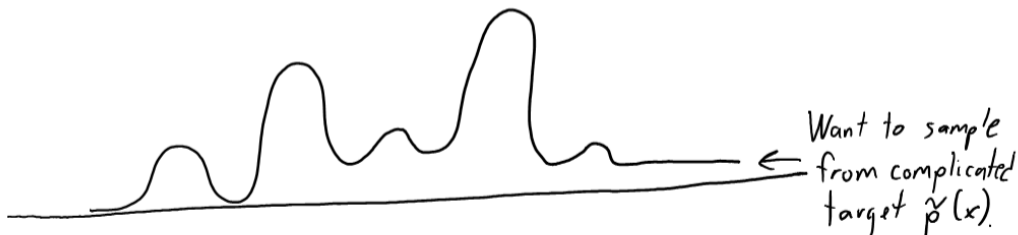
- We have already discussed **rejection sampling to do conditional sampling**:
 - Example: sampling from a Gaussian subject to $x \in [-1, 1]$.



- Generate Gaussian samples, throw out (“reject”) the ones that aren’t in $[-1, 1]$.
 - The remaining samples will follow the conditional distribution.
- Can be used to **generate IID samples from conditional** distributions.

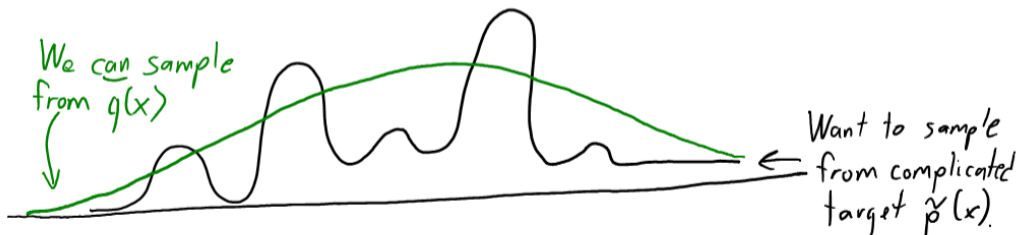
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to “sample area under the graph”:



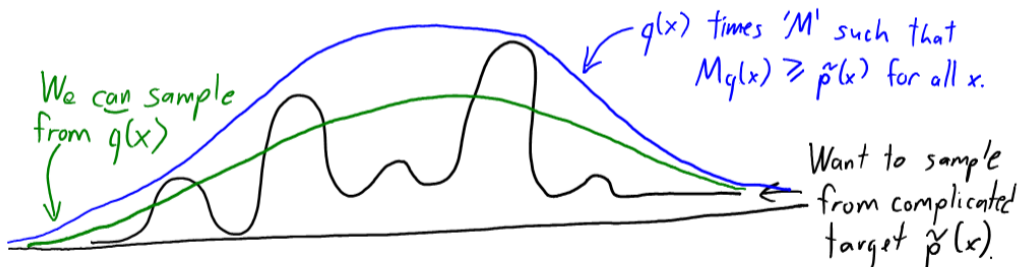
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to “sample area under the graph”:



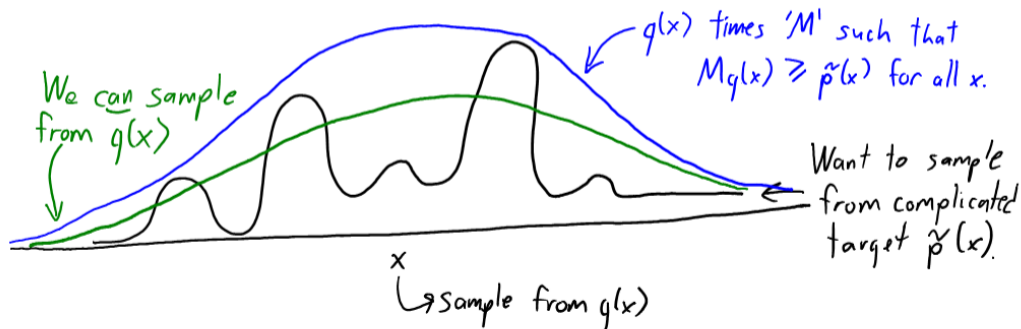
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to “sample area under the graph”:



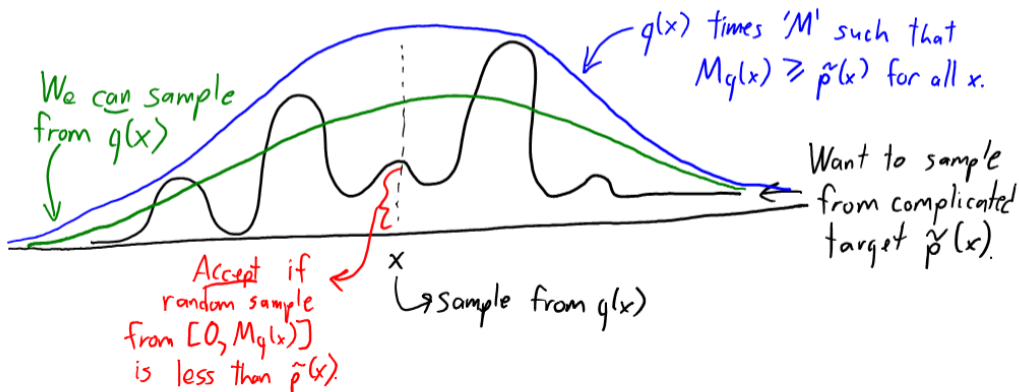
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to “sample area under the graph”:



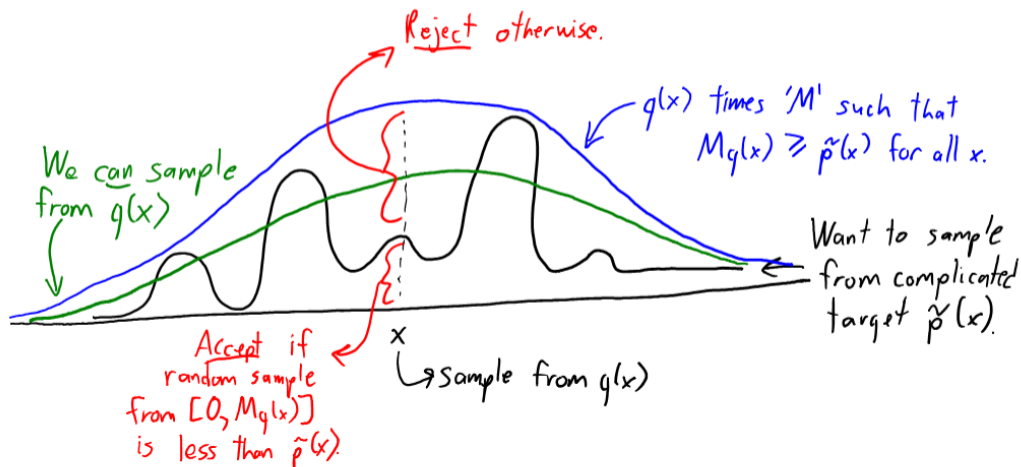
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to “sample area under the graph”:



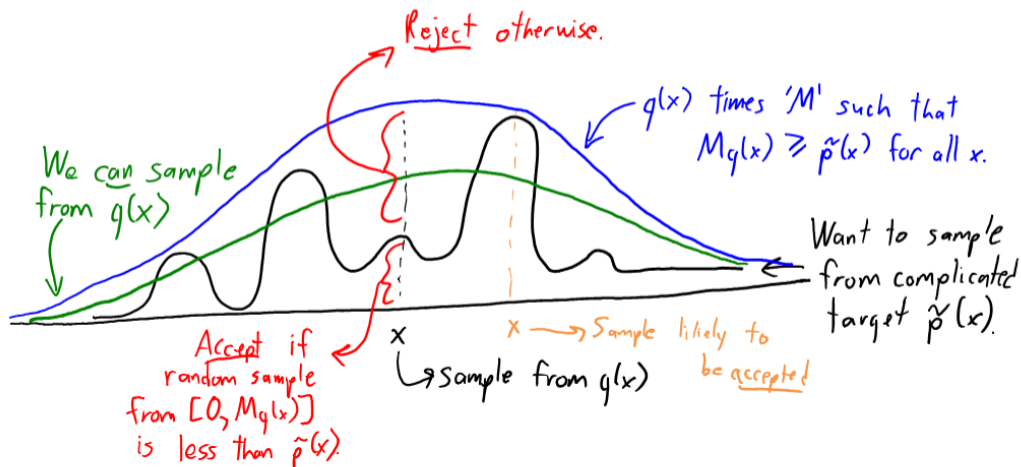
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to "sample area under the graph":



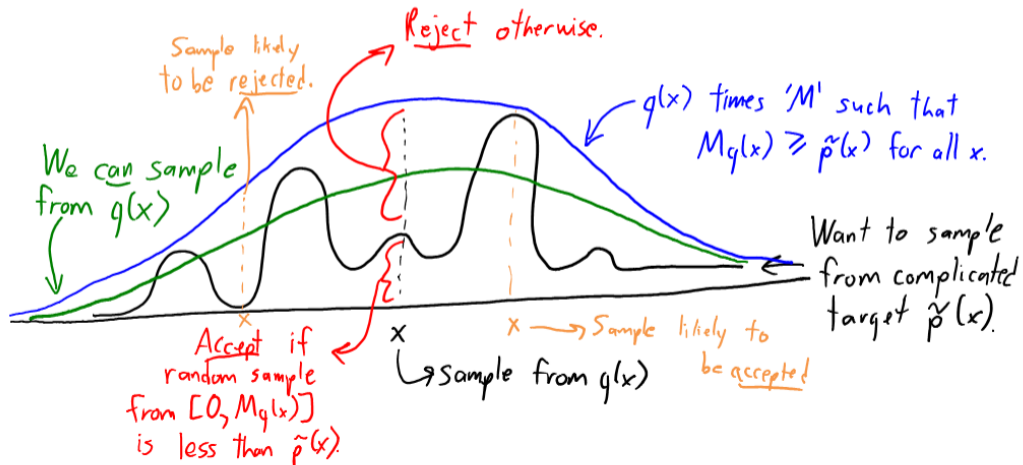
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to "sample area under the graph":



General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to “sample area under the graph”:



General Rejection Sampling Algorithm

- Ingredients of the general **rejection sampling** algorithm:

- ① Ability to evaluate unnormalized $\tilde{p}(x)$,

$$p(x) = \frac{\tilde{p}(x)}{Z}.$$

- ② A distribution q that is easy to sample from.
- ③ An **upper bound** M on $\tilde{p}(x)/q(x)$.

- **Rejection sampling** algorithm:

- ① Sample x from $q(x)$.
- ② Sample u from $\mathcal{U}(0, 1)$.
- ③ Keep the sample if $u \leq \frac{\tilde{p}(x)}{Mq(x)}$.

- The accepted samples will be from $p(x)$.

General Rejection Sampling Algorithm

- For Bayesian logistic regression, we could use rejection sampling as follows:
 - Sample from prior to sample from posterior ($M = 1$ for discrete x),

$$\tilde{p}(\theta | x) = \underbrace{p(x | \theta)}_{\leq 1} p(\theta),$$

- Would tend to accept high-likelihood samples and reject low-likelihood samples.
- Drawbacks of rejection sampling:
 - You **need to know a bound M** on $q(x)/p(x)$ (may be hard/impossible to find).
 - If x is unbounded and p has heavier tails than q , no M exist.
 - You may **reject a large number of samples**.
 - Most samples are rejected for high-dimensional complex distributions.
- If $-\log p(x)$ is **convex** and x is 1D there is a fancier version:
 - **Adaptive rejection sampling** refines piecewise-linear q after each rejection.

Importance Sampling

- Importance sampling is a variation that accepts all samples.
 - Reasoning behind importance sampling:

$$\begin{aligned}\mathbb{E}_p[f(x)] &= \sum_x p(x)f(x) \\ &= \sum_x q(x) \frac{p(x)}{q(x)} f(x) \\ &= \mathbb{E}_q \left[\frac{p(x)}{q(x)} f(x) \right] \approx \frac{1}{n} \sum_{i=1}^n \frac{p(x)}{q(x)} f(x),\end{aligned}$$

where the last line uses Monte Carlo approximation with IID samples from q .

- Replace sum over x with integral for continuous distributions.
- We can sample from q but reweight by $p(x)/q(x)$ to compute expectation.
- Only assumption is that q is non-zero when p is non-zero.
- If you only know unnormalized $\tilde{p}(x)$, a variant gives approximation of normalizer Z .
 - You could use this to approximate marginal likelihood in Bayesian logistic regression.

Summary

- **Bayesian Linear Regression**
 - Gaussian conditional likelihood and Gaussian prior gives Gaussian posterior.
 - Posterior predictive is also Gaussian (“regression with error bars”).
- **Empirical Bayes for linear regression**
 - Can use marginal likelihood to noise variance(s) and regularization parameters(s).
 - Can also select which non-linear transforms to use.
 - Bayesian Occam’s razor: can encourage sparsity and simplicity.
- **Bayesian logistic regression**
 - Gaussian prior is not conjugate so need approximations.
- **Rejection sampling**: generate exact samples from complicated distributions.
 - Tends to reject too many samples in high dimensions.
- **Importance sampling**: reweights samples from the wrong distribution.
 - Tends to have high variance in high dimensions.
- Next time: approximating integrals with optimization.

Gradient of Validation/Cross-Validation Error

- It's also possible to do **gradient descent on λ to optimize validation/cross-validation error** of model fit on the training data.
- For L2-regularized least squares, define $w(\lambda) = (X^T X + \lambda I)^{-1} X^T y$.
- You can use chain rule to get **derivative of validation error E_{valid} with respect to λ** :

$$\frac{d}{d\lambda} E_{\text{valid}}(w(\lambda)) = E'_{\text{valid}}(w(\lambda)) w'(\lambda).$$

- For more complicated models, you can use **total derivative** to get gradient with respect to λ in terms of gradient/Hessian with respect to w .

Bayesian Feature Selection

- Classic feature selection methods don't work when $d \gg n$:
 - AIC, BIC, Mallows's, adjusted- R^2 , and L1-regularization return very different results.
- Here maybe all we can hope for is **posterior probability of $w_j = 0$** .
 - Consider all models, and weight by posterior the ones where $w_j = 0$.
- If we fix λ and use L1-regularization, posterior is **not sparse**.
 - Probability that a variable is exactly 0 is zero.
 - L1-regularization only leads to sparse MAP, not sparse posterior.

Bayesian Feature Selection

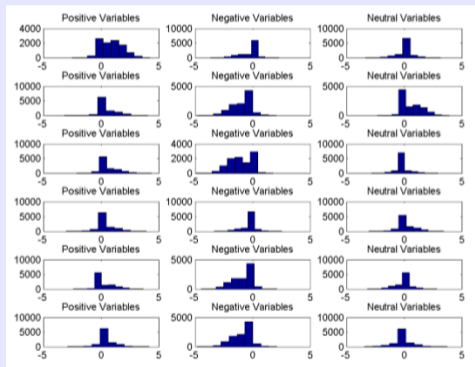
- Type II MLE gives sparsity because posterior variance goes to zero.
 - But this **doesn't give probability** of individual w_j values being 0.
- We can encourage sparsity in Bayesian models using a **spike and slab** prior:



- Mixture of Dirac delta function at 0 and another prior with non-zero variance.
- Places non-zero posterior weight at exactly 0.
- Posterior is still non-sparse, but answers the question:
 - “What is the probability that variable is non-zero”?

Bayesian Feature Selection

- Monte Carlo samples of w_j for 18 features when classifying '2' vs. '3':
 - Requires “trans-dimensional” MCMC since dimension of w is changing.



- “Positive” variables had $w_j > 0$ when fit with L1-regularization.
- “Negative” variables had $w_j < 0$ when fit with L1-regularization.
- “Neutral” variables had $w_j = 0$ when fit with L1-regularization.

Bayes Factors for Bayesian Hypothesis Testing

- Suppose we want to **compare hypotheses**:
 - E.g., “this data is best fit with linear model” vs. a degree-2 polynomial.
- **Bayes factor** is ratio of marginal likelihoods,

$$\frac{p(y | X, \text{degree } 2)}{p(y | X, \text{degree } 1)}$$

- If very large then data is much more consistent with degree 2.
 - A common variation also puts **prior on degree**.
- A more **direct method of hypothesis testing**:
 - No need for null hypothesis, “power” of test, p-values, and so on.
 - As usual only says which model is more likely, not whether any are correct.

- American Statistical Association:
 - “Statement on Statistical Significance and P-Values”.
 - <http://amstat.tandfonline.com/doi/pdf/10.1080/00031305.2016.1154108>
- “Hack Your Way To Scientific Glory”:
 - <https://fivethirtyeight.com/features/science-isnt-broken>
- “Replicability crisis” in social psychology and many other fields:
 - https://en.wikipedia.org/wiki/Replication_crisis
 - <http://www.nature.com/news/big-names-in-statistics-want-to-shake-up-much-maligned-p-value-1.22375>
- “T-Tests Aren't Monotonic” : <https://www.naftaliharris.com/blog/t-test-non-monotonic>
- Bayes factors don't solve problems with p-values and multiple testing.
 - But they give an alternative view, are more intuitive, and make assumptions clear.
- Some notes on various issues associated with Bayes factors:
 - <http://www.aarondefazio.com/aderazio-bayesfactor-guide.pdf>