CPSC 440/540: Advanced Machine Learning

Bayesian Linear Regression, Approximate Inference

Danica Sutherland (building on materials from Mark Schmidt)

University of British Columbia

Winter 2023

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 as $\hat{y}^i = w^T x^i$.
- A common training objective is L2-regularized least squares,

$$\arg\min_w \frac{1}{2\sigma^2}\|\mathbf{X}w - \mathbf{y}\|^2 + \frac{\lambda}{2}\|w\|^2.$$

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

$$Y \sim \mathcal{N}(w^T X, \sigma^2), \quad w \sim \mathcal{N}(0, \lambda^{-1}\mathbf{I}).$$

- The unique MAP estimate is given by:

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

Bayesian Linear Regression

- Keep linear a Gaussian likelihood and prior,

$$Y \sim \mathcal{N}(w^\mathsf{T} X, \sigma^2), \quad w \sim \mathcal{N}(0, \lambda^{-1}\mathbf{I}).$$

- Can use Gaussian identities to work out that the posterior has the form

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

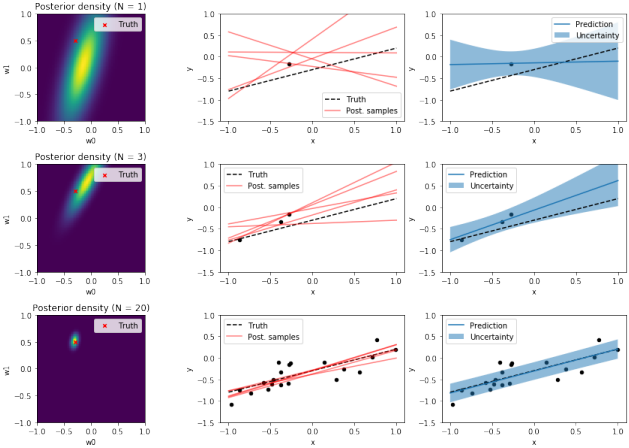
  which is a Gaussian centered at the MAP estimate.
  - The variance tells us how much variation we have around the MAP estimate.
    - In other models, the posterior mode (MAP) is usually not the posterior mean.
- By more tedious Gaussian identities the posterior predictive has the form

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

- Posterior predictive mode is the MAP prediction (also special for Gaussians).
  - But working with the full posterior predictive gives us variance of predictions.

Bayesian Linear Regression

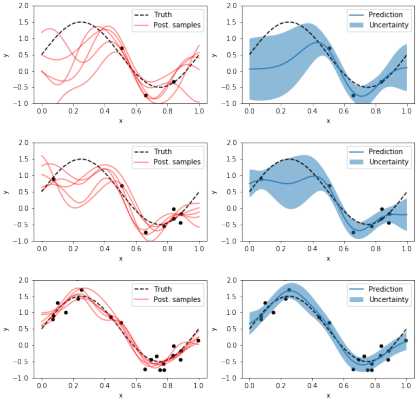
- Bayesian perspective gives us variability in w and predictions:



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

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: Kernelized Bayesian Linear Regression

bonus!

- In CPSC 340 you may have seen the kernel trick
- We can also do that here: with $\Theta = \frac{1}{\sigma^2}\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I} \in \mathbb{R}^{d\times d}$, we can rewrite

$$\begin{aligned}\tilde{y} \mid \tilde{x}, \mathbf{X}, \mathbf{y} &\sim \mathcal{N}\left(\frac{1}{\sigma^2}\tilde{x}^\mathsf{T}\Theta^{-1}\mathbf{X}\mathbf{y}, \tilde{x}^\mathsf{T}\Theta^{-1}\tilde{x}\right) \\ &= \mathcal{N}\left(\frac{1}{\lambda}\tilde{x}^\mathsf{T}\mathbf{X}A^{-1}\mathbf{y}, \frac{1}{\lambda}\tilde{x}^\mathsf{T}\tilde{x} - \frac{1}{\lambda^2}\tilde{x}^\mathsf{T}\mathbf{X}^\mathsf{T}A^{-1}\mathbf{X}\tilde{x}\right)\end{aligned}$$

where $A = \lambda^{-1}\mathbf{X}\mathbf{X}^\mathsf{T} + \sigma^2\mathbf{I} \in \mathbb{R}^{n\times n}$ is a regularized kernel matrix
and $\mathbf{X}\tilde{x} \in \mathbb{R}^n$ is the train-to-test kernel evaluations

- Allows us to efficiently use some exponential-sized or infinite-sized feature sets.
- Uses e.g. Woodbury matrix identity to rewrite

Digression: Gaussian Processes

bonus!

- Another view as a Gaussian process (GP)

- Notation: a stochastic process is an infinite collection of random variables.
  - One way to view is as a random function f
- Gaussian process is a stochastic process where any finite sample is Gaussian.
  - $\big(f(x_1), \ldots, f(x_T)\big) \in \mathbb{R}^T$ is multivariate normal for any choice of $x_1, \ldots, x_T$
  - Defined in terms of a mean function and a covariance function.
    - $\mathbb{E}\, f(x) = m(x)$, $\mathrm{Cov}(f(x_1), f(x_2)) = k(x_1, x_2)$
    - k is a valid covariance function if and only if it's a valid kernel function.
  - GP prior + Gaussian likelihood gives a GP posterior
  - Predictive distribution exactly agrees with (kernelized) Bayesian linear regression
- A popular book on this topic if you want to read more:
  - `http://www.gaussianprocess.org/gpml/chapters/RW.pdf`

- We'll assume we have explicit features, but you could use 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,

$$\hat{\sigma}^2, \hat{\lambda} \in \arg\max_{\sigma^2, \lambda} p(\mathbf{y} \mid \mathbf{X}, \sigma^2, \lambda).$$

- The marginal likelihood integrates over the parameters w,

$$p(\mathbf{y} \mid \mathbf{X}, \sigma^2, \lambda) = \int_w p(\mathbf{y}, w \mid \mathbf{X}, \sigma^2, \lambda) \mathrm{d}w = \int_w p(\mathbf{y} \mid \mathbf{X}, w, \sigma^2) p(w \mid \lambda) \mathrm{d}w \quad (w \perp\!\!\!\perp X).$$

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

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

  - 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,

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

- Too expensive for cross-validation, but can still do empirical Bayes.
  - You can do projected gradient descent to optimize the λ_j.
    - Or parameterize as $\lambda_j = \exp(\ell_j)$ and use unconstrained optimization.

- Weird fact: this yields sparse solutions.
  - It can send some $\lambda_j \to \infty$, concentrating posterior for w_j at exactly 0.
  - This is L2 regularization, but empirical Bayes naturally encourages sparsity.
    - Called "Automatic relevance determination" (ARD)

- Non-convex, and theory isn't 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.

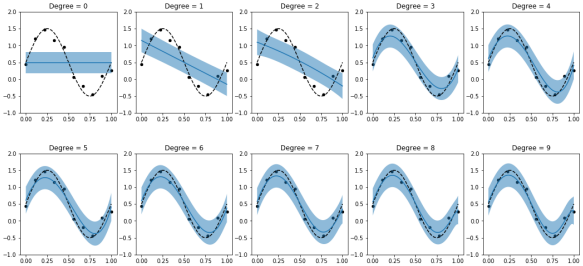
- The "automatic relevance determination" selects training examples ($\sigma_i \to \infty$).
  - This is like the support vectors in SVMs, but tends to be much more sparse.

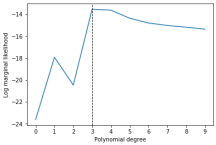
- Can also use Empirical Bayes to learn kernel parameters.
  - Do gradient descent on the σ values (or $\log \sigma$) in the Gaussian kernel.

- 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.
  - $p(\mathbf{y} \mid \mathbf{X}, \sigma^2, \lambda, k)$ is smaller for degree 4 polynomials since they can fit more datasets.
  - Non-monotonic: prefers degree 1 and 3 over degree 2.
  - Model selection criteria like BIC are approximations to marginal likelihood as $n \to \infty$.

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):

$$p(\mathbf{y} \mid \mathbf{X}) = \int_{w_0} \int_{w_1} \int_{w_2} \int_{w_3} p(\mathbf{y} \mid \mathbf{X}, w) p(w \mid \lambda) \mathrm{d}w$$

  - Marginal likelihood for degree 7:

$$p(\mathbf{y} \mid \mathbf{X}) = \int_{w_0} \int_{w_1} \int_{w_2} \int_{w_3} \int_{w_4} \int_{w_5} \int_{w_6} \int_{w_7} p(\mathbf{y} \mid \mathbf{X}, w) p(w \mid \lambda) \mathrm{d}w.$$

  - 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:

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

- This is the idea behind Bayes factors for hypothesis testing (see bonus slides).
  - Alternative to classic hypothesis tests like t-tests.

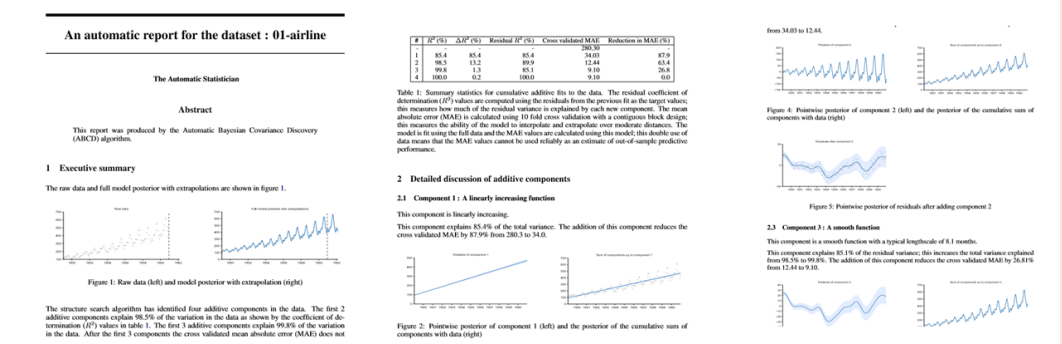
- Usual warning: empirical Bayes can sometimes becomes degenerate.
  - May need a non-vague prior on the hyper-parameters.

- 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

bonus!

- 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

Motivation: Bayesian Logistic Regression

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

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

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

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

- In Bayesian logistic regression, we'd work with the posterior.
  - But the posterior is not a Gaussian, so this is not a conjugate prior.
  - We don't 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

$$
\begin{aligned}
p(\tilde{y}^i \mid \tilde{x}^i, \mathbf{X}, \mathbf{y}, \lambda) &= \int_w p(\tilde{y}^i \mid \tilde{x}^i, w)\, p(w \mid \mathbf{X}, \mathbf{y}, \lambda) \mathrm{d}w \\
&= \mathbb{E}_w[p(\tilde{y}^i \mid \tilde{x}^i, w) \mid \mathbf{X}, \mathbf{y}, \lambda].
\end{aligned}
$$

- If we could sample from the posterior, we could compute this with Monte Carlo!
  - But we don't know how to generate IID samples from this posterior.

- Later, we'll cover MCMC, which is a standard method in scenarios like this.

- But we'll start 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.

Rejection Sampling for Conditionals

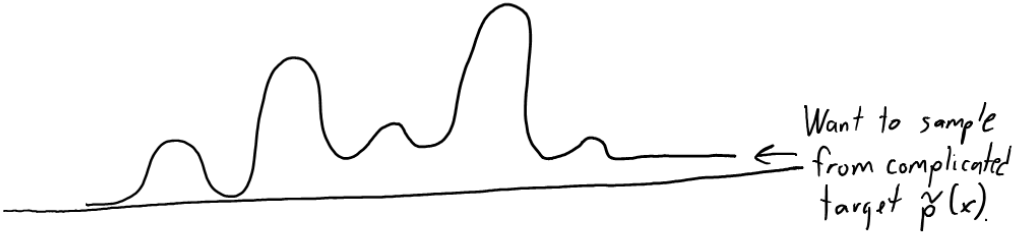
- We already mentioned rejection sampling for conditional sampling:
  - Example: sampling from a Gaussian conditional on knowing $x \in [-1, 1]$.

  - 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.

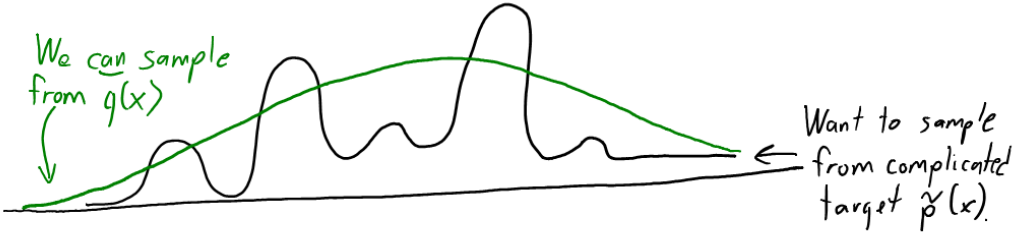
General Rejection Sampling Algorithm

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



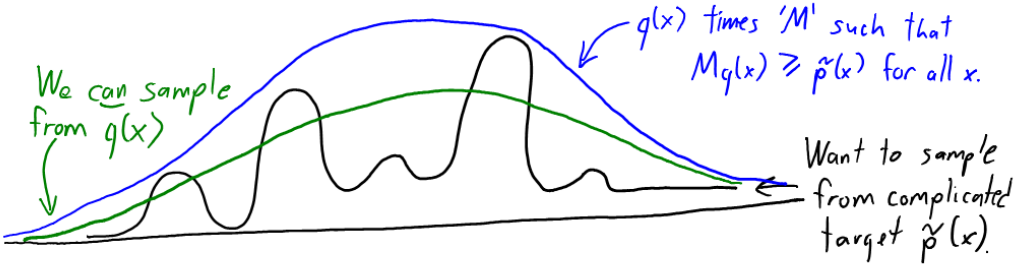
General Rejection Sampling Algorithm

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



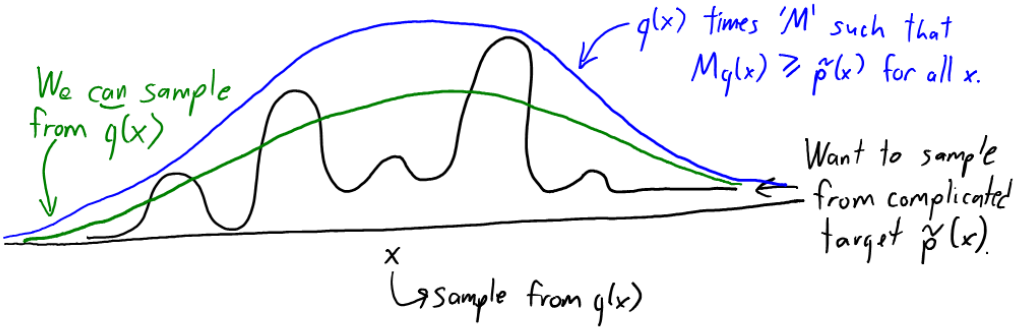
General Rejection Sampling Algorithm

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



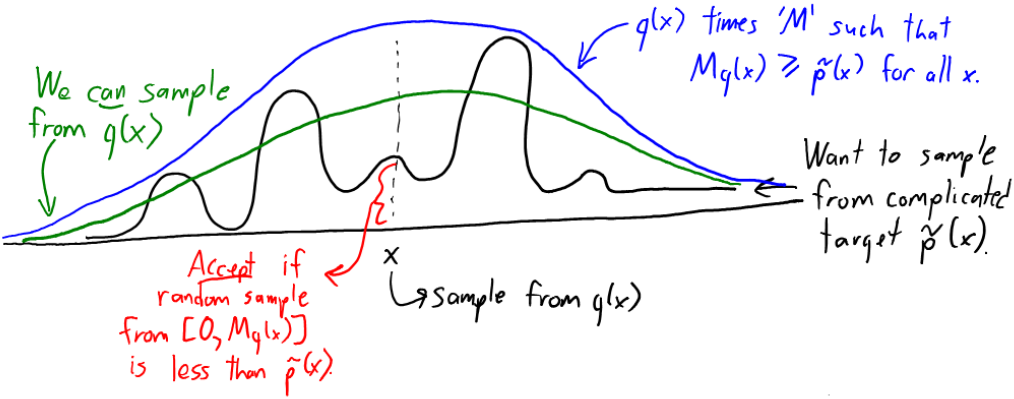
General Rejection Sampling Algorithm

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



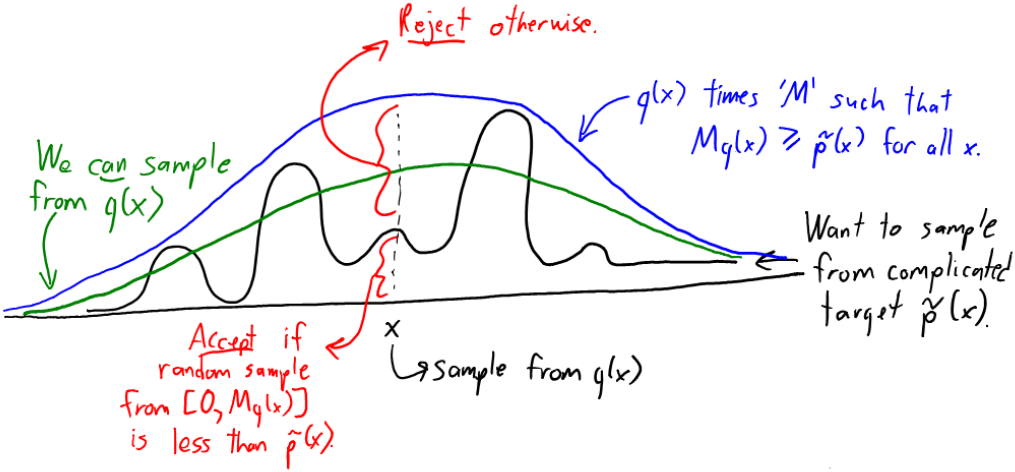
General Rejection Sampling Algorithm

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



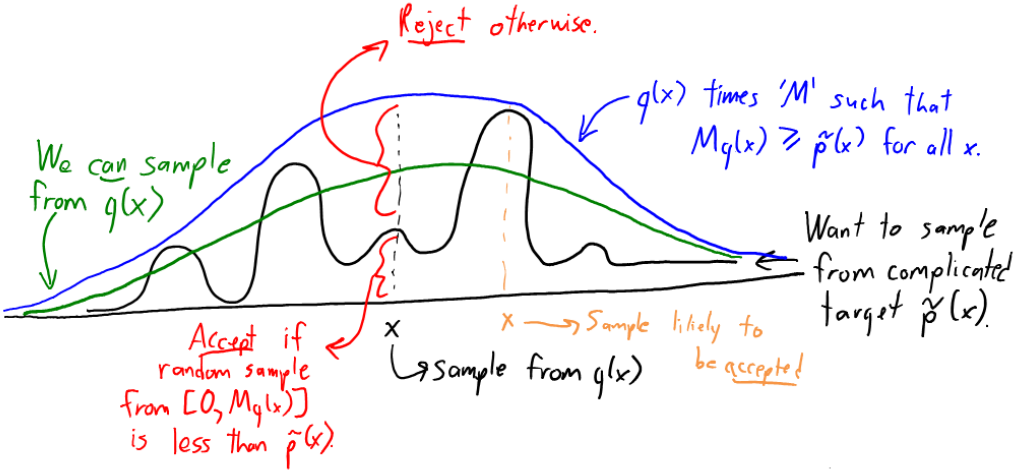
General Rejection Sampling Algorithm

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



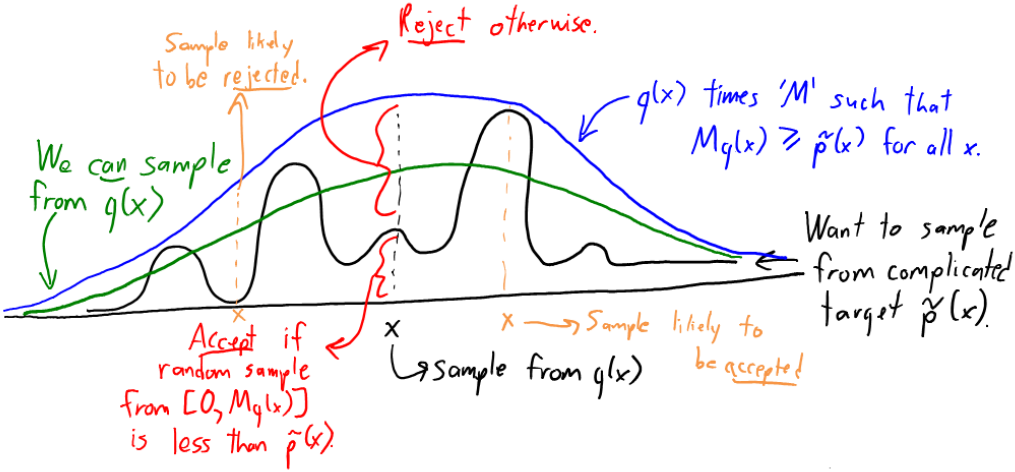
General Rejection Sampling Algorithm

- 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 we can sample from.
  3. An upper bound M on $\tilde{p}(x)/q(x)$.

- Rejection sampling algorithm:
  1. Sample x from $q(x)$.
  2. Keep the sample with probability $\tilde{p}(x)/(Mq(x))$:
     - Sample u from $\mathcal{U}(0, 1)$.
     - Keep the sample if $u \leq \tilde{p}(x) \,/\, (Mq(x))$.

- The accepted samples will be from $p(x)$ (as long as M is a valid upper bound).

General Rejection Sampling Algorithm

- For Bayesian logistic regression, we could propose samples from the prior:

$$\tilde{p}(w \mid \mathbf{X}, \mathbf{y}) = p(\mathbf{y} \mid \mathbf{X}, w)p(w) \qquad q(w) = p(w)$$

$$\frac{\tilde{p}(w \mid \mathbf{y}, \mathbf{X})}{q(w)} = \frac{p(\mathbf{y} \mid \mathbf{X}, w)p(w)}{p(w)} = p(\mathbf{y} \mid \mathbf{X}, w) \leq 1$$

  - Recall $\mathbf{y}$ is discrete here, so $p(\mathbf{y} \mid \mathbf{X}, w) \leq 1$ and can use $M = 1$
  - w sampled from prior would tend to be kept if they explain the data well.
- Drawbacks of rejection sampling:
  - You need to know a bound M on $\tilde{p}(x)/q(x)$ (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 instead accepts all samples.
- Derivation:

$$\begin{aligned}\mathbb{E}_{X\sim p}[f(X)] &= \int p(x)f(x)\,\mathrm{d}x \\ &= \int q(x)\frac{p(x)}{q(x)}f(x)\,\mathrm{d}x \\ &= \mathbb{E}_{X\sim q}\left[\frac{p(X)}{q(X)}f(X)\right] \approx \frac{1}{n}\sum_{i=1}^{n}\frac{p(x^i)}{q(x^i)}f(x^i),\end{aligned}$$

  using a Monte Carlo approximation with IID samples from q.
  - Replace integral with a sum for discrete distributions.
- We can sample from q, but reweight by $p(x)/q(x)$ to compute expectation.
- Only assumption is that q is always non-zero if p is non-zero.

Self-Normalized Importance Sampling

- What if we just have $\tilde{p}$, with $p(x) = \tilde{p}(x)/Z$? Letting $r(x) = \tilde{p}(x)/q(x)$:

$$\begin{aligned}\mathbb{E}_{X\sim p}[f(X)] &= \int p(x)f(x)\,\mathrm{d}x = \frac{1}{Z}\int q(x)\frac{\tilde{p}(x)}{q(x)}f(x)\,\mathrm{d}x \\ &= \frac{\mathbb{E}_{X\sim q}[r(X)f(X)]}{\int \tilde{p}(x)\,\mathrm{d}x} = \frac{\mathbb{E}_{X\sim q}[r(X)f(X)]}{\int q(x)\frac{\tilde{p}(x)}{q(x)}\,\mathrm{d}x} = \frac{\mathbb{E}_{X\sim q}[r(X)f(X)]}{\mathbb{E}_{X\sim q}[r(X)]}\end{aligned}$$

- Can use Monte Carlo estimator based on n samples from q:

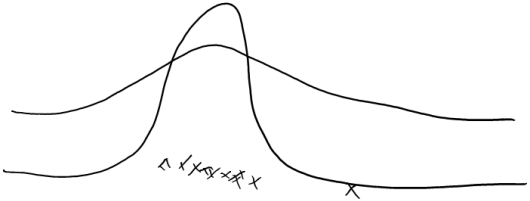
$$\mathbb{E}_{X\sim p}[f(X)] \approx \frac{\frac{1}{n}\sum_{i=1}^{n} r(x^i)f(x^i)}{\frac{1}{n}\sum_{i=1}^{n} r(x^i)}$$

  - Weighted mean, normalized by $r(x^i) = \tilde{p}(x^i)/q(x^i)$
  - Biased estimator: $\mathbb{E}\,\frac{1}{Z} > \frac{1}{Z}$ for non-constant distributions (Jensen's inequality)

Importance Sampling

- Importance sampling is only efficient if q is close to p.
- Otherwise, weights will be huge for a small number of samples.
  - Even though unbiased, variance can be huge.

- Can be problematic if q has lighter "tails" than p:
  - You rarely sample the tails, so those samples get huge weights.



- As with rejection sampling, does not tend to work well in high dimensions.
  - There's room, though, to cleverly design q.
    - Like "alternate between sampling two Gaussians with different variances".

Outline

Overview of Bayesian Inference Tasks

- Bayesian inference requires computing expectations with respect to posterior,

$$E[f(\theta)] = \int_\theta f(\theta)p(\theta \mid x)\mathrm{d}\theta.$$

- Examples:
  - If $f(\theta) = p(\tilde{x} \mid \theta)$, we get posterior predictive.
  - If $f(\theta) = \mathbb{I}(\theta \in S)$ we get probability of S (e.g., marginals or conditionals).
  - If $f(\theta) = 1$ and we use $\tilde{p}(\theta \mid x)$, we get marginal likelihood.

- But posterior often doesn't have a closed-form expression.
  - We don't just want to flip coins and multiply Gaussians.

- Our two main tools for aproximate inference:
  1. Monte Carlo methods.
  2. Variational methods.
- Classic ideas from statistical physics, that revolutionized Bayesian stats.

Approximate Inference

Two main strategies for approximate inference:

1. Monte Carlo methods:
   - Approximate p with empirical distribution over samples,

$$p(x) \approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(x^i = x).$$

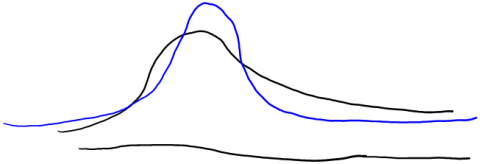
   - Turns inference into sampling.
2. Variational methods:
   - Approximate p with "closest" distribution q from a tractable family,

$$p(x) \approx q(x).$$

     - Gaussian, product of Bernoulli, any other model with easy inference. . . .
   - Turns inference into optimization.

Variational Inference Illustration

- Approximate non-Gaussian p by a Gaussian q:



- Variational methods try to find simple distribution q that is closest to target p.
  - Unlike Monte Carlo, does not converge to true solution.
    - A Gaussian may not be able to perfectly model posterior.
  - Variational methods quickly give an approximate solution.
    - Sometimes all we need.
    - Sometimes, approximation is better than any reasonable amount of Monte Carlo!

Laplace Approximation

- A classic variational method is the Laplace approximation.
  1. Find an x that maximizes $p(x)$,

$$x^* \in \arg\min_x \{-\log p(x)\}.$$

  2. Computer second-order Taylor expansion of $f(x) = -\log p(x)$ at x^*.

$$-\log p(x) \approx f(x^*) + \underbrace{\nabla f(x^*)}_{0}^{\mathsf{T}}(x - x^*) + \frac{1}{2}(x - x^*)^{\mathsf{T}} \nabla^2 f(x^*)\,(x - x^*).$$

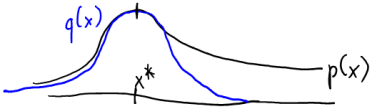
  3. Use distribution q that has this $-\log q(x)$ everywhere:

$$-\log q(x) = f(x^*) + \frac{1}{2}(x - x^*)\nabla^2 f(x^*)(x - x^*),$$

     This means the distribution q is exactly $\mathcal{N}(x^*, \nabla^2 f(x^*)^{-1})$.
     - Same approximation as used by Newton's method in optimization.

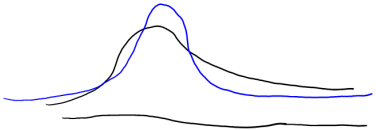
Laplace Approximation

- Laplace approximation replaces a complicated p with a Gaussian q.
  - Centered at the mode, and agrees with 1st/2nd-derivatives of log-likelihood there:



- Now you only need to compute Gaussian integrals (linear algebra for many f).
  - Very fast: just solve an optimization (compared to super-slow Monte Carlo).
  - Bad approximation if posterior is heavy-tailed, multi-modal, skewed, and so on.

- It might not even give you the "best" Gaussian approximation:



- We'll discuss fancier variational methods later.

Summary

- Bayesian Linear Regression
  - Gaussian conditional likelihood and Gaussian prior gives Gaussian posterior.
  - Posterior predictive is also Gaussian ("regression with error bars").
- Empirical Bayes to choose hyperparameters based on marginal likelihood.
  - Bayesian Occam's razor: can encourage sparsity and simplicity.
- Bayesian logistic regression: Gaussian prior isn't conjugate; 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.
- Variational methods approximate p with a simpler distribution q.
- Laplace approximation simple variationl inference method.
  - Use Gaussian centered at MAP that agrees with first two derivatives of NLL.
- Next time: the exponential family.

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- To get MLE for Σ we re-parameterize in terms of precision matrix $\Theta = \Sigma^{-1}$,

$$
\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Sigma^{-1}(x^i-\mu)+\frac{n}{2}\log|\Sigma| \\
=&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Theta(x^i-\mu)+\frac{n}{2}\log|\Theta^{-1}| && \text{(ok because } \Sigma \text{ is invertible)} \\
=&\frac{1}{2}\sum_{i=1}^{n}\mathrm{Tr}\left((x^i-\mu)^\top\Theta(x^i-\mu)\right)+\frac{n}{2}\log|\Theta|^{-1} && \text{(scalar } y^\top Ay=\mathrm{Tr}(y^\top Ay)) \\
=&\frac{1}{2}\sum_{i=1}^{n}\mathrm{Tr}((x^i-\mu)(x^i-\mu)^\top\Theta)-\frac{n}{2}\log|\Theta| && (\mathrm{Tr}(ABC)=\mathrm{Tr}(CAB))
\end{aligned}
$$

- Where the trace $\mathrm{Tr}(A)$ is the sum of the diagonal elements of A.
  - That $\mathrm{Tr}(ABC) = \mathrm{Tr}(CAB)$ when dimensions match is the cyclic property of trace.

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- From the last slide we have in terms of precision matrix Θ that

$$= \frac{1}{2} \sum_{i=1}^{n} \mathsf{Tr}((x^i - \mu)(x^i - \mu)^\top \Theta) - \frac{n}{2} \log |\Theta|$$

- We can exchange the sum and trace (trace is a linear operator) to get,

$$= \frac{1}{2} \mathsf{Tr} \left(\sum_{i=1}^{n} (x^i - \mu)(x^i - \mu)^\top \Theta \right) - \frac{n}{2} \log |\Theta| \qquad \sum_i \mathsf{Tr}(A_i B) = \mathsf{Tr} \left(\sum_i A_i B \right)$$

$$= \frac{n}{2} \mathsf{Tr} \left(\left(\underbrace{\frac{1}{n} \sum_{i=1}^{n} (x^i - \mu)(x^i - \mu)^\top}_{\text{sample covariance } `S'} \right) \Theta \right) - \frac{n}{2} \log |\Theta|. \qquad \left(\sum_i A_i B \right) = \left(\sum_i A_i \right) B$$

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- So the NLL in terms of the precision matrix Θ and sample covariance S is

$$f(\Theta) = \frac{n}{2}\text{Tr}(S\Theta) - \frac{n}{2}\log|\Theta|, \text{ with } S = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^\top$$

- Weird-looking but has nice properties:
  - $\text{Tr}(S\Theta)$ is linear function of Θ, with $\nabla_\Theta \text{Tr}(S\Theta) = S$.

    (it's the matrix version of an inner-product $s^\top\theta$)
  - Negative log-determinant is strictly convex, and has $\nabla_\Theta \log\det\Theta = \Theta^{-1}$.

    (generalizes $\nabla \log|x| = 1/x$ for for $x > 0$).
- Using these two properties the gradient matrix has a simple form:

$$\nabla f(\Theta) = \frac{n}{2}S - \frac{n}{2}\Theta^{-1}.$$

Trace Regularization and L1-regularization

bonus!

- A classic regularizer for Σ is to add a diagonal matrix to S and use

$$\Sigma = S + \lambda I,$$

which satisfies $\Sigma \succ 0$ because $S \succeq 0$ (eigenvalues at least λ).

- This corresponds to L1-regularization of diagonals of precision.

$$\begin{aligned}
f(\Theta) &= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{j=1}^{d} |\Theta_{jj}| && \text{(Gauss. NLL plus L1 of diags)} \\
&= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{j=1}^{d} \Theta_{jj} && \text{(Diagonals of pos. def. matrix are } > 0\text{)} \\
&= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \text{Tr}(\Theta) && \text{(Definition of trace)} \\
&= \text{Tr}(S\Theta + \lambda\Theta) - \log|\Theta| && \text{(Linearity of trace)} \\
&= \text{Tr}((S + \lambda I)\Theta) - \log|\Theta| && \text{(Distributive law)}
\end{aligned}$$

- Taking gradient and setting to zero gives $\Sigma = S + \lambda$.
  - But doesn't set to exactly zero as log-determinant term is too "steep" at 0.

Gradient of Validation/Cross-Validation Error

bonus!

- 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 $w(\lambda) = (X^TX + \lambda I)^{-1}X^Ty$.
- 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

bonus!

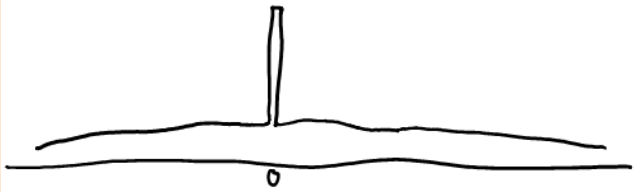
- Classic feature selection methods don't work when $d >> n$:
  - AIC, BIC, Mallow's, adjusted-R^2, and L1-regularization return very different results.
- Here maybe all we can hope for is posterior probability of $w_j = 0$.
  - Consider all models, and weight by posterior the ones where $w_j = 0$.
- 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

bonus!

- 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:

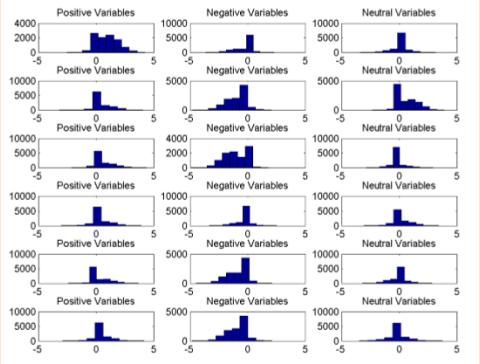


  - Mixture of Dirac delta function at 0 and another prior with non-zero variance.
  - Places non-zero posterior weight at exactly 0.
  - Posterior is still non-sparse, but answers the question:
    - "What is the probability that variable is non-zero"?

Bayesian Feature Selection

bonus!

- 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

bonus!

- 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,

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

  - 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.

bonus!

- American Statistical Assocation:
  - "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/adefazio-bayesfactor-guide.pdf