Back to Maximum Likelihood

- Given a generative model

\[ f(x, y = k) = \pi_k f_k(x) \]

- Using a generative modelling approach, we assume a parametric form for 
  \( f_k(x) = f(x; \phi_k) \) and compute the MLE \( \hat{\theta} \) of \( \theta = (\pi_k, \phi_k)_{k=1}^K \) based on the 
  training data \( \{x_i, y_i\}_{i=1}^n \).

- We then use a plug-in approach to perform classification

\[
p(Y = k|X = x, \hat{\theta}) = \frac{\hat{\pi}_k f(x; \hat{\phi}_k)}{\sum_{j=1}^K \hat{\pi}_j f(x; \hat{\phi}_j)}
\]

- Even for simple models, this can prove difficult; e.g. for LDA,
  \( f(x; \phi_k) = \mathcal{N}(x; \mu_k, \Sigma) \), and the MLE estimate of \( \Sigma \) is not full rank for \( p > n \).

- One answer: simplify even further, e.g. using axis-aligned covariances, 
  but this is usually too crude.

- Another answer: regularization.
Naïve Bayes

- Return to the spam classification example with two-class naïve Bayes

\[ f(x_i; \phi_k) = \prod_{j=1}^{p} \phi_{kj}^{x_{ij}} (1 - \phi_{kj})^{1-x_{ij}}. \]

The MLE estimates are given by

\[ \hat{\phi}_{kj} = \frac{\sum_{i=1}^{n} \mathbb{1}(x_{ij} = 1, y_i = k)}{n_k}, \quad \hat{\pi}_k = \frac{n_k}{n} \]

where \( n_k = \sum_{i=1}^{n} \mathbb{I}(y_i = k) \).

- If a word \( j \) does not appear in class \( k \) by chance, but it does appear in a document \( x_* \), then \( p(x_*|y_* = k) = 0 \) and so posterior \( p(y_* = k|x_*) = 0 \).

- Worse things can happen: e.g., probability of document under all classes can be 0, so posterior is ill-defined.
The Bayesian Learning Framework

- **Bayes Theorem**: Given two random variables $X$ and $\Theta$,

\[
p(\Theta|X) = \frac{p(X|\Theta)p(\Theta)}{p(X)}
\]

- **Likelihood**: $p(X|\Theta)$
- **Prior**: $p(\Theta)$
- **Posterior**: $p(\Theta|X)$
- **Marginal likelihood**: $p(X) = \int p(X|\Theta)p(\Theta) d\Theta$

- Treat parameters as random variables, and process of learning is just computation of posterior $p(\Theta|X)$.
- Summarizing the posterior:
  - **Posterior mode**: $\hat{\theta}^{\text{MAP}} = \arg\max_\theta p(\theta|X)$. Maximum a posteriori.
  - **Posterior mean**: $\hat{\text{mean}} = \mathbb{E}[\Theta|X]$.
  - **Posterior variance**: $\text{Var}[\Theta|X]$.

- How to make decisions and predictions? Decision theory.
- How to compute posterior?
Simple Example: Coin Tosses

- A very simple example: We have a coin with probability $\phi$ of coming up heads. Model coin tosses as iid Bernoullis, $1 =$head, $0 =$tail.
- Learn about $\phi$ given dataset $D = (x_i)_{i=1}^n$ of tosses.
  \[
  f(D|\phi) = \phi^{n_1} (1 - \phi)^{n_0}
  \]
  with $n_j = \sum_{i=1}^n \mathbb{1}(x_i = j)$.
- Maximum likelihood
  \[
  \hat{\phi}^{\text{ML}} = \frac{n_1}{n}
  \]
- Bayesian approach: treat unknown parameter as a random variable $\Phi$. Simple prior: $\Phi \sim U[0, 1]$. Posterior distribution:
  \[
  p(\phi|D) = \frac{1}{Z} \phi^{n_1} (1 - \phi)^{n_0}, \quad Z = \int_0^1 \phi^{n_1} (1 - \phi)^{n_0} d\phi = \frac{(n + 1)!}{n_1!n_0!}
  \]
  Posterior is a $\text{Beta}(n_1 + 1, n_0 + 1)$ distribution.
Simple Example: Coin Tosses

Posterior becomes peaked at true value $\phi^* = .7$ as dataset grows.
Simple Example: Coin Tosses

- Posterior distribution captures all learnt information.
  - Posterior mode:
    \[ \hat{\phi}_{\text{MAP}} = \frac{n_1}{n} \]
  - Posterior mean:
    \[ \hat{\phi}_{\text{mean}} = \frac{n_1 + 1}{n + 2} \]
  - Posterior variance:
    \[ \frac{1}{n + 3} \hat{\phi}_{\text{mean}} (1 - \hat{\phi}_{\text{mean}}) \]

- Asymptotically, for large \( n \), variance decreases as \( 1/n \) and is given by the inverse of Fisher’s information.
- Posterior distribution converges to true parameter \( \phi^* \) as \( n \to \infty \).
Simple Example: Coin Tosses

- What about test data?
- The **posterior predictive distribution** is the conditional distribution of $x_{n+1}$ given $(x_i)_{i=1}^{n}$:

\[
p(x_{n+1}|(x_i)_{i=1}^{n}) = \int_0^1 p(x_{n+1}|\phi, (x_i)_{i=1}^{n})p(\phi|(x_i)_{i=1}^{n}))d\phi
\]

\[
= \int_0^1 p(x_{n+1}|\phi)p(\phi|(x_i)_{i=1}^{n}))d\phi
\]

\[
= (\hat{\phi}_{\text{mean}})^{x_{n+1}}(1 - \hat{\phi}_{\text{mean}})^{1-x_{n+1}}
\]

- We predict on new data by **averaging** the predictive distribution over the posterior. Accounts for uncertainty about $\phi$. 

Simple Example: Coin Tosses

- Posterior distribution is a known analytic form. In fact posterior distribution is in the same beta family as the prior.
- An example of a **conjugate prior**.
- A beta distribution $\text{Beta}(a, b)$ with parameters $a, b > 0$ is an exponential family distribution with density

$$p(\phi|a, b) = \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)} \phi^{a-1}(1 - \phi)^{b-1}$$

where $\Gamma(t) = \int_{0}^{\infty} u^{t-1}e^{-u}du$ is the gamma function.
- If the prior is $\phi \sim \text{Beta}(a, b)$, then the posterior distribution is

$$p(\phi|D, a, b) = \propto \phi^{a+n_1-1}(1 - \phi)^{b+n_0-1}$$

so is $\text{Beta}(a + n_1, b + n_0)$.
- Hyperparameters $a$ and $b$ are **pseudo-counts**, an imaginary initial sample that reflects our prior beliefs about $\phi$. 

---

*Posterior distribution is a known analytic form. In fact posterior distribution is in the same beta family as the prior. An example of a conjugate prior. A beta distribution $\text{Beta}(a, b)$ with parameters $a, b > 0$ is an exponential family distribution with density

$$p(\phi|a, b) = \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)} \phi^{a-1}(1 - \phi)^{b-1}$$

where $\Gamma(t) = \int_{0}^{\infty} u^{t-1}e^{-u}du$ is the gamma function. If the prior is $\phi \sim \text{Beta}(a, b)$, then the posterior distribution is

$$p(\phi|D, a, b) = \propto \phi^{a+n_1-1}(1 - \phi)^{b+n_0-1}$$

so is $\text{Beta}(a + n_1, b + n_0)$.

Hyperparameters $a$ and $b$ are pseudo-counts, an imaginary initial sample that reflects our prior beliefs about $\phi.*
Beta Distributions
Bayesian Inference for Multinomials

- Suppose \( x_i \in \{1, \ldots, K\} \) instead, and we model \( (x_i)_{i=1}^n \) as iid multinomials:

\[
p(D | \pi) = \prod_{i=1}^{n} \pi_{x_i} = \prod_{k=1}^{K} \pi_k^{n_k}
\]

with \( n_k = \sum_{i=1}^{n} \mathbb{1}(x_i = k) \) and \( \pi_k > 0, \sum_{k=1}^{K} \pi_k = 1 \).

- The conjugate prior is the Dirichlet distribution. \( \text{Dir}(\alpha_1, \ldots, \alpha_K) \) has parameters \( \alpha_k > 0 \), and density

\[
p(\pi) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k-1}
\]

on the probability simplex \( \{\pi : \pi_k > 0, \sum_{k=1}^{K} \pi_k = 1\} \).

- The posterior is also Dirichlet, with parameters \( (\alpha_k + n_k)_{k=1}^{K} \).

- Posterior mean is

\[
\hat{\pi}_k^{\text{mean}} = \frac{\alpha_k + n_k}{\sum_{j=1}^{K} \alpha_j + n_j}
\]
Dirichlet Distributions

(A) Support of the Dirichlet density for $K = 3$.
(B) Dirichlet density for $\alpha_k = 10$.
(C) Dirichlet density for $\alpha_k = 0.1$. 

267
Text Classification with (Less) Naïve Bayes

- Under the Naïve Bayes model, the joint distribution of labels $y_i \in \{1, \ldots, K\}$ and data vectors $x_i \in \{0, 1\}^p$ is

$$\prod_{i=1}^{n} p(x_i, y_i) = \prod_{i=1}^{n} \prod_{k=1}^{K} \left( \prod_{j=1}^{p} \pi_k \phi_{kj}^{x_{ij}} (1 - \phi_{kj})^{1-x_{ij}} \right)^{\mathbb{1}(y_i=k)}$$

$$= \prod_{k=1}^{K} \pi_k^{n_k} \prod_{j=1}^{p} \phi_{kj}^{n_{kj}} (1 - \phi_{kj})^{n_k-n_{kj}}$$

where $n_k = \sum_{i=1}^{n} \mathbb{1}(y_i = k)$, $n_{kj} = \sum_{i=1}^{n} \mathbb{1}(y_i = k, x_{ij} = 1)$.

- For conjugate prior, we can use $\text{Dir}((\alpha_k)_{k=1}^{K})$ for $\pi$, and $\text{Beta}(a, b)$ for $\phi_{kj}$ independently.

- Because the likelihood factorizes, the posterior distribution over $\pi$ and $(\phi_{kj})$ also factorizes, and posterior for $\pi$ is $\text{Dir}((\alpha_k + n_k)_{k=1}^{K})$, and for $\phi_{kj}$ is $\text{Beta}(a + n_{kj}, b + n_k - n_{kj})$. 
Text Classification with (Less) Naïve Bayes

- For prediction give $D = (x_i, y_i)_{i=1}^n$ we can calculate

$$p(x_0, y_0 = k|D) = p(y_0 = k|D)p(x_0|y_0 = k, D)$$

with

$$p(y_0 = k|D) = \frac{\alpha_k + n_k}{n + \sum_{l=1}^{K} \alpha_l}$$

$$p(x_{0j} = 1|y_0 = k, D) = \frac{a + n_{kj}}{a + b + n_k}$$

- Predicted class is

$$p(y_0 = k|x_0|D) = \frac{p(y_0 = k|D)p(x_0|y_0 = k, D)}{p(x_0|D)}$$

- Compared to ML plug-in estimator, pseudocounts help to regularize probabilities away from extreme values.
Bayesian Learning and Regularization

- Consider a Bayesian approach to logistic regression: introduce a multivariate normal prior for $b$, and uniform (improper) prior for $a$. The prior density is:

$$p(a, b) = (2\pi\sigma^2)^{-\frac{p}{2}} e^{-\frac{1}{2\sigma^2} \|b\|^2}$$

- The posterior is

$$p(a, b|D) \propto \exp \left( -\frac{1}{2\sigma^2} \|b\|^2 - \sum_{i=1}^{n} \log(1 + \exp(-y_i(a + b^T x_i))) \right)$$

- The posterior mode is the parameters maximizing the above, equivalent to minimizing the $L_2$-regularized empirical risk.

- Regularized empirical risk minimization is (often) equivalent to having a prior and finding the maximum a posteriori (MAP) parameters.
  - $L_2$ regularization - multivariate normal prior.
  - $L_1$ regularization - multivariate Laplace prior.

- From a Bayesian perspective, the MAP parameters are just one way to summarize the posterior distribution.
Bayesian Learning – Discussion

- Clear separation between models, which frame learning problems and encapsulates prior information, and algorithms, which computes posteriors and predictions.
- Bayesian computations — Most posteriors are intractable, and algorithms needed to efficiently approximate posterior:
  - Monte Carlo methods (Markov chain and sequential varieties).
  - Variational methods (variational Bayes, belief propagation etc).
- No optimization — no overfitting (!) but there can still be model misfit.
- Tuning parameters $\Psi$ can be optimized (without need for cross-validation).

$$p(X|\Psi) = \int p(X|\theta)p(\theta|\Psi)d\theta$$

$$p(\Psi|X) = \frac{p(X|\Psi)p(\Psi)}{p(X)}$$

- Be Bayesian about $\Psi$ — compute posterior.
- Type II maximum likelihood — find $\Psi$ maximizing $p(X|\Psi)$. 

271
Bayesian Learning – Further Readings

- Gelman et al. Bayesian Data Analysis.
Suppose we are given a dataset consisting of $n$ inputs $\mathbf{x} = (x_i)^n_{i=1}$ and $n$ outputs $\mathbf{y} = (y_i)^n_{i=1}$.

Regression: learn the underlying function $f(x)$. 

Gaussian Processes
Gaussian Processes

- We can model response as noisy version of an underlying function $f(x)$:
  \[ y_i | f(x_i) \sim \mathcal{N}(f(x_i), \sigma^2) \]

- Typical approach: parametrize $f(x; \beta)$, and learn $\beta$, e.g.,
  \[ f(x) = \sum_{j=1}^{d} \beta_d \phi_j(x) \]

- More direct approach: since $f(x)$ is unknown, we take a Bayesian approach, introduce a prior over functions, and compute a posterior over functions.

- Instead of trying to work with the whole function, just work with the function values at the inputs
  \[ f = (f(x_1), \ldots, f(x_n))^\top \]
Gaussian Processes

- The prior $p(f)$ encodes our prior knowledge about the function.
- What properties of the function can we incorporate?
  - Multivariate normal assumption:
    $$ f \sim \mathcal{N}(0, G) $$
  - Use a kernel function $\kappa$ to define $G$:
    $$ G_{ij} = \kappa(x_i, x_j) $$
  - Expect regression functions to be smooth: If $x$ and $x'$ are close by, then $f(x)$ and $f(x')$ have similar values, i.e. strongly correlated.
    $$ \begin{pmatrix} f(x) \\ f(x') \end{pmatrix} \sim \mathcal{N} \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \kappa(x, x) & \kappa(x, x') \\ \kappa(x', x) & \kappa(x', x') \end{pmatrix} \right) $$
    In particular, want
    $$ \kappa(x, x') \approx \kappa(x, x) = \kappa(x', x'). $$

- Model:
  $$ f \sim \mathcal{N}(0, G) $$
  $$ y_i | f_i \sim \mathcal{N}(f_i, \sigma^2) $$
Gaussian Processes

- What does a multivariate normal prior mean?
- Imagine $x$ forms a very dense grid of data space. Simulate prior draws

$$ f \sim \mathcal{N}(0, G) $$

Plot $f_i$ vs $x_i$ for $i = 1, \ldots, n$.

- The prior over functions is called a **Gaussian process** (GP).

http://www.gaussianprocess.org/
Gaussian Processes

- Different kernels lead to different function characteristics.

Gaussian Processes

\[ f|\mathbf{x} \sim \mathcal{N}(0, G) \]
\[ y|f \sim \mathcal{N}(f, \sigma^2 I) \]

- Posterior distribution:
  \[ f|y \sim \mathcal{N}(G(G + \sigma^2 I)^{-1}y, G - G(G + \sigma^2 I)G) \]

- Posterior predictive distribution: Suppose \( \mathbf{x}' \) is a test set. We can extend our model to include the function values \( f' \) at the test set:
  \[ \begin{pmatrix} f' \end{pmatrix}|\mathbf{x}, \mathbf{x}' \sim \mathcal{N} \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K_{xx} & K_{xx'} \\ K_{x'x} & K_{x'x'} \end{pmatrix} \right) \]
  \[ y|f \sim \mathcal{N}(f, \sigma^2 I) \]

where \( K_{zz'} \) is matrix with \( ij \)th entry \( \kappa(z_i, z_j') \). \( K_{xx} = G \). 

- Some manipulation of multivariate normals gives:
  \[ f'|y \sim \mathcal{N} \left( K_{x'x}(K_{xx} + \sigma^2 I)^{-1}y, K_{x'x}' - K_{x'x}(K_{xx} + \sigma^2 I)^{-1}K_{xx'} \right) \]
Gaussian Processes