

Machine Learning (Week 8 Lecture)

Naive Bayes Classifier and Bayesian Classification

Debasis Ganguly

Debasis.Ganguly@glasgow.ac.uk

School of Computing Science
University of Glasgow

November 20, 2025

Classification under Uncertainties

Introduction

D. Ganguly

Refresher on Bayes Theorem

Naive Bayes

NB for Categorical data

NB for Non-categorical data

Bayesian Logistic Regression

Laplace approximation

MCMC sampling

- ▶ Last 2 weeks of Classification:
 - ▶ Binary and multi-class parametric approaches - Logistic Regression and Softmax Regression
 - ▶ Non-parametric approach: K-NN (Doesn't learn any parameters).
- ▶ Today about application of Bayesian:
 - ▶ As a **Classifier**: Naive Bayes (NB) - Learns an abstraction from the data but does not learn parameters by gradient descent.
 - ▶ As a **Model calibrator**: Address uncertainties in data/model by **calibrating model confidences**.

- ▶ Two identical looking bins: A (●, ●, ●, ●, ●), and B (●, ●, ●, ●)
- ▶ You are blind-folded and asked to select a ball from a bin (you don't know which bin that is).

Menti-Quiz (Code: 2131-2255)

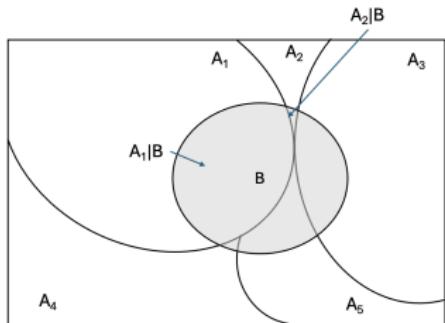
You observe a ● ball. Which likely bin did the ball come from?

- ▶ Two identical looking bins: A (●, ●, ●, ●, ●), and B (●, ●, ●, ●)
- ▶ You are blind-folded and asked to select a ball from a bin (you don't know which bin that is).

Menti-Quiz (Code: 2131-2255)

You observe a ● ball. Which likely bin did the ball come from?

Bayes Theorem



- ▶ A_i - Hypotheses/Causes
(forms a partition over the set of all possibilities)
- ▶ B - Evidence/Effect, i.e., one that is observed.

Most likely hypothesis that has led to an observation

$$P(A_i|B) = \frac{\underbrace{P(B|A_i)}_{\text{reverse causation}} \underbrace{P(A_i)}_{\text{prior}}}{P(B)}$$

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)}$$

Solution: Which bin did the ball come from?

Compute the priors

- ▶ $P(\bullet|A) = 3/5$, $P(\bullet|B) = 2/5$, $P(\bullet|B) = 1/5$,
 $P(\bullet|B) = 4/5$
- ▶ $P(A) = P(B) = 1/2$ (No other information is given)

Apply Bayes Theorem to compute posteriors

- ▶ $P(B|\bullet) = P(\bullet|B)P(B)/P(\bullet)$
- ▶ $= P(\bullet|B)P(B)/ (P(\bullet|A)P(A) + P(\bullet|B)P(B))$
- ▶ $= \frac{1/5 \times 1/2}{3/5 \times 1/2 + 1/5 \times 1/2} = \frac{1/10}{3/10 + 1/10} = 1/4.$

From drawing balls to building a classification model

For classification

- ▶ Bins (A_i): Hypothesis \equiv Class
- ▶ Ball (B): An observed Value of a **Categorical Feature**

Menti-Quiz (Code: 2131-2255)

Revisit the ball sampling problem: You now draw two balls blindfolded, and you observe both are red (•, •). Will your estimate change? Why?

From drawing balls to building a classification model

For classification

- ▶ Bins (A_i): Hypothesis \equiv Class
- ▶ Ball (B): An observed Value of a **Categorical Feature**

Menti-Quiz (Code: 2131-2255)

Revisit the ball sampling problem: You now draw two balls blindfolded, and you observe both are red (●, ●). Will your estimate change? Why?

- ▶ Balls \equiv Observed features (e.g., terms in an email that you received).
- ▶ You then have to find the likely class \equiv Predict a class of an instance (e.g., predict if the email is spam or not).

Independence Assumption

- ▶ Labelled training data \equiv knowing the content of the two bins.
- ▶ Each bin \equiv a class (spam or not spam).

- ▶ Labelled training data \equiv knowing the content of the two bins.
- ▶ Each bin \equiv a class (spam or not spam).
- ▶ Estimate probabilities of the form:
 $P(x = \text{cheap} | y = \text{SPAM})$.
- ▶ You need to **aggregate** these probabilities \equiv assuming that drawing a ball is independent of drawing another.

- ▶ Labelled training data \equiv knowing the content of the two bins.
- ▶ Each bin \equiv a class (spam or not spam).
- ▶ Estimate probabilities of the form:
 $P(x = \text{cheap} | y = \text{SPAM})$.
- ▶ You need to **aggregate** these probabilities \equiv assuming that drawing a ball is independent of drawing another.
- ▶ Here it means:
 - ▶ A term in an observed document **is independent of another**.
- ▶ Naive: Indicates this simplistic assumption.

- ▶ At inference time, you don't know the y value. You're given an instance \mathbf{x} .
- ▶ You need to compute $P(y = k|\mathbf{x})$ for each k and then take the maximum.
- ▶ In our example, choose the max between $P(y = 1|\mathbf{x})$ and $P(y = 2|\mathbf{x})$.

Bayes Theorem - Invert class priors to class posteriors

- ▶
$$P(y = k|\mathbf{x}) = \frac{P(\mathbf{x}|y=k).P(y=k)}{Z}$$
- ▶ $Z = \sum_{k'=1}^M P(\mathbf{x}|y = k').P(y = k')$.
- ▶ $P(\mathbf{x}|y = k) = \prod_{x \in \mathbf{x}} P(x|y = k)$ (**features are independent**, i.e., sampling a blue ball is independent of sampling an yellow ball).

- ▶ Naive Bayes: Classifier that's somewhere in the middle of a parametric (logistic regression) and non-parametric classification (K-NN).
- ▶ Unlike a non-parametric classifier it learns a representation from the training set.
- ▶ Unlike a parametric classifier, it doesn't learn via back-propagation of error (i.e., doesn't learn from mistakes).

- ▶ Works for discrete-valued data instances, i.e., where the values of each component of a data instance are categorical.
- ▶ Examples of classifiers working with discrete-valued data:
 - ▶ A combination of age-range and income-bracket \mapsto sanction loan (0/1).
 - ▶ Documents (comprised of terms) \mapsto Spam (0/1).

Categorical Data Example: Loan Sanction Classifier

- Age-range $\in \{ \text{Young (20-35)}, \text{Middle_aged (36-55)}, \text{Seniors (>55)} \}$, i.e., one of {Y, M, S} (category names). You can think of these as integers, i.e., Y=1 and so on.

Categorical Data Example: Loan Sanction Classifier

- ▶ Age-range $\in \{ \text{Young (20-35)}, \text{Middle_aged (36-55)}, \text{Seniors (>55)} \}$, i.e., one of {Y, M, S} (category names). You can think of these as integers, i.e., Y=1 and so on.
- ▶ Income-bracket $\in \{ \text{Low_income (1)}, \text{Average_income (2)}, \text{Above_average_income (3)}, \text{High_income (4)} \}$.

Categorical Data Example: Loan Sanction Classifier

- ▶ Age-range $\in \{ \text{Young (20-35)}, \text{Middle_aged (36-55)}, \text{Seniors (>55)} \}$, i.e., one of {Y, M, S} (category names). You can think of these as integers, i.e., Y=1 and so on.
- ▶ Income-bracket $\in \{ \text{Low_income (1)}, \text{Average_income (2)}, \text{Above_average_income (3)}, \text{High_income (4)} \}$.
- ▶ Given training data of the following form:

Age-range	Income-bracket	Loan sanctioned
1	1	0
2	2	1
	...	
3	4	0

Categorical Data Example: Loan Sanction Classifier

- ▶ Age-range $\in \{ \text{Young (20-35)}, \text{Middle_aged (36-55)}, \text{Seniors (>55)} \}$, i.e., one of {Y, M, S} (category names). You can think of these as integers, i.e., Y=1 and so on.
- ▶ Income-bracket $\in \{ \text{Low_income (1)}, \text{Average_income (2)}, \text{Above_average_income (3)}, \text{High_income (4)} \}$.
- ▶ Given training data of the following form:

Age-range	Income-bracket	Loan sanctioned
1	1	0
2	2	1
	...	
3	4	0

- ▶ Prediction Task:

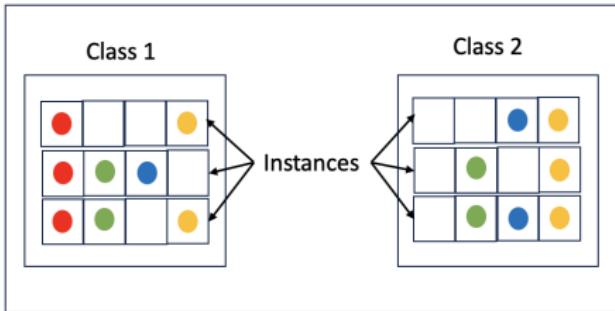
Age-range	Income-bracket	Loan sanctioned
3	2	?

Categorical Data Example: Document Classification

- ▶ $V = \{\text{Set of all unique terms}\}$ (called **vocabulary**).
- ▶ Each document is a $|V|$ -sized vector, each index corresponds to a term.
- ▶ Value 1 if term present, else 0.
- ▶ Example: consider the following four term world.
 - ▶ $V = \{buy, product, cheap, review\}$.
 - ▶ $D_1 = \{buy, cheap, product\}, \mathbf{d}_1 = (1, 1, 1, 0), SPAM(D_1) = 1$.
 - ▶ $D_2 = \{product, review\}, \mathbf{d}_2 = (0, 0, 1, 1), SPAM(D_2) = 0$.
 - ▶ Example task: Predict if a document is spam or not.

Sampling with replacement

Training Set (we know class labels of each instance)



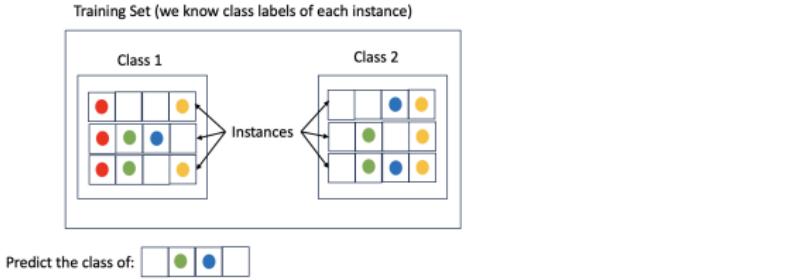
Predict the class of:

▶ Intuitions:

1. Yellow/Green absence or presence doesn't really matter?
2. Red not present \approx Class 2.
3. Blue present \approx Class 2 (sort of puts more confidence that this is indeed class 2).

▶ We now need to **formalise the intuitions**.

Modeling the class priors



Refresher on Bayes Theorem

Naive Bayes

NB for Categorical data

NB for Non-categorical data

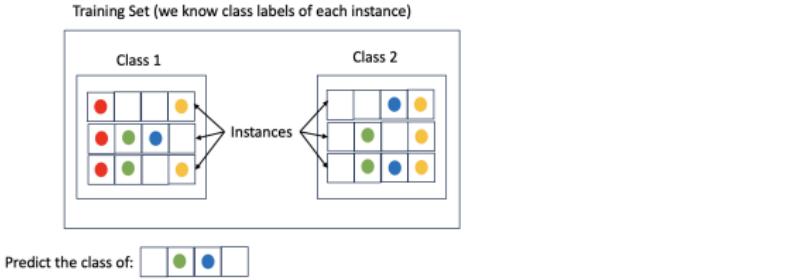
Bayesian Logistic Regression

Laplace approximation

MCMC sampling

- ▶ Compute the **class priors** for each feature (balls).
- ▶ $P(x = \text{Red}|y = 1) = 3/3$
 - ▶ 3 red balls out of 3 total instances in class 1.
 - ▶ You can also estimate this to be 3/8 (3 red balls out of a total of 8 balls)
- ▶ $P(x = \text{Red}|y = 2) = 0$.

Modeling the class priors



- ▶ Compute the **class priors** for each feature (balls).
- ▶ $P(x = \text{Red}|y = 1) = 3/3$
 - ▶ 3 red balls out of 3 total instances in class 1.
 - ▶ You can also estimate this to be 3/8 (3 red balls out of a total of 8 balls)
- ▶ $P(x = \text{Red}|y = 2) = 0$.
- ▶ Repeat for all colors.

Refresher on Bayes Theorem

Naive Bayes

NB for Categorical data

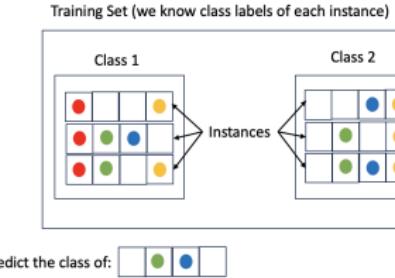
NB for Non-categorical data

Bayesian Logistic Regression

Laplace approximation

MCMC sampling

Modeling the class priors



- ▶ Compute the **class priors** for each feature (balls).
- ▶ $P(x = \text{Red}|y = 1) = 3/3$
 - ▶ 3 red balls out of 3 total instances in class 1.
 - ▶ You can also estimate this to be 3/8 (3 red balls out of a total of 8 balls)
- ▶ $P(x = \text{Red}|y = 2) = 0$.
- ▶ Repeat for all colors.
- ▶ You now have a table of the following form:

Class	Color (Feature value)	Probability
Red	1	3/3 = 1
Green	2	2/3

Refresher on Bayes Theorem

Naive Bayes

NB for Categorical data

NB for Non-categorical data

Bayesian Logistic Regression

Laplace approximation

MCMC sampling

- \mathbf{x}_{test} (we simply write as \mathbf{x}) = {G, B}.

Feature → Class Priors

$$P(\mathbf{x} = \{G, B\} | y = 1)$$

$$= P(G|y = 1) P(B|y = 1)$$

$$= \frac{2}{3} \times \frac{1}{3} = \frac{2}{9}$$

$$P(\mathbf{x} = \{G, B\} | y = 2)$$

$$= P(G|y = 2) P(B|y = 2)$$

$$= \frac{2}{3} \times \frac{2}{3} = \frac{4}{9}$$

Class Priors

$$P(y = 1) = \frac{3}{6} = \frac{1}{2}$$

$$P(y = 1) = \frac{3}{6} = \frac{1}{2}$$

Numerical Example of posterior computation

- ▶ Compute the normalisation constant (denominator in Bayes):
- ▶ $Z = \frac{2}{9} \frac{1}{2} + \frac{4}{9} \frac{1}{2} = \frac{1}{3}$.

$$P(y = 1 | \mathbf{x} = \{G, B\})$$

$$= P(\mathbf{x}|y = 1) P(y = 1) / Z$$

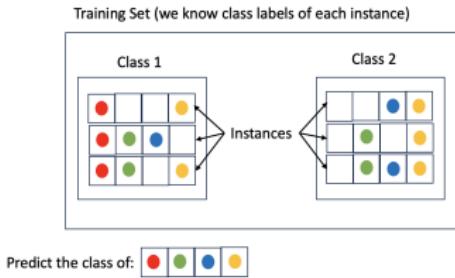
$$= \frac{2}{9} \frac{1}{2} \times 3 = \frac{1}{3}$$

$$P(y = 2 | \mathbf{x} = \{G, B\})$$

$$= P(\mathbf{x}|y = 1) P(y = 1) / Z$$

$$= \frac{4}{9} \frac{1}{2} \times 3 = \frac{2}{3}$$

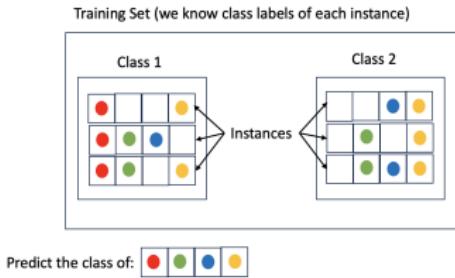
- ▶ So, we predict $\mathbf{x} = \{G, B\}$ to class 2 (and that conforms to our intuitions).



- ▶ What do we predict for $\mathbf{x}_{test} = \{R, G, B, Y\}$?

Menti-Quiz (Code: 2131-2255)

$P(x = Red | y = 2) = 0$. Is that good?



- ▶ What do we predict for $\mathbf{x}_{test} = \{R, G, B, Y\}$?

Menti-Quiz (Code: 2131-2255)

$P(x = Red | y = 2) = 0$. Is that good?

- ▶ Intuitively class 2. But what if it's actually class 1 (not that unlikely).
- ▶ Our prediction depends too much on the absence of Red.
- ▶ We would still like to have some small probability of sampling a red from class 2.
- ▶ Otherwise, we overfit on our training data.

- ▶ Add 1 to the numerator and K (#categories) to the denominator.
- ▶ In our example, number of categories is 4 (Size of the set $\{R, G, B, Y\}$).
- ▶ $P(x = Red|y = 2) = \frac{1 + \text{Count}(x=Red|y=2)}{\# \text{categories} + \text{Count(class2)}} = \frac{1+0}{4+7}.$
- ▶ And all other probabilities need to be adjusted in the same way.
- ▶ **Homework:**
 - ▶ Work out the posterior probability estimates with this modified notion of defining the smoothed probabilities.
 - ▶ Predict the class of $x_{test} = \{R, G, B, Y\}$.

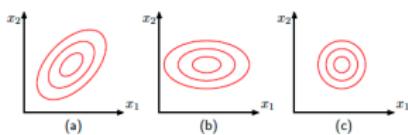
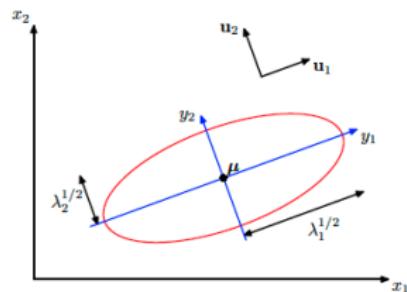
- ▶ Independence assumption requires computing:
 - ▶ $P(\mathbf{x}|y = k) = \prod_{x \in \mathbf{x}} P(x|y = k).$
- ▶ Leads to floating point arithmetic underflows with large dimensions. Why?
- ▶ Because we're multiplying numbers that are all < 1 .
- ▶ An elegant engineering solution is to work with the $\log(1 + P)$ s, and predict the log-likelihoods.

A refresher on Gaussians

m-dimensional Gaussian

$$P(\mathbf{x}|\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})) \propto \exp(-(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}))$$

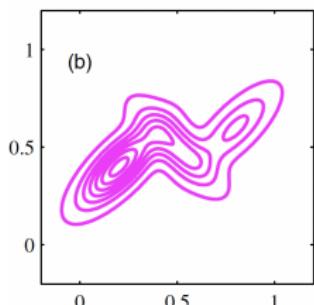
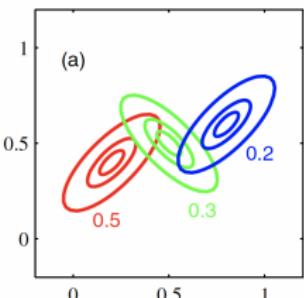
- ▶ $\boldsymbol{\mu} \in \mathbb{R}^m$ - the centroid is a vector of m dimensions
- ▶ $\boldsymbol{\Sigma} \in \mathbb{R}^m \times \mathbb{R}^m$ - the spread matrix is of size $m \times m$



- ▶ Figure on the left shows a 2D example.
- ▶ Effect of the spread matrix:
 - ▶ a) $((1, 1), (1, 1))$ – Anisotropic w/ cross correlation
 - ▶ b) $((5, 0), (0, 1))$ – Anisotropic w/o cross correlation
 - ▶ c) $((1, 0), (0, 1))$ – Isotropic

Colab notebook: ↗

Mixture of Gaussians (MoG)

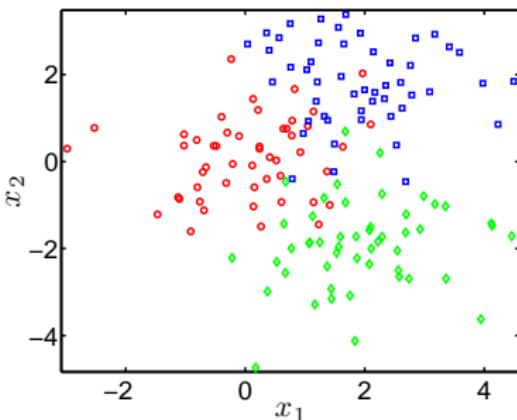


Gaussian Mixture distribution

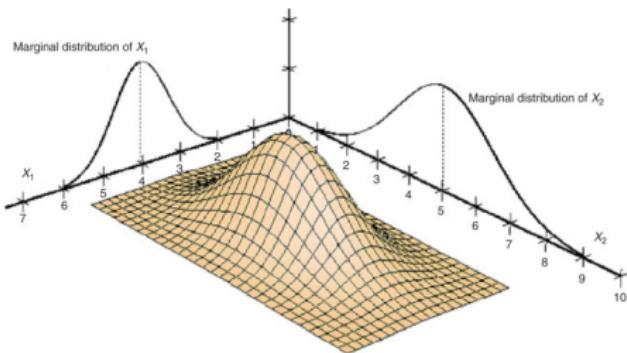
$$\sum_{i=1}^k p_i \mathcal{N}(\mu_i, \Sigma_i)$$

- ▶ p_i : Prior belief that an observed point \mathbf{x} is generated by the i^{th} component.
- ▶ Posterior: Given \mathbf{x} , the likelihood that it is generated from the i^{th} component -

- ▶ $P(Y = i|\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)$

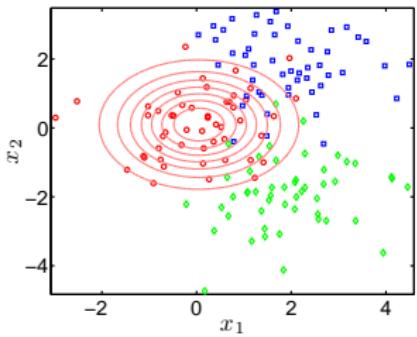


- ▶ 2D data: $\mathbf{x} = (x_1, x_2)$.
- ▶ $K = 3$ classes.
- ▶ $P(y = k) = 1/\#\text{classes}$ – uniform class priors.



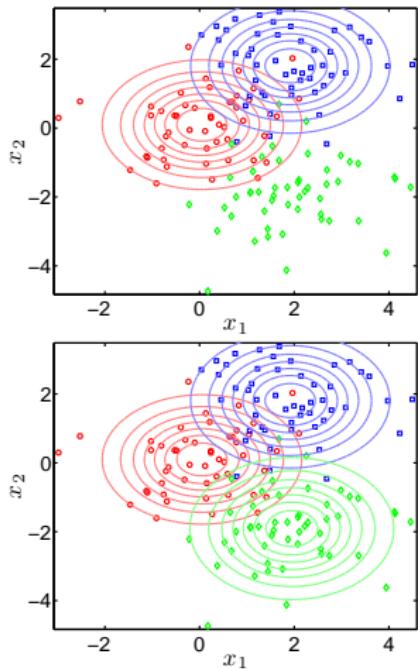
- ▶ A marginal of a Gaussian is a projection on to a lower dimension (analogous to a shadow) .
 - ▶ The projections themselves are Gaussians.
 - ▶ The two projections combined is **not equivalent** to the mixture distribution (except when cross-correlation components are 0s)
 - ▶ In NB, we work with the projections.

NB Algorithm (Training)



- ▶ Each component a Gaussian with parameters μ and Σ .
- ▶ For NB, Σ is w/o cross-correlation (independence of features).
- ▶ Hence, for dimension d we work with d number of **univariate (1-d) Gaussians**, each with parameters μ_i and σ_i ($i = 1, \dots, d$).
- ▶ In the example, for the given red points, calculate μ^R and σ^R .
 - ▶ $\mu^R = \frac{1}{|R|} \sum_{\mathbf{x}: y(x)=R} \mathbf{x}$
 - ▶ $\sigma_i^R = \sqrt{\frac{1}{|R|} \sum_{\mathbf{x}: y(x)=R} (\mathbf{x}_i - \mu_i^R)^2}$

NB Algorithm (Training)

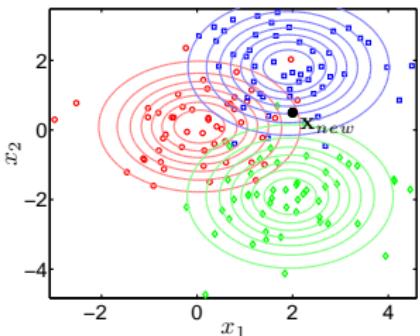


- ▶ Repeat the same steps for the green and the blue components.

$$\▶ \mu^G = \frac{1}{|G|} \sum_{\mathbf{x}:y(x)=G} \mathbf{x}$$

$$\▶ \sigma_i^G =$$

$$\frac{1}{|G|} \sqrt{\sum_{\mathbf{x}:y(x)=G} (\mathbf{x}_i - \mu_i^G)^2}$$



- ▶ For each given point x_{new} :
- ▶ Compute the posterior likelihoods for each component, e.g., for red:
- ▶
$$P(x_{new} = R) = (\mathbf{x} - \boldsymbol{\mu}^R)^T ((\sigma_1^R, 0)(0, \sigma_2^R))^{-1}(\mathbf{x} - \boldsymbol{\mu}^R)$$
- ▶ Similarly,
$$P(x_{new} = G) = (\mathbf{x} - \boldsymbol{\mu}^G)^T ((\sigma_1^G, 0)(0, \sigma_2^G))^{-1}(\mathbf{x} - \boldsymbol{\mu}^G)$$
, and so on.
- ▶ Multiply all these posteriors with the class priors, i.e., $P(x_{new} = R) \leftarrow P(x_{new} = R) \times P(R)$, and so on.
- ▶ Classify the point to be in the class with the highest probability.

Calibration of Model Confidence

Introduction

D. Ganguly

Menti-Quiz (Code: 2131-2255)

Let's see how confident humans are!

Refresher on Bayes Theorem

Naive Bayes

NB for Categorical data

NB for Non-categorical data

Bayesian Logistic Regression

Laplace approximation

MCMC sampling

Calibration of Model Confidence

Introduction

D. Ganguly

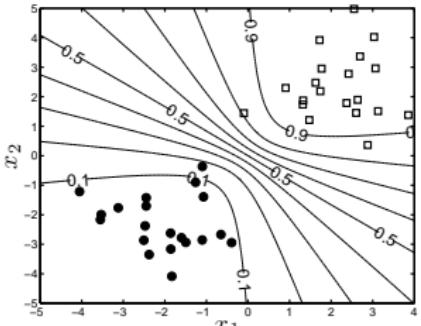
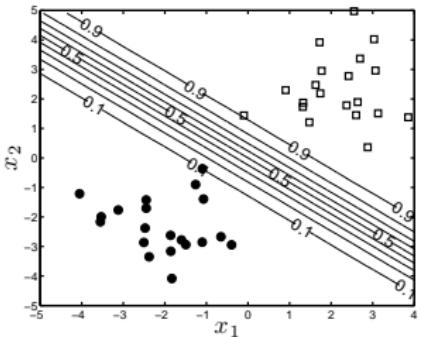
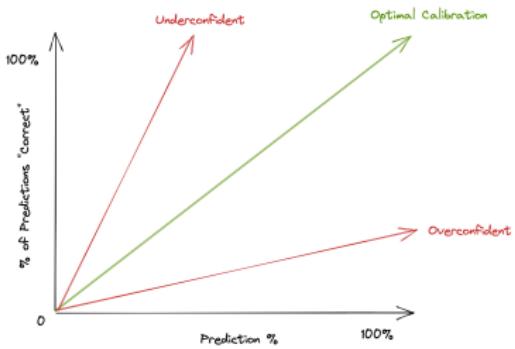
NB for Categorical data

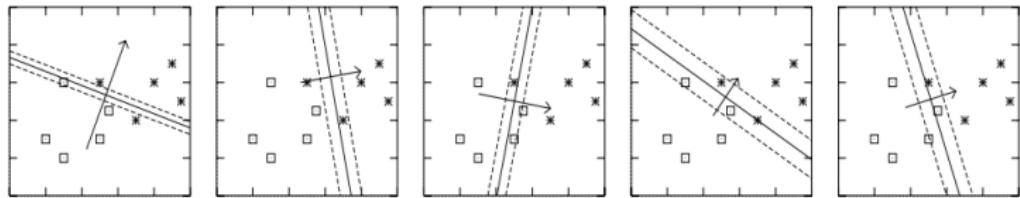
NB for Non-categorical data

Bayesian Logistic Regression

Laplace approximation

MCMC sampling





- ▶ Which decision boundary is better?
- ▶ Can estimations change if we see more data samples?

The Core Equation

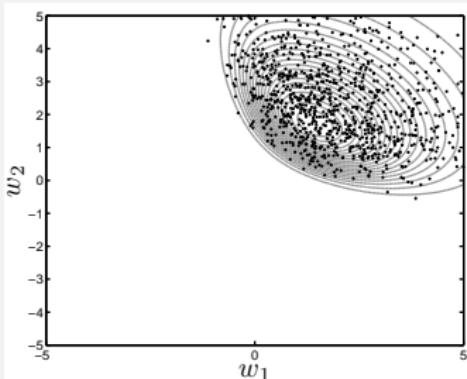
$$\underbrace{P(Y|X, \mathcal{D})}_{\text{Overall belief}} = \int \underbrace{P(Y|X, \theta)}_{\text{Belief with one model}} \underbrace{P(\theta|\mathcal{D})d\theta}_{\text{Model likelihood}}$$

- ▶ Aggregate the different beliefs of explaining your data samples by various models $P(Y|X, \theta)$ weighted by their priors $P(\theta|\mathcal{D})$.

Bayesian: Addresses Model Uncertainty

A Bayesian approach

- ▶ Doesn't estimate a single point, but rather a distribution.
- ▶ In the **data space**: Observed data is sampled from an underlying distribution (e.g., an MoG).
- ▶ Analogously, points in the parameter space are also sampled from a distribution \Rightarrow **not a single solution but a distribution that yields potential solutions.**



Variational Approximation

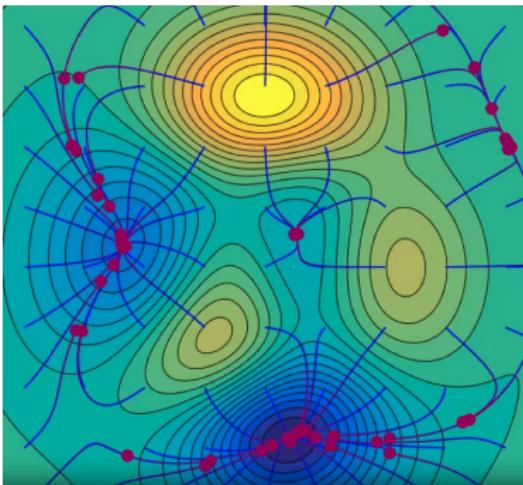
- ▶ Approximate $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ with another distribution.

- ▶ $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$: Model parameter uncertainty, i.e., the **distribution of model parameters** given the training data.

Variational Approximation

► Approximate $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ with another distribution.

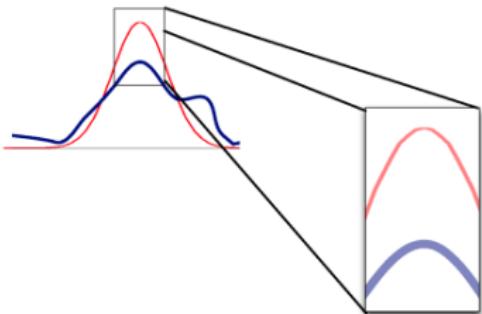
- $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$: Model parameter uncertainty, i.e., the **distribution of model parameters** given the training data.
- Realise that: with data perturbation or a different starting point of MAP (gradient descent), we may converge on different decision boundaries.



Variational Approximation by Laplace Algorithm

Variational distribution $q(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$

- ▶ Should be **similar in behaviour** to the distribution that we want to approximate.
- ▶ We then need an algorithm to estimate the parameters of this variational distribution q .



- ▶ Fit the variational distribution at the **mode** (**most likely region**) of the original distribution.
 - ▶ This is the region where we want a close fit.
 - ▶ Rest doesn't matter that much.

Laplace approximation

- ▶ Approximate $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ with a Gaussian:

$$q(\mathbf{w}|\mathbf{X}, \mathbf{t}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- ▶ We now need to estimate the values of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

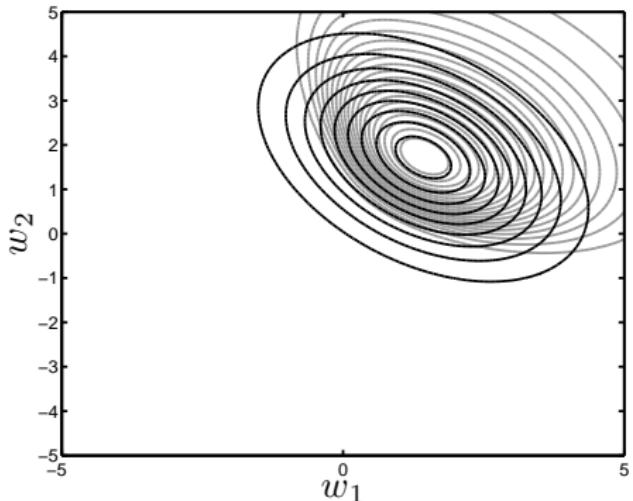
Posterior likelihood

$g = p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)$ - distribution of model predictions

$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \log g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$ – the mode of this distribution

Consider the point estimate as the mode

$$\boldsymbol{\mu} = \hat{\mathbf{w}}, \quad \boldsymbol{\Sigma}^{-1} = -\underbrace{\frac{\partial \log g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)}{\partial \mathbf{w} \partial \mathbf{w}^\top}}_{\text{spread of the variational}}$$



- ▶ Dark lines – **approximation**.
- ▶ Light lines – $\propto p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$.
- ▶ Approximation is OK.
- ▶ As expected, it gets worse as we travel away from the mode.

- ▶ We have $\mathcal{N}(\mu, \Sigma)$ as an approximation to $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$.
- ▶ Can we use it to make predictions?
- ▶ Need to evaluate:

$$\begin{aligned} P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) &= \mathbb{E}_{\mathcal{N}(\mu, \Sigma)} \{P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})\} \\ &= \int \mathcal{N}(\mu, \Sigma) \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x}_{\text{new}})} d\mathbf{w} \end{aligned}$$

- ▶ We can't quantitatively evaluate the integral. So, what was the point?

Sampling from $\mathcal{N}(\mu, \Sigma)$ is **easy**

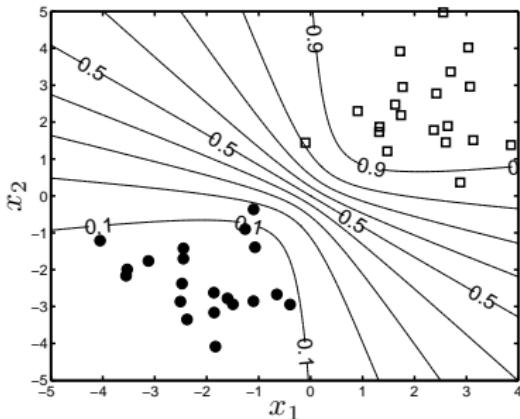
And we can approximate an expectation with samples!

Predictions with the Laplace approximation

Expectation over samples

Draw S samples $\mathbf{w}_1, \dots, \mathbf{w}_S$ from $\mathcal{N}(\mu, \Sigma)$

$$\mathbf{E}_{\mathcal{N}(\mu, \Sigma)} \{P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})\} \approx \frac{1}{S} \sum_{s=1}^S \frac{1}{1 + \exp(-\mathbf{w}_s^T \mathbf{x}_{\text{new}})}$$

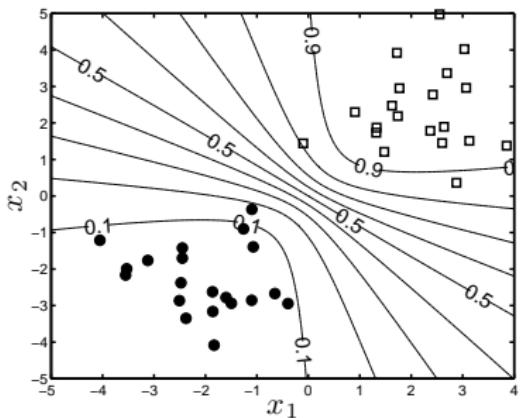


- ▶ Contours of $P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$.
- ▶ Better than those from the point prediction?

Predictions with the Laplace approximation

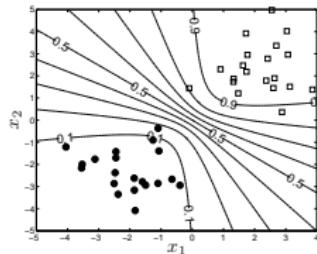
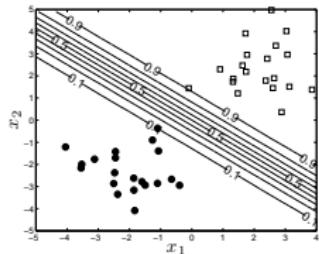
Predictions averaged over many classifiers

$$\mathbf{E}_{\mathcal{N}(\mu, \Sigma)} \{P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})\} \approx \frac{1}{S} \sum_{s=1}^S \frac{1}{1 + \exp(-\mathbf{w}_s^T \mathbf{x}_{\text{new}})}$$



- ▶ This classifier is **less certain** in its predictions.
- ▶ Only if a point is **close to a centroid of a class**, the classifier's posterior probability is high.

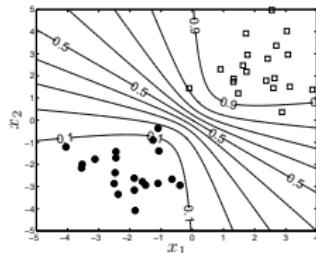
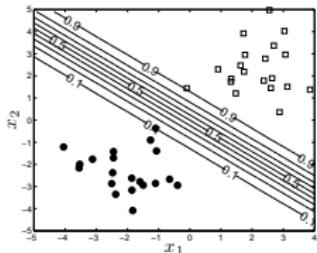
Point prediction v Laplace approximation



Menti-Quiz (Code: 2131-2255)

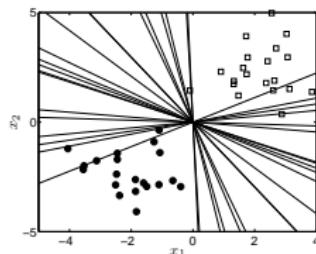
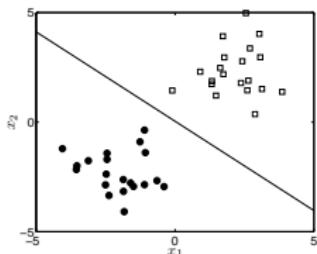
Why the difference? Explain

Point prediction v Laplace approximation



Menti-Quiz (Code: 2131-2255)

Why the difference? Explain



Laplace uses a distribution ($\mathcal{N}(\mu, \Sigma)$) over \mathbf{w} (and therefore a distribution over decision boundaries).

Laplace Algorithm

- ▶ Computationally expensive!
- ▶ Suffers from curse of dimensionality.
- ▶ We rely on sampling for the eventual predictions.
 - ▶ Do we then need to estimate a parametric distribution to be able to sample from it?
 - ▶ No - let's motivate the idea of **Rejection Sampling**.

The idea of Rejection Sampling

Analogy

- ▶ Curve that defines a quadrant of a circle
≡ True distribution of w .
- ▶ Both are unknown.
- ▶ Use a criterion to decide whether a sample is to be considered or rejected.
- ▶ Compute stats on the acceptance vs. rejection to estimate the desired distribution.

Refresher on Bayes Theorem

Naive Bayes

NB for Categorical data

NB for Non-categorical data

Bayesian Logistic Regression

Laplace approximation

MCMC sampling

Metropolis-Hastings

MH Sampling

Similar to the idea of accepting or rejecting if a point lies within a unit radius or not for computing π .

Proposal and acceptance

- ▶ Produces a sequence of samples – $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_s, \dots$
- ▶ Consider the last sample accepted: \mathbf{w}_{s-1}
- ▶ First *propose* a **candidate** \mathbf{w}_s ($\widetilde{\mathbf{w}}_s$) based on \mathbf{w}_{s-1} .
- ▶ Then decide whether or not to *accept* $\widetilde{\mathbf{w}}_s$
 - ▶ If accepted, $\mathbf{w}_s = \widetilde{\mathbf{w}}_s \implies$ we moved in the parameter space!
 - ▶ If not, $\mathbf{w}_s = \mathbf{w}_{s-1} \implies$ try again!

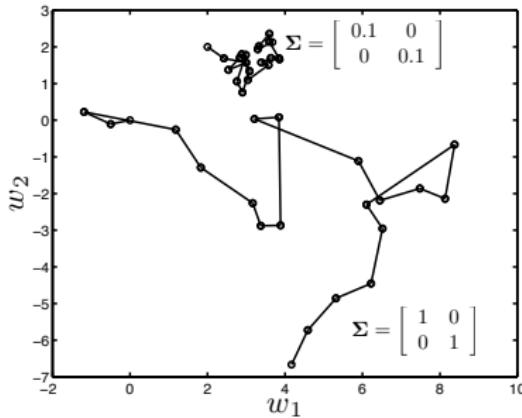
MH – proposal

- ▶ Treat $\tilde{\mathbf{w}}_s$ as a random variable **conditioned on \mathbf{w}_{s-1}**
- ▶ Define $p(\tilde{\mathbf{w}}_s | \mathbf{w}_{s-1})$
 - ▶ Note that this does not necessarily have to be similar to posterior we're trying to sample from.
- ▶ Can be **any distribution!** So, choose a parametric distribution that is easy to work with.

MH – proposal

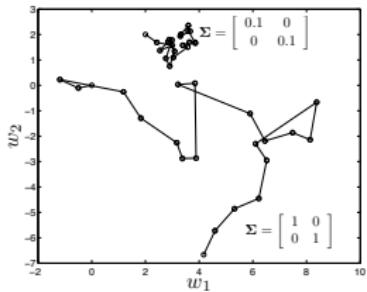
- ▶ Treat $\tilde{\mathbf{w}_s}$ as a random variable **conditioned on** \mathbf{w}_{s-1}
 - ▶ Define $p(\tilde{\mathbf{w}_s} | \mathbf{w}_{s-1})$
 - ▶ Note that this does not necessarily have to be similar to posterior we're trying to sample from.
 - ▶ Can be **any distribution!** So, choose a parametric distribution that is easy to work with.
 - ▶ Use a Gaussian centred on \mathbf{w}_{s-1} with some covariance:

$$p(\widetilde{\mathbf{w}_s} | \mathbf{w}_{s-1}, \boldsymbol{\Sigma}_p) = \mathcal{N}(\mathbf{w}_{s-1}, \boldsymbol{\Sigma}_p)$$



Random Walk in MH

- ▶ Start from a random point on the parameter space.
- ▶ Estimations get better with a higher number of iterations.



- ▶ What is the role of σ in MH sampling?
- ▶ High σ :
 - ▶ **Aggressive** exploration.
 - ▶ Effective if point estimate $\hat{\mathbf{w}}$ is bad.
- ▶ Low σ :
 - ▶ **Conservative** exploration.
 - ▶ More exploitation: works well if point estimate $\hat{\mathbf{w}}$ is good.

MH Acceptance for Symmetric Variational

- p : Distribution that we want to approximate.
- Acceptance probability r .

Acceptance probability r

$$r = \min\left(1, \frac{p(\widetilde{\mathbf{w}}_s | \mathbf{X}, \mathbf{t}, \sigma^2)}{p(\mathbf{w}_{s-1} | \mathbf{X}, \mathbf{t}, \sigma^2)}\right).$$

- If more probable (better) sample drawn ($p(\widetilde{\mathbf{w}}_s)$) then **always accept it**.
- Else accept it with some probability r .
- Small r means that:
 - $p(\widetilde{\mathbf{w}}_s)$ is small – this model is not likely to be effective.
Hence accept only with a small probability.
 - $p(\mathbf{w}_{s-1} | \mathbf{X}, \mathbf{t}, \sigma^2)$ is high – we already have got good solutions; risky to explore more.

MH Acceptance for Asymmetric Variational

Acceptance probability r

$$r = \min\left(1, \frac{p(\tilde{\mathbf{w}}_s | \mathbf{X}, \mathbf{t}, \sigma^2)}{p(\mathbf{w}_{s-1} | \mathbf{X}, \mathbf{t}, \sigma^2)} \frac{p(\mathbf{w}_{s-1} | \tilde{\mathbf{w}}_s, \Sigma_p)}{p(\tilde{\mathbf{w}}_s | \mathbf{w}_{s-1}, \Sigma_p)}\right).$$

- ▶ The ratio of $p(\mathbf{w}_{s-1} | \tilde{\mathbf{w}}_s, \Sigma_p)$ and $p(\tilde{\mathbf{w}}_s | \mathbf{w}_{s-1}, \Sigma_p)$ is called the Hastings correction.
- ▶ High $p(\mathbf{w}_{s-1} | \tilde{\mathbf{w}}_s, \Sigma_p)$: We can explain our existing model with the new model (**Good Exploitation**).
- ▶ Low $p(\tilde{\mathbf{w}}_s | \mathbf{w}_{s-1}, \Sigma_p)$: The new model is different from what we have seen so far (**Good Exploration**).

MH for Logistic Regression

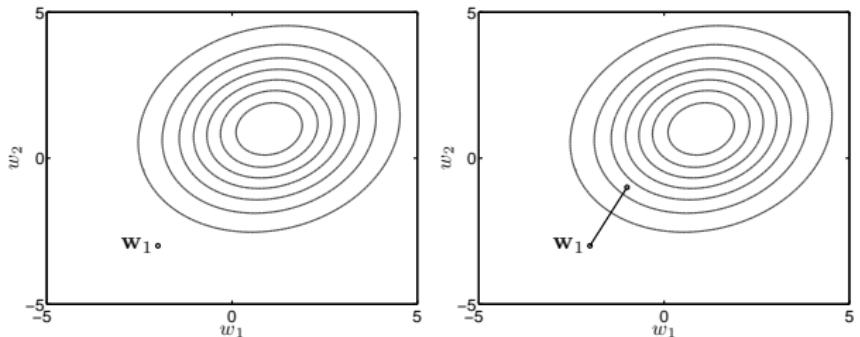
Generic Formula

$$r = \min\left(1, \frac{p(\tilde{\mathbf{w}}_s | \mathbf{X}, \mathbf{t}, \sigma^2)}{p(\mathbf{w}_{s-1} | \mathbf{X}, \mathbf{t}, \sigma^2)} \frac{p(\mathbf{w}_{s-1} | \tilde{\mathbf{w}}_s, \Sigma_p)}{p(\tilde{\mathbf{w}}_s | \mathbf{w}_{s-1}, \Sigma_p)}\right).$$

In Logistic Regression

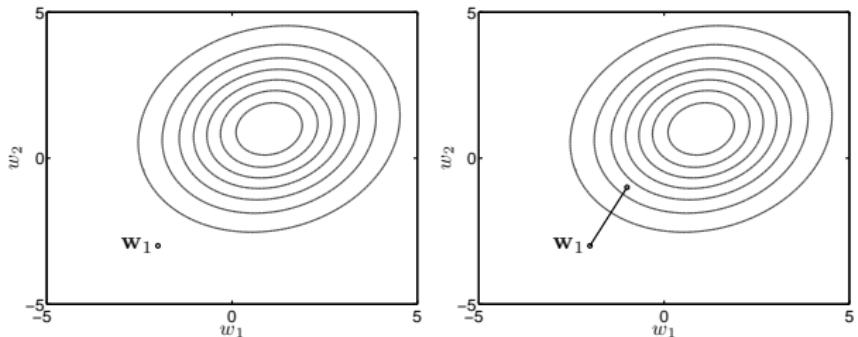
$$r = \min\left(1, \frac{g(\tilde{\mathbf{w}}_s; \mathbf{X}, \mathbf{t}, \sigma^2)}{g(\mathbf{w}_{s-1}; \mathbf{X}, \mathbf{t}, \sigma^2)} \frac{p(\mathbf{w}_{s-1} | \tilde{\mathbf{w}}_s, \Sigma_p)}{p(\tilde{\mathbf{w}}_s | \mathbf{w}_{s-1}, \Sigma_p)}\right).$$

- ▶ Replace $p(\mathbf{w} | \mathbf{X})$ with the cross-entropy posterior g (as we did for Laplace Approximation).



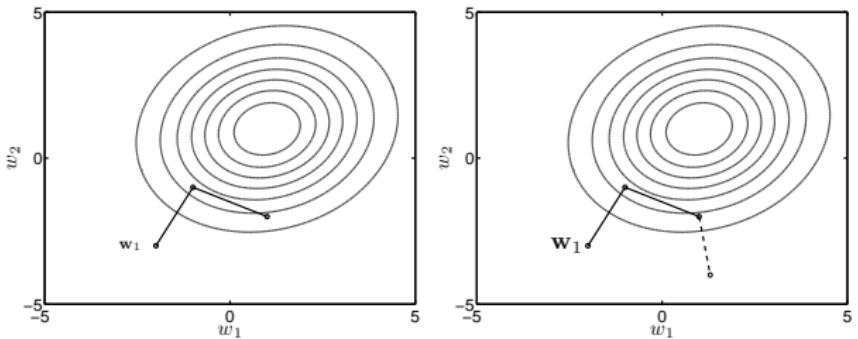
Menti-Quiz (Code: 2131-2255)

Why do we accept the first sample?



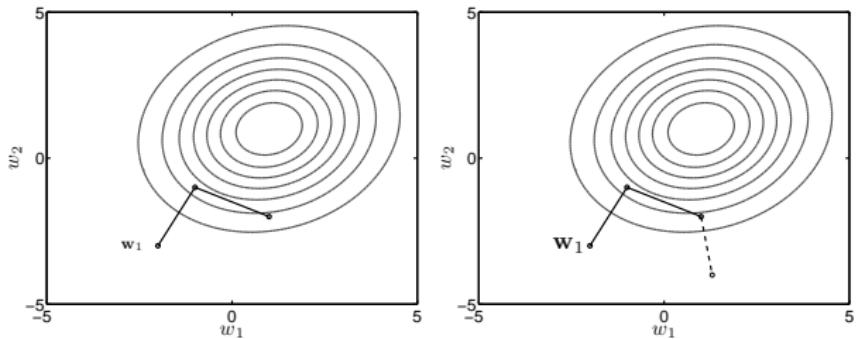
Menti-Quiz (Code: 2131-2255)

Why do we accept the first sample?



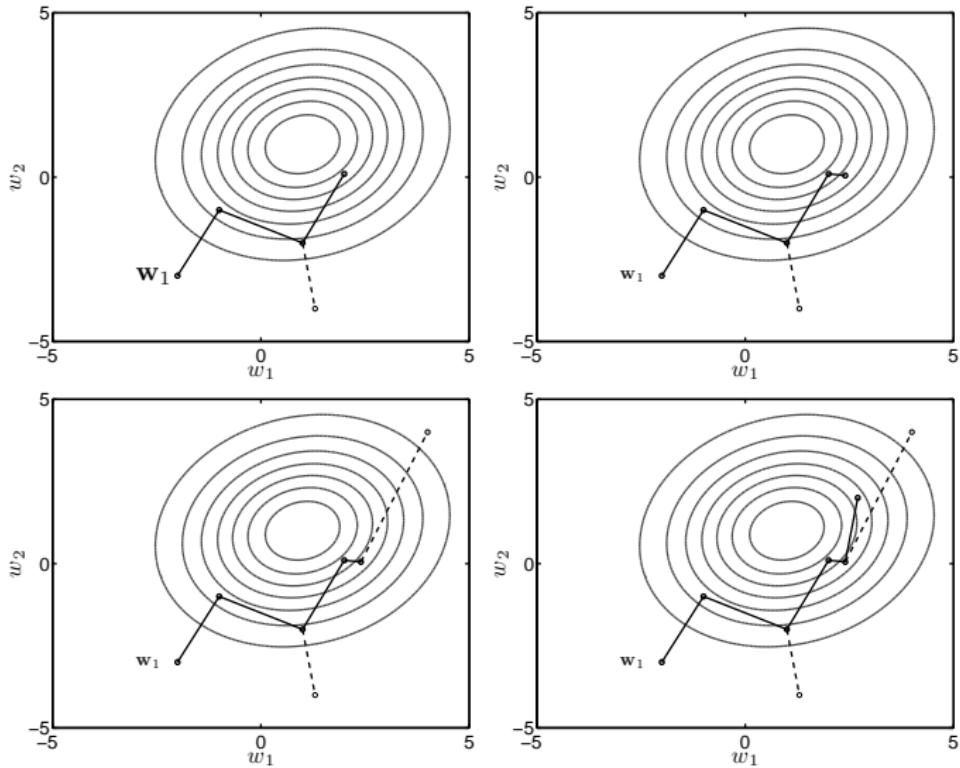
Menti-Quiz (Code: 2131-2255)

Why do we reject the third sample?



Menti-Quiz (Code: 2131-2255)

Why do we reject the third sample?



Refresher on Bayes Theorem

Naive Bayes

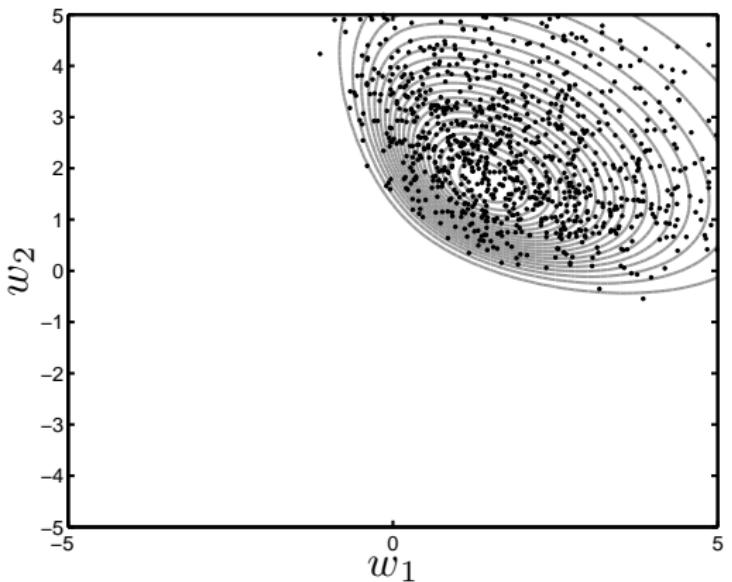
NB for Categorical data

NB for Non-categorical data

Bayesian Logistic Regression

Laplace approximation

MCMC sampling

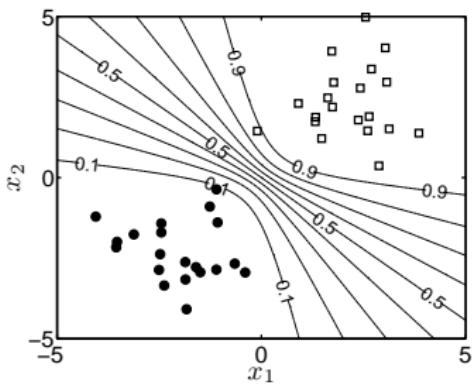


- ▶ 1000 samples from the posterior using MH.

Predictions with MH

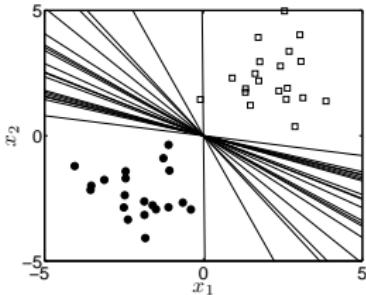
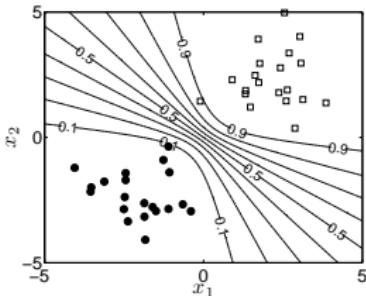
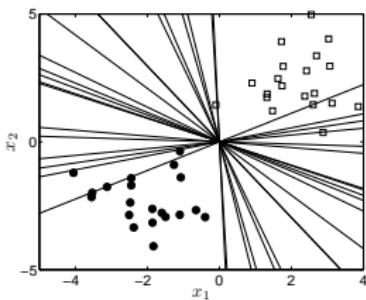
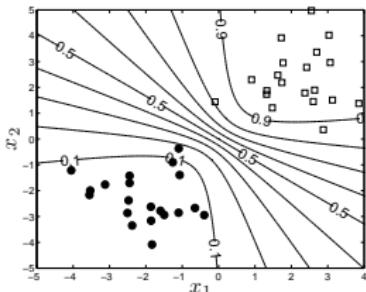
- MH provides us with a set of samples – $\mathbf{w}_1, \dots, \mathbf{w}_S$.
- These can be used like the samples from the Laplace approximation:

$$\begin{aligned} P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}, \sigma^2) &= \mathbf{E}_{p(\mathbf{w} | \mathbf{X}, \mathbf{t}, \sigma^2)} \{P(t_{\text{new}} | \mathbf{x}_{\text{new}}, \mathbf{w})\} \\ &\approx \frac{1}{S} \sum_{s=1}^S \frac{1}{1 + \exp(-\mathbf{w}_s^\top \mathbf{x}_{\text{new}})} \end{aligned}$$



- Contours of $P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}, \sigma^2)$

Laplace v MH



- ▶ Laplace approximation allows some *bad* boundaries.
- ▶ Hence over-compensates for uncertainty modeling.
- ▶ See how close the 0.9 contours are to the centroids.

- ▶ MH is more selective.
- ▶ Achieves the right amount of calibration for uncertainty.
- ▶ See that the 0.9 contours aren't too close to the centroids.

- ▶ Two applications of Bayes Theorem in Machine Learning.
- ▶ First: To build a classifier based on the independence assumption of features.
- ▶ Second: Calibrate model confidences for uncertainties.
- ▶ Two specific algorithms for extending the MAP estimate.
 - ▶ Laplace Approximation - Involves **estimating parameters of a distribution** and samples.
 - ▶ Metropolis Hastings - **No need to estimate parameters**. Find the distribution by rejection sampling.
- ▶ **Next week:**
 - ▶ Bayesian is a calibration approach - no direct modeling of uncertainty in the objective function.
 - ▶ Will discuss a **more direct approach** of modeling uncertainty - **Support Vector Machines**.
 - ▶ **Change in the objective function** itself.