Advanced Statistical Inference
Maurizio Filippone

Lecture1: Introduction

Recap on linear algebra & probability theory.

Lecture 2: Bayesian Linear Regression

Preliminaries

Probabilities

Sum rule: $p(x) = \int p(x,y)dy$

Product rule: $p(x,y) = p(x|y)p(y) = p(y|x)p(x)$

Expectations

Discrete: $\tilde{x} = \mathbb{E}_{p(x)}(x) = \sum xp(x)$

Continuous: $\tilde{x} = \mathbb{E}_{p(x)}(x) = \int xp(x)dx$. In general, $\mathbb{E}_{p(x)}[f(x)] = \int f(x)p(x)dx$

Mean and Covariance:

$\mu = \mathbb{E}_{p(x)}[x]$

$\sigma^2 = \mathbb{E}_{p(x)}[(x-\mu)^2]$

$cov(x) = \mathbb{E}_{p(x)}[(x-\mu)(x-\mu)^T] = \mathbb{E}_{p(x)}[xx^T] - \mu\mu^T$

The Gaussian Distribution

$p(v|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left\{-\frac{1}{2\sigma^2}(v-\mu)^2\right\}$

The Multivariate Gaussian Distribution

$p(\boldsymbol{v}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{v}|\boldsymbol{\mu},\boldsymbol{\Sigma})$ where $\mathcal{N}(\boldsymbol{v}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\boldsymbol{v}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{v}-\boldsymbol{\mu})\right\}$

The eigenvalues of the covariance give us the alignment of the distribution.

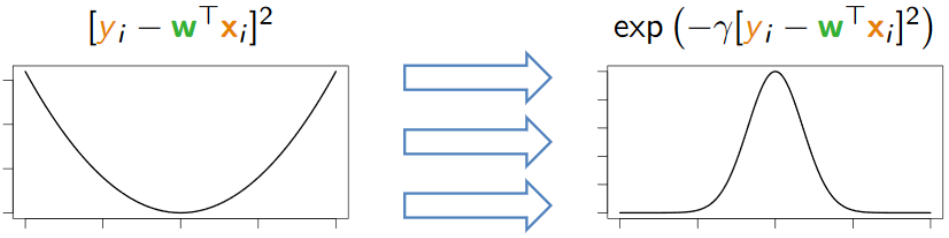
Loss Minimization in Linear Regression

Linear Models for Regression: $f(\boldsymbol{x}) = \sum w_i\phi_i(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x})$

It is a linear model because the parameters appear in a linear way, not because of the linear basis functions.

Quadratic loss function: $\mathcal{L} = \sum(y_i - \boldsymbol{w}^T\boldsymbol{x_i})^2 = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2$. Solution: $\nabla_{\boldsymbol{w}}\mathcal{L} = 0 \Rightarrow \widehat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$

Probabilistic Interpretation of Loss: minimizing the quadratic loss is equivalent to minimizing the Gaussian likelihood function $\exp(-\gamma\mathcal{L}) = \exp(-\gamma\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2) \propto \mathcal{N}\left(\boldsymbol{y}|\boldsymbol{X}\boldsymbol{w}, \frac{1}{2\gamma}\right)$.



Model Selection

Loss minimization/likelihood maximization is not sufficient to select the best model because we can have generalization problems due to overfitting.

Cross-validation: evaluate models on randomly picked data using validation loss or validation log-likelihood.

Bayesian Inference

Model parameters are considered as probability distributions. Going from $p(\boldsymbol{w})$ to $p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y})$.

Bayes rule: $p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})p(\boldsymbol{w})}{\int p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})p(\boldsymbol{w})dw} = \frac{p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{y}|\boldsymbol{X})}$

Posterior density $p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y})$: the distribution over parameters after observing the data.

Likelihood $p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})$: measure of fitness.

Prior density $p(\boldsymbol{w})$: anything we know about parameters before we see any data.

Marginal likelihood $p(\boldsymbol{y}|\boldsymbol{X})$: normalization constant.

When can we compute the posterior? When the multiplication results is a posterior of same type of density as the prior.

Why it is important? Because we do not have to calculate $p(\boldsymbol{y}|X)$ because we know the form of $p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y})$.

Finding posterior parameters

The posterior must be Gaussian:

$$p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{\sigma}^2) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{w}-\boldsymbol{\mu})\right\} \propto \exp\left\{-\frac{1}{2}(\boldsymbol{w}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{w} - 2\boldsymbol{w}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right\}$$

On the other hand, we multiply the likelihood and the prior:

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})p(\boldsymbol{w}) &\propto \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})\right\}\exp\left\{-\frac{1}{2}\boldsymbol{w}^T\boldsymbol{S}^{-1}\boldsymbol{w}\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left(\boldsymbol{w}^T\left[\frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{S}^{-1}\right]\boldsymbol{w} - \frac{2}{\sigma^2}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y}\right)\right\} \end{aligned}$$

We extract parameters:

Covariance: $\boldsymbol{\Sigma} = \left(\frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{S}^{-1}\right)^{-1}$

Mean: $\boldsymbol{\mu} = \frac{1}{\sigma^2}\boldsymbol{\Sigma}\boldsymbol{X}^T\boldsymbol{y}$

Introducing basis functions

We replace X of size (N, D) by $\phi(X)$ of size (N, D') (expending or compressing the features).

Covariance: $\boldsymbol{\Sigma} = \left(\frac{1}{\sigma^2}\boldsymbol{\phi}^T\boldsymbol{\phi}+\boldsymbol{S}^{-1}\right)^{-1}$

Mean: $\boldsymbol{\mu} = \frac{1}{\sigma^2}\boldsymbol{\Sigma}\boldsymbol{\phi}^T\boldsymbol{y}$

Prediction: $p(y_*|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_*,\sigma^2) = \mathcal{N}\left(y_*|\boldsymbol{\phi}(\boldsymbol{x}_*)^T\boldsymbol{\mu}, \sigma^2 + \boldsymbol{\phi}(\boldsymbol{x}_*)^T\boldsymbol{\Sigma}\boldsymbol{\phi}(\boldsymbol{x}_*)\right)$

Lecture 3: Gaussian Processes

Introduction

How to choose which basis functions (polynomials, trigonometric, etc.) to use?

Gaussian Processes learn a probabilistic combination of an infinite set of basis functions.

Weight Space View

Recap:

Modeling observations as noisy realizations of a linear combination of features $p(\boldsymbol{y}|\boldsymbol{w},\boldsymbol{X},\sigma^2) = \mathcal{N}(\boldsymbol{X}\boldsymbol{w},\sigma^2\boldsymbol{I})$.

Gaussian prior over model parameters: $p(w) = \mathcal{N}(\boldsymbol{0},\boldsymbol{S})$.

BLR as a Kernel Machine

Working with $\psi(.)$ costs $O(D^2)$ storage and $O(D^3)$ time

Working with $k(.,.)$ costs $O(N^2)$ storage and $O(N^3)$ time

We can pick $k(.,.)$ so that $\psi(.)$ is infinite dimensional.

$k(\boldsymbol{x},\boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|}{2}\right) = \psi(\boldsymbol{x})^T\psi(\boldsymbol{x}')$ where $\psi(\boldsymbol{x}) = \exp\left(-\frac{\boldsymbol{x}^2}{2}\right)\left(1, \boldsymbol{x}, \frac{\boldsymbol{x}^2}{\sqrt{2!}}, \dots\right)^T$

When working with a kernel, we are implicitly working with a finite number of basis functions.

Using Woodbury identity, we can write:

$$\boldsymbol{\Sigma} = \left(\frac{1}{\sigma^2}\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \boldsymbol{S}^{-1}\right)^{-1} = \boldsymbol{S} - \boldsymbol{S}\boldsymbol{\Phi}^T(\sigma^2\boldsymbol{I} + \boldsymbol{\Phi}\boldsymbol{S}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{S}$$

We can rewrite the variance:

$$\sigma^2 + \phi_*^T\boldsymbol{\Sigma}\phi_* = \sigma^2 + \phi_*^T\boldsymbol{S}\phi_* - \phi_*^T\boldsymbol{S}\boldsymbol{\Phi}^{\mathrm{T}}(\sigma^2\boldsymbol{I} + \boldsymbol{\Phi}\boldsymbol{S}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{S}\phi_* = \sigma^2 + k_{**} - \boldsymbol{k}_*^T(\sigma\boldsymbol{I} + \boldsymbol{K})^{-1}\boldsymbol{k}_*$$

We can also rewrite the mean by applying Woodbury identity twice (no thanks!).

$$\phi_*^T\mu = \cdots = \phi_*^T\boldsymbol{S}\boldsymbol{\Phi}^T(\sigma^2\boldsymbol{I} + \boldsymbol{\Phi}\boldsymbol{S}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{t} = \boldsymbol{k}_*^T(\sigma\boldsymbol{I} + \boldsymbol{K})^{-1}\boldsymbol{t}$$

Where

$$\psi(\boldsymbol{x}) = \boldsymbol{S}^{1/2}\phi(\boldsymbol{x})$$
$$k_{**} = k(\boldsymbol{x}_*,\boldsymbol{x}_*) = \psi(\boldsymbol{x}_*)^T\psi(\boldsymbol{x}_*)$$
$$(\boldsymbol{k}_*)_i = k(\boldsymbol{x}_*,\boldsymbol{x}_i) = \psi(\boldsymbol{x}_*)^T\psi(\boldsymbol{x}_i)$$
$$(\boldsymbol{K})_{ij} = k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \psi(\boldsymbol{x}_i)^T\psi(\boldsymbol{x}_j)$$

$\boldsymbol{\phi}$ is a D by infinite matrix.

Function Space View

We will consider an infinite number of random variables $f(\boldsymbol{x})$ indexed by $\boldsymbol{x}$. The covariance of those variables is an infinite by infinite zero matrix except on the diagonal where we have the variance of each variable.

$$k(\boldsymbol{x},\boldsymbol{x}') = \alpha\exp(-\beta\|\boldsymbol{x}-\boldsymbol{x}'\|^2)$$

Using the distance kernel we can impose nearby variables to have high covariance.

And the we select N variables by selecting the corresponding rows and columns from the covariance matrix.

Marginal distribution $\boldsymbol{f} = \left(f(x_1),\dots,f(x_N)\right)^T$ is $p(\boldsymbol{f}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{0},\boldsymbol{K})$

Conditional distribution of f_* given $\boldsymbol{f}$ is $p(f_*|f,\boldsymbol{x}_*,\boldsymbol{X}) = \mathcal{N}(\bar{m},\bar{s}^2)$

Where $\bar{m} = \boldsymbol{k}_*^T\boldsymbol{K}^{-1}\boldsymbol{f}$

$\bar{s}^2 = \boldsymbol{k}_{**} - \boldsymbol{k}_*^T\boldsymbol{K}^{-1}\boldsymbol{k}_*$

(posterior and prediction part is missing)

Lecture 4: Bayesian Classification

Classification algorithms:

- Probabilistic: Bayes classifier, Logistic regression.
- Non-probabilistic: K-nearest neighbors, Support Vector Machines.
- Many others ...

Probabilistic classifiers produce a probability of class membership: $P(t_{new} = k|x_{new},X,t)$

Non-probabilistic classifiers produce a hard assignment, e.g. $t_{new} = 1$ or $t_{new} = 0$.

Probabilities provide us with more information: $P(t_{new} = 1) = 0.6$ is more useful than $t_{new} = 1$.

This is very important when misclassification cost is high and imbalanced (diseased/healthy person).

Logistic Regression

We can apply a logistic function $h(.)$ To $f(x_{new};w) = w^Tx_{new}$ to squash it between 0 and 1.

We get $P(t_{new} = k|x_{new},X,t) = h\big(f(x_{new};w)\big) = h(w^Tx_{new}) = \frac{1}{1+\exp(-w^Tx_{new})}$

Defining a prior: $p(w) = \prod_{d=1}^{D}\mathcal{N}(0,\sigma^2)$

Defining a likelihood:

Assume independence: the noises of observations are independent: $p(t, X, w) = \prod p(t_n|x_n, w)$

For a binary classification: $P(t_n = 0|x_n, w) = 1 - P(t_n = 1|x, w) = 1 - h(w^T x_n)$

Posterior: $p(w|X, t, \sigma^2) = \frac{p(t|X,w)p(w)}{p(t|X)} = \frac{p(t|X,w)p(w)}{\int p(t|X,w)p(w)dw}$ (can't calculate the integral $\int \left(\frac{1}{1+exp}\right)^n$ ...)

We can compute $p(t|X, w)p(w) = g(w, X, t)$

We have then three options:

1. Find the most likely value of w
2. Approximate $p(w|X, t)$
3. Sample from $p(w|X, t)$

MAP estimate

$g(w, X, t) \propto p(w|X, t)$. So, if $\hat{w}$ maximizes g it also maximizes the prior (gradient descent).

Once we have $\hat{w}$, we make prediction with $h(\hat{w}^T x_{new})$

We get a linear boundary (linear model): $h(w^T x_{new}) = 0.5 \Leftrightarrow w^T x_{new} = 0$.

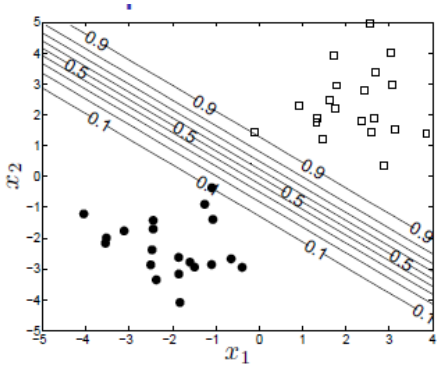
We can approximate $p(w|X, t)$ with a Gaussian $q(w|X, t) = \mathcal{N}(\mu, \Sigma)$, where:

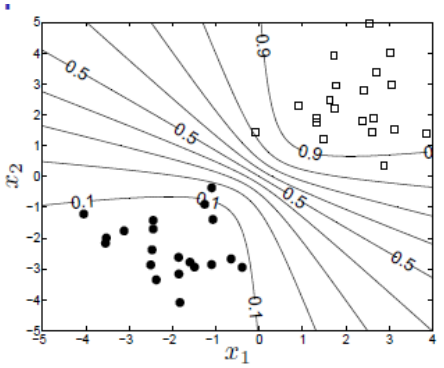
- $\mu = \hat{w} = \arg\max \log g$ ($\arg\max p$ should be a good approximation of the Gaussian mean)
- $\Sigma^{-1} = -\frac{\partial^2 \log g}{\partial w \partial w^T}$ (if g is Gaussian, we get directly the covariance matrix)

Prediction: $P(t_{new} = 1|x_{new}, X, t) = \mathbb{E}_{\mathcal{N}(\mu,\Sigma)}\big(P(t_{new} = 1|x_{new}, w)\big) = \int \mathcal{N}(\mu, \Sigma) \frac{1}{1+\exp(-w^T x_{new})} dw$ (still can't calculate this integral).

The solution is to sample (we know how to sample from Gaussian) and then average.

Draw S samples $\{w_1, \dots, w_S\}$ from the distribution: $\mathbb{E}_{\mathcal{N}(\mu,\Sigma)}\big(P(t_{new} = 1|x_{new}, w)\big) \approx \frac{1}{S}\sum \frac{1}{1+\exp(-w_s^T x_{new})}$





Bayesian Classifier

Based on Bayes rule: $P(t_{new} = k|\boldsymbol{X}, \boldsymbol{t}, \boldsymbol{x}_{new}) = \frac{P(\boldsymbol{x}_{new}|t_{new} = k, \boldsymbol{X}, \boldsymbol{t})P(t_{new}=k)}{\sum P(\boldsymbol{x}_{new}|t_{new} = j, \boldsymbol{X}, \boldsymbol{t})P(t_{new}=j)}$

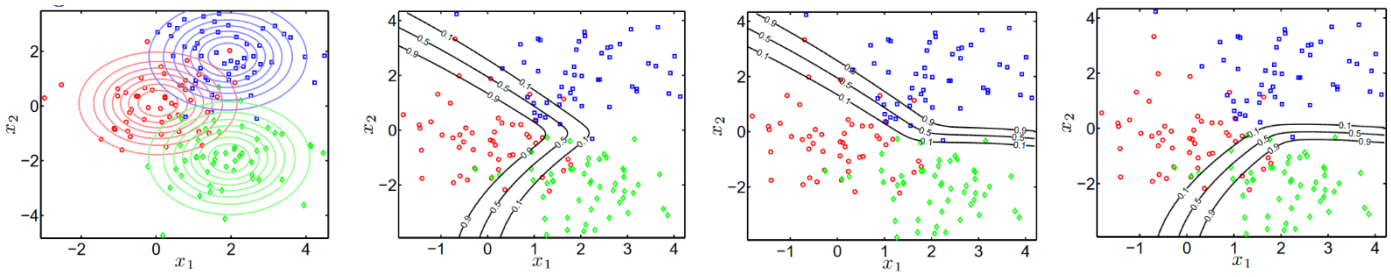
- Likelihood $P(x_{new}|t_{new} = k, X, t)$: how likely to observe x_{new} if it's in class k? We can choose this distribution as we like depending on our data (Gaussian, binomial likelihood) → Training data with $t = k$ is used to determine params of likelihood for class k (e.g. mean and covariance).
- Prior $P(t_{new} = k)$: this is not related to data (there is no x_{new}).
  - There are fewer instances of class $k = 0$ means $P(t_{new} = 0) < P(t_{new} = 1)$.
  - No prior preference means $P(t_{new} = 0) = P(t_{new} = 1)$.
  - Class 0 is very rare means $P(t_{new} = 0) \ll P(t_{new} = 1)$.

Naïve-Bayes

We add the assumption that all features of x are independent.

$$p(x) = p(x_d|x_{d-1}, \dots, x_1)p(x_{d-1}|x_{d-2}, \dots, x_1) \dots p(x_1) = p(x_d)p(x_{d-1}) \dots p(x_1)$$

Step1: fitting the class-conditional densities, i.e. find features' means and variances for each class.
Step2: evaluate densities at test point.



Performance Evaluation

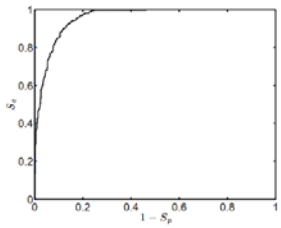
0/1 loss

proportion of times classifier is wrong. Mean loss is defined as $\frac{1}{N}\sum \delta(t_n \neq t_n^*)$

+ Used for binary or multiclass classification.
+ Simple to compute and give a single value.
− Does not consider class imbalance

Sensitivity and specificity

True/False Negatives/Positives: true means correctly classified, and positives refer to class 1.
Sensitivity $S_e = \frac{TP}{TP+FN}$ (proportion of diseased classified as diseased).
Specificity $S_p = \frac{TN}{TN+FP}$ (proportion of healthy classified as healthy).
We would like both to be as high as possible



ROC Analysis

The Receiver Operating Characteristic curve shows how S_e and $1 - S_p$ vary as the threshold changes.

We can quantify the performance by computing the area under the ROC curve (AUC).

Confusion Matrix

		True Class	
		1	0
Predicted Class	1	TP	FP
	0	FN	TN

Lecture 5: Variational Inference

The aim of this part is to approximate the posterior $p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{t})$

1- Introduce $q(\boldsymbol{w})$

For simplicity $q(\boldsymbol{w}) = \prod q(w_i) = \prod \mathcal{N}(w_i|\mu_i,\sigma_i^2)$ (it's a choice and other alternatives can be used).

2- Distance between $q(\boldsymbol{w})$ and $p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{t})$

A possible distance can be $KL[q(\boldsymbol{w})\|p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{t})] = \mathbb{E}_{q(\boldsymbol{w})}\left[\log\frac{q(\boldsymbol{w})}{p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{t})}\right] = \mathbb{E}_{q(\boldsymbol{w})}[\log q(\boldsymbol{w})] - \mathbb{E}_{q(\boldsymbol{w})}[\log p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{t})]$

This is not a distance! Not symmetric! Does not satisfy triangular equality.

The second term is problematic because the posterior is intractable.
After rearranging: $\log p(\boldsymbol{t}|\boldsymbol{X}) - KL[q(\boldsymbol{w})\|p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{t})] = \mathbb{E}_{q(\boldsymbol{w})}[\log p(\boldsymbol{t}|\boldsymbol{w},\boldsymbol{X})] - KL[q(\boldsymbol{w})\|p(\boldsymbol{w})]$
We maximize the $\text{objective} = \mathbb{E}_{q(\boldsymbol{w})}[\log p(\boldsymbol{t}|\boldsymbol{w},\boldsymbol{X})] - KL[q(\boldsymbol{w})\|p(\boldsymbol{w})]$ wrt $q(\boldsymbol{w})$, i.e. change q params.

3- Optimization

If $p(w) = \prod \mathcal{N}(w_i|0,s^2)$ we obtain $KL[q(\boldsymbol{w})\|p(\boldsymbol{w})] = \frac{1}{2}\sum\left[\log\frac{s^2}{\sigma_i^2} - 1 + \frac{\sigma_i^2}{s^2} + \frac{\mu_i^2}{s^2}\right]$
$\mathbb{E}_{q(\boldsymbol{w})}[\log p(\boldsymbol{t}|\boldsymbol{w},\boldsymbol{X})] = \int q(\boldsymbol{w})\log p(\boldsymbol{t}|\boldsymbol{w},\boldsymbol{X})\,d\boldsymbol{w} \approx \frac{1}{N_{MC}}\sum_{h=1}^{N_{MC}}\log p\left(\boldsymbol{t}|\widetilde{\boldsymbol{w}}^{(h)},\boldsymbol{X}\right)$ where $\widetilde{\boldsymbol{w}}^{(h)}$ are sampled from $q(\boldsymbol{w})$ such that $\left(\widetilde{\boldsymbol{w}}^{(h)}\right)_i = \mu_i + \epsilon_i\sigma_i$.
Now we maximize $\widetilde{\text{objective}} = \frac{1}{N_{MC}}\sum_{h=1}^{N_{MC}}\log p\left(\boldsymbol{t}|\widetilde{\boldsymbol{w}}^{(h)},\boldsymbol{X}\right) - KL[q(\boldsymbol{w})\|p(\boldsymbol{w})]$ with gradient-based optimization.

$$\text{vapr}' = \text{vpar} + \frac{\alpha_t}{2}\nabla_{\text{vpar}}\left(\widetilde{\text{objective}}\right);\ \alpha_t \to 0$$

Lecture 6: Bayesian Unsupervised Learning

K-means

Each cluster is defined by a position in the input space $\boldsymbol{\mu_k} = [\mu_{k1},\mu_{k2}]^T$.
Each $\boldsymbol{x_n}$ is assigned to its closest cluster.
Euclidean distance is usually used: $d_{nk} = (\boldsymbol{x_n} - \boldsymbol{\mu_k})^T(\boldsymbol{x_n} - \boldsymbol{\mu_k})$

There is no analytical solution, so we use an iterative algorithm:

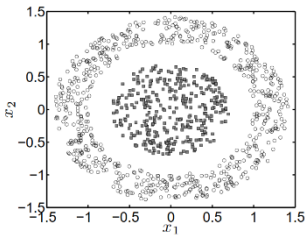
1. Randomly pick $\boldsymbol{\mu_1},\dots,\boldsymbol{\mu_K}$
2. Assign each $\boldsymbol{x_n}$ to the closest $\boldsymbol{\mu_k}$
3. Define $z_{nk} = 1$ if $\boldsymbol{x_n}$ is assigned to $\boldsymbol{\mu_k}$
4. Update $\boldsymbol{\mu_k}$ to the average of $\boldsymbol{x_n}$'s $\boldsymbol{\mu_k} = \frac{\sum z_{nk}\boldsymbol{x_n}}{\sum z_{nk}}$
5. Return to 2 until assignment do not change

Kernelizing K-means

$$\begin{aligned} d_{nk} = (\boldsymbol{x_n} - \boldsymbol{\mu_k})^T(\boldsymbol{x_n} - \boldsymbol{\mu_k}) &= \left(\boldsymbol{x_n} - \frac{\sum z_{nk}\boldsymbol{x_n}}{\sum z_{nk}}\right)^T\left(\boldsymbol{x_n} - \frac{\sum z_{nk}\boldsymbol{x_n}}{\sum z_{nk}}\right) \\ &= \left(\boldsymbol{x_n} - N_k^{-1}\sum z_{nk}\boldsymbol{x_n}\right)^T\left(\boldsymbol{x_n} - N_k^{-1}\sum z_{nk}\boldsymbol{x_n}\right) \\ &= \boldsymbol{x_n^T x_n} - 2N_k^{-1}\sum_m z_{mk}\boldsymbol{x_m^T x_n} + N_k^{-2}\sum_{m,l} z_{mk}z_{lk}\boldsymbol{x_m^T x_l} \\ &= k(\boldsymbol{x_n},\boldsymbol{x_n}) - 2N_k^{-1}\sum_m z_{mk}k(\boldsymbol{x_n},\boldsymbol{x_m}) + N_k^{-2}\sum_{m,l} z_{mk}z_{lk}k(\boldsymbol{x_m},\boldsymbol{x_l}) \end{aligned}$$

Algorithm:

1. Choose a kernel and any necessary params
2. Start with random assignments z_{nk}
3. For each x_n assign it to the nearest center using the new distance
4. Return to 3 until assignment do not change



Mixture Models

Can we hypothesis a function that could have generated the data?