APUNTES-COMPLETOS-para-el-final.pdf

Icorsan

Aprendizaje Automático II

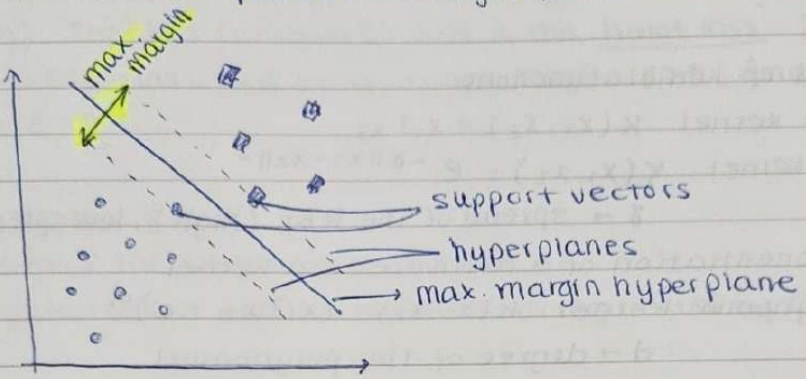
3º Grado en Ciencia e Ingeniería de Datos

Escuela Politécnica Superior. Campus de Leganés
Universidad Carlos III de Madrid

MACHINE LEARNING 2

SUPPORT VECTOR MACHINE

GOAL: Divide the points with a mathematical function so that every point is in the right side of the boundary. We also want ~~that~~ to separate the test data as much as possible, so that there is the max. possible margin from the line.



SVM wants to create the best decision boundary to divide an n-dimensional space into ~~boundaries~~ classes so that we want classify new data in the future correctly. This boundary is the hyperplane, and the extreme cases (points) that help creating the hyper plane are the support vectors.

~~Support vectors~~
Using only the subset of SV instead of all the training data, we will obtain the same classifier.
The whole classification is the sum of all the SVs.

Non linearly separable case "kernel" refers to kernel trick
Find a mapping into a linearly separable space is tricky so to avoid all these computational costs, we use kernel SVM.
With kernel functions, we perform nonlinear classifications: the kernel corresponding to the similarity of two vectors projected into a high dimensional space.
Replacing dot product by kernel functions (symmetric and positive semidefinite) permits obtaining all the pros of feature mapping.

Every Kernel function induces a mapping $h(\cdot)$ into a feature space H such that the evaluation of a Kernel between two observations x_i and x_j:

$$x_i \rightarrow h(x_i) \qquad x_j \rightarrow h(x_j)$$

$$k(x_i, x_j) = h(x_i)^T h(x_j)$$

↳ we compute the kernel without knowing $h(\cdot)$

Examples of Kernel functions

> Linear Kernel: $K(x_1, x_2) = x_1^T x_2$

> RBF Kernel: $K(x_1, x_2) = e^{-\gamma \|x_1 - x_2\|^2}$

$\gamma \rightarrow$ spread of the RBF (high γ, low spread or curvature)

> Exponentiation of a distance is a kernel

> Polynomial Kernel: $K(x_1, x_2) = (x_1^T x_2 + c)^d$

$d \rightarrow$ degree of the polynomial

> Combinations of Kernels are Kernels

Soft margin and slacks

Slack variables are introduced to allow certain constraints to be violated, to achieve smoother classification boundaries.

The idea is simple: to allow some points to be mistaken (not well-classified) and keep margin as wide as possible. Goal: to generalise better for the classification of unseen points.

> C is the hyperparameter that controls tradeoff between margin maximization and missclassifications in the training set.

(BUEN EJEMPLO EN EL PDF DE NOTION)

Large C → large error

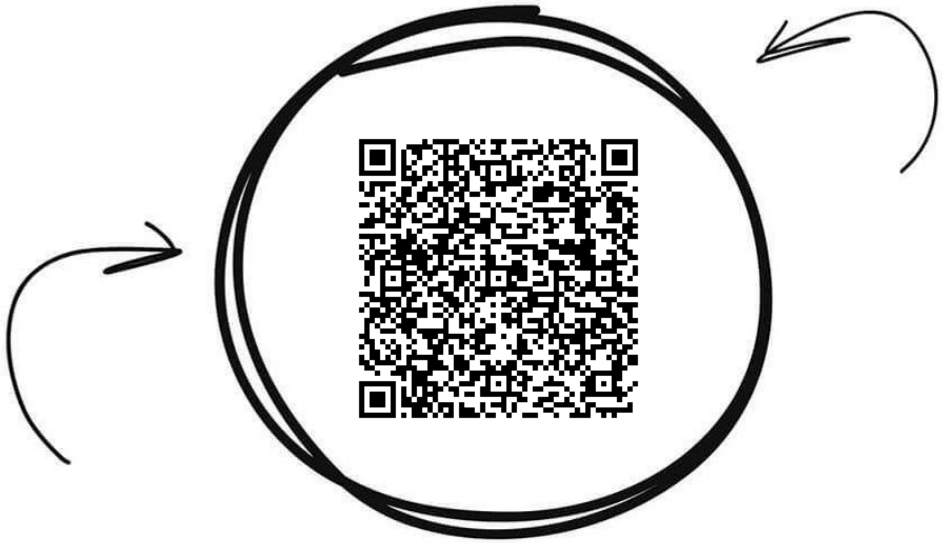
low C → low error

The SVM is the result of a robust, CONVEX QUADRATIC program: the solution is unique and there are no local minima.

The SVM is sparse: it only depends on the support vectors that lie in the margin.


ANTES DE SER ENEMIGOS FUERON HERMANOS
Disney
MUFASA
EL REY LEÓN
20 DE DICIEMBRE
SOLO EN CINES
ENTRADAS YA A LA VENTA
©2024 Disney Enterprises, Inc. Pendiente de calificación por edades.

Aprendizaje Automático II



Banco de apuntes de la

WUOLAH

Comparte estos flyers en tu clase y consigue más dinero y recompensas

1. Imprime esta hoja
2. Recorta por la mitad
3. Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes
4. Llévate dinero por cada descarga de los documentos descargados a través de tu QR

RBF Kernel

Captures radial symmetry: all the points at the same Euclidean distance of a SV receive the same activation.

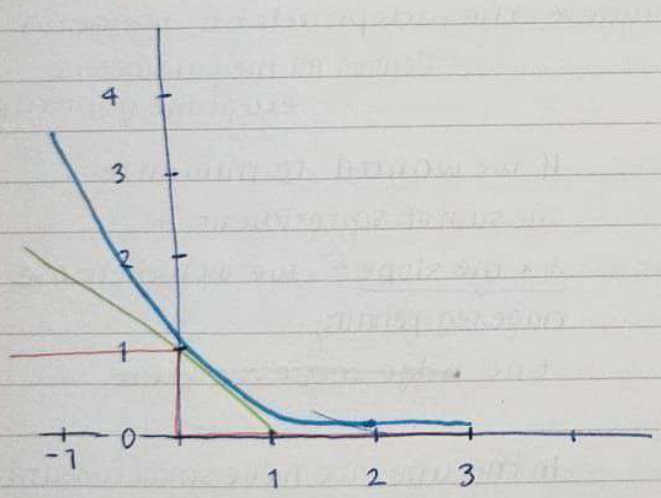
SVM loss function

All classifiers involve a loss function that depends on the no. of errors in the training set + some regularization (to avoid overfitting). The loss function for SVM is the hinge loss $\Sigma_1^n \varepsilon_i$.
The wrong-classified observations have a $\varepsilon > 0$, and right ones have $\varepsilon = 0$.

SVM Regularization

L_2 norm of the weight vector $||w||^2$
The regularization and loss function is regulated by C.

Types of loss



Zero-one loss → not differentiable, counts classification errors.

Hinge loss → Not differentiable but also continuous

Square loss → Diff. and cont. but strong penalty to outliers

Regularization imposes a bound on the weight vector to facilitate a good generalization. Each misclassified point augments the values of its Lagrange Multiplier.
C is the maximum effort ~~of~~ the weight vector will make in the direction of x_i ($W = \Sigma_{i=1}^n y_i \alpha_i x_i$) to get this instance correctly classified.
C controls the penalty imposed on the observations that lie outside the ε-insensitive region (margin) in SVR, and prevents overfitting (regularization) → repeated later in its corresponding point

SUPPORT VECTOR REGRESSION

Linear regression finds weights w and intercept w_0 in the model:

$$\hat{y} = f(x) = w^T x + w_0$$

(w^T: slope; w_0: y-intercept)

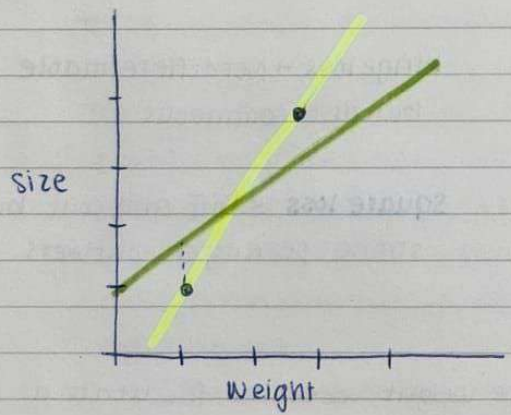
and minimizes the Mean Squared Error (MSE) in the training set.

Regularized linear regression

MSE forces a strong dependence of the value of w with the outliers. To control the influence of the outliers in the final regression function is to introduce a regularization term: the minimization of the norm of the weight vector.

Ridge Regression

It is a tuning method of estimating the coefficients of multiple regression models in scenarios where the independent variables are highly correlated.



Contain All the parameters except the y-intercept

If we wanted to minimize the sum of sq. residuals + $\lambda \times$ the slope^2 (ridge reg. penalty), we would choose the ridge regression line.

In this line, we have small amount of bias but less variance than in least squares. (SEE STATQUEST VIDEO)

The ridge regression line has smaller slope (larger λ) so it is less sensitive to changes in weight than the L.S. line.

We will choose (with K-Fold) the λ value which results in smaller variance.

We can use least squares to minimize sums, and f. ex. if we have three parameters, we will need at least three observations to fit an optimal plane.

On the contrary, with Ridge Regression you need fewer samples.

KERNEL RIDGE REGRESSION

Kernel Ridge Regression combines Ridge Regression with the kernel trick. It thus learns a linear function in the space induced by the respective kernel and the data. For non-linear kernels, this corresponds to a non-linear function in the original space.

→ Kernel Ridge is identical to Support Vector Regression. However, different loss functions are used: KRR uses squared error loss while SVR uses ϵ-insensitive loss, both combined w/ l2 regularization. In contrast with SVR, KRR can be done in a closed-form and it is typically faster in medium-sized datasets.
On the other hand, the learned model is non-sparse and thus slower than SVR, which learns a sparse model $\epsilon > 0$, at prediction time.

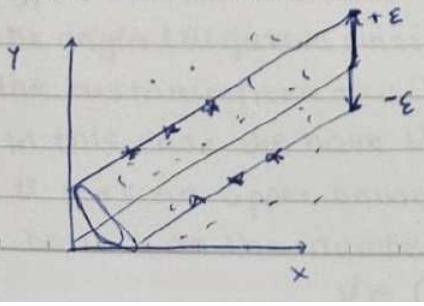
To sparse in KRR, one trick is to select some n training data as centers of the kernel functions $\{x_{c_1}, x_{c_2}, \dots, x_{c_n}\}$ (centroids ~~of each~~ are each of these centers).
later, construct a mapping $h(x)$ into a Feature space of n dimensions where each component of $h()$ is the evaluation of a kernel function centered in one of the centroids. Construct a data matrix H and solve a linear RR with h as a data matrix and y as target vector.

Motivation of SVR

Choosing centroids at random can be problematic to choose the no. of centroids and where to pick them.

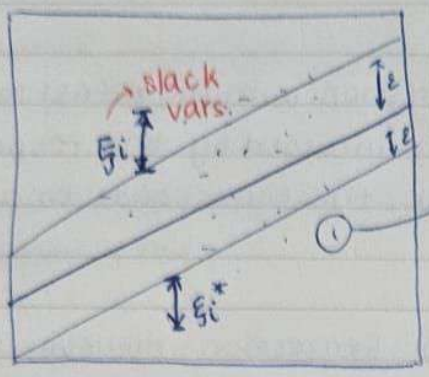
USEFUL TERMS: Kernel, hyperplane, support vectors

ϵ-insensitive loss function: Ignores errors that are within the ϵ distance of the observed value by treating them as equal to 0. The loss is measured as the distance between the observed value y and the ϵ boundary. The higher ϵ, the higher tolerance to error (less sensitive to errors, more robust). see in SVM *



The ϵ-insensitive loss tries to fit a tube of width 2ϵ to the data. The support vectors stand in the walls of the ϵ-tube

Example: Current exchange rate



regularizes the error
higher C, more weight to minimize the errors

$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i + \xi_i^*$ errors

flatness = minimal error

Problem!
How much are we going to allow slack variables (ξ_i) = errors?

Here we can still exchange even if we lose.

This is a convex problem that can be solved with quadratic programming, and a global optimum

soft margin: similar to SVM "soft margin", we consider the points that lay outside the ε-margin, allowing some regression errors up to the values ξ_i and ξ_i^* yet still satisfying the required conditions (exchange occur. in the training set for smoothness ⇒ better general

C is the constraint, a positive value that controls the penalty imposed on observations that lay outside the ε-insensitive area and helps prevent overfitting. This value determines the tradeoff between the flatness of $f(x)$ and the amount up to which deviations larger than ε are tolerated.

In smooth SVR, slack variables also turn into Support Vectors. to address some infeasibilities in the constraints

Kernel methods for Unsupervised learning: ONE CLASS SVM and SPECTRAL CLUSTERING

ONE CLASS SVM

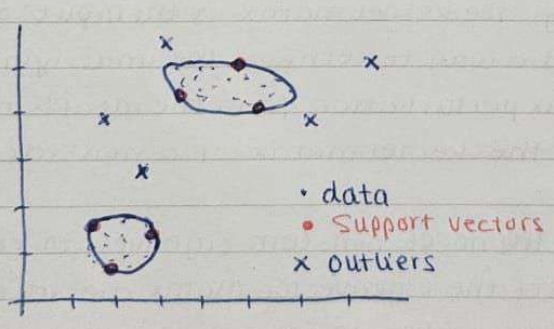
One-class SVM is an unsupervised model for outliers or anomaly detection. Unlike supervised SVM, OCSVM has no target labels for the model training process. Instead, it learns the boundaries of normal data points and classifies the ones outside of them as anomalies.

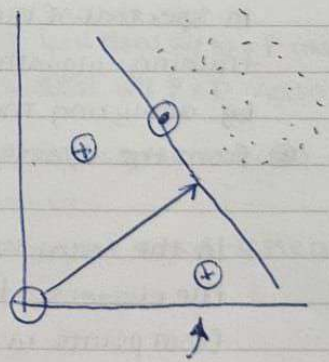
Novelty detection is used in two scenarios:

- Clean datasets removing outliers before training.
- Decide when the statistics that define the problem have changed.

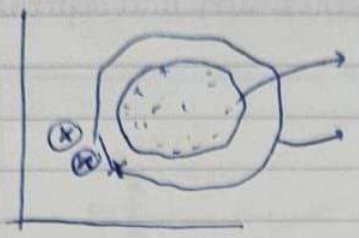
FORMULATIONS

- Enclosing hypersphere: The kernel induces a mapping onto a Feature Space. The algorithm determines the center and radius of the smallest possible hypershape in the Feat. Space that encloses all points, and the ones left out are outliers.
  This hypershere has a center a and a radius r, and we want to minimize the sphere to minimize the ~~cost of~~ effect of incorporating outliers to the solution. To create a soft margin, slack variables and the penalty parameter C, as well as Lagrange Multipliers, are used.





- Hyperplane formulation (classification): Separates all the data from the origin (negative class) in the feature space F and maximizes the distance from this hyperplane to the origin.
  In this case we have $\nu(m)$ instead of C as a parameter.
  ν sets an upper bound on the number of outliers and a lower bound on the number of training examples used as SVs.
  $\nu \in (0, 1]$



High ν (allow more outliers)

Low ν (allow less outliers)

ν is a regularization parameter.

In classification, the points left outside the plane and closer to the origin are outliers.

Eg: if $\nu = 0.3$, at least 30% of the training data will be SVs (in line in the margin or outside of it, in the wrong side). So, at most 30% of data will be outliers on the wrong side of the classification boundary.

SPECTRAL CLUSTERING

Intuition: Data samples that are similar should be in the same groups, and data samples that are different should be in different groups.

K Means

1. Initialize the K centroids at random
2. Assign the X observations to the closest centroid (clusters)
3. Recompute the centroids (mean of all the current clusters)
4. 2 and 3 until convergence

In Spectral Clustering, the kernel matrix is the input of the training algorithm and a way to retrieve the underlying clusters (*) by analyzing the matrix perturbation from the ideal situation (*) from the spectrum of the kernel matrix of a non-ideal kernel.

In the ideal kernel, the no. of non-zero eigenvalues coincide w/ the clusters, the rows of the eigenvector matrix are identical from points in the same cluster. In reality, kernels of instances from different clusters are not zero. ~~and from~~

The affinity matrix (similarity matrix) and training data samples define a graph in which the nodes are the training samples and the edges connecting two nodes mean a ① value in A(i,j).
Finding clusters = cutting the graph in subgraphs by removing weak edges, ~~connecting heavily~~
(Finding the optimal normalized cut = eigenvalue problem w/ laplacian matrix.)

DIFFERENCES BETWEEN SPECTRAL CLUSTERING and KMEANS

- SC: data points as nodes that form a connected graph, and finding clusters by partitioning the graph (based on spectral decomposition) into subgraphs

- K-Means: divide the objects into K clusters such that the metric relative to the centroids is minimized.
K-Means (as a data-clustering algorithm) is ideal for discovering GLOBULAR CLUSTERS, where all points from each cluster are in close proximity to each other.
Spectral clustering is a graph-clustering technique where you don't cluster data points directly in their native data space, instead form a similarity matrix where the (i,j)-th entry is some similarity distance between such points.

- Practical considerations: In K-Means you factorize the input data matrix while in SC you factorize the laplacian matrix, so in P data points w/ N features each < in KMeans you deal w/ N x P matrix; in SC you deal w/ P x P matrix.

In SC, you are indifferent to the no. of features you use, so it is a problem to apply it to large datasets.
SC deals better w/ non-linear separable input and KMeans with linearly separable cases.

GAUSSIAN PROCESSES

→ A gaussian process is a probability distribution over possible functions

ELEMENTS

- Scenario: data D, model $y = h()$ unknown and task (learn $h()$ from D)
- Hypothesis space H
- Observations D
- Version space: subset of H consistent with D.
- Likelihood: $p(D|h)$ → probability of D if $h()$ is the right hypot.
- Prior $p(h)$ → doesn't change with data, reflects BELIEFS prior to data (how natural is h compared to other hypotheses).
- Posterior $p(h|D) = p(D|h)\,p(h)/p(D)$ balances between prior (no data) and posterior (knowledge increases with data). It converges to MAP with more data, and to MLE when likelihood overwhelms the prior.
- Posterior predictive distribution → the distribution of possible unobserved values conditional on the observed ones

$$p(y^*|x^*, D) = \sum_h p(y^* = h^*(x)|x^*, h)\,p(h|D)$$

Bayes model averaging

It is the weighted average of the predictions of all hypotheses.
If we have enough data $p(h|D)$ converges to δ on the MAP hypothesis (only one hypothesis survive).

BMA over MAP learning: BMA always narrow the prediction with more data on h, but MAP can broaden it.

Uncertainty and noise: Uncertainty comes from the partial knowledge about the hypothesis space, while noise is present because of noisy variables in the joint distribution.

Bayesian regression aims to return a pdf on w and σ_n by introducing prior knowledge on w (independent from X)
Bayesian regression models tend to perform better in smaller dataset (than frequent standard regression). This is especially true when there is external information that you can incorporate to your model prior.

(+) uncertainty can be represented as a set of possible outcomes and its likelihoods ~~but in some cases it is not enough~~. Regression offers a whole distrib. over target var.
↳ compute view!

(+) simple linear regression estimates the parameters ~~to~~ and use them to make predictions, while Bayesian regression estimates distributions over the parameters and predictions.
With Bayesian regression you recover a whole range of inferential solutions (*) instead of a point estimate and a confidence interval.

PASAR A SUDS

(-) The Bayesian approach has no way to represent and handle uncertainty within the background knowledge and prior prob. function (limitation)

(-) High computational cost.

In Bayesian Regression we assume that there is a linear model with added noise that can serve to solve ~~of~~ the problem, and you need to find W and σ_n such that

$$y = W^T x + \varepsilon \quad \varepsilon \sim N(0, \sigma_n)$$

is a good approximation for the observations that you are receiving.

! If you don't choose an appropriate kernel, performance is poor and convergence is slow.
~~The Bay~~

(*) The Bayesian model gives the pdf of the linear regression model predicted for a test observation X_t. This probabilistic model is full of RVs that follow a joint prob. distribution.

The Bayesian linear model ~~that~~ can be easily extended to nonlinear models using kernel trick.

Kernels for GPs: RBF, ARD, constant, white noise, absolute exponential, matern (w/ parameters K_ν which is a Bessel function and $\Gamma()$ which is the gamma function*; ν controls the smoothness of the function). (As $\nu \to \infty$, Matern converges to RBF kernel,)
rational quadratic kernel, exp-sine-squared and dot-product.

BAYESIAN CLASSIFICATION

The way to address the binary classification problem is to learn the posterior probability of one of the output classes, either $p(y=1|x)$ or $p(y=-1|x)$. The way to learn so is to learn the linear model squashed with a sigmoid function.

$$p(y=1|x) = \sigma(W^T x)$$

→ ~~It~~ Gaussian process classification is a method of probabilistic modeling that assumes that all data points are generated by a latent function plus some noise.

The Gaussian process is used as the prior to learn this underlying function

Latent variables are inferred indirectly through a mathematical model from other observable variables that can be measured.

MODELS FOR DISCRETE AND CONTINUOUS DATA

Discriminative models draw boundaries in data space (models the decision boundaries), while generative models includes the distribution of the data itself and how likely a given example is.

Parametric models have a fixed number of parameters, while nonparametric models have the no. of parameters grown as the size of the data D (N) grows.

A statistic is sufficient with respect to a statistical model and its associated unknown parameter if no other statistic that can be calculated from the same sample provides any additional information of the parameter.

BETA BINOMIAL MODEL → The probability of success in each of a fixed or known number of Bernoulli trials is unknown or random, and is drawn from a beta distribution.

$$\text{Prior} \rightarrow p(\theta) = \text{Beta}(\theta|a,b) = \frac{1}{B(a,b)}\theta^{a-1}(1-\theta)^{b-1}$$

$$\text{Posterior} \rightarrow p(\theta|D) = \text{Beta}(N_1+a, N_0+b)$$

$$\text{MAP} \rightarrow \hat{\theta}_{MAP} = \frac{N_1+a-1}{N_0+b+a-2}$$

$$\text{ML} \rightarrow \hat{\theta}_{ML} = \frac{N_1}{N}$$

The multinomial distribution is a generalization of a binomial distribution. For n independent trials each of which leads to a success of exactly one of the K categories (with each category having a success probability), the multinomial distribution gives the probability of any particular combination of no. of successes for the various categories.
Eg: It models the prob. of counts for each side of a k-sides dice rolled n times.

The Dirichlet Distribution (denoted as $\text{Dir}(\alpha)$) is a family of multivariate probability distributions parametrized by a vector α of positive reals. Dirichlet is the conjugate prior of the multinomial distribution.

Likelihood

$$\text{Prior} \rightarrow \text{Dir}(\theta|\alpha) = \frac{1}{B(\alpha)} \prod \theta_k^{\alpha_k - 1}$$

$$\text{Posterior} \rightarrow \text{Dir}(\theta|N+\alpha)$$

$$\text{MAP} \rightarrow \hat{\theta}_k = \frac{N_k + \alpha_k - 1}{N + \alpha_0 - k} \qquad \text{ML} \rightarrow \hat{\theta}_k = \frac{N_k}{N}$$

LINEAR GAUSSIAN SYSTEM

It transforms a RV x ~~~~ that follows a Gaussian pdf in another RV y (also Gaussian). The mean of y depends on x and the covariance of y is independent of x.

Wishart distribution → Generalization of the gamma distribution to positive definite matrices.

A way to obtain a point estimate is to define a cost/loss function $l(\hat{\theta}, \theta)$ and minimizing risk

$$\hat{\theta}_{MAP} = \underset{\theta}{\text{argmax}}\ p(\theta|D)$$

$$\hat{\theta}_{MMSE} = \mathbb{E}\{p(\theta|D)\}$$

$$\hat{\theta}_{MMAE} = \underset{\text{med}}{\mathbb{E}}\{p(\theta|D)\}$$

MIXTURE MODEL: to represent a probability distribution of a dataset as a mixture of multiple components distribution. It is a probabilistic model ~~that~~ for representing the presence of subpopulations within an overall population without requering that ~~an~~ observed data identifies the subpopulation to which ~~indi~~ an individual observation belongs.

They are used to classify data into categories based on the probability distribution.

K-Means vs GMM

K-Means classifies data points using distance from the cluster centroid and GMM uses probabilistic assignment of data points to clusters.

k-Means try to minimize $(x-\mu_k)^2$ and GMM min. $\frac{(x-\mu_k)^2}{\sigma^2}$

so GMM takes variance into consideration. (so computes a "weighted" distance not just Euclidean one).

Advantages and disadvantages of GMM.

⊕ Does not assume clusters to be any geometry, works well with non-linear geometric distributions.

⊕ Does not bias cluster sizes to specific structures as K-M (circular) have

⊖ Uses all components to which has access so when dimensionality is high, it is difficult to initialize the clusters.

⊖ Assume normal distribution for features (still less restrictive than circula

⊖ Hard for categorical features

Expectation Maximization (EM) → Generative model ~~(K-M is discriminative)~~

It is a method to find ~~a~~ the good parameter ~~to~~ to maximize the likelihood function when there are latent variables.

1. Expectation: Assign probabilistically each point to a cluster.
2. Update the parameters for each cluster based on the points in the cluster weighted by their probability from 1) (MAXIMIZATION)
3. Repeat until convergence

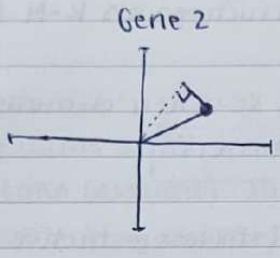
KNN → lazy learner KMeans → Eager learner

PRINCIPAL COMPONENT ANALYSIS

↳ For large datasets with many dimensions/features

The goal is to project the data onto a space with smaller dimensionality while minimizing the norm-2 error in the reconstruction of the original data (projections of the PCAs).

PCA finds the best fitting line by maximizing the SSR from the projected points to the origin



PCAs are a linear combination of variables
PCAs are eigenvectors of the data's cov. matrix.

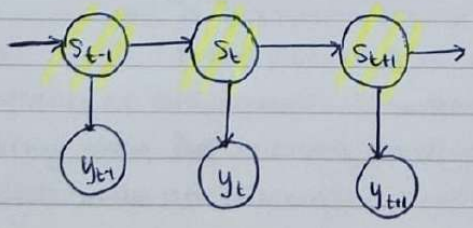
1. Compute the covariance matrix of the data D, S.
2. Compute the M dominant eigenvectors: $U_{D \times M}$
3. Project each datapoint onto the space of M dimensions defined by the basis U.
4. If centered data: $\tilde{x}_i = Uz_i$

→ Probabilistic PCA is a dimensionality reduction technique that analyzes data via a lower dimensional latent space.
It is very used when there are data missing.

MARKOV MODELS

Discrete x_t (s_t): HMM Continuous x_t: State Space Model (SSM)

Hidden Markov Model



$S: \{S_1, S_2, \dots, S_t\}$ hidden state sequence
$Y: \{y_1, y_2, \dots y_t\}$ observed continuous sequence
$A: \{a_{ij} : a_{ij} = p(S_{t+1} = j \mid S_t = i)\}$: state transition probabilities
$B: \{b_i : p_{b_i}(y_t) = p(y_t \mid S_t = i)\}$: observation emission probabilities
$\pi: \{\pi_i : \pi_i = p(S_1 = i)\}$: initial state probability distribution
$\theta: \{A, B, \pi\}$ model parameters

Example of HMM: Automatic Speech Recognition (s is the phonemes and y are the features extracted from speech signal)

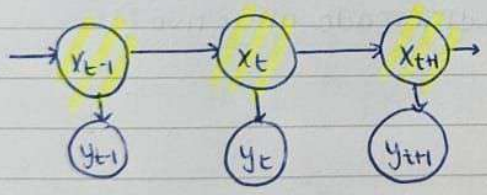
> Forward algorithm in this context is used to calculate a "belief state" (the probability of a state at a certain time), given the history of evidence.

> Forward-backward algorithm is an inference algorithm which computes the posterior marginals of all hidden state variables given a sequence of observations.

> Viterbi algorithm is an algorithm for obtaining the MAP estimate of the most likely sequence of hidden states (Viterbi path) that results in a sequence of observed events.

> Baum-Welch algorithm is a special case of EM Algorithm used to find the unknown parameters of a HMM. It uses the Forward-Backward algorithm for the maximization step.

Linear ~~Sdm~~ State Space Model Eg: positioning, target tracking, signal pred.



$$x_t = Ax_{t-1} + Cu_t + \epsilon_t$$
$$y_t = Bx_{t-1} + Du_t + z_t$$

↳ Equations to see how states change over time

$X = \{x_1, x_2, \dots, x_t\}$ Hidden state sequence
$Y = \{y_1, y_2, \dots, y_t\}$ Observed sequence
$U = \{u_1, \dots, u_t\}$ Control sequence
$E = \{\epsilon_1, \dots, \epsilon_t\}$ System noise sequence
$Z = \{z_1, \dots, z_t\}$ Observation noise sequence
A: State transition matrix
B: observation matrix
C, D: control matrices

> Kalman Filter: It is used to estimate the state of a system from noisy measurements. It is a correction-prediction approach: it uses the current estimates of the states to predict what states will be in the next step. Then it compares this prediction to the actual measurements and adjusts the estimates accordingly. (Repeats over t)

Conjugate prior: when the prior distribution is in the same family as the posterior, so it is possible to derive a closed expression for the posterior distribution without numerical approximations.

⊕ Analytical tractability
⊕ Conjugate priors give insight on the structure of posterior.
⊖ They may not capture well enough complexity
⊖ Overly restrictive

Why to center in PCA?

1. With centering, the PC1 corresponds to the direction in which data varies the most, not the direction in which it is more shifted.
2. So that variances are not influenced by the mean of data
3. More interpretable results as we compare in the same scale.

MARKET SIZE

Situation - Task - Action - Result

1. Preguntas
2. Top-down
3. Reasonable answer?
4. Implicaciones de la respuesta - ¿mercado atractivo?


ANTES DE SER ENEMIGOS FUERON HERMANOS
Disney
MUFASA
EL REY LEÓN
20 DE DICIEMBRE
SOLO EN CINES
ENTRADAS YA A LA VENTA
©2024 Disney Enterprises, Inc. Pendiente de calificación por edades.