Machine Learning

Probabilistic KNN.

Mark Girolami

girolami@dcs.gla.ac.uk

Department of Computing Science
University of Glasgow

Probabilistic KNN

UNIVERSITY *of* GLASGOW

- KNN is a remarkably simple algorithm with proven error-rates

Probabilistic KNN

UNIVERSITY
of
GLASGOW

- KNN is a remarkably simple algorithm with proven error-rates
- One drawback is that it is not built on any probabilistic framework

Probabilistic KNN

UNIVERSITY of GLASGOW

- KNN is a remarkably simple algorithm with proven error-rates
- One drawback is that it is not built on any probabilistic framework
- No posterior probabilities of class membership

Probabilistic KNN

UNIVERSITY of GLASGOW

- KNN is a remarkably simple algorithm with proven error-rates
- One drawback is that it is not built on any probabilistic framework
- No posterior probabilities of class membership
- No way to infer number of neighbours or metric parameters probabilistically

Probabilistic KNN

UNIVERSITY of GLASGOW

- KNN is a remarkably simple algorithm with proven error-rates
- One drawback is that it is not built on any probabilistic framework
- No posterior probabilities of class membership
- No way to infer number of neighbours or metric parameters probabilistically
- Let us try and get around this 'problem'

Probabilistic KNN

- The first thing which is needed is a likelihood

Probabilistic KNN

UNIVERSITY *of* GLASGOW

- The first thing which is needed is a likelihood
- Consider a finite data sample $\{(t_1, \mathbf{x}_1), \cdots, (t_N, \mathbf{x}_N)\}$ where each $t_n \in \{1, \cdots, C\}$ denotes the class label and D-dimensional feature vector $\mathbf{x}_n \in \mathbb{R}^D$. The feature space $\mathbb{R}^D$ has an associated metric with parameters $\boldsymbol{\theta}$ denoted as $\mathcal{M}_{\boldsymbol{\theta}}$.

Probabilistic KNN

UNIVERSITY of GLASGOW

- The first thing which is needed is a likelihood
- Consider a finite data sample $\{(t_1, \mathbf{x}_1), \cdots, (t_N, \mathbf{x}_N)\}$ where each $t_n \in \{1, \cdots, C\}$ denotes the class label and D-dimensional feature vector $\mathbf{x}_n \in \mathbb{R}^D$. The feature space $\mathbb{R}^D$ has an associated metric with parameters $\boldsymbol{\theta}$ denoted as $\mathcal{M}_{\boldsymbol{\theta}}$.
- A likelihood can be formed as

$$p(\mathbf{t}|\mathbf{X}, \beta, k, \boldsymbol{\theta}, \mathcal{M}) \approx \prod_{n=1} \frac{\exp\left\{\frac{\beta}{k} \sum_{j \sim n|k}^{\mathcal{M}_{\boldsymbol{\theta}}} \delta_{t_n t_j}\right\}}{\sum_{c=1}^{C} \exp\left\{\frac{\beta}{k} \sum_{j \sim n|k}^{\mathcal{M}_{\boldsymbol{\theta}}} \delta_{c t_n}\right\}}$$

Probabilistic KNN

UNIVERSITY of GLASGOW

- The number of nearest neighbours is k and β defines a scaling variable. The expression

$$\sum_{j \sim n|k}^{\mathcal{M}_{\boldsymbol{\theta}}} \delta_{t_n t_j}$$

denotes the number of the nearest k neighbours of $\mathbf{x}_n$, as measured under the metric $\mathcal{M}_{\boldsymbol{\theta}}$ within $N-1$ samples from $\mathbf{X}$ remaining when $\mathbf{x}_n$ is removed which we denote as $\mathbf{X}_{-n}$, and have the class label value of t_n, whilst each of the terms in the summation of the denominator provides a count of the number of the k neighbours of $\mathbf{x}_n$ which have class label equaling c.

Probabilistic KNN

- Likelihood formed by product of terms

$$p(t_n|\mathbf{x}_n, \mathbf{X}_{-n}, \mathbf{t}_{-n}, \beta, k, \boldsymbol{\theta}, \mathcal{M})$$

Probabilistic KNN

UNIVERSITY of GLASGOW

- Likelihood formed by product of terms

$$p(t_n|\mathbf{x}_n, \mathbf{X}_{-n}, \mathbf{t}_{-n}, \beta, k, \boldsymbol{\theta}, \mathcal{M})$$

- This is a Leave-One-Out (LOO) predictive likelihood, where $\mathbf{t}_{-n}$ denotes the vector $\mathbf{t}$ with the n'th element removed

Probabilistic KNN

UNIVERSITY of GLASGOW

- Likelihood formed by product of terms

$$p(t_n|\mathbf{x}_n, \mathbf{X}_{-n}, \mathbf{t}_{-n}, \beta, k, \boldsymbol{\theta}, \mathcal{M})$$

- This is a Leave-One-Out (LOO) predictive likelihood, where $\mathbf{t}_{-n}$ denotes the vector $\mathbf{t}$ with the n'th element removed
- Approximate joint likelihood provides an overall measure of the LOO predictive likelihood

Probabilistic KNN

UNIVERSITY of GLASGOW

- Likelihood formed by product of terms

$$p(t_n|\mathbf{x}_n, \mathbf{X}_{-n}, \mathbf{t}_{-n}, \beta, k, \boldsymbol{\theta}, \mathcal{M})$$

- This is a Leave-One-Out (LOO) predictive likelihood, where $\mathbf{t}_{-n}$ denotes the vector $\mathbf{t}$ with the n'th element removed
- Approximate joint likelihood provides an overall measure of the LOO predictive likelihood
- Should exhibit some resiliance to overfitting due to the LOO nature of the approximate likelihood

Probabilistic KNN

UNIVERSITY
of
GLASGOW

- Posterior inference will follow by obtaining the parameter posterior distribution $p(\beta, k, \boldsymbol{\theta}|\mathbf{t}, \mathbf{X}, \mathcal{M})$

Probabilistic KNN

UNIVERSITY *of* GLASGOW

- Posterior inference will follow by obtaining the parameter posterior distribution $p(\beta, k, \boldsymbol{\theta}|\mathbf{t}, \mathbf{X}, \mathcal{M})$
- Predictions of the target class label t_* of a new datum $\mathbf{x}_*$ are made by posterior averaging such that $p(t_*|\mathbf{x}_*, \mathbf{t}, \mathbf{X}, \mathcal{M})$ equals

$$\sum_k \int p(t_*|\mathbf{x}_*, \mathbf{t}, \mathbf{X}, \beta, k, \boldsymbol{\theta}, \mathcal{M}) p(\beta, k, \boldsymbol{\theta}|\mathbf{t}, \mathbf{X}, \mathcal{M}) d\beta d\boldsymbol{\theta}$$

Probabilistic KNN

UNIVERSITY *of* GLASGOW

- Posterior inference will follow by obtaining the parameter posterior distribution $p(\beta, k, \boldsymbol{\theta} | \mathbf{t}, \mathbf{X}, \mathcal{M})$
- Predictions of the target class label t_* of a new datum $\mathbf{x}_*$ are made by posterior averaging such that $p(t_* | \mathbf{x}_*, \mathbf{t}, \mathbf{X}, \mathcal{M})$ equals

$$\sum_k \int p(t_* | \mathbf{x}_*, \mathbf{t}, \mathbf{X}, \beta, k, \boldsymbol{\theta}, \mathcal{M}) p(\beta, k, \boldsymbol{\theta} | \mathbf{t}, \mathbf{X}, \mathcal{M}) d\beta d\boldsymbol{\theta}$$

- Posterior takes an intractable form so MCMC procedure is proposed so that the following Monte-Carlo estimate is employed

$$\hat{p}(t_* | \mathbf{x}_*, \mathbf{t}, \mathbf{X}, \mathcal{M}) = \frac{1}{N_s} \sum_{s=1}^{N_s} p(t_* | \mathbf{x}_*, \mathbf{t}, \mathbf{X}, \beta^{(s)}, k^{(s)}, \boldsymbol{\theta}^{(s)}, \mathcal{M})$$

Probabilistic KNN

- Posterior sampling algorithm simple Metropolis algorithm

Probabilistic KNN

- Posterior sampling algorithm simple Metropolis algorithm
- Assume priors on k and β are uniform over all possible values (integer & real)

Probabilistic KNN

UNIVERSITY of GLASGOW

- Posterior sampling algorithm simple Metropolis algorithm
- Assume priors on k and β are uniform over all possible values (integer & real)
- Proposal distribution for β_{new} is Gaussian i.e. $\mathcal{N}(\beta^{(i)}, h)$

Probabilistic KNN

UNIVERSITY of GLASGOW

- Posterior sampling algorithm simple Metropolis algorithm
- Assume priors on k and β are uniform over all possible values (integer & real)
- Proposal distribution for β_{new} is Gaussian i.e. $\mathcal{N}(\beta^{(i)}, h)$
- Proposal distribution for k is uniform between `Min` & `Max` values

$$index \sim U(0, k_{step} + 1)$$
$$k_{new} = k_{old} + k_{inc}(index);$$

Probabilistic KNN

UNIVERSITY
of
GLASGOW

- Need to accept this new move using Metropolis ratio

Probabilistic KNN

UNIVERSITY *of* GLASGOW

- Need to accept this new move using Metropolis ratio

$$\min\left\{1, \frac{p(\mathbf{t}|\mathbf{X}, \beta_{new}, k_{new}, \boldsymbol{\theta}_{new}, \mathcal{M})}{p(\mathbf{t}|\mathbf{X}, \beta, k, \boldsymbol{\theta}, \mathcal{M})}\right\}$$

Probabilistic KNN

UNIVERSITY *of* GLASGOW

- Need to accept this new move using Metropolis ratio

$$\min\left\{1, \frac{p(\mathbf{t}|\mathbf{X}, \beta_{new}, k_{new}, \boldsymbol{\theta}_{new}, \mathcal{M})}{p(\mathbf{t}|\mathbf{X}, \beta, k, \boldsymbol{\theta}, \mathcal{M})}\right\}$$

- Builds up a Markov Chain whose stationary distribution is $p(\beta, k, \boldsymbol{\theta}|\mathbf{t}, \mathbf{X}, \mathcal{M})$

Probabilistic KNN

- Need to accept this new move using Metropolis ratio

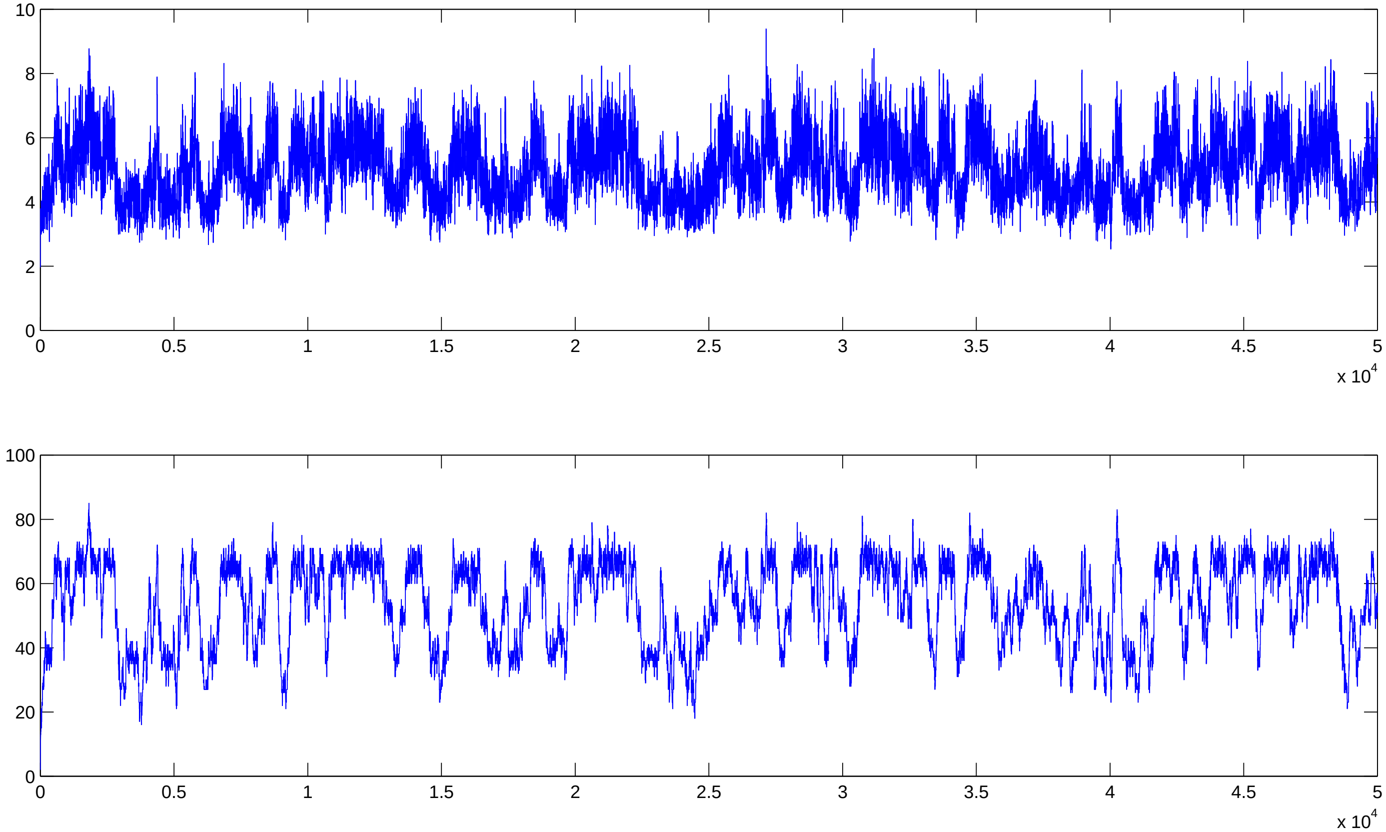
$$\min \left\{ 1, \frac{p(\mathbf{t}|\mathbf{X}, \beta_{new}, k_{new}, \boldsymbol{\theta}_{new}, \mathcal{M})}{p(\mathbf{t}|\mathbf{X}, \beta, k, \boldsymbol{\theta}, \mathcal{M})} \right\}$$

- Builds up a Markov Chain whose stationary distribution is $p(\beta, k, \boldsymbol{\theta}|\mathbf{t}, \mathbf{X}, \mathcal{M})$
- Very simple algorithm to implement - Matlab and C implementations available

Probabilistic KNN

UNIVERSITY *of* GLASGOW

- Trace of Metropolis Sampler for β & k



Probabilistic KNN

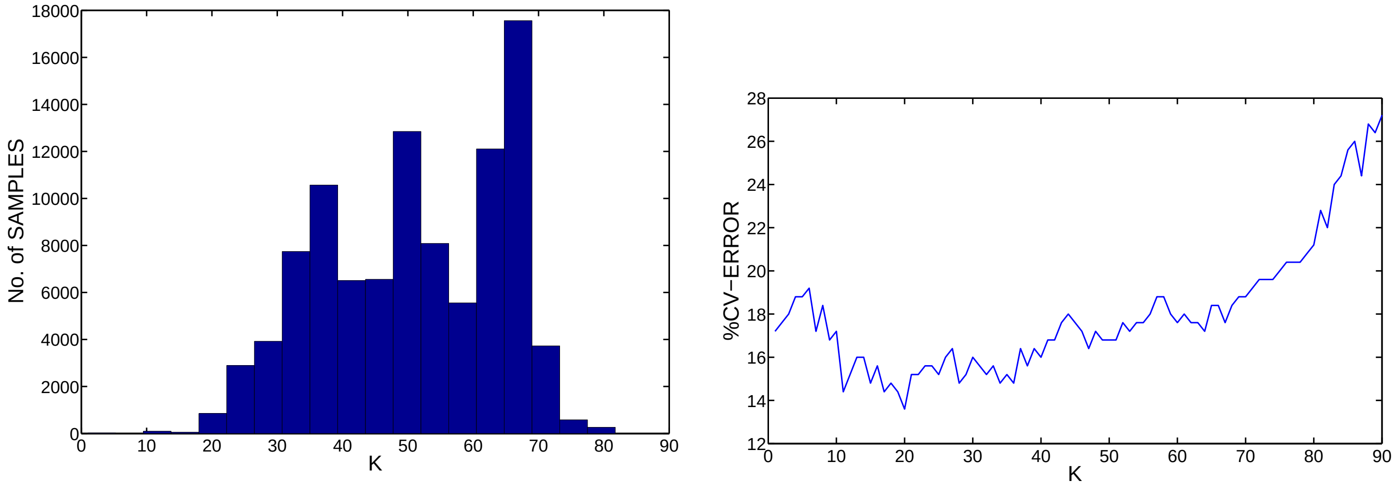


Figure 1: The top graph shows a histogram of the marginal posterior for K on the synthetic Ripley dataset and the bottom shows the 10CV error against the value of K.

Probabilistic KNN

UNIVERSITY
of
GLASGOW

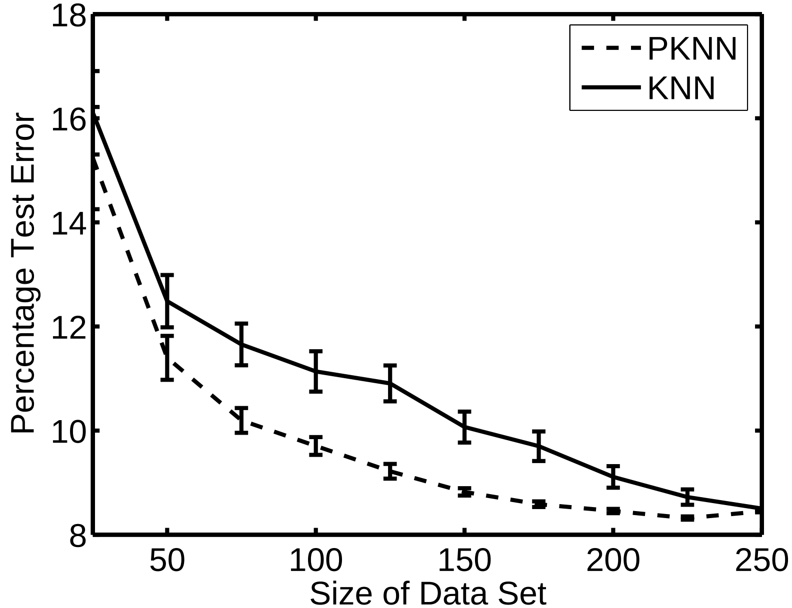


Figure 2: The percentage test error obtained with training sets of varying size from 25 to 250 data points. For each sub-sample size, 50 random subsets were sampled and each of these used to obtain a KNN and PKNN classifier which were then used to make predictions on the 1000 independent test points. The mean percentage performance and associated standard error obtained for each training set are shown in the above figure for each classifier.

Probabilistic KNN

UNIVERSITY *of* GLASGOW

Data	KNN	PKNN	P-Value
Glass	29.91 ± 9.22	26.67 ± 8.81	0.517
Iris	5.33 ± 5.25	4.00 ± 5.62	0.537
Crabs	15.00 ± 8.82	19.50 ± 6.85	0.240
Pima	27.00 ± 8.88	24.00 ± 14.68	0.645
Soybean	14.50 ± 16.74	4.50 ± 9.56	0.155
Wine	3.922 ± 3.77	3.37 ± 2.89	0.805
Balance	11.52 ± 2.99	10.23 ± 3.02	0.324
Heart	15.18 ± 5.91	15.18 ± 4.43	1.000
Liver	33.60 ± 6.98	36.26 ± 12.93	0.705
Diabetes	25.91 ± 7.15	25.25 ± 8.11	0.970
Vehicle	36.28 ± 5.16	37.22 ± 4.53	0.732

Probabilistic KNN

UNIVERSITY of GLASGOW

Data	KNN	PKNN
Glass	39.55	243.52
Iris	7.58	91.8
Crabs	21.99	156.30
Pima	24.10	103.60
Soybean	1.16	38.38
Wine	27.9	144.90
Balance	609.86	555.72
Heart	96.11	145.22
Liver	116.71	189.73
Diabetes	1643.09	567.03
Vehicle	4226.69	1063.13

Probabilistic KNN

UNIVERSITY
of
GLASGOW

- PKNN is a fully Bayesian method for KNN classification

Probabilistic KNN

UNIVERSITY of GLASGOW

- PKNN is a fully Bayesian method for KNN classification
- Requires MCMC therefore slow

Probabilistic KNN

UNIVERSITY of GLASGOW

- PKNN is a fully Bayesian method for KNN classification
- Requires MCMC therefore slow
- Possible to learn metric though this is computationally demanding

Probabilistic KNN

UNIVERSITY *of* GLASGOW

- PKNN is a fully Bayesian method for KNN classification
- Requires MCMC therefore slow
- Possible to learn metric though this is computationally demanding
- Predictive probabilities more useful in certain applications - e.g. clinical prediction

Probabilistic KNN

UNIVERSITY of GLASGOW

- PKNN is a fully Bayesian method for KNN classification
- Requires MCMC therefore slow
- Possible to learn metric though this is computationally demanding
- Predictive probabilities more useful in certain applications - e.g. clinical prediction
- On 0-1 loss no statistically significant difference with CV & KNN