Motivation
- Nearest-neighbor-like methods for time series classification widely used in practice, often with outstanding performance
- Little theoretical development for characterizing performance in terms of:
  - How much training data to use?
  - How much of time series do we observe?

Goal: develop theory to explain performance of nearest-neighbor-like methods for time series classification, and relate theory to practice (forecasting which news topics will go viral on Twitter)

Hypothesis: in many real time series datasets, there are only a few possible patterns (latent sources) relative to how many time series we can collect (a news topic goes viral on Twitter only in a few ways yet we can collect time series for a huge number of news topics)

A Latent Source Model
1. Choose random latent source
2. Add i.i.d. zero mean sub-Gaussian noise & random integer time shift $(0,1,...,\Delta_{max})$
3. Observe time steps $1,2,...,T$

Measurement
Latent sources
- true label
- $p(k)$

Oracle MAP estimator (if noise is Gaussian and we knew the latent sources)

\[
\theta = \arg\max_{\theta} \log p(D_{train} | \theta)
\]

Likelihood ratio test:
\[
\frac{\sum_{i \in \epsilon} \exp\left(\frac{1}{\sigma^2} (s_i - x_i - \Delta t_i)^2\right)}{\sum_{i \notin \epsilon} \exp\left(\frac{1}{\sigma^2} (s_i - x_i - \Delta t_i)^2\right)} \geq 1
\]

If true, declare label of $s$ to be $+1$

We don't actually know the latent sources!
- use training data $x_1,...,x_T$ generated from latent source model as proxy (assume each is observed for all time and come with ground truth labels)

Weighted majority voting
\[
\sum_{i \in \epsilon} \exp\left(\frac{1}{\sigma^2} (s_i - x_i - \Delta t_i)^2\right) + \sum_{i \notin \epsilon} \exp\left(\frac{1}{\sigma^2} (s_i - x_i - \Delta t_i)^2\right) \geq 1
\]

If true, declare label of $s$ to be $+1$

\[
\hat{t} = \arg\min_{t} \sum_{i \in \epsilon} (s_i - x_i - \Delta t_i
\]

Declare label of $s$ to be the same as that of $\hat{t}$

Theorem: Under the latent source model, with $n = \Theta(k \log T)$ training time series, if
\[
\left| \sum_{i \in \epsilon} (s_i - x_i - \Delta t_i) \right| \geq \Omega(\sqrt{2}T),
\]
then weighted majority voting (with $\gamma = \frac{1}{2n}$) and nearest-neighbor classification each classify time series $s$ correctly with probability at least
\[
1 - \delta \text{ once we've seen the first } T = \Omega\left(\log(2\Delta_{max} + 1) + \log\gamma\right) \text{ time steps of } s.
\]

Why not just learn the latent sources?
- For Gaussian noise and no time shifts, existing results on learning Gaussian mixture models require more training data than what our results require or require more stringent assumptions on separation of mixture components

Binary Classification

Time series $s$ to be classified

Measurement

Oracle MAP estimator (if noise is Gaussian and we knew the latent sources)

\[
\theta = \arg\max_{\theta} \log p(D_{train} | \theta)
\]

Likelihood ratio test:
\[
\frac{\sum_{i \in \epsilon} \exp\left(\frac{1}{\sigma^2} (s_i - x_i - \Delta t_i)^2\right)}{\sum_{i \notin \epsilon} \exp\left(\frac{1}{\sigma^2} (s_i - x_i - \Delta t_i)^2\right)} \geq 1
\]

If true, declare label of $s$ to be $+1$

We don't actually know the latent sources!
- use training data $x_1,...,x_T$ generated from latent source model as proxy (assume each is observed for all time and come with ground truth labels)

Weighted majority voting
\[
\sum_{i \in \epsilon} \exp\left(\frac{1}{\sigma^2} (s_i - x_i - \Delta t_i)^2\right) + \sum_{i \notin \epsilon} \exp\left(\frac{1}{\sigma^2} (s_i - x_i - \Delta t_i)^2\right) \geq 1
\]

If true, declare label of $s$ to be $+1$

\[
\hat{t} = \arg\min_{t} \sum_{i \in \epsilon} (s_i - x_i - \Delta t_i
\]

Declare label of $s$ to be the same as that of $\hat{t}$

Theorem: Under the latent source model, with $n = \Theta(k \log T)$ training time series, if
\[
\left| \sum_{i \in \epsilon} (s_i - x_i - \Delta t_i) \right| \geq \Omega(\sqrt{2}T),
\]
then weighted majority voting (with $\gamma = \frac{1}{2n}$) and nearest-neighbor classification each classify time series $s$ correctly with probability at least
\[
1 - \delta \text{ once we've seen the first } T = \Omega\left(\log(2\Delta_{max} + 1) + \log\gamma\right) \text{ time steps of } s.
\]

Why not just learn the latent sources?
- For Gaussian noise and no time shifts, existing results on learning Gaussian mixture models require more training data than what our results require or require more stringent assumptions on separation of mixture components

Experimental Results

Synthetic data
- $k = 280$ latent sources, $\frac{1}{5}$ with label $+1$, $\frac{4}{5}$ with label $-1$; generate each from Gaussian process: $N(0.100)$ smoothed w/10 Gauss filter [scale 30]
- Noise is $N(0,1)$, max shift is $\Delta_{max} = 100$
- Sample $n = 50 \log k$ training time series

- Weighted majority voting ($\gamma = 1/8$) and nearest-neighbor classification have similar performance for large $T$ in agreement with theory
- For small $T$, weighted majority voting outperforms nearest-neighbor classification $\Rightarrow$ weighted majority voting better suited than NN classification for online time series classification

Forecasting which news topics will go viral on Twitter
- 500 news topics that go viral; 500 that do not $\Rightarrow$ split 50%-50% train-test
- Forecast trends on test dataset
- Online time series classification

This work was supported in part by the Army Research Office under MURI Award 58153-MA-MURI. GHC was supported by an NDSEG fellowship.