COMP 330: Support Vector Machines

Chris Jermaine and Kia Teymourian

Rice University

Alternatives to Logistic Regression

The "Big Three" for classification

- Logistic regression
- Support Vector Machines
- kNN

Problem With Un-Regularized Logistic Regression

When the data are "linearly separable"

- Possible to choose model get infinite LLH
- Just choose ANY cutting plane between classes
- Then pump up the magnitude of regression coefs arbitrarily
- Bad: Not clear which plane is preferred
- Bad: model fails in easiest case!

SVMs: Geometric, Not Probabilistic

Starts with question:

- What should classifier do in this super-esay case?
- Just put the widest strip possible between two classes
- Future points above center of strip are "yes"
- Below center of strip are "no"
- Points that keep strgin from expanding are "support vectors"

Basic Formulation

Any line/plane/hyperplane can be described by a normal vector w, distnace b

- The line/plane/hyperplane is all points x where $w \cdot x - b = 0$

Basic Formulation

SVM chooses two planes $w \cdot x - b = 1$, $w \cdot x - b = -1$

- Distance betwen them is $2/||w||$
- Use $||w||$ to denote the l_2 norm of the vector w

Where $w \cdot x_i - b \geq 1$ when x_i is "yes"

Where $w \cdot x_i - b \leq -1$ when x_i is "no"

- in general, where $y_i(w \cdot x_i - b) \geq 1$

Basic Formulation

So in the end we have...

Choose w to maximize $2/||w||$ (alt., to minimize $||w||$)

Subject to $y_i(w \cdot x_i - b) \geq 1$

That's all a SVM is!

One Issue: What If Not Linearly Separable?

Then no solution to above problem!

- Solution: don't requre a "hard margin"
- Allow some error
- Training points on wrong side of cutting plane get a penalty

"Soft Margin" Formulation

Choose w to minimize $||w||^2 + c\sum_i \epsilon_i$

Subject to $y_i(w \cdot x_i - b) \geq 1 - \epsilon_i$

That's all a soft-margin SVM is!

How To Solve?

This is a constrained optimization problem

- Everything we've done (GD, MCMC, Newton's) has been un-constrained
- How to deal with?
- Turns out, using some math, can re-write as unconstrained
- Simply minimize

$$\frac{\lambda}{2}||w||^2 + \frac{1}{n}\sum_i \max(0, 1 - y_i(w \cdot x_i))$$

- where $\lambda = \frac{1}{n \times c}$
- Amenable to gradient descent (one issue: non-smooth max function)

The Kernel Trick

Motivation:

- SVMs (and logistic regression) are linear
- But many classification problems are not...
- Kernel trick: map into higher-D, use linear classifier there
- Not just for SVMs, but closely linked with them

The "Trick"

Possible to learn an SVM without explicitly performing mapping

To do this, start with the "dual" formulation of the problem:

- Maximize

$$\sum_i \alpha_i \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j)$$

- Subject to

$$0 \leq \alpha_i \leq \frac{1}{\lambda}$$

Mathematically equivalent to the last one

Key observaltion: each x_i only used as input to dot product

So no need to explicitly map to high-D

- As long as have n by n matrix
- Where each entry is pairwise dot product in high-D, we're cool!

The "Trick"

Math tells us:

- ANY mapping from pairs to matrix entries is associated with SOME high-D space
- As long as we get a positive semi-definite matrix

This mapping is called a "kernel"

Standard Kernels

"Polynomial kernel"

- Replace $(x_i \cdot x_j)$ with $(1 + (x_i \cdot x_j))^d$ for $d > 0$

"Gaussian kernel"

- Replace $(x_i \cdot x_j)$ with $\mathbf{exp}(-||x_i - x_j||^2/(2\sigma^2))$

Kernel Trick Plus/Minus

Good: can give better classification accuracy

- Often used in practice for this reason!!

Bad: often sensitive to parameter (σ in Gaussian kernel, for example)

Bad: computationally more complex... dual formulation not easily amenable to GD

- Means kernels useful mostly for smaller problems

Finally: LR or SVM?

SVM naturally regularizing

But not much difference between linear SVM and regularized LR

- Regularized LR seems more common in "big data"

But kernel trick makes SVM standard choice for small data

Questions?