Abstrato

Scikit-learn é um módulo Python que integra uma ampla gama de algoritmos de aprendizado de máquina de última geração para problemas supervisionados e não supervisionados de média escala. Este pacote se concentra em levar o aprendizado de máquina para não especialistas usando uma linguagem de alto nível de uso geral. A ênfase é colocada na facilidade de uso, desempenho, documentação e consistência da API. Possui dependências mínimas e é distribuído sob a licença BSD simplificada, incentivando seu uso em ambientes acadêmicos e comerciais. O código-fonte, binários e documentação podem ser baixados em <http://scikit-learn.sourceforge.net>.

1. Introdução

A linguagem de programação Python está se estabelecendo como uma das linguagens mais populares para computação científica. Graças à sua natureza interativa de alto nível e seu ecossistema em desenvolvimento de bibliotecas científicas, é uma escolha atraente para o desenvolvimento algorítmico e a análise exploratória de dados (Dubois, 2007; Milmann e Avaizis, 2011). No entanto, como uma linguagem de uso geral, é cada vez mais usada não apenas em ambientes acadêmicos, mas também na indústria.

O Scikit-learn aproveita esse ambiente rico para fornecer implementações de última geração de muitos algoritmos de aprendizado de máquina bem conhecidos, enquanto mantém uma interface fácil de usar totalmente integrada com a linguagem Python. Isso atende à necessidade crescente de análise de dados estatísticos por não especialistas nas indústrias de software e web, bem como em áreas fora da ciência da computação, como biologia ou física. O Scikit-learn difere de outras caixas de ferramentas de aprendizado de máquina em Python por vários motivos:

i) é distribuído sob a licença BSD ii) incorpora código compilado para eficiência, ao contrário do MDP (Zito et al., 2008) e pybrain (Schaul et al., 2010), iii) depende apenas de numpy e scipy para facilitar distribuição fácil, ao contrário do pymvpa (Hanke et al., 2009) que possui dependências opcionais como R e shogun e iv) concentra-se na programação imperativa, ao contrário do pybrain que usa um framework de fluxo de dados. Embora o pacote seja principalmente escrito em Python, ele incorpora as bibliotecas C ++ LibSVM (Chang e Lin, 2001) e LibLinear (Fan et al., 2008) que fornecem implementações de referência de SVMs e modelos lineares generalizados com licenças compatíveis. Os pacotes binários estão disponíveis em um rico conjunto de plataformas, incluindo Windows e quaisquer plataformas POSIX. Além disso, graças à sua licença liberal, ele foi amplamente distribuído como parte das principais distribuições de software livre, como Ubuntu, Debian, Mandriva, NetBSD e Macports e em distribuições comerciais como a “Enthought Python Distribution”.

4. Design de código

Objetos especificados por interface, não por herança. Para facilitar o uso de objetos externos com o scikit-learn, a herança não é imposta; em vez disso, as convenções de código fornecem uma interface consistente. O objeto central é um estimador, que implementa um método de ajuste, aceitando como argumentos uma matriz de dados de entrada e, opcionalmente, uma matriz de rótulos para problemas supervisionados. Estimadores supervisionados, como classificadores SVM, podem implementar um método de previsão. Alguns estimadores, que chamamos de transformadores, por exemplo, PCA, implementam um método de transformação, retornando dados de entrada modificados. Os estimadores também podem fornecer um método de pontuação, que é uma avaliação crescente da qualidade do ajuste: uma probabilidade logarítmica ou uma função de perda negada. O outro objeto importante é o iterador de validação cruzada, que fornece pares de índices de teste e trem para dividir os dados de entrada, por exemplo K-fold, deixar um de fora ou validação cruzada estratificada.

Seleção de modelo. O Scikit-learn pode avaliar o desempenho de um estimador ou selecionar parâmetros usando validação cruzada, opcionalmente distribuindo o cálculo para vários núcleos. Isso é feito envolvendo um estimador em um objeto GridSearchCV, onde “CV” significa “validação cruzada”. Durante a chamada para ajustar, ele seleciona os parâmetros em uma grade de parâmetros especificada, maximizando uma pontuação (o método de pontuação do estimador subjacente). prever, pontuar ou transformar são então delegados ao estimador ajustado. Este objeto pode, portanto, ser usado de forma transparente como qualquer outro estimador. A validação cruzada pode ser mais eficiente para certos estimadores, explorando propriedades específicas, como reinicializações a quente ou caminhos de regularização (Friedman et al., 2010). Isso é feito por meio de objetos especiais, como o LassoCV.

Finalmente, um objeto Pipeline pode combinar vários transformadores e um estimador para criar um estimador combinado para, por exemplo, aplicar a redução de dimensão antes do ajuste. Ele se comporta como um estimador padrão e, portanto, o GridSearchCV ajusta os parâmetros de todas as etapas.

5. Alto nível, mas eficiente: algumas compensações

Embora o scikit-learn se concentre na facilidade de uso e seja principalmente escrito em uma linguagem de alto nível, foi tomado cuidado para maximizar a eficiência computacional. Na Tabela 1, comparamos o tempo de computação para alguns algoritmos implementados nos principais kits de ferramentas de aprendizado de máquina acessíveis em Python. Usamos o conjunto de dados Madelon (Guyon et al., 2004), 4400 instâncias e 500 atributos. O conjunto de dados é bastante grande, mas pequeno o suficiente para a maioria dos algoritmos serem executados. SVM. Embora todos os pacotes comparem a chamada libsvm em segundo plano, o desempenho do scikit-learn pode ser explicado por dois fatores. Primeiro, nossos vínculos evitam cópias de memória e têm até 40% menos sobrecarga do que os vínculos Python libsvm originais. Em segundo lugar, corrigimos o libsvm para melhorar a eficiência em dados densos, usar uma área de cobertura de memória menor e usar melhor o alinhamento de memória e os recursos de pipelining dos processadores modernos. Esta versão corrigida também oferece recursos exclusivos, como definir pesos para amostras individuais.

LARS. Refinar iterativamente os resíduos em vez de recomputá-los dá ganhos de desempenho de 2 a 10 vezes sobre a implementação de R de referência (Hastie e Efron, 2004). Pymvpa usa essa implementação por meio das ligações Rpy R e paga um preço alto pelas cópias de memória.

Rede Elástica. Comparamos as implementações de descida por coordenadas do scikit-learn do Elastic Net. Ele atinge a mesma ordem de desempenho que a versão altamente otimizada do Fortran glmnet (Friedman et al., 2010) em problemas de média escala, mas o desempenho em problemas muito grandes é limitado, uma vez que não usamos as condições KKT para definir um conjunto ativo.

kNN. A implementação do classificador de k-vizinhos mais próximos constrói uma árvore de bolas (Omohundro, 1989) das amostras, mas usa uma busca de força bruta mais eficiente em grandes dimensões.

PCA. Para conjuntos de dados médios a grandes, o scikit-learn fornece uma implementação de um PCA truncado com base em projeções aleatórias (Rokhlin et al., 2009).

k-means. O algoritmo k-means do scikit-learn é implementado em Python puro. Seu desempenho é limitado pelo fato de que as operações de array numpy realizam várias passagens sobre os dados.

6. Conclusão

O Scikit-learn expõe uma ampla variedade de algoritmos de aprendizado de máquina, supervisionados e não supervisionados, usando uma interface consistente e orientada a tarefas, permitindo assim uma comparação fácil de métodos para um determinado aplicativo. Uma vez que se baseia no ecossistema científico Python, pode ser facilmente integrado em aplicativos fora da faixa tradicional de análise de dados estatísticos. É importante ressaltar que os algoritmos, implementados em uma linguagem de alto nível, podem ser usados como blocos de construção para abordagens específicas a um caso de uso, por exemplo, em imagens médicas (Michel et al., 2011). O trabalho futuro inclui o aprendizado online, para dimensionar grandes conjuntos de dados.